



**KiCad**

**April 25, 2019**

# Contents

<b>1</b>	<b>イントロダクション</b>	<b>1</b>
1.1	KiCad	1
1.2	KiCad のファイルとフォルダ	2
<b>2</b>	<b>インストールと設定</b>	<b>4</b>
2.1	ディスプレイ選択	4
2.2	デフォルト設定の初期化	4
2.3	デフォルト設定の変更	4
2.4	パスの設定	5
2.5	外部ユーティリティの初期化	6
2.5.1	テキスト・エディターの選択	6
2.5.2	PDF ビューアの選択	6
2.6	新規プロジェクトの作成	6
2.7	プロジェクトのインポート	7
<b>3</b>	<b>KiCad マネージャーの使用方法</b>	<b>8</b>
3.1	プロジェクト・マネージャー・ウィンドウ:	8
3.2	ユーティリティ起動ペイン	9
3.3	プロジェクト・ツリー・ビュー	9
3.4	トップ・ツールバー	10
<b>4</b>	<b>プロジェクト・テンプレート</b>	<b>11</b>
4.1	テンプレートの使用	11
4.2	テンプレートが置かれる場所:	12
4.3	テンプレートの作成	13
4.3.1	必須ファイル:	14
4.3.2	オプション・ファイル:	15

---

<b>I</b>	<b>バージョン4からバージョン5へのアップグレード</b>	<b>16</b>
<b>5</b>	<b>回路図シンボル・ライブラリー</b>	<b>18</b>
5.1	グローバル・シンボル・ライブラリー・テーブル	18
5.2	シンボル・ライブラリー・テーブルのマッピング	19
5.3	再マッピング・サーチの順番	20
5.4	シンボル名とシンボル・ライブラリー・ニックネームにおける制限	20
<b>6</b>	<b>シンボル・キャッシュ・ライブラリーの有効性</b>	<b>21</b>
<b>7</b>	<b>基板ファイル・フォーマットの変更</b>	<b>22</b>
7.1	グローバル・フットプリント・ライブラリー・テーブル	22

---

## リファレンス・マニュアル

### 著作権

このドキュメントは以下の貢献者により著作権所有 © 2010-2018 されています。あなたは、GNU General Public License ( <http://www.gnu.org/licenses/gpl.html> ) のバージョン 3 以降、あるいはクリエイティブ・コモンズ・ライセンス ( <http://creativecommons.org/licenses/by/3.0/> ) のバージョン 3.0 以降のいずれかの条件の下で、配布または変更することができます。

このガイドの中のすべての商標は、正当な所有者に帰属します。

### 貢献者

Jean-Pierre Charras, Fabrizio Tappero.

### 翻訳

Asuki Kono <asukiaaa AT gmail.com>, 2018. starfort <starfort AT nifty.com>, 2017-2019. Norio Suzuki <nosuzuki AT postcard.st>, 2015. yoneken <yoneken AT kicad.jp>, 2011-2015. silvermoon <silvermoon AT kicad.jp>, 2011-2015.

### フィードバック

バグ報告や提案はこちらへお知らせください:

- KiCad のドキュメントについて: <https://github.com/KiCad/kicad-doc/issues>
- KiCad ソフトウェアについて: <https://bugs.launchpad.net/kicad>
- KiCad ソフトウェアの翻訳について: <https://github.com/KiCad/kicad-i18n/issues>

### 発行日とバージョン

2015 年 5 月 21 日

# Chapter 1

## イントロダクション

### 1.1 KiCad

KiCad は回路図と PCB アートワーク作成のためのオープン・ソース・ソフトウェアです。KiCad は、以下の統一された外観のツール群がエレガントに調和しています。

- **KiCad:** プロジェクト・マネージャー
- **Eeschema:** 回路図エディターとコンポーネント・ライブラリー・エディター
- **Pcbnew:** 回路基板レイアウト・エディターとフットプリント・エディター
- **GerbView:** ガーバー・ビューア

さらに3つのユーティリティが含まれます:

- **Bitmap2Component:** ロゴ用のコンポーネント作成。ビットマップ画像から回路図コンポーネントとフットプリントを作成します。
- **PCBCalculator:** レギュレーターの分圧抵抗や電流に対する配線幅、伝送線路などの計算機
- **Pl\_Editor:** 図枠の編集

これらのツールは通常プロジェクト・マネージャーから起動されますが、独立したツールとして実行することもできます。

KiCad は基板サイズに制限はなく、最大 32 層までの配線層と 14 層のテクニカル層、4 層の補助層を扱えます。

KiCad はプリント基板作成に必要なすべてのファイルを作成できます。含まれるものは:

- フォトプロッタ用ガーバー・ファイル
- 穴あけファイル
- 部品配置ファイル

KiCad は GPL ライセンスに基づくオープン・ソースのため、オープン・ソース志向の電子機器作成プロジェクトに理想的なツールです。

KiCad は Linux、Windows と Apple macOS で使用できます。

---

## 1.2 KiCad のファイルとフォルダ

KiCad は特定の拡張子を持ったファイル（とフォルダ）を回路図と基板の編集に使用します。

プロジェクト・マネージャー・ファイル:

*.pro	現在のプロジェクトに関連するいくつかのパラメーターと、コンポーネント・ライブラリー・リストを保存する小さなファイル。
-------	--

回路図エディター:

*.sch	回路図ファイル。ただし使用しているコンポーネント自身の情報は含まない。
*.lib	コンポーネント・ライブラリー・ファイル。各コンポーネントの情報 (コンポーネント記号、ピンとフィールド情報) を含む。
*.dcm	コンポーネントライブラリーの説明ドキュメント。各コンポーネントに対するコメント、キーワード、データシートへの参照。
*_cache.lib	コンポーネントライブラリーのキャッシュファイル。回路図で使用しているコンポーネントのコピーを保存する。
sym-lib-table	シンボル・ライブラリーのリスト。(シンボル・ライブラリー・テーブル): 回路図エディターで読み込まれるシンボル・ライブラリーのリスト。

