# *Syscomp Computer Controlled Instruments*
# CGR-101 CircuitGear Manual

Syscomp Electronic Design Limited
`http:\\www.syscompdesign.com`
Revision: 1.09
May 25, 2010


CIRCUITGEAR

## Revision History

| Version | Date | Notes |
|---|---|---|
| 1.00 | September 2008 | First revision |
| 1.01 | October 2008 | Added VNA, XY Plot,Spectrum Analysis |
| 1.02 | May 2009 | Added reference to Ubuntu and Mac installs. Corrected errors. Added pinout diagram for rear panel connector. Added data record. Added hyperterminal note. |
| 1.03 | June 2009 | Updated network analyser description |
| 1.04 | July 2009 | Minor corrections |
| 1.05 | Sept 2009 | Expanded Mac and Linux install notes |
| 1.06 | Oct 2009 | Added rear connector info, exernal trigger, waveform math, offset calibration and amplitude calibration. |
| 1.07 | December 2009 | Corrected Mac OS-X install procedure |
| 1.08 | January 2010 | Documented new GUI V1.13 installation procedures.<br>Reorganized Install and Troubleshooting |
| 1.09 | May 2010 | Documented V1.14 software, new features |

## Contents

# List of Figures

Caution: Never connect this instrument to the AC line. Doing so may result in personal injury and extreme damage to the operator, the instrument and to an attached computer. See section 10 on page 31.

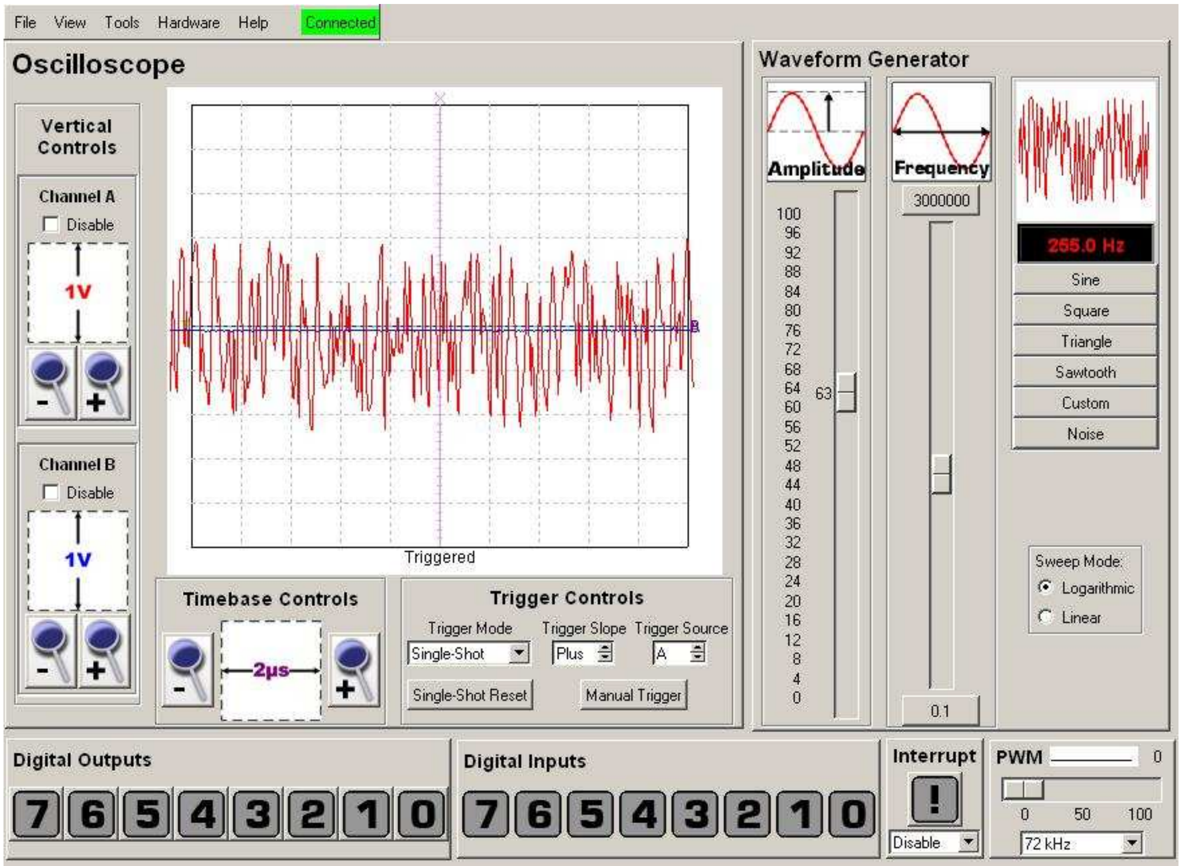



Figure 1: CircuitGear Graphical User Interface (GUI)

# 1 Overview

The Syscomp CircuitGear CGR-101 is a combination of three electronic instruments: a two-channel digital storage oscilloscope, a waveform generator, and a digital input-output port. Host software can operate the instrument as a spectrum analyser and as a vector-network analyser (Bode plotter). CGR-101 is one of a series of instruments from Syscomp Electronic Design.

CircuitGear includes a small hardware module and display software that runs on a host PC. Figure 1 shows the user interface.

## 1.1 Oscilloscope

The oscilloscope is a dual-channel, 20MSample/sec oscilloscope with 10 bit A/D conversion, digital storage and display.

Channels A and B are sampled simultaneously and stored in the oscilloscope memory before being sent for display to the host computer. Consequently, the signals are always time aligned and associated with the same trigger signal. Triggering is accomplished by digital circuitry so it is precise and consistent. Trigger controls include *Mode: Auto, Normal, Single-shot, Manual, Source: A or B*, and *Slope: Positive or Negative*.

The trigger point is continuously adjustable so that the operator can display the signal before and/or after the trigger event.

The oscilloscope timebase frequency is derived from a crystal oscillator, so it can be expected to be precise and stable. The displayed amplitude is determined by 1% resistors and analog-digital conversion. The vertical preamplifier is gain-switched to optimize the signal-noise ratio and will accept x10 scope probes.

The initial software release includes basic oscilloscope functions. Subsequent releases[1] will include additional features such as spectrum analysis.

## 1.2 Waveform Generator

The waveform generator is a direct-digital synthesis (DDS) based device with frequency range between 0.1Hz and 3MHz. The frequency can be manually adjusted continuously, without range switching over that entire range or some subrange.

For automatic sweep, a *Vector Network Analyser* (VNA) (aka Bode Plotter) program is available. The VNA software operates the oscilloscope and generator sections in concert to sweep a network over a specified range and plot the amplitude and phase of the response[2].

The usual sine, square, triangle and sawtooth waveforms are supplied with the instrument. The generator can also load and produce an arbitrary waveform. A GUI-based program *Wavemaker* is available for the construction of arbitrary waveforms.

The generator includes a random-noise source with white spectrum, useable to 2MHz.

## 1.3 Digital Input-Output

The digital I/O section includes an 8 bit output port and 8 bit input port. Outputs are controlled by 8 GUI buttons. Inputs are displayed on 8 GUI indicators.

In addition, there is an *Interrupt* indicator input and *PWM variable frequency* output.

In combination, these controls form the basis for digital controls and displays for basic digital exercises or more advanced control systems.

## 1.4 General

The hardware is in a pocket-sized package that can easily be carried in a student backpack or with a laptop computer. Power and control signals are provided to the hardware via a single serial-emulated USB connection with the host PC.

The PC host displays a graphical user interface for the oscilloscope with frequency readouts, sliders, clickable buttons and various other controls.

The GUI software is written in the Tcl/Tk language. The software is open source and entirely in Tcl/Tk. There are no operating-system specific routines. The GUI software will operate under Linux, Mac or Windows operating systems.

Tcl/Tk is an interpreted language, so reading and modifying the source code is straightforward.

The applications programming interface (API) is documented, so the hardware can be accessed by other computer programs and languages. The only requirement is that the language be able to communicate with a serial port.

[1] Scheduled for Fall 2008

[2] VNA software available Fall 2008.

## 2 Applications

In addition to the usual operation of oscilloscope and signal generator, here are some possible applications of the CircuitGear unit.

- **Logic Net** The digital controls supply the functionality of a digital *Exerciser Unit*, which can apply a stimulus to a digital circuit and measure the output.

  For example the 8 bit digital lines can be used as inputs and indicators for a logic net. Students set up various combinations of input signals to the net and record the outputs to generate logic equations or a truth table for the logic net.

- **State Machine Exerciser** A single manual output line and the PWM output can be used as a pulser for counter and state machine circuits. The manual output exercises the circuit at low speeds, where the behaviour can be observed on the GUI indicators. The PWM output is then used to operate the circuit at higher frequencies, and the oscilloscope can be used to observe faster events.

- **Mixed Analog and Digital Circuits** The digital outputs control a MDAC (multiplying D-A converter) which sets the centre frequency of a bandpass filter. The generator and oscilloscope function as a Vector Network Analyser, showing how the frequency response changes as the digital value is adjusted.

- **Switching Power Supply** The PWM output controls a power MOSFET and LC network which functions as a simple switching power supply. Similarily, PWM output can modulate the power to a DC motor as a simple method of speed control.

- **PWM DAC** It is common for the PWM output of a microprocessor to be used as the basis for a low-cost digital-analog converter. The PWM signal is filtered to produce a variable analog control signal. In this exercise students design the PWM filter and then measure the ripple using the CircuitGear oscilloscope. They can also operate the PWM signal at various frequencies to illustrate the effect of frequency on ripple.

# 3 Features and Specifications

The features and performance specifications are as follows:

**Oscilloscope**

| | |
|---|---|
| Channels | 2 independent channels sampled simultaneously |
| Sampling Frequencies | 20 MSamples/second maximum |
| Vertical Resolution | 10 bits per channel (1:1024) |
| Vertical Bandwidth | 2 MHz |
| Vertical Input Range | $\pm$250mV to $\pm$25V full scale |
| Vertical Gain Settings | 7 settings, 50mV/div to 5V/div, in 1:2:5 sequence |
| Vertical Scale | 10 major divisions |
| Vertical Preamp Ranges | 2 |
| Horizontal Time Settings | 20 settings, 100mSec/div to 50nSec/div, in 1:2:5 sequence |
| Horizontal Scale | 10 major divisions |
| Input Impedance | 1M Ohm parallel 27pF |
| Triggering | Digital comparison with input signal |
| Trigger View | Pre and Post trigger simultaneously viewable |
| Trigger Controls | Source (A, B, Manual), level and slope select |
| Memory Depth | 1K Samples each channel |
| Software, Initial Release | Export to Postscript<br>Selectable screen capture<br>Cursor Readouts |
| Subsequent Release | Spectrum Analysis (FFT)<br>Real Time Histogram<br>X-Y Plot<br>Export to Postscript<br>Selectable screen capture<br>Data record to CSV file<br>Save/Load Settings<br>Cursor Readouts<br>Auto Measurements<br>Vertical Calibration<br>Labview Drivers |

**Waveform Generator**

| | |
|---|---|
| Frequency Range | 0.1Hz to 2MHz |
| Output Amplitude | +/-3V |
| Amplitude Control | Hardware |
| Vertical Resolution | 8 bits at all amplitude settings |
| Output Impedance | 150 ohms |
| Waveforms | Sine, Square, Triangle, Ramp, Arbitrary, Noise |
| Arbitrary Waveform | 8 bit resolution vertical, 256 time points<br>Constructed with *Wavemaker* software |
| Noise | Pseudo-random, 8 bit analog noise, sample rate 12.5Mz, sequence length 21 seconds |

**Digital I/O**

| | |
|---|---|
| Output | 8 bits, GUI (Graphical User Interface) controlled, 5 volt, HCMOS |
| Input | 8 bits, GUI indicators, 5 or 3 volt, HCMOS |
| Pulse waveform | Variable duty cycle at constant frequency<br>35Hz to 72kHz in steps of x2 |
| Interrupt | Selectable level and slope, illuminates ’Interrupt’ indicator. |

| | |
|---|---|
| **Other** | |
| Indicators | Power LED (Green) |
| | Activity LED (Red) |
| Interface | USB 2.0: Emulated serial port |
| Physical Dimensions | 3" x 5" |
| GUI Source Code | Tcl/Tk language |
| | Open source, OSI Compliant |
| | Windows, Linux, Mac operating systems |
| Current Consumption | Scope Only, 230mA |
| | Function Generator Amplitude 100% Open Circuit load, 240mA |
| | Function Generator Amplitude 100% Output Shorted, 260mA |

# 4 Installation

The installation should work correctly under modern versions of Windows, Linux and Macintosh operating systems.

## 4.1 Windows Installation

The installation procedure unpacks and installs the Syscomp Graphical User Interface (GUI) program that interacts with the hardware. It also install two Future Technology Devices International (FTDI) virtual COM port (VCP) drivers necessary to communicate with the hardware.

Screen shots of the install procedure are in figure 2 on page 7.

1. Turn on the computer.
2. Place the Syscomp install CD in the CD player and close the drive.
3. After a brief pause, the screen of figure 2(a) appears. Click `Next`.
4. The screen of figure 2(b) appears. Click `Next`.
5. The `Select Start Menu Folder` of figure 2(c) appears. Click `Next`.
6. The `Select Additional Tasks` screen of figure 2(d) appears. We recommend enabling `Create a Desktop Icon`. Click on `Create a Desktop Icon` until a check mark appears next to it. Click `Next`.
7. The `Ready to Install` screen of figure 2(e) appears. If you are satisfied with the install directories, click on `Install`. Otherwise, click `Back` and revise the necessary entries.
8. The `Installing` screen and progress bar of figure 2(f) appears. Wait while that completes.
9. The `USB Device Driver` install screen of figure 2(g) appears. Click `Next`.
10. The `Installing drivers` screen of figure 2(h) appears.
11. The `Congratulations! You are finished installing drivers...` screen appears, figure 2(i). Click `Finish`.
12. The `Completing the Syscomp Setup Wizard` screen appears, figure 2(j). Click `Finish`.
13. The software is installed. You may eject the CD.


(a) Welcome to Setup Wizard

(b) Select Destination

(c) Select Start Menu Folder

(d) Additional Tasks

(e) Ready to Install

(f) Installing

(g) Welcome to Driver Install

(h) Installing Drivers

(i) Finished Drivers

(j) Completing the Install

Figure 2: Windows Installation Screens

### 4.1.1 First Time Operation

1. Using the supplied USB cable (or similar one of your own choosing), plug the oscilloscope into a computer USB port. If you have more than one USB port, you can chose any port.
2. The green LED on the hardware unit should illuminate, and the computer will make its USB boing noise.
3. There should be a Syscomp icon on the desktop. The exact name will vary, but it should be something like `Syscomp CircuitGear`. Start the oscilloscope program by double clicking on this icon.
4. A screen `Unable to Connect to Device. Examine Connection Settings?` appears. Click on `Yes`.
5. The `Port Settings` control panel appears. Click on `Autodetect`.
6. The program scans through the available COM ports, 1 through 99.
7. If it finds an available port, a notification screen will pop up saying something like `CGR-101 found on COM6`.

   If it does not find an available COM port, try clicking on `Autodetect` again. If this also fails, you'll need to set the COM port manually, section 4.1.2 below.
8. Assuming that the system finds an available COM port, click on `OK`. A `Port Settings Listing` window shows a list of the unavailable COM ports and the one that was found. Click on `Back`
9. The Port Settings control panel shows which port has been selected. Click on `Save and Exit`. This causes a small text file `scopeport.cfg` to be written to the directory where the scope program was launched. This file contains the connection port number.
10. At this point, the `Connected` indicator at the top of the GUI screen show show a green colour. The unit is operational.

From now on, whenever you double-click the program icon, the program should start and automatically connect.

### 4.1.2 Setting the `COM` Port Manually

1. Using a USB cable plug the oscilloscope into a computer USB port. If you have more than one USB port, you can chose any port.
2. The green LED on the hardware unit should illuminate, and the computer will make its USB boing noise.
3. Go to: Start `->` Settings `->` Control Panel `->` System `->` Hardware `->` Device Manager
4. On the Device Manager panel, click on `Ports (COM and LPT)`. You should see an entry like `USB Serial Port (COM 5)`. This is the COM port that the operating system has assigned to the oscilloscope for this session. Make note of the port, in this example `COM 5`.
5. On the oscilloscope GUI, open Hardware `->` Connect. Select the COM port that you found previously. In our example, that would be COM 5. Click on `Save and Exit`.
6. The oscilloscope GUI status message at the top of the screen should show `Connected`. This is a Happy Moment, because your computer is now talking properly to the oscilloscope hardware.

7. Operate the frequency, amplitude or offset controls on the oscilloscope GUI. As you do so, the GUI sends commands to the hardware and you should see flashing from the `Activity` LED on the hardware front panel.

From now on, it should be sufficient to boot up your computer, plug in the oscilloscope and double-click on the desktop icon for the oscilloscope.

## 4.2 Macintosh Install

This section describes how to install Syscomp instruments on a MacBook running OSX. These instructions describe a new 'one button' procedure for software version 1.13 and following. This procedure is much simpler than the previous version.

### 4.2.1 Checking the Host Machine

The Syscomp software is interpreted by the Tcl/Tk language interpreter, so it is essential that the machine is capable of running Tcl/Tk.

The ActiveState website (originators of Tcl/Tk) indicate that Tcl/Tk should run correctly on platforms with 'Mac OS X 10.2+'. There is no mention of requirement for Intel or Power PC processors [3].

