



kiCad



kiCad

KiCad

July 18, 2018

Contents

1	イントロダクション	1
1.1	KiCad	1
1.2	KiCad のファイルとフォルダ	2
2	インストールと設定	4
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
3	KiCad マネージャーの使用方法	8
3.1	プロジェクト・マネージャ・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
4	プロジェクト・テンプレート	11
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

I	Upgrading from Version 4 to Version 5	16
5	Schematic Symbol Libraries	18
5.1	Global Symbol Library Table.	18
5.2	Symbol Library Table Mapping	18
5.3	Remapping Search Order	19
5.4	Symbol Names and Symbol Library Nickname Limitations	20
6	Symbol Cache Library Availability	21
7	Board File Format Changes	22
7.1	Global Footprint Library Table.	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. 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・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・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と、コンポーネント・ライブラリ・リストを保存する小さなファイル。
-------	--

回路図エディタ:

*.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・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・ライブラリのリスト。(シンボル・ライブラリ・テーブル): 回路図エディタで読み込まれるシンボル・ライブラリのリスト。

基板エディタ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・フォルダー。フォルダー自体がライブラリになっています。
*.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。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・ライブラリのリスト (フットプリント・ライブラリ・テーブル): 基板エディタやフットプリント・エディター、CvPcb で読み込まれるフットプリント・ライブラリのリスト。含まれるフットプリント・ライブラリには、様々なフォーマットのライブラリを指定できます。

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・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（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・リストを作成するために一度だけ使用されます; さもなければ、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 マネージャから Pcbnew を起動します。フットプリント・ライブラリ・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・テーブルというライブラリ・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・ファイルがあります: ホームディレクトリにある 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
- Pcbnew フットプリント・ライブラリ
- 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 *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されるテンプレート。この変数を使用する場合には、ユーザーによる定義が必要です。
KICAD_SYMBOL_DIR	シンボル・ライブラリのベース・パス。
KIGITHUB	フットプリント・ライブラリ・テーブルでよく使用されます。この変数を使用する場合には、ユーザーによる定義が必要です。
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・フォルダのベース・パス。フットプリント・ライブラリ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。



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・フォルダ内にある *connect.pretty* フォルダ (pretty フットプリント・ライブラリ) 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を作成する。ディレクトリの作成は、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を使用する。単一のディレクトリ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 (回路図で使用されるライブラリのリストなど) 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:

<code>example.pro</code>	プロジェクト管理ファイル。
<code>example.sch</code>	メインの回路図ファイル。
<code>example.kicad_pcb</code>	プリント基板ファイル。
<code>example.net</code>	ネットリスト・ファイル。
<code>example.*</code>	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。
<code>example-cache.lib</code>	回路図エディタが自動的に作成するライブラリ・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

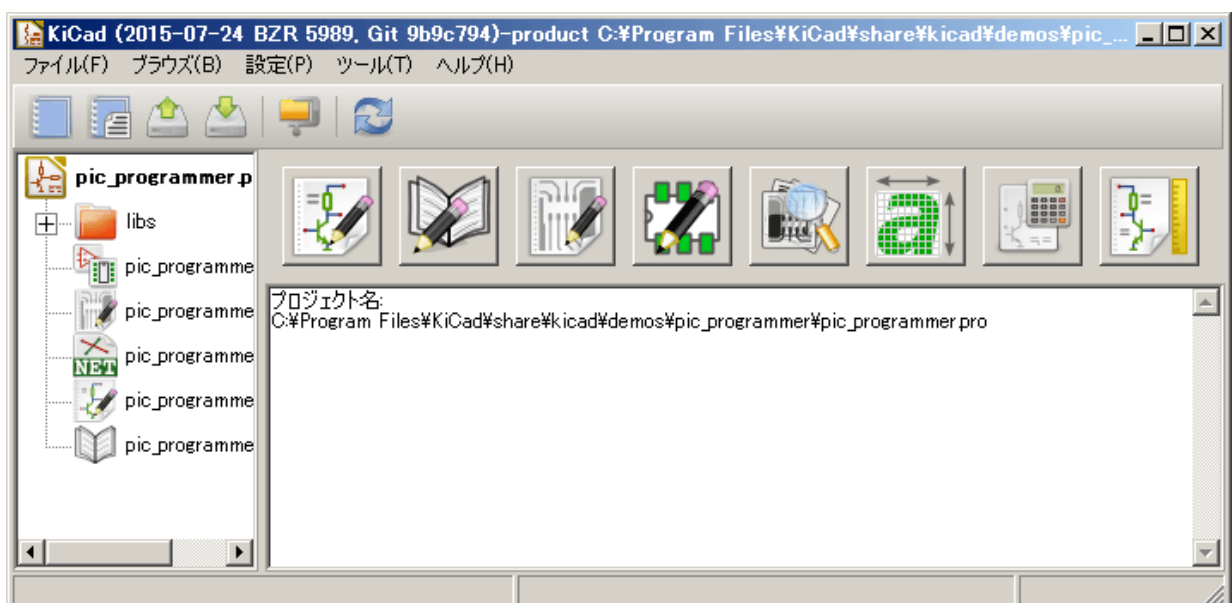
KiCad マネージャ (kicad または kicad.exe) は、デザインに必要な他のツール (エディターやガーバー・ビューア、その他ユーティリティ) を簡単に起動するツールです。