基板エディターのファイルとフォルダ:

*.kicad_pcb	図枠以外のすべての基板情報を含む基板ファイル。
*.pretty	フットプリント・ライブラリー・フォルダー。フォルダー自体がライブラリーになっています。
*.kicad_mod	フットプリント・ファイル。それぞれが一つのフットプリントの記述を含んでいます。
*.brd	古いフォーマットの基板ファイル。現在の基板エディターでも読み込み可能ですが、書き込みはできません。
*.mod	古いフォーマットのフットプリント・ライブラリー。フットプリント・エディターと基板エディターで読み込み可能ですが、書き込みはできません。
fp-lib-table	フットプリント・ライブラリーのリスト (フットプリント・ライブラリー・テーブル): 基板エディターやフットプリント・エディター、CvPcb で読み込まれるフットプリント・ライブラリーのリスト。含まれるフットプリント・ライブラリーには、様々なフォーマットのライブラリーを指定できます。

共通ファイル:

*.kicad_wks	独自の外観を持ったワークシートを使用したいユーザー向けの図枠設定ファイル。
*.net	回路図エディターが作成するネットリスト・ファイル。基板エディターはこのファイルを読み込みます。このファイルは、コンポーネントとフットプリントの対応を別ファイルにしておきたいユーザー向けに.cmp ファイルと関連付けられています。

スペシャル・ファイル:

*.cmp	回路図で使用しているコンポーネントとフットプリントの対応を保存しています。Pcbnew がこのファイルを作成し、Eeschema が取り込みます。例えば Pcbnew 上でフットプリントを変更コマンドでフットプリントを変更したとします。この変更を回路図に反映するため Eeschema で読み込みます。
-------	---

その他のファイル:

次のファイルは製造や文書化のために KiCad が作成します。

*.gbr	ガーバー・ファイル、製造用
*.drl	ドリル・ファイル (Excellon 形式)、製造用
*.pos	ポジション・ファイル (アスキー形式)、自動部品挿入用
*.rpt	レポート・ファイル (アスキー形式)、文書化用
*.ps	プロット・ファイル (postscript)、文書化用
*.pdf	プロット・ファイル (pdf 形式)、文書化用
*.svg	プロット・ファイル (svg 形式)、文書化用
*.dxf	プロット・ファイル (dxf 形式)、文書化用
*.plt	プロット・ファイル (HPGL 形式)、文書化用

## Chapter 2

# インストールと設定

### 2.1 ディスプレイ選択

Pcbnew と Gerbview はハードウェアで加速されたレンダラーを利用しているため、OpenGL v2.1 以上をサポートするビデオ・カードが必要です。

### 2.2 デフォルト設定の初期化

kicad/template にある **kicad.pro** がデフォルト設定ファイルです。このファイルは新規プロジェクトのテンプレートとして提供され、Eeschema がロードするライブラリー・ファイルのリストを設定するために使用されます。Pcbnew に関する他のパラメーター（デフォルトのテキスト・サイズ、デフォルトの伝送線路の太さ、など）もここに保持されます。

別のデフォルト設定ファイル **fp-lib-table** が存在することがあります。これはフットプリント・ライブラリー・リストを作成するために一度だけ使用されます; さもないと、このリストはゼロから作成されるでしょう。

### 2.3 デフォルト設定の変更

デフォルトの **kicad.pro** ファイルは、必要であれば自由に変更できます。

kicad/template/kicad.pro への書き込み権限があることを確認して下さい。

KiCad を起動して **kicad.pro** プロジェクトを読み込みます。

KiCad マネージャーから Eeschema を起動します。新しいプロジェクトを作るときに使いたいライブラリーのリストを設定するため Eeschema の設定を変更します。

KiCad マネージャーから Pcbnew を起動します。フットプリント・ライブラリー・リストなど Pcbnew の設定を変更します。Pcbnew は、フットプリント・ライブラリー・テーブルというライブラリー・リスト・ファイルを作成または更新します。二箇所に (fp-lib-table という名前の) ライブラリー・ファイルがあります: ホーム・ディレクトリーにある fp-lib-table ファイルは、すべてのプロジェクトで使用されます。そしてプロジェクトのディレクトリーにもある場合は、そのプロジェクト専用で使われます。

---

## 2.4 パスの設定

環境変数を使って KiCad が使うパスを定義できます。いくつかの環境変数は KiCad 自身によって内部で設定され、使用するライブラリーや 3D シェイプなどへのパスを指定するために使われます。

これは、（プロジェクトを別のコンピュータに送るときなど）絶対パスを事前に決められなかったり変更の可能性がある場合、また一つのベース・パスを多くの同様なアイテムで共有するような場合に役立ちます。いろいろな場所へインストールされる以下のものについて考慮してください:

- Eeschema コンポーネント・ライブラリー
- Pcbnew フットプリント・ライブラリー
- フットプリントの定義で使用される 3D シェイプ・ファイル

例えば、フットプリント・ライブラリー *connect.pretty* へのパスは、環境変数 **KISYSMOD** を使って定義すると、`${KISYSMOD}/connect.pretty` となるでしょう。