The first step is to determine that the computer is capable of running this software. To do that:

1. Click on the Apple icon in the top left corner of the screen.
2. Select 'About this Mac'

For example, on the local Mac machine the Mac information is:

- Mac OSX 10.5.6
- 2.4GHz Intel Core 2 Duo
- 2GB 667MHz RAM

The key items are that the operating system is OSX and it's version is later than 10.2. So the software should run on this machine.

### 4.2.2 Installing Software from the CD

1. Start the Mac.
2. Plug in the Syscomp CGR-101 hardware. The green LED on the hardware should illuminate.
3. Insert the install CD.
4. The CGR-101 icon appears on the desktop. Open it.
5. Open `setup.htm`.
6. Click on `Click here for the DMG installer`.
7. Open `CircuitGear.dmg`

[3] `http://www.tcl.tk/software/tcltk/platforms.html`

8. A screen appears on the desktop, showing the Applications folder and the Syscomp CircuitGear unit. Drag and drop the Syscomp image onto the Applications folder. A progress bar appears and completes.

9. Go to the Applications folder:
   `Finder -> Macintosh HD -> Applications`.

10. Find the entry for the Syscomp CircuitGear. Double click on it. The CGR-101 GUI should appear. It will probably show 'Not Connected' in red. A dialog box will pop up saying 'Examine hardware settings?'. Click on Yes.

11. Examine the hardware settings. Select the entry that says `cu-usbserial-xxxx` where `xxxx` is some serial number. The gui indicator should turn green and show `Connected`.

    If the GUI does not connect, exit from the gui. Unplug the CGR hardware and plug it in again. Restart the GUI and this time the hardware should connect properly.

12. The program should indicate `Connected`, some activity should be visible on the scope trace, the red LED on the scope hardware should flash. The unit is operational.

There's no `eject` button. Steve Jobs doesn't like buttons, apparently. Right click on the CD icon, and select `Eject CD`.

### 4.3 Linux Installation

1. Insert the CD into a drive. An image of the drive should appear on the computer display.

2. Change to the `Source` directory. Copy all files and folders from the /Source directory on the CDROM in to a suitable directory on your machine, for example:
   `/home/yourname/Syscomp/CGR101`

3. Copy the appropriate binary file from the /Source/bin/ directory into the folder you selected *or* execute the appropriate shell script in the Source directory. For example:

   ```
   sh Install-Linux-32-bit.sh
   ```

   *or*

   ```
   cp /home/yourname/Syscomp/CGR101/bin/linux/linux_ix86/CGR101-linux-x86
                  /home/yourname/Syscomp/CGR101/
   ```

   (The above command should be executed on one line.)

4. Execute the binary executable to start the software. For example, use these two commands:

   ```
   cd /home/yourname/Syscomp/CGR101
   ./CGR101-linux-x86
   ```

#### 4.3.1 Hardware Installation

1. Plug the hardware device into an available USB port.
2. Run the program `dmesg` to show which serial port the instrument has been attached. For example, /dev/ttyUSB0.
3. Start the binary executable (the `GUI`, Graphical User Interface) as described above.
4. On the GUI, use the `Hardware-->Port Settings` menu in the instrument software GUI to manually connect to the serial port we found using `dmesg`, eg, `/dev/ttyUSB0`.
5. The GUI should indicate `Connected` and the oscilloscope trace should show activity.

Note that currently, Linux assigns USB serial ports in the order in which the instruments are plugged in. You may have to manually reassign the port settings in the GUI intrumentation software if multiple Syscomp Instruments are used simultaneously. Use `dmesg` each time you attach an instrument to the computer to determine its serial port assignment.

Refer to the technical manual in the `/Documentation` folder of the CDROM for more information on how to determine which USB serial port is being used by the device with help from the dmesg command.

#### 4.3.2 Optional: Running the GUI from Source

You can execute the source code directly using the Tcl interpreter `wish`. The Tcl/Tk language must be installed on your machine. This language is commonly included with most Linux distributions. If it is not installed on your machine, visit ActiveState (the maintainers of the Tcl language) to download it. You can determine if Tcl/Tk is installed on your system by executing the command `which wish`.

Please visit our website for more information on running and editing source code.

# 5 Oscilloscope, Basic Controls

The oscilloscope *Graphical User Interface* (*GUI*) is shown in figure 3. The exact design of the GUI is subject to change and development as features are added, but figure 3 will provide some guidance.

Many of the scope controls are similar in function to those of the classic analog oscilloscope. Other controls are unique to the CGR-101. They provide functions - such as pre-trigger display - that are only possible in a digital oscilloscope.

The scope controls divide into various groups: *amplitude*, *timebase*, *triggering* and *display*.

## 5.1 Amplitude

There are two input channels, A and B, each with identical controls.

- **Disable** Click on this button to disable display of the corresponding waveform. This is useful to reduce clutter on the display.
- **Scale** Sets the vertical scale factor of the display between 50mV per division and 5 volts per division in the traditional oscilloscope 1:2:5 sequence. The amplitude may be read off the display by measuring the number of divisions and multiplying by the scale factor. Alternatively, the *display amplitude cursors* may be used (see *display* below.)

  Changes to the *scale* control adjust the sensitivity of the front-end .of the oscilloscope, adjust the preamp gain and change the software scale factor.

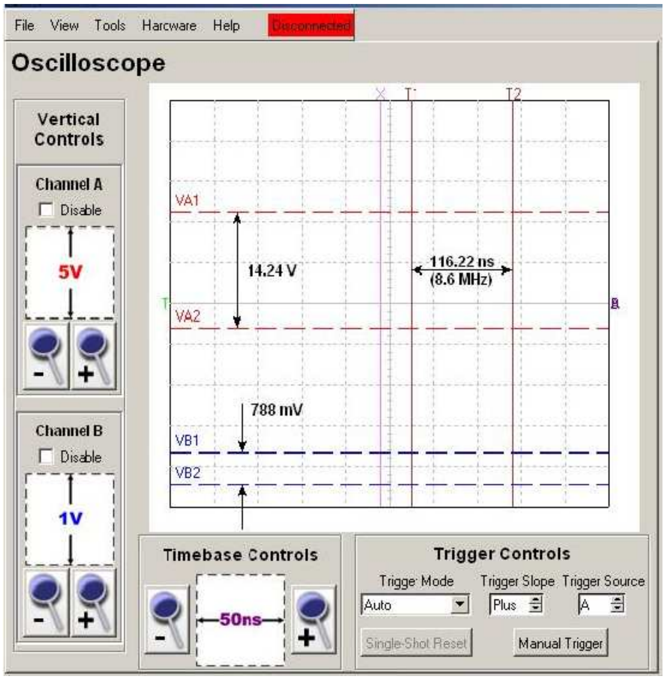



Figure 3: Oscilloscope GUI

### 5.2 Timebase

The *Main Time Base* (*MTB*) can be varied between 50nsec/division and 100msec/division in the traditional oscilloscope 1:2:5 sequence.

### 5.3 Triggering

In order to present a stable waveform display, each display update must start at the same point on a waveform. The trigger functions determine how that *trigger point* on the waveform is selected.

The trigger controls must meet certain requirements in order to generate trigger signals for waveform capture. If these controls are not set properly, it is possible that the scope will not capture waveforms and the display will appear to be frozen. Alternatively, the display may appear to jump between captures, without a stable waveform display.

- **Trigger Time** The *trigger time* is marked by a vertical cursor with an **X** symbol at the top. The trigger point may be dragged left or right to reveal more or less of the waveform preceeding or following the trigger point. This ability to view the waveform preceeding the trigger point is one of the advantages of a digital oscilloscope.

- **Trigger Level** The *trigger level* is marked by a horizontal cursor with a **T** symbol at the leftmost edge of the screen. The trigger level may be dragged vertically to set the amplitude on a waveform that establishes the trigger point. In order to cause triggering, the trigger level cursor must be positioned within the amplitude of the triggering waveform.

  As the trigger is dragged, an accompanying readout displays the trigger amplitude in volts.

- **Trigger Mode: Auto/Normal/Single**
  - In the **Normal** position, the scope hardware *must* get a proper trigger signal in order to display a new waveform. Without a trigger signal, the display simply waits. (The **Manual Trigger** button can be used to force a trigger event, that is, display one capture.)
  - In the **Auto** position, if there is a trigger signal, the scope uses the trigger signal to synchronize waveform capture. If there is no trigger signal the scope hardware waits for a period of time and then generates a trigger signal internally. That way, there are periodic updates to the waveform display, even if triggering is not occurring from an input waveform.

    In general, the most convenient position is **Auto**. However, there are two situations where **Normal** triggering is necessary:
    - For very low frequency waveforms, the trigger signals occur infrequently. If **Auto** triggering is enabled, the scope will decide that trigger signals are not present and generate them internally. This is not what is wanted: the scope should wait for a waveform trigger signal.
    - If the scope is being used to capture a single-shot event, then it should not trigger itself: it should wait for a waveform trigger signal, regardless of how long it takes for that trigger signal to occur.
  - In the **Single** position, the scope waits for a trigger signal. (This is known as the **Armed** state.) When a trigger signal occurs, the software captures and displays that waveform and disables further triggers. The **Single-Shot Reset** button clears the display and returns the scope to the **Armed** state.
- **Trigger Slope** The *trigger slope* control selects a positive-going or negative-going slope at the trigger point. This allows one to trigger off the leading or trailing edge of a positive pulse waveform, for example.
- **Trigger Source** The trigger signal may be derived from the Channel A waveform or the Channel B waveform. Generally, it is easier to obtain a stable trigger signal from the simpler of the two waveforms.
- **Manual Trigger** Actuating this button generates a trigger signal. This is sometimes useful to cause the scope to capture one waveform.
- **Trigger Level (Readout)** This display shows the amplitude of the trigger level setting.
- **Display Lag** For low frequency waveforms displayed at slow timebase settings, there is a noticeable lag between display updates. This occurs because each display is shipped back to the host as a complete waveform and it takes time for the 1k memory to fill with data.

## 5.4 Display

Refer to figure 3 on page 12.

- **Vertical Position** A waveform may be moved in vertical position. At startup, the **A** and **B** cursors - which are the zero reference for the channel – are placed at centre screen. Using the mouse, drag the letter **A** or **B** at the right edge of the display area up or down to change the vertical position of the corresponding waveform.
- **Time Cursors** It is possible to enable and disable various time and amplitude cursors.

  Right click in the display area. A menu appears:

| | |
|---|---|
| **Toggle Time Cursors** | Left Click to enable and disable vertical cursor lines that mark the time between two locations on screen. |
| **Toggle Channel A (or B) Cursors** | Left Click to enable and disable horizontal cursor lines that mark the amplitude between two locations on screen. |
| **Grid** | Left Click to select the appearance of the graticule grid in the display area. |
| **Auto Measure** | Follow the menu heirarchy to select auto-measurement of average amplitude and/or frequency on Channel A or B |

Notice that the readout text can be dragged to different positions on the screen. This is useful when setting up a screen display for capture in a document.

# 6 Additional Features

Section 5 described the basic controls of the oscilloscope. In this section we describe additional controls for more specialized measurements.

## 6.1 Screen Capture

It is extremely useful to be able to capture oscilloscope screen shots. One or more screen shots may be used to document a particular measurement situation as a record of the measurement or to capture the result for a larger document.

The oscilloscope main window or any of its subsidiary windows (measurements panel, histogram, spectrum display) can be captured to a JPEG image file. [4].

Selecting the menu item `Tools -> Screen Capture (jpg)` brings up a small dialog.

Clicking on one of the selections then brings up the standard `File Save` menu, and the file may be named and saved where required.

The screen shot can only be saved in JPEG format. If it is needed in another format, load it into a drawing or image processing program and save it in that format.

Under the Windows operating system, you can use `Paint` for this purpose.

Under Linux, the program `Imagemagick` can convert to a variety of image formats.

## 6.2 Export to Postscript

Section 6.1 on page 14 described the oscilloscope the `Screen Capture` features. These features support the capture of any oscilloscope screen window to a JPEG image file.

It is also possible to capture the waveform display area of the main oscilloscope display as a postscript file.

Select the menu item `Tools -> Export to Postscript (PS) File` brings up the standard `File Save` menu, and the file may be named and saved where required.

What are the relative advantages of JPEG screen capture vs exporting the display area to a Postscript file?

- `Screen Capture` grabs everything inside the selected window. This may be useful if you need to document the control settings. Furthermore, `Screen Capture` can capture any of the scope windows.

  `Export to Postscript` grabs only the waveform display area on the main oscilloscope screen.

- Certain writing tools may require image format to be postscript. It is entirely possible to convert a JPEG image to Postscript, but the file size is quite large. If you need the waveform in Postscript format and file size is an issue, then you might be better off capturing just the waveform area in Postscript format.

[4] Acknowledgement: Special thanks to John Foster who helped develop the screen capture code.

- Both the JPEG capture and Postscript capture are in colour.

## 6.3 Scope Offset Calibration

The CircuitGear CGR-101 includes provision for measuring and substracting any DC offset into the vertical channels. This calibration is performed when units are tested, so in general it should not be necessary to do this calibration repeatedly.

The calibration panel is started from the `Tools` menu and appears as in figure 4.

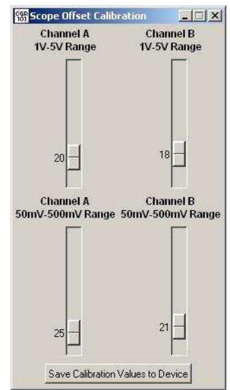



Figure 4: Scope Offset Calibration Panel

To operate the calibration procedure:

- Place the scope in `Auto` trigger mode.
- Remove any inputs from scope channel A.
- Set the Channel A vertical gain to 1V/div.
- Disable Channel B.
- Drag the Channel A trace up and down. You will see a dashed line separately from the Channel A trace for a brief instant before the scope triggers. This red dashed line is the zero reference. Move the zero reference for channel A vertically to the centre of the screen.
- Adjust the offset adjustment slider `Channel A, 1V-5V Range` control until the channel A trace is over top of the red dashed line.
- Set the Channel A vertical gain to 500mV/div.
- Adjust the slider `Channel A, 50mV-500mV Range` control until the channel A trace is over top of the red dashed line.
- That completes the offset adjustment for Channel A.
- Repeat the procedure for Channel B.
- Click on `Save Calibration Values to Device`. This transfers the offset values to the EEPROM memory in the hardware.
- Close the Offset Calibration display window.

## 6.4 Scope Amplitude Calibration

The `Scope Vertical Calibration` panel allows one to set the vertical scale factor to a precisely correct value. This calibration is performed when units are tested, so in general it should not be necessary to do this calibration repeatedly.

The calibration panel is started from the `Tools` menu and appears as in figure 5.

To operate the calibration procedure:

- Important. Check the offsets and re-calibrate them if necessary.
- Disable Channel B.
- Set the Channel A vertical gain to 1V/div.
- Move the trace vertically until it coincides with one of the display major divisions.
- Apply a known voltage that is within the display capability of the channel setting.
- Adjust the amplitude adjustment slider `Channel A, High Range` until the trace deflection is exactly correct.
- Set the Channel A vertical gain to 500mV/div.
- Apply a known voltage that is within the display capability of the channel setting.
- Adjust the slider `Channel A, Low Range` control until thetrace deflection is exactly correct.
- That completes the offset adjustment for Channel A.
- Repeat the procedure for Channel B.
- Click on `Save Calibration Values to Device`. This transfers the offset values to the EEPROM memory in the hardware. This does not affect the factory default settings.
- If you decide that you wish to return to the factory default settings, click on `Restore Defaults`. The hardware will retrieve the factory settings from the EEPROM memory in the hardware.

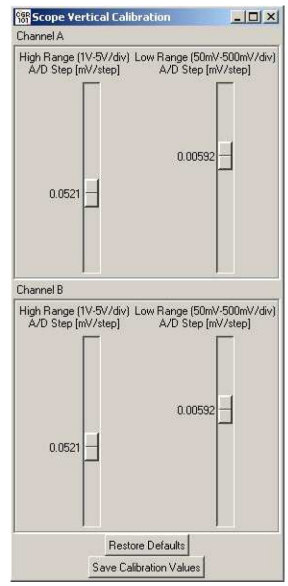



Figure 5: Scope Vertical Calibration Panel

## 6.5 Vector Network Analyser

An electrical network, such as a lowpass filter, is characterized by its amplitude and phase response. The amplitude response is a plot against frequency of the *gain* of the network: the ratio of output signal amplitude to input signal amplitude. The phase response is a plot of the *phase* of the network: the difference between the output phase and input phase. The test signal is a sine wave that is swept over a range of frequencies, taking care not to overload the network.