このオプションは、環境変数でパスを定義したり、もし必要ならば個別にパス定義するためにユーザー自身の環境変数を追加することができます。

**KiCad 環境変数:**

KICAD_PTEMPLATES	プロジェクトの作成で使用するテンプレート。(バージョン 5.0.0-rc2 以降は非推奨、代わりに KICAD_TEMPLATE_DIR を使用のこと) この変数を使用する場合は、ユーザー定義が必要です。
KICAD_SYMBOL_DIR	シンボル・ライブラリーのベース・パス。
KIGITHUB	フットプリント・ライブラリー・テーブルでよく使用されます。この変数を使用する場合は、ユーザー定義が必要です。
KISYS3DMOD	3D シェイプ・ファイルのベース・パス。通常は絶対パスを使用しないので定義する必要があります。
KISYSMOD	フットプリント・ライブラリー・フォルダのベース・パス。フットプリント・ライブラリーの名前に絶対パスを使用しない場合は定義する必要があります。
KICAD_TEMPLATE_DIR	KiCad によってインストールされたテンプレートの位置。
KICAD_USER_TEMPLATE_DIR	ユーザーテンプレートの位置。



環境変数 **KIPRJMOD** は 常に KiCad によって内部的に定義され、現在のプロジェクトの絶対パスであることに注意して下さい。

例えば、`${KIPRJMOD}/connect.pretty` は常に 現在のプロジェクト・フォルダー内にある `connect.pretty` フォルダー (pretty フットプリント・ライブラリー) です。

パスの設定を変更した場合は、パスに関連する問題を避けるために一旦 KiCad を終了して再起動してください。

## 2.5 外部ユーティリティの初期化

お気に入りのテキスト・エディターや PDF ビューアを定義できます。これらの設定は、テキストや PDF ファイルを開く際に利用されます。

これらの設定は、設定メニューからアクセスできます:



### 2.5.1 テキスト・エディターの選択

現在のプロジェクトにあるファイルをテキスト・エディターで閲覧・編集する前に、使用するテキスト・エディターを選択しておく必要があります。

使用したいテキスト・エディターを設定するには、設定 → テキスト・エディターを設定を選びます。

### 2.5.2 PDF ビューアの選択

デフォルトの PDF ビューアまたはユーザーが指定した PDF ビューアを使用できます。

デフォルトの PDF ビューアから変更するには、PDF ビューア・プログラムを指定するために 設定 → **PDF** ビューア → **PDF** ビューアを設定を使用し、それから 設定 → **PDF** ビューア → お気に入りの **PDF** ビューアを選択します。

Linux ではデフォルトの PDF ビューアは脆弱であることが知られています。そのため、あなた自身で PDF ビューアを指定することを推奨します。

## 2.6 新規プロジェクトの作成

回路図ファイル、プリント基板ファイル、補助ライブラリー、製造に使う各種ファイル (フォト・トレース、ドリルおよび自動部品配置のファイル) からなる KiCad プロジェクトを管理するためには、次のようにプロジェクトを作成することを推奨します:

- プロジェクトの作業ディレクトリーを作成する。ディレクトリーの作成は、KiCad または通常の方法を使用する。
- このディレクトリーに、” 新規プロジェクト作成” または” テンプレートから新規プロジェクトを作成” のアイコンから、KiCad で利用するプロジェクト・ファイル（.pro 拡張子）を作成します。



**警告**

KiCad プロジェクト毎にユニークなディレクトリーを使用する。単一のディレクトリーに複数のプロジェクトを混ぜないで下さい。

KiCad は、プロジェクト管理のためのパラメーター（回路図で使用するライブラリーのリストなど）が多数記録されている拡張子が.pro というファイルを作成します。プロジェクトの名前は、メインの回路図ファイル名、プリント基板のファイル名に使用されます。例えば example.pro というプロジェクトが example ディレクトリーに作成されたとすると、デフォルトのファイルは次のように作られます：

example.pro	プロジェクト管理ファイル。
example.sch	メインの回路図ファイル。
example.kicad_pcb	プリント基板ファイル。
example.net	ネットリスト・ファイル。
example.*	他のユーティリティ・プログラムによって作成される各種ファイル。
example-cache.lib	回路図エディターが自動的に作成するライブラリー・ファイル。（回路図に使用されているコンポーネントをバックアップとして保存）

## 2.7 プロジェクトのインポート

他のソフトウェアで作られたファイルを KiCad にインポートできます。いまのところ、Eagle 6.x 以上で作られた XML フォーマットのファイルのみに対応しています。

プロジェクトをインポートするために、インポートのためのファイル選択ダイアログから回路図か基板のファイルのどちらかを選択します。インポートする回路図と基板のファイルは同じベース名（例: project.sch と project.brd）である必要があります。ファイルを選択したら、インポート後に KiCad プロジェクトとして扱われるファイルの保存先ディレクトリーを選択します。