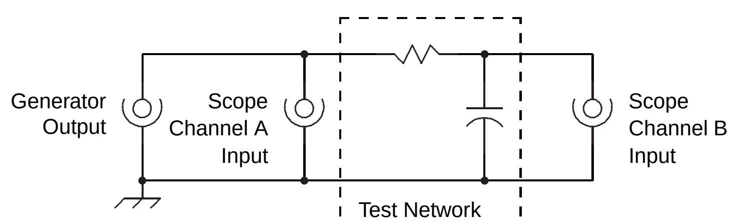



Figure 7: Network Analyser Wiring

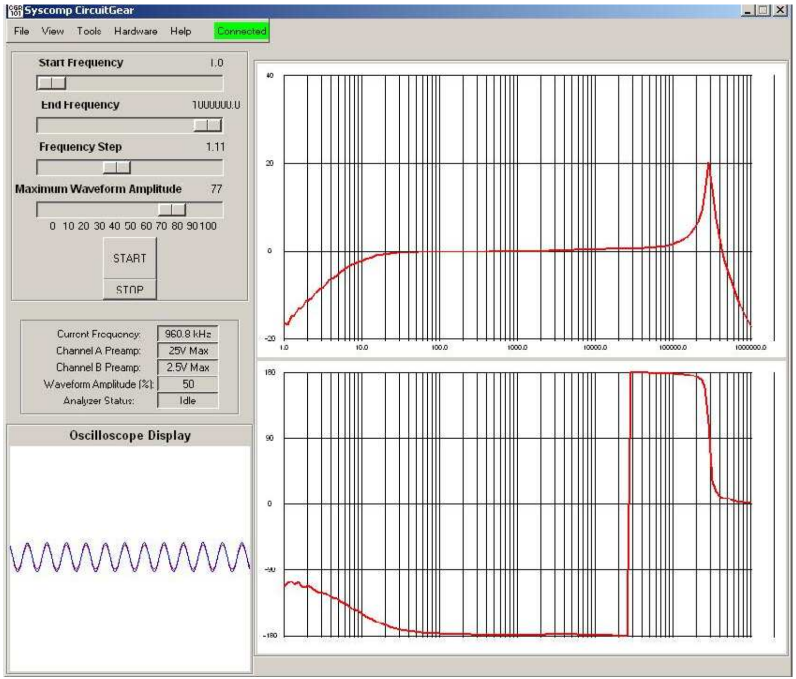



Figure 6: Vector Network Analyser

As part of their AC Circuits lab, electrical engineering students are required to plot these response curves by hand. This is a very tedious process. Each point in the plot requires setting the generator frequency, reading the input signal amplitude, reading the output signal amplitude, reading the output signal phase, and plotting the result. The CGR-101 vector network analyser does this automatically over a range of 1Hz to 1MHz, or some part thereof. This makes it practical to explore the effect of changing component values. For example, if the resistor or capacitor value in an RC lowpass filter is changed, one can immediately determine the effect on frequency and phase response.

The CGR-101 has two principal modes: as an oscilloscope and signal generator (with digital input-output), and as a network analyser. To change between them, select `Hardware -> Network Analyser Mode` or `Hardware -> Circuit Gear Mode`.

Figure 6 shows a screen shot of a VNA plot of a single-pole RC lowpass filter. Figure 7 shows how the test circuit must be wired. Notice that there is no internal connection between the generator output and the channel A input.

The display is the amplitude and phase response of a small 1:1 audio transformer. Notice that the amplitude response rolls off below 20Hz and exhibits very substantial peaking around 300kHz. (This shows up as ringing in the square wave time-domain response.) The transformer midband phase response because the transformer primary and secondary are deliberately wired out of phase. The jump in the phase response from $-180°$ to $+180°$ is correct: plus or minus 180 degrees is the same physical result.

- Two slide controls set the start and finish frequency
- A third slider sets the *frequency step*. The frequency step is the amount by which a frequency is to increase to create each new measurement frequency. Or, put another way, it's the ratio between two adjacent

measurement frequencies. The frequency step multiplies by a factor (rather than adding a constant factor) because the frequency scale is logarithmic.

- A fourth slider sets the signal amplitude.
- The *Oscilloscope Display* window shows the input and output waveforms, which should be sine waves. If the signals show clipping, reduce the signal amplitude.
- The *Start* button initiates a frequency sweep. Sweeping is slow at low frequencies and speeds up as the frequency increases.
- The display autoscales, that is, tries to select a scale that just fills the display.
- The amplitude dynamic range of the VNA is in excess of 50db. The VNA automatically adjusts the input signal attenuators of the oscilloscope section to obtain the best possible signal-noise ratio without clipping. It also uses the full 10 bit range of the oscilloscope A/D converters. In the frequency response plot of figure 6, the amplitude and phase plots become erratic at high frequencies. This occurs because the output signal from the low pass filter is extremely small in that region.
- The amplitude and phase information may be saved to a `.csv` format data file (which can be loaded into Matlab or a spreadsheet). With the VNA operational, select `Tools -> Export Waveform (CSV)`.

More information on the operation and theory of the vector network analyser is in the Syscomp Application Note *A Software-Based Network Analyser* at `http://www.syscompdesign.com/na-theory.pdf`.

## 6.6 XY Mode

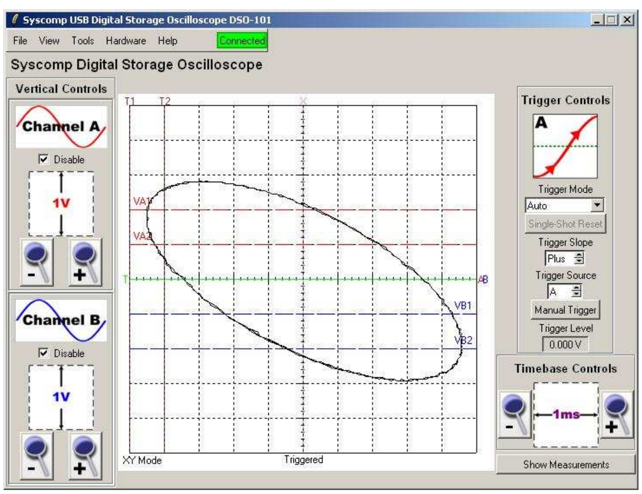



Figure 8: XY Display Mode

The usual oscilloscope display shows a plot of the two signal amplitudes, voltage on Channel A and Channel B, vs time. It is also possible to plot the two voltages against each other: Channel A as the X axis and channel B as the Y axis.

Select `View -> XY Mode` to enable the XY display.

The CGR-101 can simultaneously display both the XY display and the conventional voltage-time waveforms, which is useful in teaching situations and for debugging purposes.

### Lissajous Figures

When the two signals are sine waves of the same frequency, with a phase shift between them, the display is as shown in figure 8. This type of looping display is known as a *lissajous figure*.

If the two frequencies different but integer multiples of each other, then the lissajous figure will have multiple nodes. In the early days of oscilloscopes, lissajous figures were used in this manner for frequency measurement. The vertical amplifiers of the day could not work at high frequencies, so the signals were applied directly to the deflection plates of the cathode ray tube. The lissajous figure gave an indication of frequency ratio and relative phase.

To form a complete lissajous loop, the timebase setting must be such that both waveforms show at least one complete cycle.

### Magnitude Measurement

If the two signals are exactly in phase, the XY plot is a straight line. If the two signals are of exactly the same magnitude, the angle of the straight line is $45°$. If the magnitudes are different, then the line is at some other angle. This is a sensitive method of comparing the amplitude of two waveforms, which need not be sine waves. Any waveshape should function in this measurement.

### General Purpose Plotting

The XY Mode display may be used for a variety of applications where a plot of some kind is required.

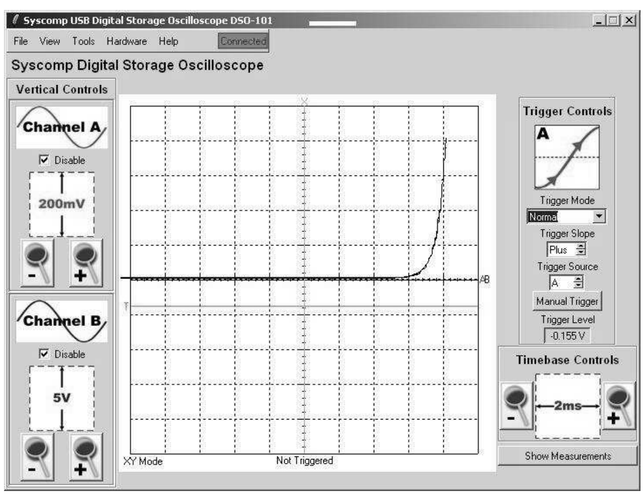



Figure 9: Diode Voltage-Current Characteristic

Figure 9 shows an example. The CGR-101 has been configured to plot the voltage-current curve of a silicon diode. Notice that the diode threshold is around 0.6 volts. The vertical scale is voltage measured across a current sensing resistance, equivalent to 10mA per division.

## 6.7 Spectrum Display

A complex waveform may be treated as being composed of a number of sinusoid waveforms. These sinusoids are of various phases, frequencies and amplitudes. The description of the magnitude, phase and frequency of these various waves is known as the *spectrum* of the signal, by analogy with the spectrum of light.

*Spectrum Analysis* or *Fourier Analysis* is the process of analysing some time-domain waveform to find its spectrum. We also say that the time domain waveform is converted into a frequency spectrum by means of the *Fourier transform*.

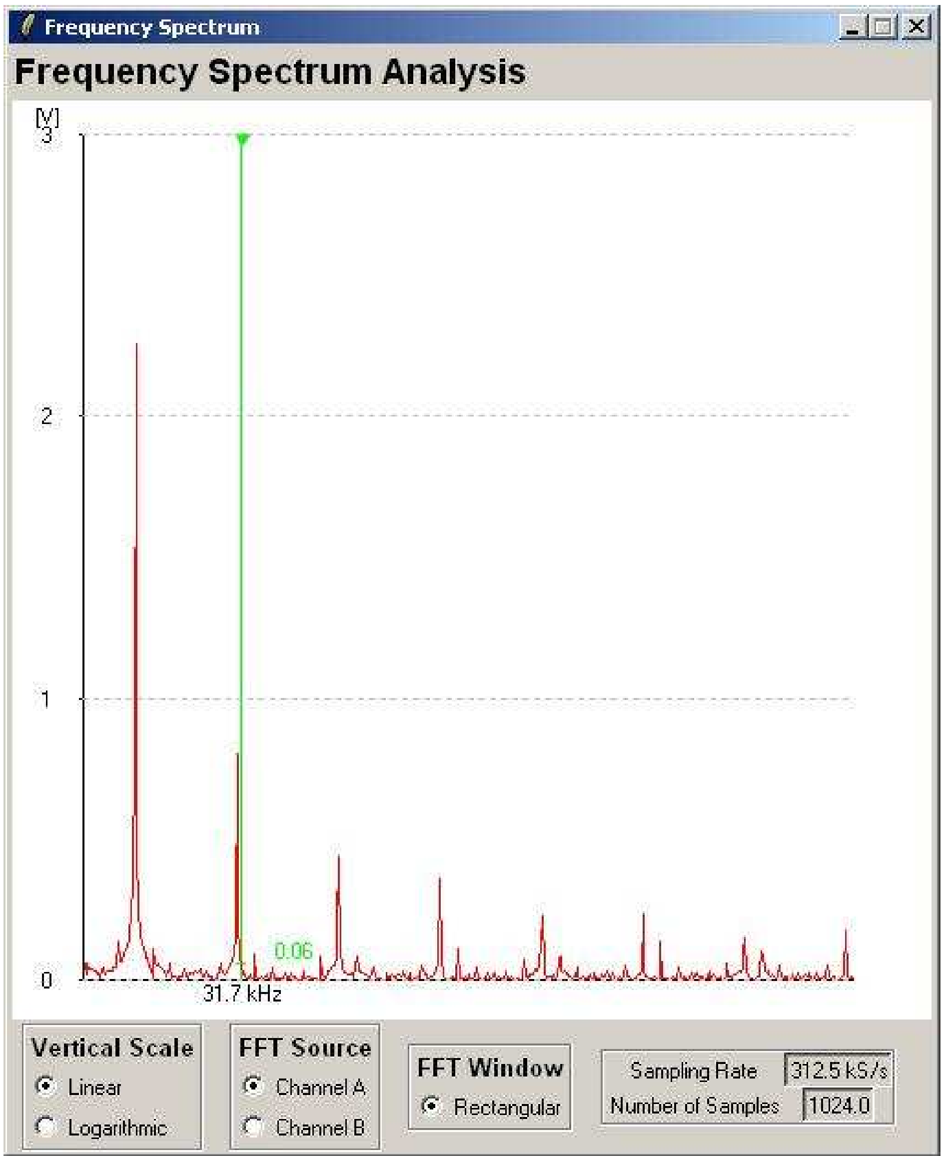



Figure 10: Square Wave Spectrum

Clicking on `Tools -> Spectrum Analysis` brings up the spectrum analysis display of figure 10. The displayed spectrum in this image is a 10kHz square wave.

The theory of Fourier Analysis shows that a square wave is composed of a fundamental of magnitude E volts at frequency $f$ (10kHz in this case) with the following harmonics: $E/3$ magnitude at frequency $3f$, $E/5$ magnitude at frequency $5f$, $E/7$ magnitude at frequency $7f$, and so on. The spectrum display shows this pattern.

Each vertical line represents one of these frequency components. The height of the line is proportional to the magnitude of that particular component. The horizontal axis is a linear scale of frequency, with zero frequency (DC) at the left edge.

The vertical cursor can be dragged horizontally to determine the frequency and magnitude of a component of the spectrum.

The spectrum display and main waveform display are active at the same time, allowing one to simultaneously observe a waveform in the time domain and frequency domain.

### Interpreting the Display

Because of fundamental limitations in a sampled-data system, it is possible for the display to be misleading. Here are some important points to keep in mind when using spectrum analysis based on digital methods:

- The `Effective Sampling Rate` is shown in a readout at the bottom right corner of the spectrum display. This is important: the sampling rate must be at least twice the frequencies being analysed to avoid aliasing. Put another way, there must not be frequency components above the Nyquist rate, which is half the sampling rate. In the example shown in figure 10, the sample rate is 200kHz. The frequency components range from 10kHz to 90kHz, below the Nyquist rate of 100kHz[5].

- A *sweeping type* analog spectrum analyser moves a bandpass filter across a range of frequencies to determine the spectrum. A digital spectrum analyser such as this one divides up the frequency range into a number of *bins* and then measures the energy in those bins.

  The main oscilloscope display of the CGR-101 is 500 points. This is padded to 512 points[6] by appending zeros to the waveform record. As a result, there are 256 frequency bins when using the main oscilloscope display.

  The centre frequency of each of these bins may not coincide exactly with the frequency components present. If that is the case, then the displayed amplitude will be incorrect and should only be regarded as an approximation of the true situation.

- As the readout cursor is dragged higher in frequency it jumps from bin to bin, reading out the centre frequency of each bin. A given frequency component may not be centred in its bin, so the frequency readout will be only approximate. For example, in figure 10, the square wave frequency (generated by a Syscomp WGM-101 waveform generator) is at a frequency 10kHz to within a fraction of a Hz. The 9th harmonic is at 90kHz. The spectrum display readout puts the 9th harmonic at 90234 Hz, which is only approximately correct.

### Frequency Scale, Bin Spacing

It is sometimes useful to be able to determine the resolution of the frequency axis. Each frequency bin has a width $\Delta f = 1/T$ Hz where $T$ is the length of the data record in seconds. If there are $N$ points in the data record, then $N/2$ points are displayed as positive frequency. (The other $N/2$ points are redundant.)

[5] Harmonics of the square wave extend to much higher frequencies but we assume their amplitude is small enough to be ignored.

[6] The FFT routine requires that the number of points be a power of 2.

**Example**

Determine the frequency resolution (bin spacing) for the case of the display of figure 10.

**Solution**

The sample rate is 200kS/sec. The sample interval $\Delta T$ is the reciprocal of this:

$$\begin{aligned}\Delta T &= \frac{1}{200 \times 10^3} \\ &= 5\ \mu\text{Sec}\end{aligned}$$

The number of points $N$ in the data record is 512 points, so the total length of the data record is:

$$\begin{aligned}T &= N\Delta T \\ &= 2.56\ \text{mSec}\end{aligned}$$

The resolution $\Delta f$ is the reciprocal of the record length:

$$\begin{aligned}\Delta F &= \frac{1}{T} \\ &= 390.625\ \text{Hz}\end{aligned}$$

For example, the 7th harmonic should appear at $f_7 = 70$kHz. The spectrum display actually puts it at 70313 Hz, which is bin 180.

$$\begin{aligned}F_7 &= 180 \times 390.625 \\ &= 70313\ \text{Hz}\end{aligned}$$

The maximum frequency $f_{max}$ on the display occurs at bin 256:

$$\begin{aligned}f_{max} &= 256 \times 390.625 \\ &= 100\ \text{kHz}\end{aligned}$$

**Windows**

Window or *weighting* functions are often applied to the time-sequence data prior to transformation into the frequency domain. All window functions taper the data down to zero at its ends. Then the discontinuity caused by a finite record length does not affect the shape of the transform.

The choice of window function depends on the application, and all window functions are a compromise of some sort. For example, some window functions provide very accurate amplitude readings, others are best for separating closely spaced frequencies. A collection of window functions is shown at `http://en.wikipedia.org/wiki/Window_function`.

The current spectrum analysis routines have only one window function, the *rectangular window*. This is in fact a non-window, it does not shape the time function, all points on the time record are weighted equally. Other weighting functions may be added in future versions of the software.

**Applications**

Spectrum analysis has a number of applications in electronics and mechanical engineering:

- A pure tone has no harmonics and will show up on a spectrum display as one single vertical line. Distortion of a sine wave will create additional harmonics. Consequently, a measure of the magnitude of the harmonics is a measure of the magnitude of the *harmonic distortion*.
- In a distortion-free (linear) system, two separate input tones (single frequencies) will emerge as the same two tones at the output. If the system is distorting (non-linear), then the system will generate other tones at the sum and difference frequencies of the input signals. A measure of these extra signals is a measure of the *intermodulation distortion*.
- The existence of certain frequencies in a signal may give some clues as to its source. For example, if a signal contains the power line frequency (eg, 60Hz in North America, 50Hz for the UK), then it is probably picking up interference from the AC power line.
- Power systems frequently manipulate waveforms by chopping them or combining them with other signals. Spectrum analysis allows one to measure the harmonic content of a signal, which may be specified as a requirement.
- The analysis of a mechanical system for resonances can be done by driving the system with a wide-band excitation signal, an impulse hammer blow or random noise from a shaker. Microphones or accelerometers convert the mechanical vibration of the system to an electrical signal. The spectrum analysis of this signal indicates the mechanical resonances in the structure.
- The extraction of signals from noise may require some knowledge of the spectrum of the signal and the noise.
- It is useful to see the spectrum diagram for modulation and other signal manipulations.

Further information on spectrum analysis is in the paper *Introduction to Digital Spectrum Analysis*, which is on the Syscomp web site.

## 6.8 Data Record

Oscilloscope waveform data recording is started with the `Data Recorder` control panel, found as menu item `Tools -> Data Recorder`, figure 11.

Choose a suitable directory into which the data should be stored. Change to that directory in the `Choose Export Directory` entry widget.

Set the `Log Interval` to whatever is required. Click on `Start Logging` and the files will be generated. The `Log Count` shows how many records have been stored.

The data recorder creates a separate file for each data record. A typical directory record is shown in figure 11. Each file name is in the `month/day/year/hour/minute/second` format.

Click on `Stop Logging` to halt data logging.

The captured data is in `{.csv}` (comma-separated values) format, which can be directly loaded into a spreadsheet such as Open-Office Calc or Microsoft Excel. An exerpt of the file is shown below. There are 512 data entries in the file.

```
"Time [s],Channel A [V],Channel B [V]"
0.0,-0.1042,0.0
2.56e-005,-0.1563,-0.0521
5.12e-005,-0.1563,-0.0521
7.68e-005,-0.1042,-0.0521
0.0001024,-0.1042,0.0
0.000128,-0.1042,0.0
0.0001536,-0.1042,0.0
0.0001792,-0.1042,-0.0521
0.0002048,-0.1042,0.0
0.0002304,-0.1042,0.0
0.000256,-0.1042,0.0
0.0002816,-0.1042,-0.0521
```

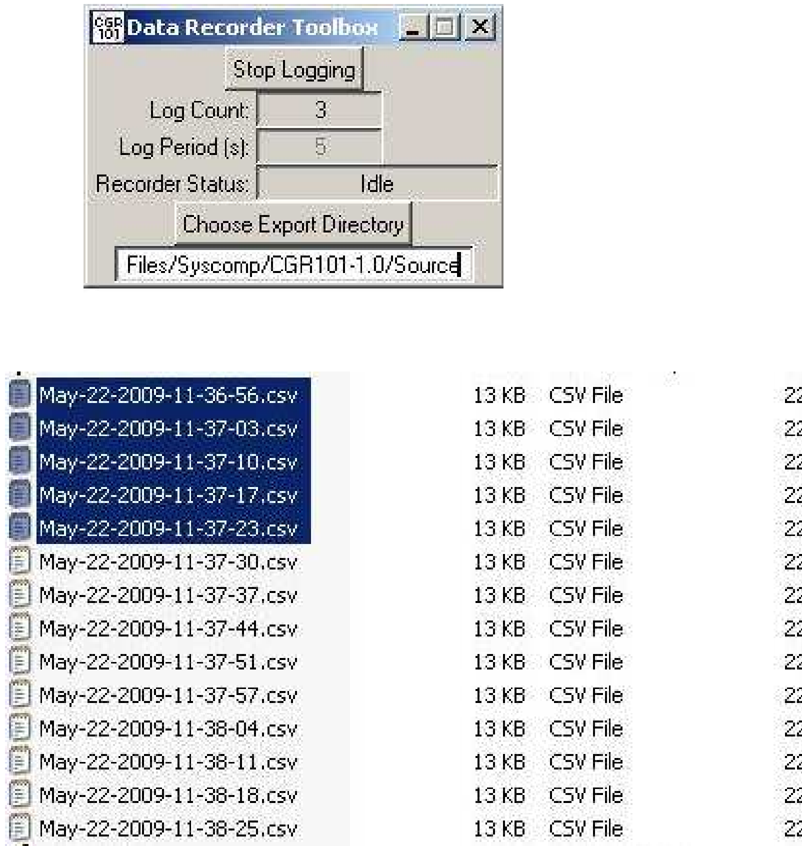
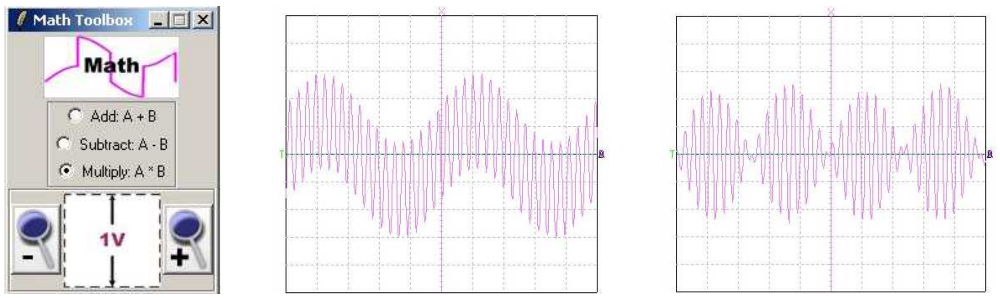



Figure 11: Data Recorder Panel

## 6.9 Waveform Math




(a) Toolbox (b) Addition (c) Multiplication

Figure 12: Waveform Math Tool

The math toolbox generates a waveform display that is some function of the waveforms on Channel A and Channel B.

Clicking on `Tools -> Math Toolbox` brings up the display of figure 12(a). At this time, there are three math functions: `Add: A+B`, `Subtract: A-B` and `Multiply: A` $\times$ `B`. The software does a point-by-point math operation on the input waveforms and generates a display of the result on the main display.

The vertical scale factor control (2 volts/div in figure 12) can be used to adjust the displayed magnitude of the result.

The display of figure 12(b) shows the addition of two sine waves. Figure 12(b) shows multiplication. In the case of multiplication where one waveform is proportional to voltage and the other to current, then the product is a waveform of power.

Subtraction of two waveforms is useful in taking a differential voltage measurement across some component, when neither terminal of the component is grounded. The input voltages must be within the amplitude capability of their respective channels.

It is a relatively simple matter to add other functions to the math toolbox. (see the source code file `math.tcl`).

# 7 Aliasing

The oscilloscope is a *sampled-data-system*. It works by taking a series of samples of the input waveform and displaying them. However, when the signal contains high frequency components compared to the sampling rate, the display may be incorrect. In theory, at least two samples per cycle of the highest frequency present in the waveform are required to reconstruct the waveform correctly.

Some sampled-data-systems have a constant sampling rate. For example, audio is typically sampled at 44.1k samples per second. In that situation, usual practice is to incorporate a low-pass filter such that frequencies above 22 kHz are prevented from entering the system[7]

Most – if not all – digital oscilloscopes do not incorporate an anti-aliasing filter. The sample rate of a digital scope varies over a wide range of frequencies, and so the cutoff frequency of the anti-aliasing filter would have to do so as well. Combined with the bandwidth requirement, that is a difficult technical challenge. Instead, the

[7] In practice, the lowpass cutoff frequency is set to somewhat less than half the sampling frequency to allow for the finite rolloff rate of the filter.

oscilloscope relies on the operator to recognize when aliasing is occurring and increase the sample rate until the effect disappears.

A useful strategy in measuring an unknown waveform is to approach it from a high sampling rate (aka time-base setting) and reduce the setting until a readable display appears. It is also required of the operator to know (approximately) the frequency of the waveform that is being observed. That is often the case.

A useful rule of thumb is this: the display must contain about 10 samples per cycle of the waveform to reconstruct it. For this oscilloscope the maximum sampling rate is 20MSamples/second, so it can usefully observe frequencies up to about 2MHz. The analog bandwidth has been designed to be 2MHz to meet this requirement.

# 8 Waveform Generator

The waveform generator controls are shown in figure 13.

The *Amplitude* control is calibrated from 0 to 100%.

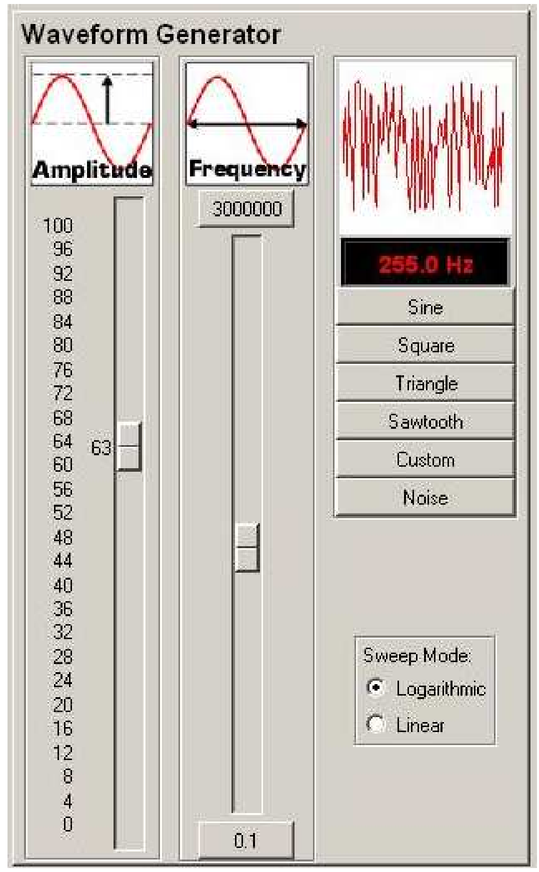



Figure 13: Generator Controls

## 8.1 Setting Frequency

The frequency control adjusts frequency between the limits shown in the two buttons at the top and bottom of the frequency slider. In the default, these limits are 0.1Hz to 3MHz. The frequency resolution is 0.1Hz. The accuracy is based on a crystal clock. A readout shows the current frequency to a resolution and accuracy of 0.1Hz.

You can set the frequency by moving the slider. Alternatively, left-click on the frequency display and enter a frequency value in the pop-up dialog.

## 8.2 Setting Sweep Limits

To change one of these slider limits, left-click on it. An entry widget appears, prompting for a new maximum or minimum frequency. Enter a new value and left-click on OK or hit `<carriage-return>`. The new value appears above or below the frequency slider.

For example, if you are sweeping an audio device, you can set the maximum and minimum frequencies to 20,000 and 20Hz. Then the full scale movement of the slider applies to that range.

As another example if you are investigating the frequency response of a 3kHz narrow-band active filter, you could set the frequency range to 3050Hz maximum and 2950Hz minimum. Then the adjustment range of the generator is 100Hz, giving effective fine-grain control of frequency.

## 8.3 Sweep Mode

The control characteristic of the frequency slider can be set to *Logarithmic* or *Linear*. The Logarithmic control increases the frequency in an exponential fashion as it is increased, which is the most convenient characteristic is most situations. In Logarithmic Mode, the physical mid-point of the scale corresponds to about 700Hz.

In Linear Mode, the control characteristic is linear and the mid-point of the control is 1.5MHz. In effect, this assigns most of the physical movement to high frequencies.

## 8.4 Manual and Automatic Sweep

The CircuitGear GUI provides frequency control of the generator with a manual frequency control. Automatic sweep is provided with a separate program: the VNA (Vector Network Analyser), aka Bode Plotter software, which operates the generator to make a sweep and the oscilloscope to plot the response of some device or network.

## 8.5 Waveform Selection

There are six possible waveform selections. Selection of a waveform (except *Noise*) causes that waveform data to be downloaded into the CircuitGear hardware. Each waveform data file consists of 256 data points. Each data point has a value between 0 and 255. The files for *Sine*, *Square*, *Triangle* and *Ramp* are supplied with the GUI software. (If you download the source code these files are named *sine.dat* and so forth.)

Selecting the *Custom* waveform pops up a file selection box so you can select any waveform. A waveform data file can be constructed manually, using a programming language (eg, Visual Basic) or from a spreadsheet (eg, Open Office *calc*). The GUI-based program *Wavemaker*, available at `http://www.syscompdesign.com/download.htm` can be used to draw a waveform and convert that drawing into a suitable data file.

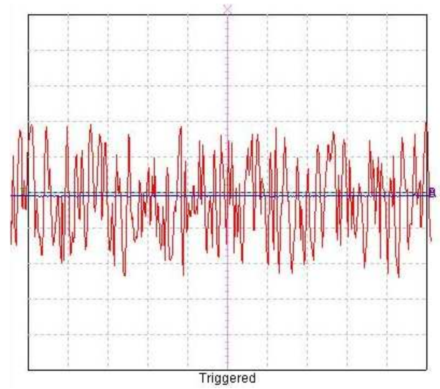



Figure 14: Noise Waveform

There is a 1 to 2 second download delay after selecting a waveform before the generator begins producing the waveform. During that time, other waveform selection buttons are locked out to prevent *button mashing*.

## 8.6 Noise

CircuitGear can generate a random noise signal (figure 14), which is useful in transfer function measurents by correlation and acoustical testing. The output spectrum is *white*, that is, equal energy per hertz bandwidth [8].

The noise is generated by a 32-bit shift register, configured to give a pseudo-random bit sequence, and clocked at 100MHz. (See `http://en.wikipedia.org/wiki/Linear_feedback_shift_register`.) Every 8 shift pulses, an 8 bit sample is extracted and converted to analog form. Consequently the noise rate is 12.5Msamples/second. The shift register sequence length is $2^{32} - 1$ shifts before repeating, which gives a repetition period of 42 seconds.

The amplitude spectrum is actually a $\sin(x)/x$ function, but for practical purposes the spectrum is flat to 2MHz.

[8] For acoustical testing, you will probably need a *pink* noise spectrum, which rolls off the amplitude at 3db/octave, a 1/f characteristic.. It is also adviseable to limit the spectrum to the audio range to avoid damage to amplifiers and tweeter loudspeakers.

# 9 Digital Input-Output Section

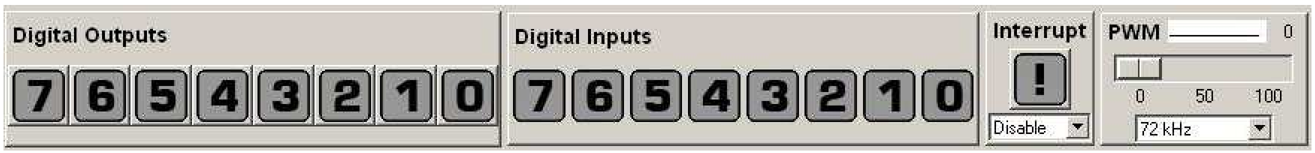



Figure 15: Digital Controls

The CircuitGear digital controls are shown in figure 15. These input and output lines can be operated from the GUI controls in figure 15 or they may be controlled by software that communicates with the CircuitGear API (applications program interface, section 14.2 on page 52).

## 9.1 8 Bit Digital Output

Clicking on an individual bit causes that bit to illuminate on the GUI and the corresponding output line to go into the high `HIGH` state. Clicking again causes the bit to extinguish and the corresponding output line to go `LOW`.

The available current to a USB device is 500mA maximum, and this current must operate the oscilloscope and signal generator as well as the digital circuitry. Consequently the digital drive current is very limited: a few milliamps per output. Load devices such as high-current LEDs or DC motors will require their own power supply.

## 9.2 8 Bit Digital Input

The GUI digital input indicators illuminate when the corresponding input level is a logic `HIGH`. The logic levels may correspond to 3V HC logic or 5V HC logic. Great care should be taken not to exceed 5 volts on any input. Inputs are buffered but all devices are surface-mount soldered, so they are not trivial to replace.

## 9.3 Interrupt

The interrupt is a 3V or 5V HC compatable logic input. A dropdown menu below the interrupt indicator/button establishes the mode of operation, one of *Disabled*, *Rising Edge*, *Falling Edge*, *High Level*, *Low Level*.

When the specified type of interrupt occurs, the `!` indicator illuminates. Left-click on the indicator to clear the interrupt.

## 9.4 PWM: Pulse Width Modulated Waveform

The PWM output is a 5 volt pulse. The duty cycle is continuously adjustable with the slider, over a range of 0% to 100%. The output frequency is set from the drop-down menu below the slider, over a range of 72kHz to 35Hz.

Again, the output drive current should be limited to a few milliamperes of current.

## 9.5 External Trigger

When the `Trigger Mode` control on the GUI is set to `External`, a positive-going transition on this pin causes the scope to trigger. To test external trigger:

1. Set the trigger mode to 'auto'.

2. Connect the output of the generator to channel A of the scope and adjust the generator amplitude so the scope shows a sine wave.

3. On the rear panel, use a jumper cable to connect the `Ext Trig pin to the Digital output 0 pin.`

4. `Under Trigger Mode select External.`

5. `From the GUI, use the mouse to click on Digital Output 0.`

6. `On the 0V to 5V transition on digital output 0, the scope should briefly display 'triggered' and a new sample of the waveform should display. On the 5V to 0V transition, triggering will not occur.`

Edge polarity is not selectable on external triggering, the trigger edge is always positive going.