## Chapter 3

# KiCad マネージャーの使用法

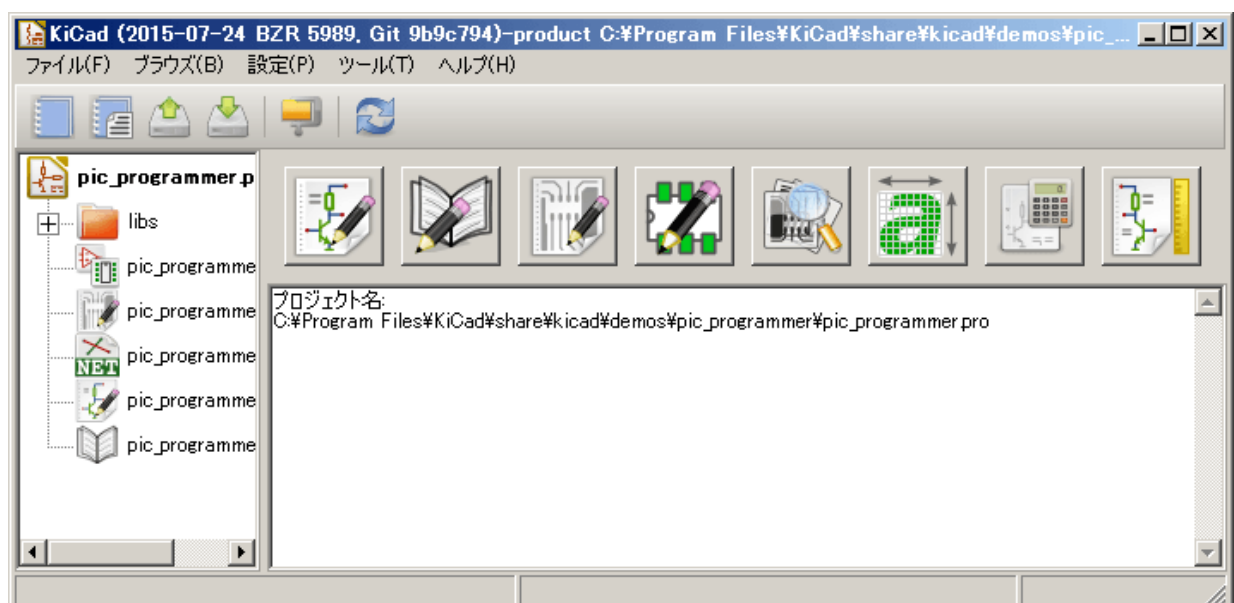
KiCad マネージャー (kicad または kicad.exe) は、デザインに必要な他のツール (エディターやガーバー・ビューア、その他ユーティリティ) を簡単に起動するツールです。

ツールを KiCad マネージャーから起動することには、利点がいくつかあります:

- 回路図エディターと基板エディター間での連携
- 回路図エディターとフットプリント選択ツール (CvPcb) 間での連携

しかしながら、編集できるのは現在のプロジェクトのファイルのみです。これらのツールを スタンド・アローン・モードで実行すると、プロジェクトに関係なく全てのファイルを開くことができますが、ツール間での連携は正しく機能しません。

### 3.1 プロジェクト・マネージャー・ウィンドウ:



KiCad のメイン・ウィンドウは、プロジェクト・ツリー・ビューおよび、様々なソフトウェア・ツールを起動するボタンのある起動ペイン、そしてメッセージ・ウィンドウで構成されています。メニューとツールバーは、プロジェクト・ファイルの作成、読み込みと保存に使用します。

## 3.2 ユーティリティ起動ペイン

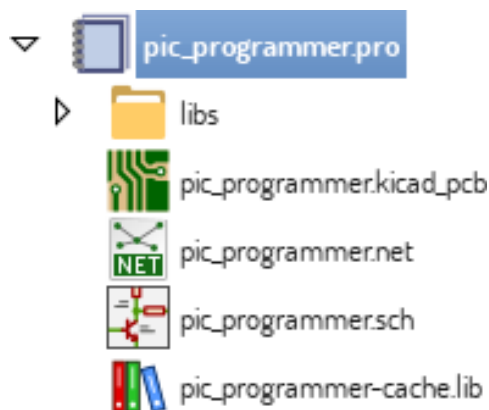
KiCad の全てのソフトウェア・ツールは、スタンドアローンでも実行できます。

起動ペインには、次のコマンドに対応する 8 つのボタン (左から右の順に 1 から 8) があります。



1	Eeschema	回路図エディター。
2	LibEdit	コンポーネント・エディターとコンポーネント・ライブラリー・マネージャー。
3	Pcbnew	プリント基板エディター。
4	FootprintEditor	フットプリント・エディターとフットプリント・ライブラリー・マネージャー。
5	Gerbview	ガーバー・ファイル・ビューア。ドリル・ファイルも閲覧可能。
6	Bitmap2component	白黒のビットマップ画像からロゴ用のフットプリントとコンポーネントを作成するツール。
7	PCB Calculator	線幅などの計算ツール。
8	P1 Editor	図枠の作成とカスタマイズをするエディター。

## 3.3 プロジェクト・ツリー・ビュー



回路図ファイルをダブル・クリックすると回路図エディターが起動し、この例では `pic_programmer.sch` というファイルを開きます。







基板ファイルをダブル・クリックするとプリント基板エディターが起動し、この例では `pic_programmer.kicad_pcb` というファイルを開きます。

プロジェクトツリーのファイル上で右クリックをすると削除など一般的などファイル操作を行えます。

### 3.4 トップ・ツールバー



KiCad のトップ・ツールバーからは、次の基本的なファイル操作を行えます。

	新規プロジェクト・ファイルの作成。kicad/template にテンプレート kicad.pro がある場合は、作業ディレクトリーにコピーされます。
	テンプレートからプロジェクトを作成。
	既存のプロジェクトを開く。
	現在のプロジェクトツリーを保存。
	プロジェクト全体の zip アーカイブを作成。回路図やライブラリー、プリント基板などが zip アーカイブにまとめられます。
	プロジェクト・ツリーの再構成と再表示。ツリーを変更した場合に必要なことがあります。