The trigger input is protected against overvoltages but we do not recommend relying on that. We recommend keeping the trigger signal between $0$ and $+5$volts.

## 9.6 Rear Panel Connector

The rear panel connector provides access to the following signals:

- Digital outputs 0 through 7
- Digital inputs 0 through 7
- Interrupt input
- PWM (Pulse Width Modulated) output
- External trigger input
- Ground

The pinout is shown on the case label of the instrument and in figure 16 below. These functions are described in section 9 above.

## 9.7 Rear Panel Connector Mating Plug

One suitable rear panel connector mating plug is as follows:

```
MODE part # 35-0202-0
20 position, 2 row, 0.1" (2.54mm) contact spacing.
Mates with 20 way 0.050" spacing flat cable.
Polarizing key (Centre bump)
Strain Relief
Active-Tech price: $0.38
```

Available from MODE Electronics, `http://www.mode-elec.com`
Distributed by Active-Tech Electronics in Canada, `http://www.active123.com`

Another suitable connector is this one:

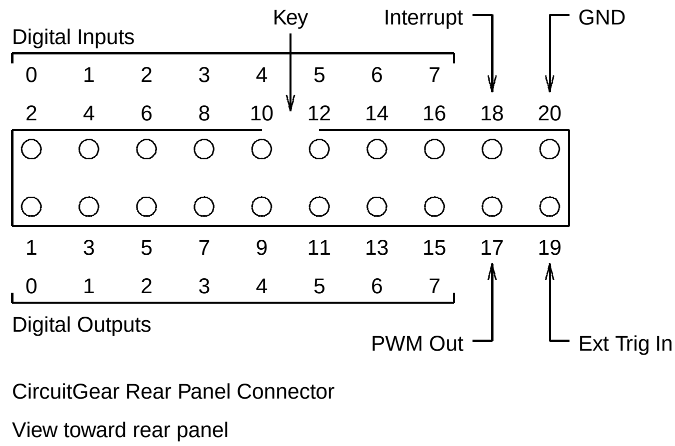



Figure 16: CircuitGear Rear Panel Connector Pinout

```
3M part # D89120-0131HK
Digikey part # MKC20E-ND
20 position, 2 row, 0.1" (2.54mm) contact spacing.
Mates with 20 way 0.050" spacing flat cable.
Polarizing key (Centre bump)
Contacts 10u gold plate
Digikey price: $0.69
```

Mating strain relief for this connector:

```
3M part # D3448-89120
Digikey part # MESR20-ND
```

Available from Digikey, `http://www.digikey.com`

Any flat cable with 0.050" conductor spacing that will mate with IDC (Insulation displacement connector) should be suitable for use with these plugs. For example:

```
3M 3302 Series
Digikey part # MCM-20M-5-ND (5 foot length)
10 colour repeat, clear carrier
Digikey price: $7.50
```

The installation of an IDC connector onto flat cable is described here:
`http://hubbard.engr.scu.edu/embedded/guide/ribbon/`

# 10 Safe Measurement Technique

These notes are included for the benefit of those who are new to using an oscilloscope. The information is not unique to this oscilloscope, but applies to most oscilloscope measurement instruments.

Rather than simply state rules and prohibitions, we explain why certain procedures are dangerous and why some techniques should be avoided. This information is provided for guidance in using the oscilloscope and is not intended to replace proper training in working around high voltage circuits.

In general, this oscilloscope may be used safely to observe signals in low-voltage circuits where the power supply is *floating* from the AC line.

## 10.1 Floating Power Supply

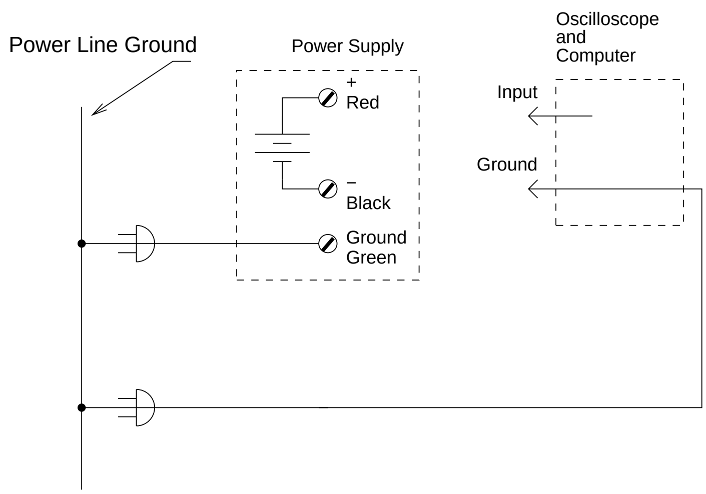



Figure 17: Floating Power Supply

In this context, *floating* power supply is one in which neither terminal is connected to the power line ground.

Consider the circuit shown in figure 17. Like many lab power supplies, the power supply has three terminals: positive, negative and ground. The ground terminal is connected to the third prong on the line cord, which connects to the power line ground wire.

The oscilloscope has two connections: the *input* terminal and *ground* terminal. On the front panel BNC connector, the inner contact is the input, the outer ring is ground. The ground connector finds its way to the AC ground line via the third prong on its line cord.

As shown in figure 17, *either* lead on the oscilloscope can be safely connected to the positive or negative terminal of the power supply. With proper care to avoid short-circuits of the power supply, this is a safe measurement situation.

## 10.2 Grounded Power Supply

Now consider that the negative terminal of the power supply is connected to its `Ground` terminal, as shown in figure 18.

If the ground terminal of the oscilloscope is connected to the ground terminal of the power supply, then the circuit is in no danger and will work properly. However, **if the ground terminal of the oscilloscope is inadvertently connected to the positive terminal of the power supply, then the power supply will be connected to a short circuit. The power supply short circuit will drive current around the ground connections of the power supply and oscilloscope.** Since lab power supplies are usually current limited to less than an ampere of current,

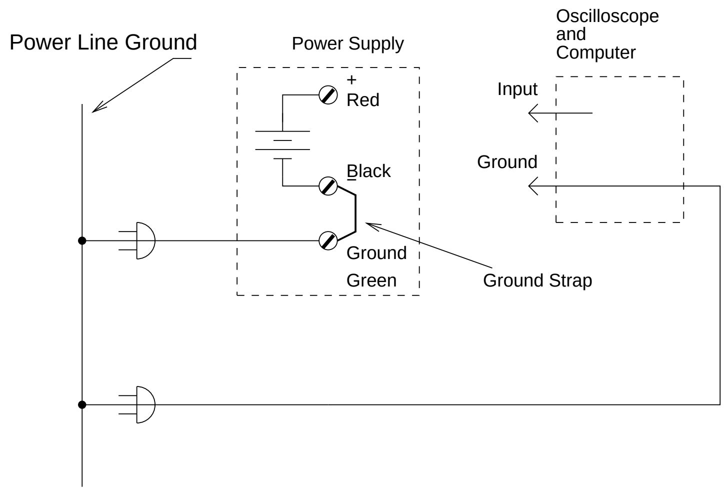



Figure 18: Floating Power Supply

the equipment will likely survive. However, the circuit will not function properly because the power supply is in a short-circuited condition.

To avoid this situation, **do not connect the positive or negative terminal of the lab power supply to the ground terminal. Leave the supply floating.**

## 10.3 Battery and AC Adaptor Power Supplies

If batteries are used to power the circuit under test, the problematic situation of section 10.2 is not likely to occur, because batteries do not normally have a ground connection to the AC line[9].

An *AC Adaptor* is essentially a small transformer coupled DC power supply. These are usually supplied with a two-prong line cord, so there is no ground connection to the AC power line ground. The supply is floating so the problem of section 10.2 cannot occur.

## 10.4 Russian Roulette and AC Line Voltage

The unsafe situation of an oscilloscope being used to measure AC line voltage, is shown in figure 19. The AC line consists of three connections: the *hot* line, the *neutral* line, and the *ground* wire. For safety reasons, the neutral and ground are connected together and to an earth ground at the system AC distribution panel. The hot and neutral line carry load current in the system – normally the ground wire does not carry any current. Because the neutral is carrying current and because the neutral wire has resistance, at any given point in the system there likely will be a small voltage difference between the neutral and ground wires.

If the ground wire of the oscilloscope is inadvertently connected to the *hot* wire of the AC line, an extremely large short circuit current will flow through the ground connection. Eventually, a circuit breaker will open, a fuse will blow, or the short-circuit current will destroy a conductor. However, until that occurs, the short circuit current can be in the order of hundreds of amperes. This current will destroy the oscilloscope and computer, and the resultant flaming debris may cause injury to nearby living organisms, including humans. It may also start a fire.

[9]The disadvantages of batteries are (a) they run down and (b) they are not current limited. A short circuited battery can do significant damage.

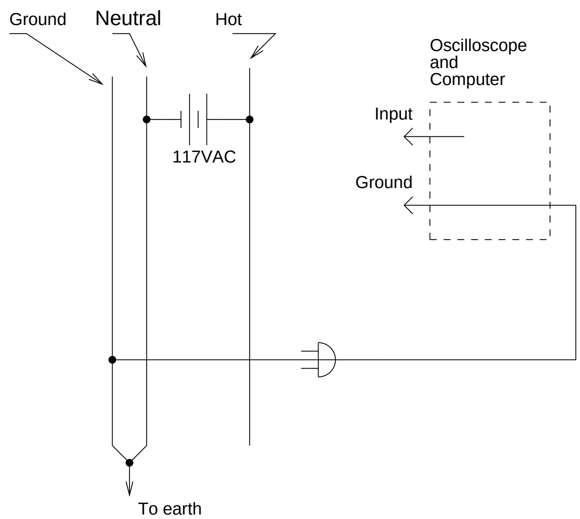



Figure 19: AC Line Voltage

Furthermore, connecting the ground lead of the oscilloscope to the AC neutral line causes another problem - it effectively connects the neutral and ground AC lines at that point. Now the neutral current has another path, and some of it will flow through the oscilloscope and computer ground leads. If this current is sufficient, it may damage the oscilloscope and computer.

Notice that *this same situation can occur with equipment that is not transformer isolated from the AC line*. For example, some electronic equipment has a direct connection to the AC line, so that the chassis is connected directly to the neutral line of the AC system. To safely observe the signals in this device with an oscilloscope *the equipment must be isolated from the AC line by a transformer*. The transformer must function as an *isolation transformer*, the secondary winding must not have an electrical connection to either of the primary leads, and the transformer must consist of a separate primary and secondary winding.

An *autotransformer* (common trade name *Variac*) is an adjustable transformer that is often used for adjusting line voltage. An auto transformer does *not* have an independent secondary winding and cannot be used to isolate electronic equipment from the AC line.

## 10.5 Removing the Ground

The potential for a short circuit is reduced if the ground connection is removed from the computer. However, this is extremely dangerous because metallic connections on the computer (such as the shell around a connector) are connected to the ground line. A connection to the AC line puts those metallic points at line potential, presenting a serious shock hazard to the user and possible short circuit if attached equipment is grounded.

The AC ground connection (the 'third prong' on a plug) is there specifically to prevent the chassis of the equipment from assuming a potential that is above ground, and therefore dangerous to a human operator. Removing that ground connection removes any grounding protection. This is a serious violation of health and safety regulations.

Similarly, a battery-powered laptop computer, when disconnected from its line-operated charger, is not connected to the ground line of the AC power system, so it is less likely to cause the kind of short circuit described in the previous section. However, it is extremely dangerous to rely on this. The laptop may itself become live at the AC line potential, which makes it hazardous to the operator and any attached equipment (such as a line-operated video monitor).

### 10.6 Observation of AC Line Voltages

If you must observe line voltage, here are the rules:

- The oscilloscope must be able to cope with the peak value of the input AC voltage. The Syscomp DS-101 is certified to reliably accept up to 50 volts on its input terminal.

  A *times-ten* oscilloscope probe increases this by a factor of ten. It is absolutely essential to use a probe that can withstand this voltage, and it essential to ensure that the probe cannot inadvertently be switched to a *times-one* setting.

  Notice that the peak value of a sinusoidal voltage is 1.41 times the RMS value. So a 117VAC line voltage will peak at around 170 volts.

- There must be no direct connection to the AC line. If the equipment is line operated, then it must be powered by an isolation transformer (see above).

- It is possible to obtain electronic probes that provide an *isolation barrier* between the line circuit and the oscilloscope. For example, the measurement signal is transferred from the AC line side to the oscilloscope side by means of an optically coupled circuit. There is no electrical connection between the oscilloscope and the AC line. The signal is transferred over a beam of light. This method removes all possibility of short-circuiting the line voltage to ground. See for example `http://www.powertekuk.com/`.

## 11 Overview of USB Operation

In general, the operation of the USB connection is seamless and invisible to the user. Operation of the oscilloscope is usually as simple as plugging it in to an USB port and running the oscilloscope GUI software. However, it may be useful to understand some of the details.

The USB interface uses a USB-Serial chip FT232BM from FTDI. This chip, with the appropriate driver software on the host PC, emulates a serial port. Consequently, the Tcl/Tk GUI software can access the hardware just as if it was accessing a device connected to the host serial port. This is orders of magnitude simpler than dealing with USB, which is extremely complicated. We refer to this as a `USB-serial` interface.

The `USB-serial` has major advantages over the traditional serial port. First, the data transfer rate is much faster (especially using the USB2.0 standard). Furthermore, power is transmitted from the host to the hardware over the USB cable so that an AC adaptor is not needed. Third, the USB system handles enumeration automatically so that multiple devices are accessed correctly without manual configuration.

Under Windows, the FTDI `USB-serial` drivers are automatically loaded by the `Install` program, so the user should not normally be required to intervene.

Under Linux, the FTDI drivers are included in the Linux kernel since 2.4, so they do not need to be installed under Linux.

When a `USB-serial` device is plugged into the host computer for the first time, the host USB system detects a new device and allocates it to a serial port. In Windows, this is a COM port. In Linux, this is a device such as `/dev/ttyUSB0`.

Thereafter, the operating system always associates that hardware with that serial port, even if it is plugged into a different USB port.

In Linux, the default permission of the `/dev/ttyUSBx` ports is set for root access only, so the permissions must be changed as described under 12.2. (The `x` in `/dev/ttyUSBx` represents a number for a ttyUSB port, something like `ttyUSB0` or `ttyUSB1`, and so on.)

# 12 Troubleshooting

## 12.1 Microsoft Windows

### 12.1.1 Manually Assigning a COM Port Number in Windows XP

It may be useful to know how to set the COM port manually.

For example, under certain circumstances, it is possible for the operating system to assign a COM port number that too high to be usable by the host software. In our situation here at the Syscomp factory, we test each instrument by plugging it into a USB port. Each time the operating system sees a new instrument, it assigns it a new COM port number. If the COM port number is greater than 9, it cannot be selected by the host software. Then you must re-assign the port number manually.

1. Plug in the problem instrument to a USB port.
2. Go to: Start `->` Settings `->` Control Panel `->` System `->` Hardware `->` Device Manager
3. Double-click on `Ports (COM and LPT)`. This opens to show any USB-serial port assignments. Let's say that it shows `USB Serial Port (COM12)`. This exceeds COM9, so we have to manually reset the COM port number.
4. Double-click on the entry `USB Serial Port (COM12)`. This opens a new dialogue box, `USB Serial Port (COM12) Properties`.
5. Select `Port Settings` and click on `Advanced`.
6. This opens a new dialog box, `Advanced Settings for COM12`. In the upper left corner, there is a scrollbox `COM Port Number`. Use the up/down arrows to scroll through the possible COM port assignments.
7. For the new COM port assignment, it's best to choose a COM port number 4 or larger. (Lower numbers may conflict with a USB keyboard or mouse). Suppose we decide to move to COM5. The scrollbox says `COM5 (in use)`. Select it anyway. Click `OK`.
8. A warning pops up that the COM port is in use and asks if you want to continue. Click on `Yes`.
9. Back out through the menues until you have closed the Device Manager panel. Re-open it and examine the `Ports (COM&LPT)`. This time it should read `COM5`.
10. Back out to a clean desktop and restart the instrument program. This time, it should connect properly.
11. If you are connecting multiple instruments, you may need to do this for each instrument. However, having once done the assignment for a given instrument, the operating system associates that instrument with the chosen COM port and connection should be automatic.

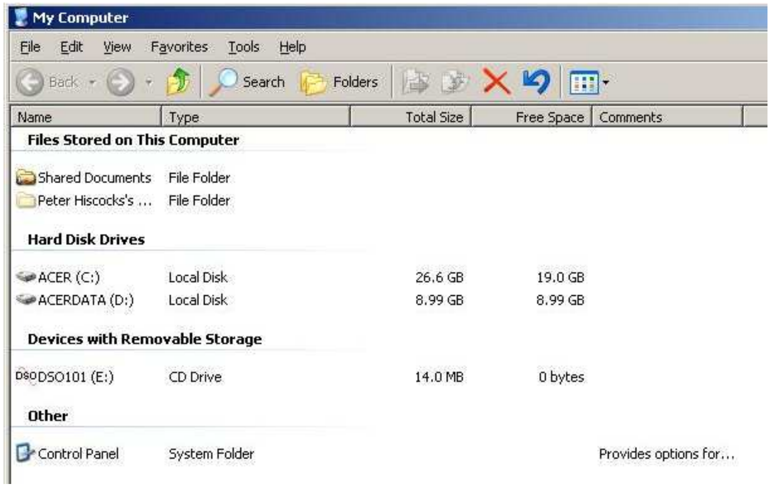



Figure 20: My Computer

### 12.1.2 Checking the Installed Files

This section details how to determine that the files are installed correctly.