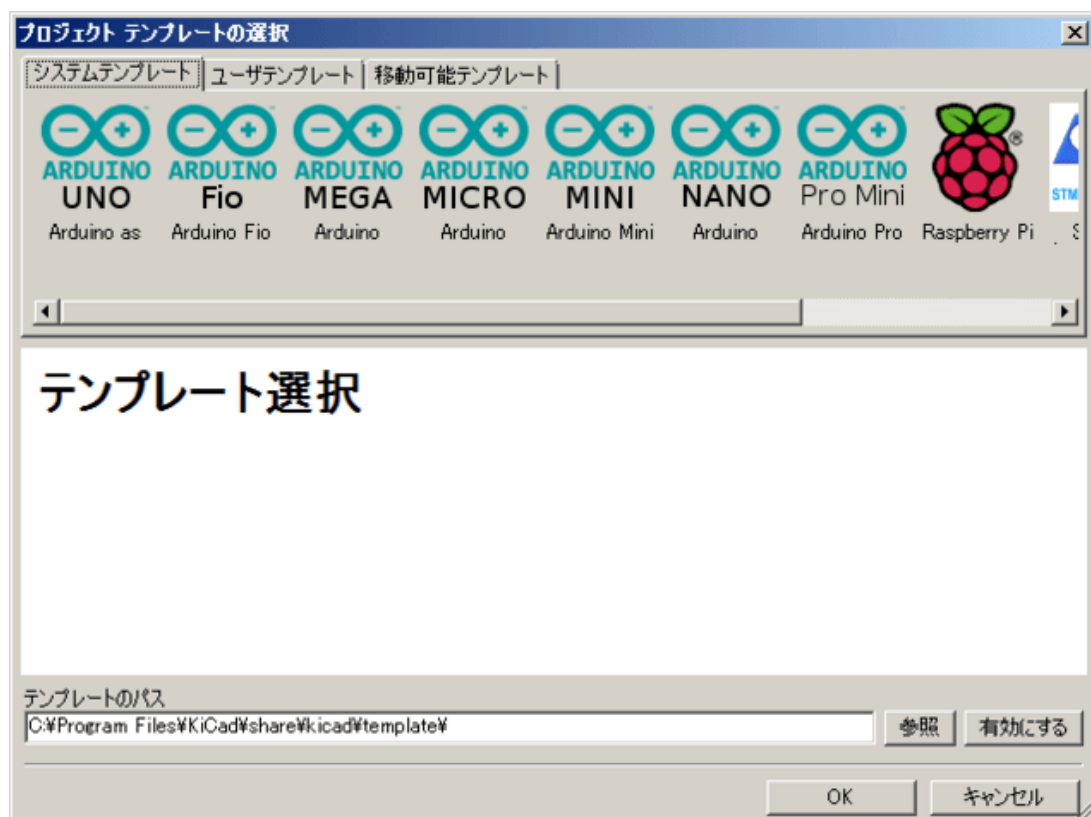
## Chapter 4

# プロジェクト・テンプレート

テンプレートを使うことで、あらかじめ設定が定義された新規プロジェクトを作成できます。テンプレートは、事前に定義された基板外形、コネクタ位置、回路要素、設計ルールなどを含んでいます。新規プロジェクト用ファイルの派生元として使われる完全な回路図や基板を含めることも可能です。

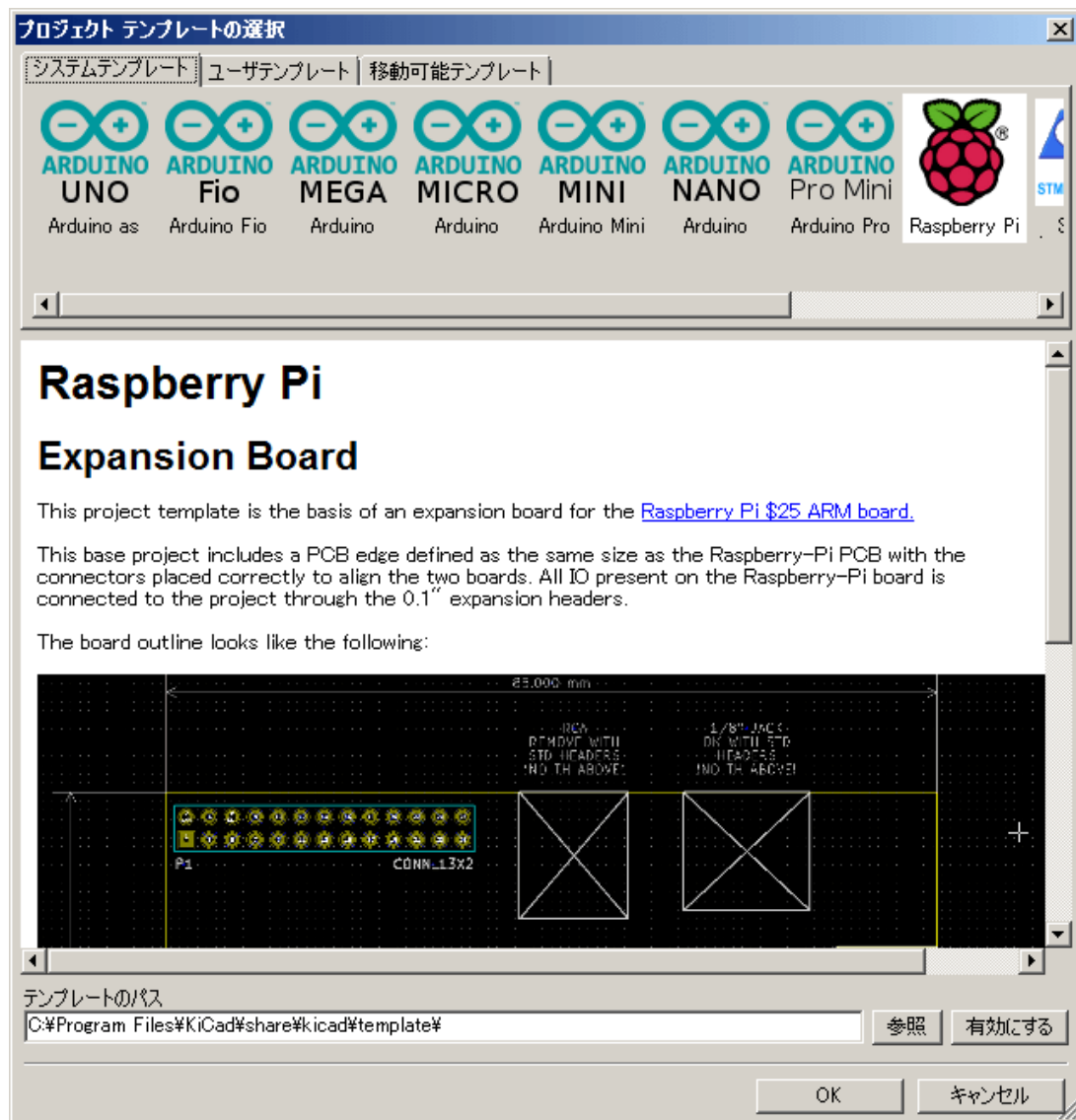
### 4.1 テンプレートの使用

ファイル → 新規プロジェクト → テンプレートから新規プロジェクトを作成とメニューをたどって、プロジェクト・テンプレートの選択ダイアログを開きます:



テンプレート・アイコンのシングル・クリックでテンプレートの情報が表示され、更に OK ボタンをクリックすると新しいプロジェクトが作成されます。テンプレート・ファイルは新規プロジェクトの場所へコピーされ、新しいプロジェクトの名前を反映したものへとリネームされます。

テンプレートを選択すると:



## 4.2 テンプレートが置かれる場所:

KiCad はこれらパスのテンプレートを読み込みます:

- 環境変数 KICAD\_USER\_TEMPLATE\_DIR に定義されたパス
- 環境変数 KICAD\_TEMPLATE\_DIR に定義されたパス
- システム・テンプレート: <kicad bin dir>/../share/kicad/template/
- ユーザー・テンプレート:

- Unix: ~/kicad/templates/
  - Windows: C:\Documents and Settings\username\My Documents\kicad\templates
  - Mac: ~/Documents/kicad/templates/
- 環境変数 KICAD\_PTEMPLATES が定義されている場合は、第 3 のタブ、ポータブルテンプレートがあります。このタブには、KICAD\_PTEMPLATES パスで見つかったテンプレートのリストが表示されます。(廃止予定)

## 4.3 テンプレートの作成

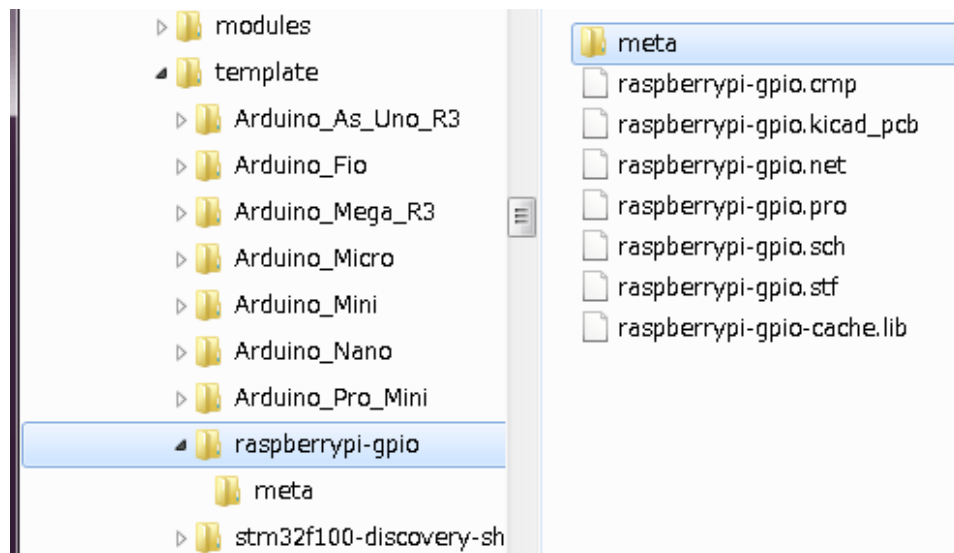
テンプレートの名前は、テンプレート・ファイルが保存されているディレクトリーの名前です。**meta** という名前のサブ・ディレクトリー内にあるメタデータ・ディレクトリーは、テンプレート情報ファイルを含んでいます。

テンプレートからプロジェクトを作成すると、**meta** を除く、テンプレートにある全てのファイルとディレクトリーは新規プロジェクトのパスにコピーされます。

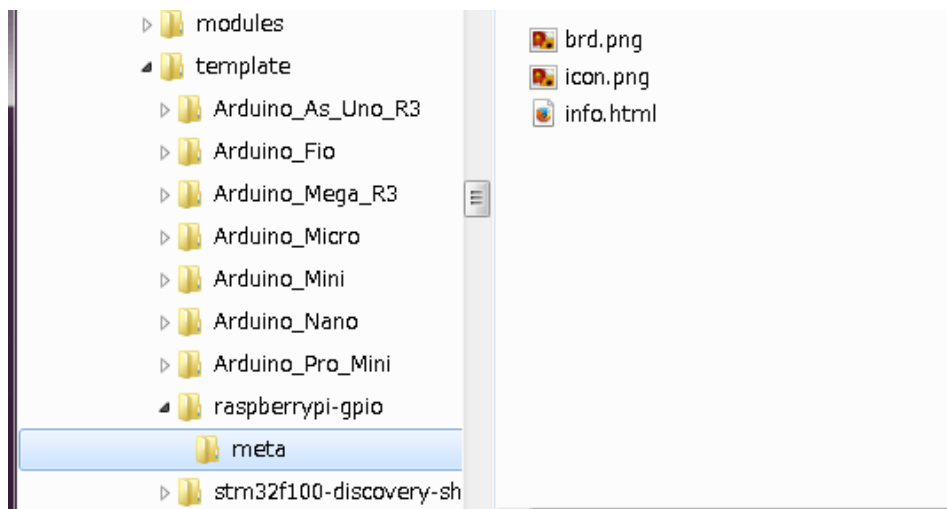
プロジェクトがテンプレートから作成された場合、テンプレートの名前で始まる全てのファイルとディレクトリーは、拡張子部分を除いて、新規プロジェクトのファイル名で名前を置換されます。