1. Double click on `My Computer` to obtain a screen like figure 20.
2. Double click on `Control Panel`, obtaining a screen like figure 21.

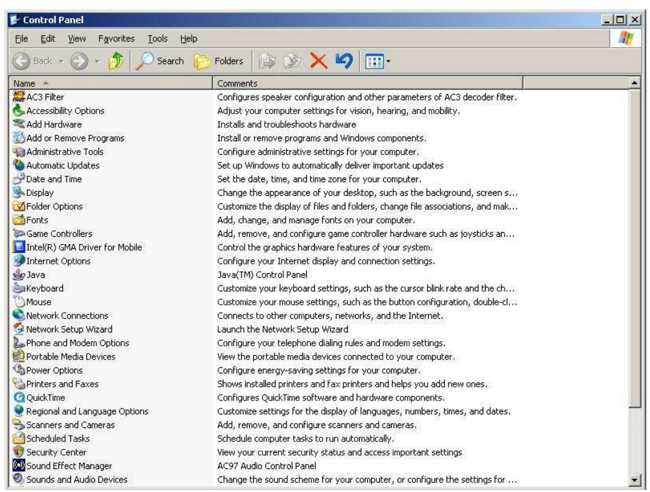



Figure 21: Control Panel

3. Double click on `Add or Remove Programs`, obtaining a screen like figure 22.

   Figure 22 lists the FTDI drivers, so they are installed. Similarily, the Tcl/Tk program for the unit should be shown on the same list under `Syscomp`.

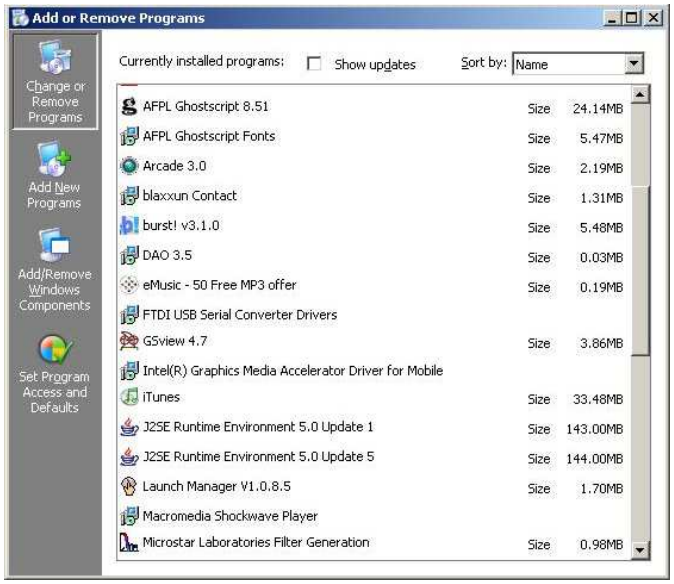



Figure 22: Add or Remove Programs

#### 12.1.3 Obtaining Details of the USB-Serial Port

1. On the `Add or Remove Programs` list of figure 22, find the entry `System`. Double click on it to obtain the `System Properties` screen of figure 23.
2. Double click on the `Hardware` tab to obtain the `Device Manager` screen of figure 24.
3. **Ensure that the hardware unit (DSO-101 oscilloscope, WGM-101 waveform generator, CGR-101 CircuitGear, etc) is plugged into a USB port.**

   In the System Properties, Hardware panel of figure 24, there should be an entry `Ports (COM & LPT)`. Double click on that and it should open to an entry like `USB Serial Port (COM4)`.

   This specifies the COM port that should be selected in the Tcl/Tk program in order to communicate with the USB hardware unit.

#### 12.1.4 Adjusting the COM Port Selection

1. **Ensure that the hardware unit (DSO-101 oscilloscope, WGM-101 waveform generator, CGR-101 CircuitGear, etc) is plugged into a USB port.**
2. In the System Properties, Hardware panel of figure 24, there should be an entry `Ports (COM & LPT)`. Double click on that and it should open to an entry like `USB Serial Port (COM4)`.
3. Double click on this last entry to obtain the Serial Port Properties screen of figure 25.
4. Click on the tab`Port Settings`. You should not need to change any of these settings.
5. Find and click on the `Advanced` button to obtain the `Advanced Serial Port Properties` screen of figure 26.


Figure 23: System Properties

6. In figure 26, the `COM Port` selection is highlighted. Click on the arrow to show a selection of `COM` ports. Use the up and down arrow to scroll through all the `COM` ports. If you wish to force the hardware to use a different `COM` port, you may select that port number here.

#### 12.1.5 Checking the Connection with Hyperterminal

We can do a very simple test to ensure that the hardware is operating.

1. As described above, use Windows Device Manager to determine which COM port the hardware was assigned to.
2. Open up Hyperterminal (`Programs->Accessories->Hyperterminal`) and create a new connection to the CircuitGear COM port.Âă
3. Set up the connection with the following settings:

| | |
|---|---|
| Baud rate | 230400bps |
| Data bits | 8 |
| Stop bits | 1 |
| Parity | None |
| Flow control | Hardware (RTS/CTS) |

4. In the terminal window type the letter "i" followed by $< enter >$.Âă The device should identify itself with a string.

## 12.2 Linux

These troubleshooting notes are specific to Suse Linux 9.2, but should apply in general. They also assume a working knowledge of Linux and its variants.


Figure 24: System Properties, Hardware (Device Manager)

**Overview**

If the software does not operate correctly, here are some things to check. They are subsequently explained in detail.

- The operating system is too old and does not contain the necessary drivers for the usb-serial ports.
- The operating system for some reason is not recognizing the usb device and assigning it to a usb-serial port. This can occur if the usb device has `root` ownership and permissions. The permissions must be changed to allow a user-mode program to access the usb port.
- The operating system is assigning the hardware to some usb-serial port but the Tcl/Tk program is not automatically selecting that particular port. You'll need to select the usb-serial port manually, using the controls in the Tcl/Tk program.
- The `wish` program, which is the interpreter for all Tcl/Tk programs, is not being found by the operating system. Locate it and change your path so that it is found.
- The device is being recognized and connects properly, but does not respond properly to certain controls. Use the instructions in section 15.2 to send commands to the hardware to determine how it is functioning.

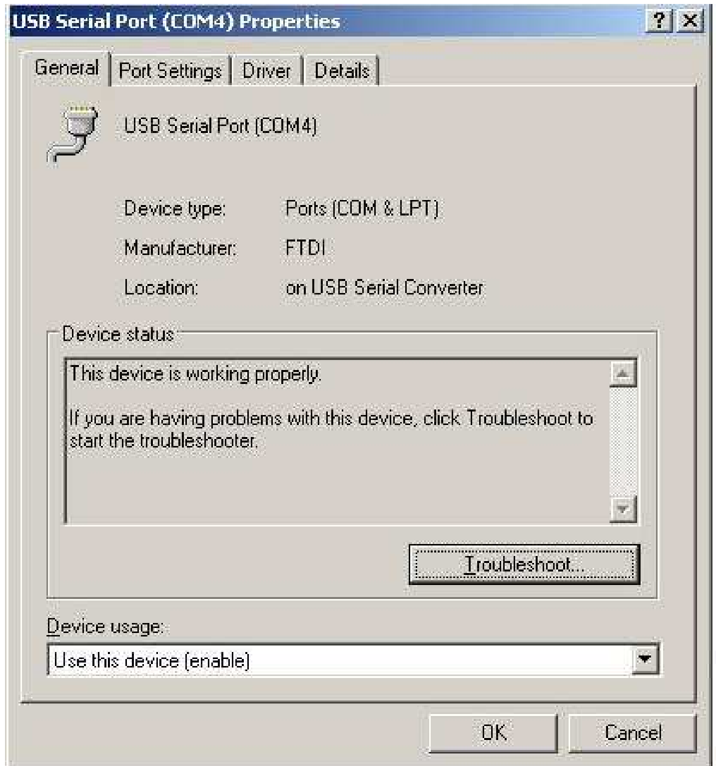



Figure 25: Serial Port Properties

1. **Check the kernel version**.
   The drivers for the FTDI USB-Serial interface are a standard part of the Linux kernel from version 2.4 onward. To check that you have a sufficiently modern kernel, run the `dmesg` command piped to the `more` command.

   ```
   phiscock@panther: dmesg | more
   ```

   Examine the first few lines, which should be something like this:

   ```
   Linux version 2.6.8-24-default (geeko@buildhost) (gcc version 3.3.4 (pre
   3.3.5 20040809)) #1 Wed Oct 6 09:16:23 UTC 2004
   ```

   In this case, the kernel is 2.6.8-24, so it contains the FTDI drivers.

   If your kernel version is older than this, you may have to upgrade the kernel or install a driver module.

2. **Install the software**:
   per the instructions on the CDROM.

3. **Determine the serial port used by the USB driver**. In this step, we'll use the `dmesg` command to determine which serial (COM) port is being assigned to the oscilloscope when it is plugged in.

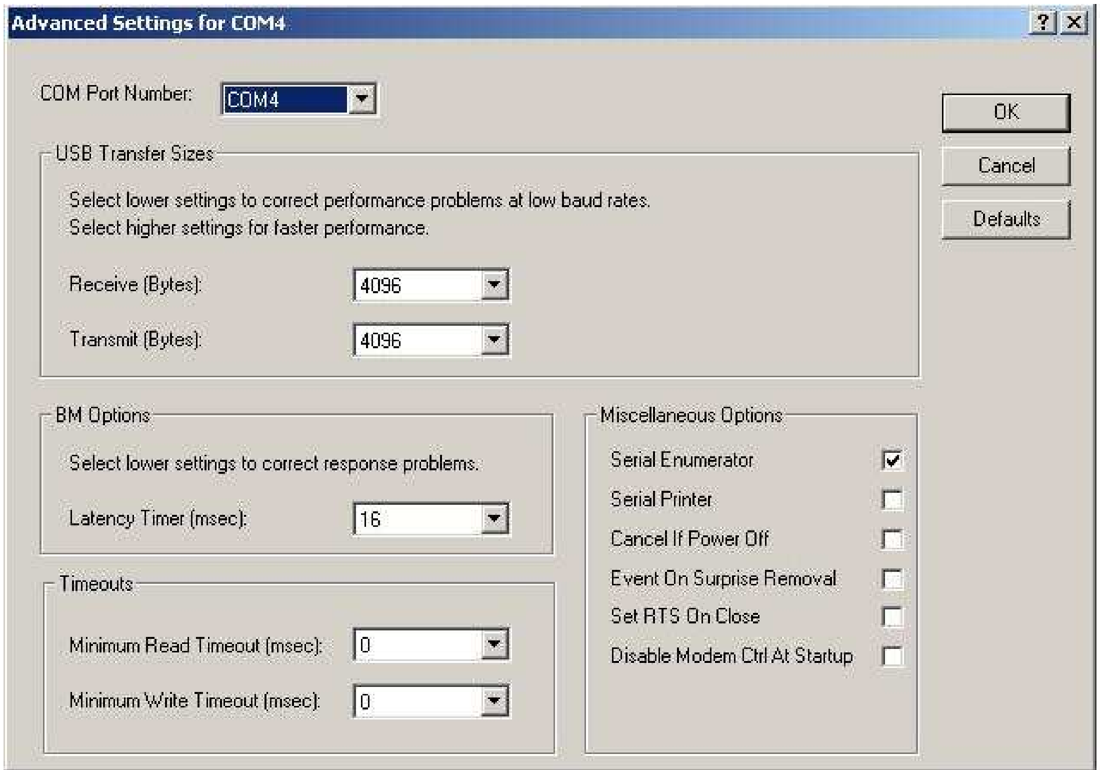



Figure 26: Serial Port Properties, Advanced

Execute `dmesg` to get an idea of the most recent kernel messages. Using the USB cable, connect the scope hardware to a USB port. Execute `dmesg` again, and you should see something like this as the last entry in the `dmesg` printout:

```
usb 4-2: new full speed USB device using address 4
usb 4-2: Product: USB <-> Serial Cable
usb 4-2: Manufacturer: FTDI
usb 4-2: SerialNumber: 00000001
ftdi_sio 4-2:1.0: FTDI FT232BM Compatible converter detected
usb 4-2: FTDI FT232BM Compatible converter now attached to ttyUSB0
```

Unplug the USB cable and run `dmesg` again and see something like this:

```
usb 4-2: USB disconnect, address 4
FTDI FT232BM Compatible ttyUSB0: FTDI FT232BM Compatible converter now
disconnected from ttyUSB0
ftdi_sio 4-2:1.0: device disconnected
```

Evidently the USB device is being assigned to device `ttyUSB0`.

This shows that the USB device is being recognized by the operating system and assigned to a usb-serial port.

4. **Set the permissions for the USB-Serial port**
   The default situation is that root is the owner of the USB serial port `ttyUSB0` and operation is restricted to root. For an ordinary user to access the port, the permissions must be changed.

   First, we will show how to do this manually in section 12.2.1. However, Linux is usually set up so that the permissions revert back to root mode every time the USB is plugged and unplugged, and every time the system is rebooted. Therefore, we need to modify the system so that this is done automatically, ie, so that the port permissions are set to user mode by default. This is shown in section 12.2.2 below.

### 12.2.1 Manually Changing Device Port Permissions

Change to the `/dev` directory.

```
phiscock@linux:~> cd /dev
```

Check the permissions on the `ttyUSB` ports:

```
phiscock@linux:/dev> ls -l ttyUSB*
crw-rw----  1 root uucp 188, 0 2005-11-07 18:25 ttyUSB0
crw-rw----  1 root uucp 188, 1 2004-10-02 01:38 ttyUSB1
<others deleted>
```

In this case, the owner (root) has read-write access. The group that root belongs to, uucp, also has read-write access. Others (that's you) have no access at all. To open up the port to user access, enter root mode using the `su` command:

```
phiscock@linux:/dev> su
```

The system asks for the root password. Enter it. Now you can change the permissions (mode) for the ports. In this case, we'll use the `chmod` command to add read and write permission for 'others'. For example, the first command below says: *change the mode of device ttyUSB0 to add read permission for 'others'*. The second command does the same for write permission.

```
linux:/dev # chmod o+r ttyUSB0
linux:/dev # chmod o+w ttyUSB0
```

Check the permissions again:

```
ls -l ttyUSB*
crw-rw-rw-  1 root uucp 188, 0 2005-11-07 18:25 ttyUSB0
crw-rw-rw-  1 root uucp 188, 1 2004-10-02 01:38 ttyUSB1
```

That's it. You should now be able to access those ports from user mode. Exit from root to user mode.

Incidentally, you may be able to change the permissions by logging in as root and then using the features of the KDE or Gnome desktop to change the permissions.

### 12.2.2 Setting Default Port Permissions to User Mode: Suse 9.2

This change will ensure that the serial-usb ports are always created with user mode access.
The default permissions for user devices are contained the file: `/etc/udev/permissions.d/50-udev.permissions` . We have to modify the entry for the `ttyUSBx` ports so that the default is user mode.

1. Change to the directory `/etc/udev/permissions.d` and check that the file `50-udev.permissions` exists.

2. If the file exists, [10] enter root mode, and copy the existing file so you have a copy of the original.

   ```
   cp 50-udev.permissions 50-udev.permissions-orig
   ```

3. Now open the file `50-udev.permissions` with your favourite editor. Find the entry that says:

   ```
   ttyUSB*:root:uucp:660
   ```

   Change that to read:

   ```
   ttyUSB*:root:uucp:666
   ```

   Save the file. Now every time a `ttyUSBx` port is created, you should be able to access that port without problems.

### 12.2.3 Setting Default Port Permissions to User Mode: Suse 10.3

Suse in their wisdom have changed the method detecting USB devices. Now, USB devices do not exist in `/dev` until they are plugged in.

Plug in the DSO-101 oscilloscope and execute 'dmesg'. You should see something like the following at the end of the message:

```
usb 1-2: new full speed USB device using uhci_hcd and address 2
usb 1-2: new device found, idVendor=0403, idProduct=6001
usb 1-2: new device strings: Mfr=1, Product=2, SerialNumber=3
usb 1-2: Product: Digital Oscilloscope DSO-101
usb 1-2: Manufacturer: Syscomp
usb 1-2: SerialNumber: DSQ3Q7ZO
usb 1-2: configuration #1 chosen from 1 choice
drivers/usb/serial/usb-serial.c: USB Serial support registered for FTDI USB
Serial Device
ftdi_sio 1-2:1.0: FTDI USB Serial Device converter detected
drivers/usb/serial/ftdi_sio.c: Detected FT232BM
usb 1-2: FTDI USB Serial Device converter now attached to ttyUSB0
usbcore: registered new interface driver ftdi_sio
drivers/usb/serial/ftdi_sio.c: v1.4.3:USB FTDI Serial Converters Driver
```

This indicates that the oscilloscope was detected and it has been assigned to the USB-Serial port `ttyUSB0`.
You now go to the director `/dev` and examine the `ttyUSB0` entry:

[10] If the file does not exist, please let us know the name of the Linux distribution and we'll look for another solution.

```
phiscock@panther:/dev> ls -l ttyUSB0
crw-rw---- 1 root uucp 188, 0 2008-04-20 12:57 ttyUSB0
```

The `uucp` group have read-write permission to this device, so the permanent solution is to add `uucp` as one of your groups. In Suse 10.3 this is done from: `Computer -> Control Center -> Open Administrator Settings`. You'll need to enter the root password.