メタデータは 1 個の必須ファイルからなり、オプション・ファイルを含んでいることもあります。全てのファイルはユーザーがテキスト・エディターを使うか既存の KiCad プロジェクト・ファイルから作成する必要があり、要求されたディレクトリー構造どおりに配置されなければなりません。

ここで例として表示するのは、**raspberrypi-gpio** テンプレート用のプロジェクト・ファイルです:



そしてこれは、メタデータ・ファイルです:



### 4.3.1 必須ファイル:

meta/info.html	HTML フォーマットで書かれたテンプレートについての情報。
----------------	--------------------------------

<title> タグは、ユーザーがテンプレートを選択する時に表示される実際のテンプレートの名前を決定します。プロジェクトのテンプレート名が長すぎると短縮されることに注意して下さい。フォントのカーニングにより、通常は7もしくは8文字表示となります。

HTML を使用することで、新しい書式を作ることなく簡単に画像を行間に挿入できます。このドキュメントでは基本的な HTML タグのみ使用できます。

これは **info.html** ファイルのサンプルです:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<HTML>
<HEAD>
<META HTTP-EQUIV="CONTENT-TYPE" CONTENT="text/html;
charset=windows-1252">
<TITLE>Raspberry Pi - Expansion Board</TITLE>
</HEAD>
<BODY LANG="fr-FR" DIR="LTR">
<P>This project template is the basis of an expansion board for the
<A HREF="http://www.raspberrypi.org/" TARGET="blank">Raspberry Pi $25
ARM board.</A> <BR><BR>This base project includes a PCB edge defined
as the same size as the Raspberry-Pi PCB with the connectors placed
correctly to align the two boards. All IO present on the Raspberry-Pi
board is connected to the project through the 0.1" expansion
headers. <BR><BR>The board outline looks like the following:
</P>
<P><IMG SRC="brd.png" NAME="brd" ALIGN=BOTTOM WIDTH=680 HEIGHT=378
BORDER=0><BR><BR><BR><BR>
</P>
<P>(c)2012 Brian Sidebotham<BR>(c)2012 KiCad Developers</P>
</BODY>
```

</HTML>

4.3.2 オプションナル・ファイル:

meta/icon.png	64 x 64 ピクセルの PNG アイコン・ファイル。テンプレート選択ダイアログでクリック可能なアイコンとして使用されます。
---------------	---

上のダイアログにおける基板ファイルの画像のような meta/info.html で使われる他の画像ファイルは、同様にこのフォルダに置かれます。

## Part I

# バージョン4からバージョン5へのアップグレード

## レード

バージョン5の開発において、旧バージョンの KiCad で作られたプロジェクトに影響を与えるような動作仕様上の変更がありました。このセクションは、KiCad バージョン5へアップグレードするにあたって、円滑に移行する方法を明記したものです。

## Chapter 5

# 回路図シンボル・ライブラリー

回路図シンボル・ライブラリーは、もうシンボル (バージョン4ではコンポーネントと呼称) 検索リストを使ってアクセスできません。シンボル・ライブラリーは、フットプリント・ライブラリー・テーブルと同様、シンボル・ライブラリー・テーブルによって管理されるようになりました。この変更は大幅な改善ですが、いくつかの回路図はバージョン5に変換する時に人手による介入が必要になるでしょう。

以前のバージョンでは、Eeschema ファイルでシンボルを配置するとき、KiCad は検索にライブラリー・ファイルのリストを使っていました。シンボルを配置するときには、各パスが検索され、シンボル名を保持している最初のライブラリーが使われていました。

バージョン5から、KiCad はシンボル名にニックネームの接頭辞を持っており、ニックネームとライブラリー・パスを一致させる検索テーブルがシンボルを保持しているライブラリーを配置するために使われます。このテーブルは”シンボル・ライブラリー・テーブル”と呼ばれ、ユーザーの KiCad 設定ディレクトリーと現在ロードされているプロジェクトのディレクトリーに保存された設定ファイルから構築されます。

KiCad プロジェクトをバージョン4からバージョン5へアップグレードするためには、全てのライブラリー・ファイルにニックネームを作成する必要があり、回路図のシンボル名は合致するニックネームの接頭辞を持つ必要があります。

### 5.1 グローバル・シンボル・ライブラリー・テーブル

Eeschema バージョン5は最初の実行時にグローバル・シンボル・テーブルを自動的に作成します。このステップをスキップして、自分自身でグローバル・シンボル・テーブルを作ることもできます。これが必要なのは、KiCad のシンボル・ライブラリーを全く使わない場合のみです。もしくは、自動的に生成されたグローバル・シンボル・テーブルを変更する方がより簡単です。

---

#### 注意

**シンボル・ライブラリー・リポジトリ**に追従させようとしても、KiCad 本体はデフォルトのグローバル・シンボル・ライブラリー・テーブルになされた変更を自動的に追跡しません。グローバル・シンボル・ライブラリー・テーブルは自分自身で最新に保つ必要があります。

---

## 5.2 シンボル・ライブラリー・テーブルのマッピング

再マッピングされていない回路図が開かれる時は、常にシンボルの再マッピングが自動実行されます。最も効果的に再マッピングするには、事前にやっておくべき幾つかのステップがあります。

---

### 注意

もし KiCad の開発版を使用しているなら、KiCad ライブラリーと共にインストールされたテンプレート・フォルダーから、あるいは、[KiCad ライブラリーリポジトリ](#)から、KiCad ユーザー設定フォルダーへ完全なデフォルトのグローバル・シンボル・ライブラリー・テーブル・ファイル (sym-lib-table) をコピーします。

---



### 警告

再マッピングされた回路図は旧バージョンの KiCad との互換性がありません。リマップ・シンボル・ダイアログは回路図ファイルのバックアップを実行するためのもので、もし自分自身で再マッピングをする場合は同じことをする必要があります。

---

1. もし可能なら、システムに KiCad バージョン4をインストールしたままにしておいて下さい。さもなければ、KiCad と一緒に配布されたシンボル・ライブラリーを使用できなくなります。
2. もし Eeschema バージョン4の開始時に欠落したライブラリーについての警告が出た場合には、回路図の再マッピングを試みる前に欠落したライブラリーを回路図にあるシンボルが含まれるように修正しなければなりません。さもなければ、正しいシンボルを見つけることができず、回路図にあるシンボルのリンクは壊れたままになってしまうでしょう。これは回路図にあるシンボルを右クリックしてテストすることができ、キャッシュされたライブラリーからロードされたシンボルではないことを確認できます。もしシンボルがキャッシュされたライブラリーからロードされていた場合、Eeschema はシステムまたはプロジェクトのシンボル・ライブラリーから部品を見つけることができません。もしキャッシュされた部品を別のプロジェクトで有効化する必要があるなら、システムまたはプロジェクトのシンボル・ライブラリー内にその部品を統合する必要があります。
3. シンボルのリカバーが再マッピングのプロセス中に要求された場合、無視してはいけません。シンボル・リカバーの失敗は回路図中の壊れたシンボルのリンクや誤ってリンクされたシンボルの原因となります。
4. 再マッピングのプロセス中、グローバル・シンボル・ライブラリー・テーブルに見つからなかったシンボル・ライブラリーは、プロジェクト固有のシンボル・ライブラリー・テーブルを作るために使用されます。もし、お好みなら、グローバル・シンボル・ライブラリー・テーブルに手作業で移動させることもできます。
5. 最も正確に再マッピングするには、プロジェクトのキャッシュ・ファイル (project-name-cache.lib) をコピーして別ファイルとして作られたプロジェクト・ライブラリーをシンボル・ライブラリー・リストの先頭に追加します。これを実行するためには、シンボル・ライブラリー・テーブルが実装される以前のバージョンの KiCad を使用しなければなりません。

---

### 注意

再マッピングで起きた問題の修正を試みるためのツールが提供されています。回路図でシンボル・ライブラリーのリンクが失われていた場合、“ ツール → シンボル・ライブラリー・リファレンスを編集…” というメニュー項目を開き、“ 孤立したコンポーネントをマップ” ボタンをクリックすることで修正できます。

---

## 5.3 再マッピング・サーチの順番

シンボルを再マッピングするとき、KiCad は以下のような順番でシンボルにライブラリーを割り当てます:

1. グローバル・シンボル・ライブラリー・テーブル: シンボルはグローバル・シンボル・ライブラリー・テーブルへと優先的にマップされます。(もし存在していれば)
2. プロジェクト固有ライブラリー: グローバル・シンボル・ライブラリー・テーブルにはない、プロジェクト・ライブラリー・リストにあるライブラリーが次に検索されます。
3. プロジェクト・キャッシュ・ファイル: もしシンボルが上のリストにあるライブラリーに存在しなかった場合、初めてレスキューが実行されます。— シンボルがこの新しいレスキュー・ライブラリーにマップされる前にキャッシュからコピーが作られ、*proj-rescue.lib* に置かれます --

## 5.4 シンボル名とシンボル・ライブラリー・ニックネームにおける制限

シンボル名に <SPACE>, ':', '/' を含めることはできません。

ライブラリー・ニックネームに <SPACE>, ':' を含めることはできません。

これらの文字を含む既存のシンボル名は、関連した回路図とライブラリー・ファイルを手作業で編集してリネームしなければなりません。

## Chapter 6

# シンボル・キャッシュ・ライブラリーの有効性

キャッシュ・ライブラリーは、もはやシンボル・ライブラリー・ビューアやシンボル・ライブラリー・エディターで見ることができません。いかなる変更も次に回路図を保存すると上書きされるので、キャッシュを編集してはなりません。

## Chapter 7

# 基板ファイル・フォーマットの変更

基板ファイル・フォーマットに影響がある、いくつかの新しい機能が Pcbnew に追加されました。基板設計でこれらの新しい機能を使用することは、以前のバージョンの Pcbnew で開くときの障害となります。

- 角を丸めた長方形のフットプリント・パッド
- カスタム形状のフットプリント・パッド
- 4文字以上のフットプリント名
- 単一レイヤー以上のキープアウト (禁止) ゾーン
- インチに代わって、ミリメートル単位で保存された 3D モデルのオフセット
- フットプリント・テキストのロック

### 7.1 グローバル・フットプリント・ライブラリー・テーブル

**フットプリント・ライブラリー・リポジトリ**に追従させようとしても、KiCad 本体はデフォルトのグローバル・フットプリント・ライブラリー・テーブルになされた変更を自動的に追跡しません。グローバル・フットプリント・ライブラリー・テーブルは自分自身で最新に保つ必要があります。