Then go to: `Security and Users -> User Management -> User and Group Administration`.

Select the user (that's you) and click on `Edit`. This brings up the `Existing Local User` page. Click on `Details`. Under `Groups` check off `uucp`. Log out and log back in, or restart the computer. You should now be able to access the USB port without having to change its permissions.

#### 12.2.4 Setting Default Port Permissions to User Mode: Fedora Core 6

This note[11] applies to Fedora Core 6, kernel 2.6.19-1.2911.fc6.

Look in `/etc/udev/rules.d/50-udev.rules` for the line:

```
KERNEL=="tty[A-Z]*", NAME="%k", GROUP="uucp", MODE="0660"
```

Change the mode value to 0666.

#### 12.2.5 Running the Program

1. Change to the directory where the program resides. The `wish` interpreter is required to run the tcl program. It is normally included with a Linux distribution, so it is probably present on your system. You can find out by issuing the `which` command.

   ```
   phiscock@linux: which wish
   /usr/bin/wish
   ```

   If this doesn't turn it up, use the 'find' command, starting at the root directory '/'. If it is on the system, then add that location to your path.

   ```
   phiscock@linux: find . -name wish
   <much deleted>
   /usr/bin/wish
   ```

2. Start `wish`.

   ```
   phiscock@linux:~/eelab/demos> wish
   ```

3. A new small window will appear. This is the container for any program that `wish` executes. The cursor remains where the `wish` command was run.

4. Click in that window and run the command:

   ```
   % source main.tcl
   ```

   using the correct name for the oscilloscope program. The oscilloscope GUI should now run correctly.

There are many other ways to start the program. For example, the command `wish scope-101.tcl` (substitute the correct name of the tcl program) can be used. As well, the KDE and Gnome graphical user interfaces be used to set up an icon on the desktop. Then clicking on that icon will start the program.

[11] Kindly supplied to us by John Foster.

### 12.2.6 Ubuntu Linux Install

This sectiondescribes how to install the DSO-101 oscilloscope software on a Ubuntu Linux system. This procedure was tested on an `Edubuntu` system, `Hardy Heron`, April 2008. Similar instructions apply to installation of the WGM-101 Waveform Generator and CGR-101 CircuitGear (with the corresponding file names).

This application note assumes some familiarity with Linux. Specifically, you must know how to use an editor to modify files. You should know how permissions work, and how to use the `chmod` command to change them. You should have some familiarity with the directory structure and the location of your home directory vs system directories such as `/etc` and `/dev`. You must know how the `ls` command works.

#### Download and Unpack the Software


Figure 27: Checking that the File has Downloaded

The software is available for download from the Syscomp website:
`www.syscompdesign.com/download.htm`

Use the `mkdir` command to reate a directory on your computer. In my case, this directory is:
`/home/peter/eelab`.

Click on the file `Linux (x86bin)` to download it to your computer, check that it is in the `eelab` directory (figure 27).

The file that you downloaded is an *archive*, from which you must extract the files. Right click on the file and select `Extract Here`.

A directory is created with the extracted files in it. Change to that directory and examine the contents (figure 28).

Check that `main.tcl` is present in the file list. This is the file to execute to start the oscilloscope program.

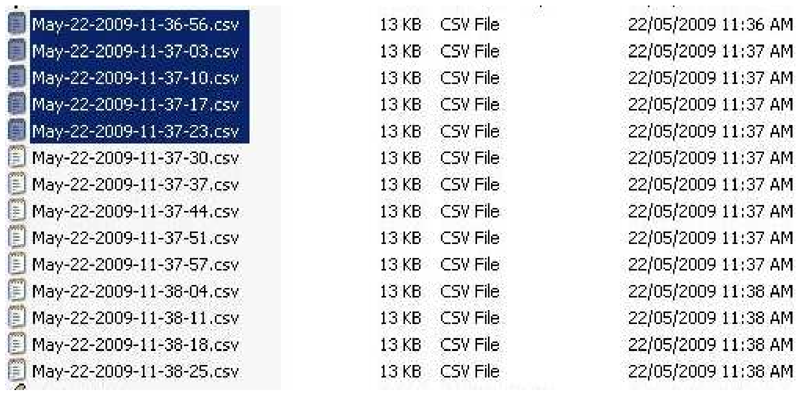
Check that the Tcl Interpreter *wish* is present on your machine:
`which wish`

If *wish* is present, the *which* command will show the directory:
`/usr/bin/wish`

Since `/usr/bin` is usually on your PATH, wish is present and should execute.

Now you can execute the DSO-101 software with the command:
`wish main.tcl`.

However, the oscilloscope software will not connect properly to a USB port until the permissions are corrected. That's the next step.



| | | | |
|---|---|---|---|
| May-22-2009-11-36-56.csv | 13 KB | CSV File | 22/05/2009 11:36 AM |
| May-22-2009-11-37-03.csv | 13 KB | CSV File | 22/05/2009 11:37 AM |
| May-22-2009-11-37-10.csv | 13 KB | CSV File | 22/05/2009 11:37 AM |
| May-22-2009-11-37-17.csv | 13 KB | CSV File | 22/05/2009 11:37 AM |
| May-22-2009-11-37-23.csv | 13 KB | CSV File | 22/05/2009 11:37 AM |
| May-22-2009-11-37-30.csv | 13 KB | CSV File | 22/05/2009 11:37 AM |
| May-22-2009-11-37-37.csv | 13 KB | CSV File | 22/05/2009 11:37 AM |
| May-22-2009-11-37-44.csv | 13 KB | CSV File | 22/05/2009 11:37 AM |
| May-22-2009-11-37-51.csv | 13 KB | CSV File | 22/05/2009 11:37 AM |
| May-22-2009-11-37-57.csv | 13 KB | CSV File | 22/05/2009 11:37 AM |
| May-22-2009-11-38-04.csv | 13 KB | CSV File | 22/05/2009 11:38 AM |
| May-22-2009-11-38-11.csv | 13 KB | CSV File | 22/05/2009 11:38 AM |
| May-22-2009-11-38-18.csv | 13 KB | CSV File | 22/05/2009 11:38 AM |
| May-22-2009-11-38-25.csv | 13 KB | CSV File | 22/05/2009 11:38 AM |

Figure 28: Directory with Files

**Configure Permissions on the USB Ports**

Plugging a Syscomp oscilloscope, waveform generator or CircuitGear unit causes the system to create a 'serial USB port' at `/dev/ttyUSB0`. (Notice that this is yew-ess-bee-zero, not yew-ess-bee-oh. If you have another usb-serial device plugged in, the number might be some other digit than zero.)

This port has read-write permissions for root and for the 'dialout' group. It does not have read-write permissions for a lowly user, so the instrument will not connect in this state.

You can change the permissions on `/dev/ttyUSB0`, but the port is transient and disappears when you shut down or disconnect. So you would need to do that every time you start the instrument.

To make this permanent, add your name to the 'dialout' group. I tried to do it using the edubuntu system administration tool, but for some reason the dialout group did not appear.

To do this the old-fashioned command-line way, use the command 'groups' to see which groups you belong to. Probably dialout does not appear. We will edit the groups file to add you. Find the file `/etc/group`. Probably read-write permission is disabled for users on this file as well. Make a note of the permissions, because we'll change it back. For example:.

```
ls -l /etc/group
-rw------- 1 root root 934 2008-05-22 10:17 /etc/group
```

In order to edit it, change the permissions to allow any user to edit that file, using the command:

```
sudo chmod 777 group
<your password here>
ls -l /etc/group
-rwxrwxrwx 1 root root 954 2008-10-24 22:02 /etc/group
```

Now you can edit and save the file. Load that file into a text editor. Find the line with 'dialout' in it. If the line ends in a colon, that means that no one yet belongs to this group. Add your login name. If there already is some other login name, add a comma and then your login name. Here's what it might look like with me (peter) added to the dialout group, after gabe.

```
man:x:12:
proxy:x:13:
kmem:x:15:
```

```
dialout:x:20:gabe,peter
fax:x:21:
voice:x:22:
.... and so on
```

Save the file. Reboot the computer. Execute the `groups` command. You should see that you are a member of the `dialout` group.

```
groups
peter dialout
```

Now you should be able to start the instrument code by executing `wish main.tcl` in the directory where `main.tcl` is located. The instrument should find and connect to the USB port.

For security, you may want to return the permissions on `/etc/group` as they were originally, as you noted previously. In my case, giving groups and users read permission as well:

```
sudo chmod 644 /etc/group
<your password here>
ls -l /etc/group
-rw-r--r-- 1 root root 934 2008-05-22 10:17 /etc/group
```

Special thanks to Gabriel Guillen, who loaned me an edubuntu system for this exploration.

### 12.2.7 64 Bit Linux

One of the packages in the Syscomp distribution on the CD is the 'Img' package. The Img package is a Tcl/Tk library that is required for the CGR-101 software. Currently, those libraries are for a 32 bit linux operating system. We are in the process of modifying the distribution to accompodate 64 bit linux, but in the meantime here is a workaround.

Delete the existing Img file. (The best way to do this is to rename it, so that you can back out if this procedure fails.) Once you've deleted it, try running the software. If your Tcl/Tk distribution has the correct Img library it should run fine. If not, download and install the ActiveState Tcl/Tk distribution for 64 bit linux.

### 12.2.8 Running under 64-bit Sidux (Debian-based Linux)

The following notes were provided by Radio College of Canada student Gabriel:

```
I am currently running the software on a 64 bit Sidux (Debian based linux)
and have noticed that the img library (the .so files in Img13Lin) are 32 bit
only.  This prevents the software from running on any 64bit linux systems.

This can easily be fixed by installing the img package in 64 bit.
the package name is libtk-img  and can be installed by running:

apt-get install libtk-img

on any Debian based OS or by using the default package manager  (this will
work for 32 and 64 bit systems)

Also, the Bwidget package should also be listed as a dependency, the package
```

```
name is bwidget  (all lower case)

apt-get install bwidget

will install it on the system
```

#### 12.2.9 Device Properties using `usbview`

In general, it's not necessary to know anything about the USB properties of the hardware in order to use it. However, if you do want to inspect those properties, `usbview` is useful.

It is likely that you will have to install `usbview` from your Linux distribution disks.

Once `usbview` is installed, (figure 29) you can use it to determine whether a USB device is recognized by the operating system USB. As a USB device is plugged and unplugged, an entry appears and disappears in the `usbview` window.

Clicking on an entry opens up a list of USB properties of the device.

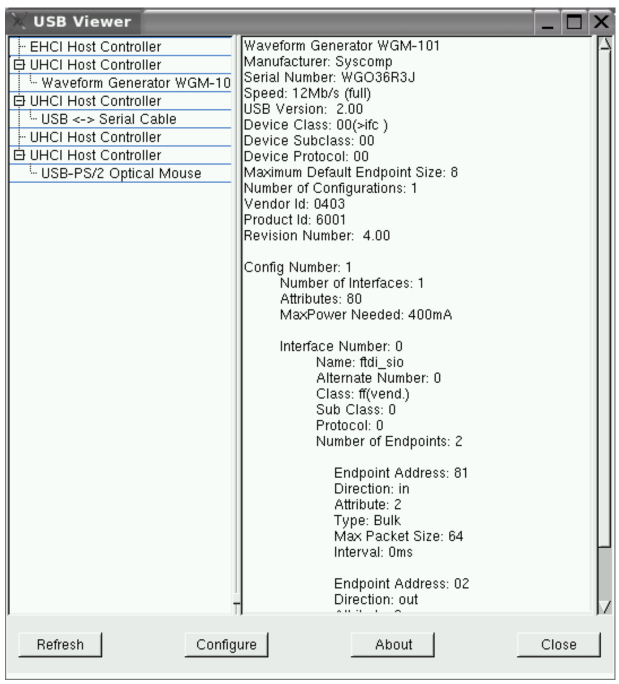



Figure 29: Usbview

Notice that `usbview` does not indicate the serial port (`/dev/ttyUSB2` or whatever) that the operating system has assigned to this device. You must use `dmesg` for that purpose.

**Acknowledgements**

Special thanks to Dorothy Rawek, Miles Pierce and Seneca Cunningham. Dorothy provided her MacBook for this investigation. Miles acted as a beta tester for this install process and supplied important feedback. Seneca showed me how to create a shell script using AppleScript.

# 13 Adjustments

The oscilloscope has been adjusted before shipping, so it should not need adjustment before use. These instructions are provide for reference purposes.

## 13.1 Input Compensation Capacitors

The schematic of the input circuitry of the oscilloscope vertical preamplifier is shown in figure 30.

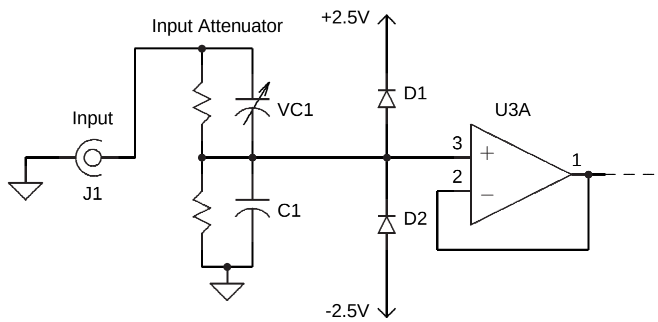



Figure 30: Scope Input Compensation Network

The two channels are identical.

Stray capacitance at the output of the voltage divider has the potential for limiting the bandwidth. To make the divider frequency independent, the resistive divider R1, R2 is accompanied by a capacitive voltage divider VC1, C1. The capacitive divider must be adjusted to have the same division ratio as the resistors. This is accomplished by adjusting VC1.

To do so, remove the circuit-board from its plastic case. Connect it to the computer via its USB cable and start the host software. Apply a square wave input signal to one of the scope channels. Adjust the magnitude of the square wave such that it makes a suitable display. Adjust the oscilloscope timebase such that the leading edge of the square wave is visible.

The two variable capacitors are relatively large, square components that rise above the other components on the board. (The current versions have orange casings.) On the current design they are designated C61 for Channel A and C81 for Channel B. Locate the variable capacitor on the circuit that corresponds to the input channel with the signal. With a tiny screwdriver, adjust that variable capacitor until the square wave shows the fastest possible rise time without overshoot. It is best to use a screwdriver with an insulated shaft because a non-insulated screwdriver will connect human body capacitance into the circuit which affects its operation. If the screwdriver is not insulated, make an adjustment and then remove the screwdriver to see the effect.

Repeat the same procedure with the second oscilloscope channel.

Reassemble the unit back in its case.

If you are using a x10 oscilloscope probe, the input capacitance of the scope channels will have changed slightly. You will need to adjust a x10 probe for best square wave response.

If the probe is a x1 x10 switchable unit, ensure that the probe is switched to the x10 position. Attach it to one of the channels. Connect the probe to a square wave source, such as the CGR-101 generator output. Find the compensation adjustment screw on the scope probe. In the economy probes we sell, the screw adjust is red and located in the base of the probe cable, near the BNC connector. In the professional probe we sell, the screw adjust is yellow and located in the probe, near the handle. Adjust this screw for best square wave response.

# 14 Oscilloscope Commands

These commands are low-level instructions to the scope hardware. The commands (and where relevant, messages back from the hardware) are ASCII strings so that they can be generated easily by software or a human operator.

There are two situations where the interface commands may be useful.

- The scope hardware may be operated directly from a terminal emulator program such as `Hyperterminal` under Windows, `Minicom` or `Seyon` under Linux.

- The commands must be known to create a scope control program with different functionality. For example, a program could be created to sweep the read the oscilloscope and plot the results on a strip-chart type of display. This new program needs to issue commands to the hardware.

  Should you decide to attempt such a project, the Tcl source code for the Oscilloscope GUI, which is provided on the accompanying CDROM, will be a useful source of ideas in controlling the scope.

  There is no requirement that the controlling program be written in the Tcl language. Any program that can issue ASCII strings to a serial port will be capable of controlling the scople hardware. (i.e `Matlab`, `Visual Basic`)

## 14.1 Using the Debug Console

When you send a command to the scope, it does not echo any confirmation back to the host terminal. This is because you are doing with the terminal exactly what is done with a control program, and responses from the scope would slow down the overall operation of the system.

Each command consists of an ascii string of characters, such as T55`<cr>` to set the trigger level to 55, where `<cr>` is a carriage return character.

If you are running under a Windows operating system, in addition to the material in the user manual, you can see commands being sent to the oscilloscope by selecting

```
View -> Debug Console.
```

This brings up a terminal screen which lists commands as they are being sent to the scope hardware, and some other debug information.

On Linux and Mac machines, the operator starts the program from a terminal window, executing a command like `wish main.tcl`. In that case, the terminal window becomes the console.

This information scrolls past rather quickly when the scope is in auto trigger mode, because it is repeatedly obtaining data from the hardware. To slow this down, put the scope in 'Manual Trigger' mode. Now each time you hit the 'manual trigger' button, the debug screen will show the commands that were sent to the scope hardware.

Now you can change control settings on the scope GUI and see the corresponding commands as they are sent to the scope hardware. For example, move the trigger level cursor on the scope screen and you will see a series of trigger level commands like T55 T43 T27 being sent to the hardware. Similarily, changing the vertical preamplifier gain settings to show the corresponding hardware commands.

Regarding the oscilloscope operation: For normal triggering, the host PC issues a request for an update. When a trigger occurs, the hardware acquires the waveform and sends it to the host. The host receives it on an asynchronous basis, using the `fileevent` command. That command then processes the data.

The waveform data is 1K samples per channel, times two channels, times two bytes per sample equals 4k bytes.

## 14.2 CircuitGear ASCII Command Set

All commands are terminated with `<carriage-return>` or a `<carriage-return><line-feed>` pair.

In the following commands, the spaces are real and must be part of the command string.

| Command | Description |
|---|---|
| i | Info: Identify |

Returns device identification string and firmware revision, preceeded by '*' and terminated by `<cr><lf>`.

| Command | Description |
|---|---|
| `W F F3 F2 F1 F0` | Waveform: Set Frequency |

The parameters F3-F0 are integer 8-bit numbers (i.e. 0-255) representing the four bytes of the phase value. The phase value is a 32-bit number. The phase value is calculated as follows:

Phase value = (Output Frequency [Hz])/ 0.09313225746 [Hz]

where 0.09313225746 is the resolution of the waveform generator.

Example for a 1kHz output: Phase value $= 1000\text{Hz}/0.09313225746 = 10737$

Break the phase value 10737 into four bytes:

| | |
|---|---|
| $0 * 16777216$ | $2^{24}$ |
| $0 * 65536$ | $2^{16}$ |
| $41 * 256$ | $2^{8}$ |
| $241 * 1$ | $2^{0}$ |

The command would be `W F 0 0 41 241`

| Command | Description |
|---|---|
| W A A0 | Waveform: Set Amplitude |

The parameter A0 is an 8-bit integer (0-255) representing the amplitude of the output waveform. i.e. A0 = 128, output amplitude = 50%

| Command | Description |
|---|---|
| W S ADDR DATA | Waveform: Program Sample |

The parameter ADDR is an 8-bit integer (0-255) representing the sample number. The parameter DATA is an 8-bit integer (0-255) representing the waveform amplitude of the sample at address ADDR. The data is stored in a waveform buffer in the generator hardware.

| Command | Description |
|---|---|
| W P | Waveform: Program waveform. |

This command takes all of the data stored in the waveform buffer (previously set up using the W S command) and programs it into the FPGA, changing the output waveform.

| Command | Description |
|---|---|
| `S T T_HIGH T_LOW` | Scope: Adjust trigger level. |

`T_HIGH` and `T_LOW` represent a 10-bit number (0-1023). The trigger voltage is calculated as follows:

Trigger Value $= 511 - \text{Gain} * (\text{Trigger Voltage})/(0.052421484375)$

Where Gain=1 for the 5V scale and Gain=10 for the 500mV scale.

Example: Setting the trigger level to 1V on the 5V scale:

Trigger Value=511-(1)*(1.0V)/( 0.052421484375V)=492

The command would be `S T 1 236`

| Command | Description |
| --- | --- |
| `S C C_HIGH C_LOW` | Scope: Post trigger sample count. |

The 8-bit numbers (0-255) make up a 10-bit number representing the number of samples that will be captured after a trigger event. For example, the command S C 1 20 represents a post-trigger count of 276, so 748 samples will be captured before the trigger event, and 276 samples will be captured after. This command can be used to examine events before and after the trigger of a given capture.

| Command | Description |
|---|---|
| `S R REG_VALUE` | Scope: Update the control register |

| `REG_VALUE` Bits | Function |
|---|---|
| 3:0 | Sample rate select. The sample rate is determined by the following formula: Sample rate $= 20$MS/s / ($2^N$) Where N is the 4-bit number (0-15) made up of bits 3 down to 0 in the scope control register |
| 4 | Trigger Source, 0: Channel A, 1: Channel B |
| 5 | Trigger Polarity, 0: Rising, 1: Falling |
| 6 | Internal or External Trigger, 0: Channel A or B as defined by bit 4 1: External |

| Command | Description |
|---|---|
| `S D DEBUG_CODE` | Scope: Debug hardware command |

| Debug Code | Function |
|---|---|
| 0 | Set RESET=0 |
| 1 | Set RESET=1 |
| 2 | Set REQ=0 |
| 3 | Set REQ=1 |
| 4 | Set `MAN_TRIG`=0 |
| 5 | Set `MAN_TRIG`=1 |

All these bits are active high, ie RESET=1 causes a hardware reset. REQ forces a capture request. `MAN_TRIG` forces a manual trigger. The manual trigger will only fire if the scope control register is set to ’external’ trigger.

| Command | Description |
|---|---|
| `S G` | Scope: Go, begin capture. |

The scope will reply with a message when the capture is complete. The message is three bytes long. The first byte is the âĂIJAâĂİ character indicating that the scope is returning the address where the capture ended in the scopeâĂŹs circular sample buffer. The second and third bytes are in binary format and form a 10-bit number representing the ending address. The trigger point can be derived by subtracting the post-trigger sample count.

| Command | Description |
|---|---|
| S B | Scope: Read scope data buffer. |

Scope will return the entire scope memory preceded by the âĂIJDâĂİ character. The data is in binary format with the following organization:

`A1a1B1b2A2a2B2b2....`

Where A1 is the upper byte and a1 is the lower byte of a 10 bit sample. For example, the value of the first data point is (256*A1)+(a1). No need to mask or shift. There is no termination character at the end of the buffer. The user should expect to receive 4097 bytes (4096 bytes for samples, plus one byte for the âĂIJDâĂİ character at the beginning of the transfer.)

| Command | Description |
|---|---|
| S S | Scope: Query scope state machine |

Returns the scope state in the following format:
State X Where âĂŸXâĂŹ is the current state of the state machine.

The possible oscilloscope states are:

| | |
|---|---|
| State 1 | Idle |
| State 2 | Initializing capture |
| State 3 | Wait for trigger signal to reset |
| State 4 | Armed, waiting for capture |
| State 5 | Capturing |
| State 6 | Done |

| Command | Description |
|---|---|
| S P [A\|a\|B\|b] | Scope: Preamp settings |
| A | Set preamp A to 1/2/5V range |
| a | Set preamp A to 500mV/200mV/100mV/50mV range |
| B | Set preamp B to 1/2/5V range |
| b | Set preamp B to 500mV/200mV/100mV/50mV range |

| Command | Description |
|---|---|
| S O | Scope: Read offset calibration |

Returns the offset values stored in eeprom in the form `OAaBb`
where A, a, B, b are 8 bit signed numbers corresponding to the offset for each channel and range setting.

The read offset command returns all four offset values: Channel A High Offset, Channel B High Offset, Channel A Low Offset, and Channel B Low Offset. These values values are fixed and do not change so you should only have to read them once. The software should know the setting of the preamp and apply the corresponding offset value.

| Command | Description |
|---|---|
| S F A a B b | Scope: Store offset calibration into eeprom |

Allows the user to store 4 8-bit numbers into the device eeprom for offset calibration. Refer to the GUI for an example.

| Command | Description |
|---|---|
| D O N | Digital: Set digital output byte to N |

The parameter N is an 8-bit integer (0-255). E.g. The command D O 3 would set bits 0 and 1 of the output port.

| Command | Description |
|---|---|
| D I | Digital: Read digital inputs. |

Returns ’Ix’ where ’x’ is an 8-bit value corresponding to the input byte.

| Command | Description |
|---|---|
| D F N | Digital: Set PWM frequency |

| N | Frequency |
|---|---|
| 0 | 72 kHz |
| 1 | 36 kHz |
| 2 | 9 kHz |
| 3 | 4.5 kHz |
| 4 | 1.125 kHz |
| 5 | 564 Hz |
| 6 | 281 Hz |
| 7 | 141 Hz |
| 8 | 70 Hz |
| 9 | 35 Hz |
| 10-255 | Disabled |

| Command | Description |
|---|---|
| D D N | Digital: Set PWM duty cycle to N/255*100% |

| Command | Description |
|---|---|
| D A | Enable auto update of digital inputs on state change |
| D a | Disable auto update of digital inputs on state change |
| D A | Enable automatic digital input update on level change |
| D a | Disable automatic digital input update on level change |

| Command | Description |
|---|---|
| D ! M | Set interrupt mode and arm interrupt |

| Mode M | Function |
|---|---|
| D | disable |
| R | rising edge |
| F | falling edge |
| H | high level |
| L | low level |
| | The Interrupt returns the charcter **!** when the interrupt condition is met. |

# 15 Manual Operation

The oscilloscope may be operated by sending it commands from a terminal emulator. This can be useful for debugging.

## 15.1 Windows

The `Hyperterminal` program that is supplied as part of Windows operating systems is suitable for this.

1. To simplify matters, it is probably best to unplug any USB devices that are connected.
2. Plug the scope hardware into a USB port on the computer.
3. Using the steps described in section 12.1.1, determine the COM port that the scope is currently connected to.
4. Start Hyperterminal:
   Start `->` Programs `->` Accessories `->` Communications `->` Hyperterminal
5. Hyperterminal starts with a 'Connection Description' popup window. Cancel the popup.
6. Select File `->` Properties
7. Operate the `Connect Using` menu to select the COM port that you found previously.
8. Select `Configure` which pops up a `Port Settings` window. Most of the port settings can be left at their default values, but the baud rate (`Bits per Second)` must be changed to the correct value. At this time, the correct value is 230400. The correct port settings are:

   | | |
   |---|---|
   | Bits per Second: | 230400 Baud |
   | Data bits: | 8 |
   | Parity: | None |
   | Stop Bit: | 1 |
   | Flow Control | ,Hardware (RTS/CTS) Handshaking |

   If you have a version of the scope Tcl program that works correctly with the scope hardware, you can verify the baud rate. Load that code into a text editor and look for the baud setting. Search for a variable `baud` or a procedure `openSerialPort`.

   Hit `OK` and back out to the hyperterminal screen.
9. Issue some command from the list in section 14. Type it in followed by the 'Enter' key. A good choice is `i`, which should result in a message from the scope indicating its version number.

## 15.2 Linux

For manual operation under the Linux operating system, you will need to communicate with the hardware using a *terminal emulator* program. There are two terminal emulators in common use under Linux: `seyon` and `minicom`. These may not be installed as part of your linux distribution. To check whether the program is installed, use the `which` command:

```
phiscock@panther:~> which seyon
/usr/X11R6/bin/seyon
```

Seyon must be properly configured to be used. This is described in the document *Seyon: Quick Start Guide* which is on the system CDROM.

Information on using `minicom` may be found at the following location:

*Using Minicom and Seyon*
Chapter 11 of *Learning Debian GNU/Linux*
Bill McCarty
O'Reilly Books, 1999
`http://www.oreilly.com/catalog/debian/chapter/book/ch11_07.html`

1. Plug the scope hardware into a USB port on the computer.

2. Run the command `dmesg` to identify the serial port that the is allocted to this USB device. It may be a few seconds before the operating system finishes its allocation, so run dmesg repeatedly until you see the serial port number, which will be something like `\ttyUSB1`.

3. Start the terminal communications program (Seyon or Minicom):

4. Set the port settings to:

| | |
|---|---|
| Bits per Second: | 230400 |
| Data bits: | 8 |
| Parity: | None |
| Stop Bit: | 1 |
| Flow Control | Hardware |

   If you have a version of the oscilloscope Tcl program that works correctly with the scope hardware, you can verify the baud rate. Load that code into a text editor and look for the baud setting. Search for a variable `baud` or a procedure `openSerialPort`.

5. Issue some command from the list in section 14.2. Type it in followed by the 'Enter' key. A good choice is `i`, which should result in a message from the oscilloscope indicating its version number.

   Now you can type in other commands.

## 16 Modifying the Tcl/Tk Software

The host software for the scope (and other Syscomp instruments in this series) is released in plain text format under the GPL (Gnu Public Lincense). Consequently, it is legal to modify the program in whatever way you may find useful. We also encourage you to share your work with others.

The software is written in Vanilla Tcl/Tk, that is, in the Tcl/Tk language without any third party packages or linked libraries. This makes it extremely simple to modify.

The program itself is in text form. The code does not need to be compliled or linked, it is executed directly from the text form by the `wish` interpreter.

Although the scope program is fairly complicated, it is possible to create a powerful Tcl/Tk program, with a sophisticated user interface, with only a few lines of Tcl/Tk code.

To set up for development, ensure that the `wish` interpreter is installed on your computer. Under Linux, it is probably already there. Under Windows, you need to download and install a file from the ActiveState website:

http://www.activestate.com/Products/languages.plex?tn=1
Download the appropriate .exe file for your operating system. Run the program to install Tcl/Tk.

Make a copy of the original code, of course, and put it in a safe place. Then, using a text editor, read and modify the existing code. When you want to test the code, run the `wish` interpreter. Execute the `source` command with the name of your file, and the program will execute. Repeat this cycle until you have the desired result.

We'd like to hear about your work. Check out our web page for the latest contact information.

# 17 Software Revisions, Version 1.14

In this section, we document revisions to controls and features that occurred with version 1.14 of the software. For bug fixes, consult the complete change log.

### `Change Log` added to `Help` menu

The `Change Log` (record of changes with each version) is now available as a `Help` menu item.

### Manual Available from `Help` Menu

The CircuitGear manual in .pdf format is now available from the `Help` menu. Clicking on that menu item automatically invokes Acrobat Reader.

### Channel Disable/Invert

The original `Channel Disable` button (one on each channel) has been moved to an `Option` menu on each channel, so we could add a `Channel Invert` invert function (figure 31).

There are now on-screen indicators of `Channel Disable` and `Channel Invert`.

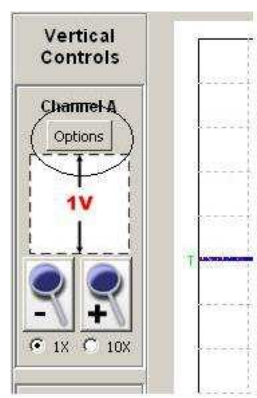



Figure 31: Option Menu

### Spectrum Display Cursor Control

For fine control over the spectrum frequency readout. the left and right cursor buttons on the keyboard now move the spectrum display cursor one pixel to the left or right.

## Network Analyser Labels and Phase Range

These items apply to the CGR-101 when in Network Analyser mode: `Hardware -> Network Analyser Mode`

The network analyser frequency labels can now be viewed in fixed-point or scientific notation. This is selectable under the `View` menu.

When the phase is close to the top or the bottom of the display, small phase changes can cause it to *jitter* in an annoying fashion from one limit to the other. For a signal that changes over a wide range of phase, this cannot be entirely prevented. However, it can be reduced by adjusting the vertical scale of the phase display.

Right click in the area of the phase measurement and select one of five possible phase ranges for the display.

## Check for Updates

The software will now check the Syscomp website and advise if there is a later version of the software. You can do that manually by selecting `Check for Update` in the `Help` menu. Alternatively, you can configure the software to do that automatically every time it starts. Select the `Check for Updates on Startup` menu item.

# 18 Sources of Information

## 18.1 News Groups

`comp.lang.tcl`
Internet News Group

## 18.2 Websites

`http://www.syscompdesign.com`
Latest information on Syscomp instruments and supporting software.
Many useful application notes and project descriptions.

`http://www.tcl.tk/`
Home of the Tcl Developer Xchange. Pointers to information and software downloads.

`http://www.activestate.com/`
ActiveState is a commercial firm that sells various programming tools provides a home for the Tcl/Tk language. Free versions of Tcl/Tk are available for download from their site.

## 18.3 Paper

*Scripting: Higher Level Programming for the 21st Century*
John K. Ousterhout
IEEE Computer magazine, March 1998
Currently at: `http://home.pacbell.net/ouster/scripting.html`
Also at: `http://www.tcl.tk/doc/scripting.htm`
The definitive paper on Tcl/Tk and scripting languages in general. Ousterhoot shows a Table of Applications which have been coded in Tcl/Tk and in the C language, and the relative effort and time required for each implementation.

## 18.4 Textbooks

*Practical Programming in Tcl and Tk, 4th Edition*
Brent B. Welch & Ken Jones with Jeffery Hobbs
Prentice Hall PTR, 2003
The definitive reference for Tcl and Tk. Includes CDROM with Tcl and examples.

*Tcl and the Tk Toolkit*
John K. Ousterhout
Addison-Wesley, 1994
Now somewhat dated, but a still useful introduction to Tcl/Tk by the inventor of the language.

*Graphical Applications with Tcl & Tk, 2nd Edition*
Eric Foster-Johnson
M&T Books, 1997

Very accessible introductory textbook.

*Tcl/Tk Tools*
Mark Harrison
O'Reilly, 1997
Information on a number of extensions to Tcl/Tk.

*Effective Tcl/Tk Programming*
Mark Harrison, Michael McLennan
Addison Wesley, 1998
Techniques of design for Tcl/Tk programs.

*Tcl/Tk for Programmers*
Adrian Zimmer
IEEE Computer Society, 1998
An textbook on Tcl/Tk with an academic tone and exercises.

*Tcl/Tk for Dummies*
Tim Webster
IDG Books, 1997
A useful introduction to Tcl/Tk.

*Tcl/Tk for Real Programmers*
Clif Flynt
Academic Press, 1999
Medium to high-level material on Tcl/Tk

*Audio-Radio Handbook*
Section 2.17.2: Pink Noise Generator
Editor, Martin Giles
National Semiconductor, 1980
Passive pink noise filter design
Available from `http://www.audioxpress.com/bksprods/products/bkaa59.htm`