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:

- 回路図エディタと基板エディタ間での連携
- 回路図エディタ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・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

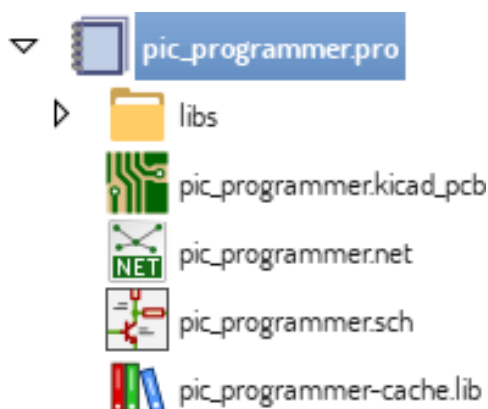
KiCad の全てのソフトウェア・ツールは、スタンドアローンでも実行できます。

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・エディタとコンポーネント・ライブラリ・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・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	Pl 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、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。

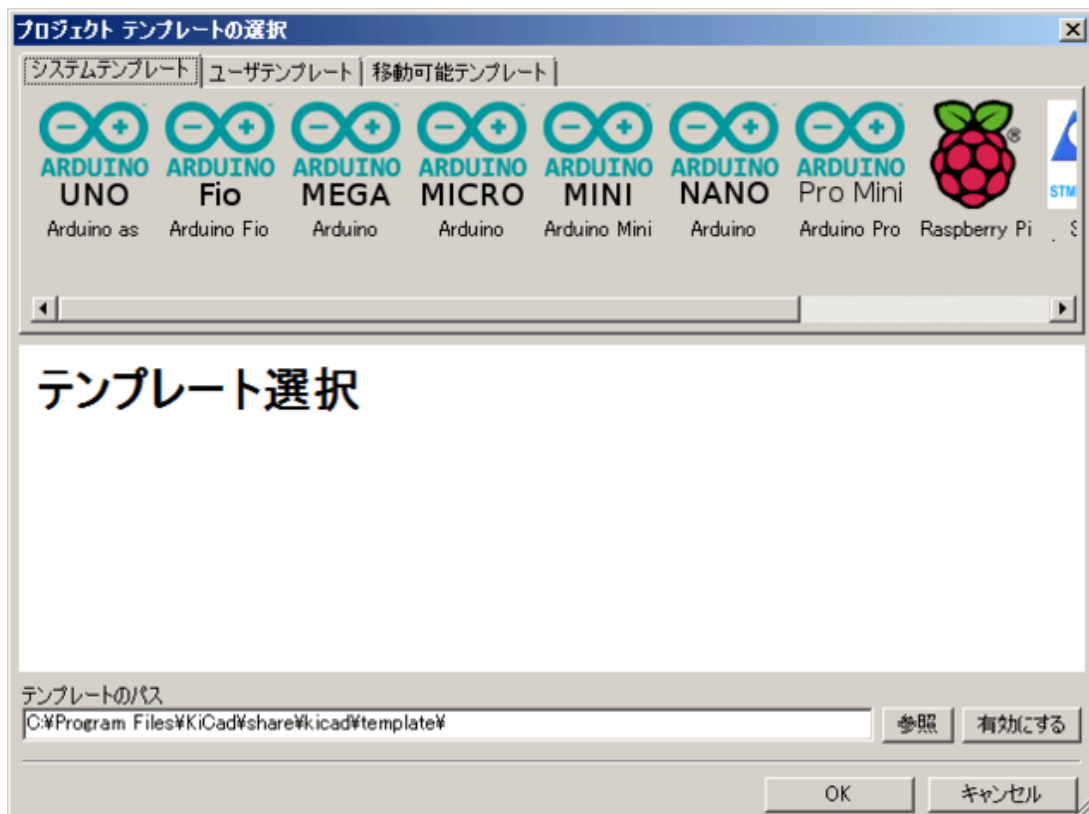
Chapter 4

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。

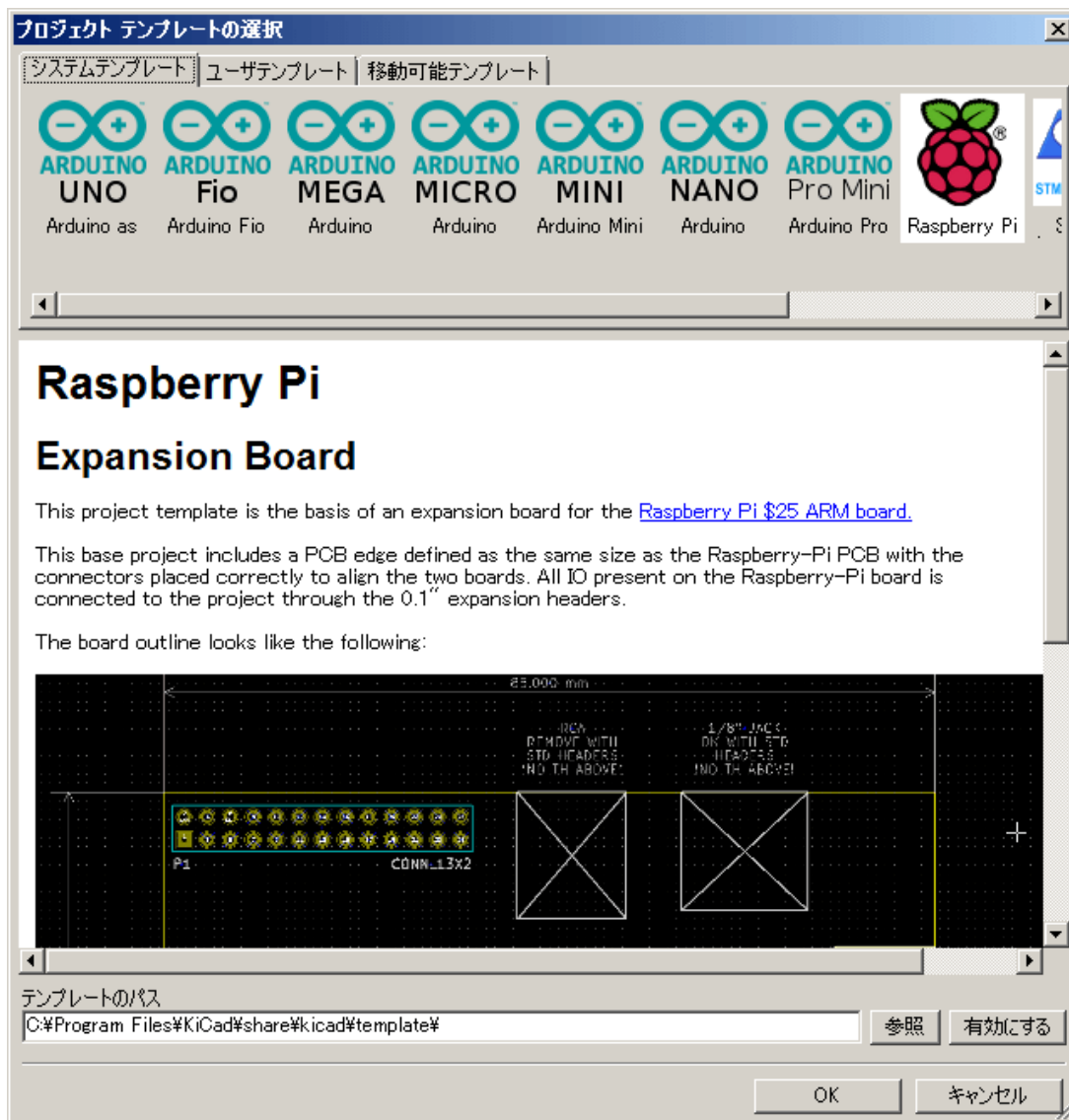
4.1 テンプレートの使用

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:

- 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、移動可能テンプレートのタブもあります。この移動可能テンプレート・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。

4.3 テンプレートの作成

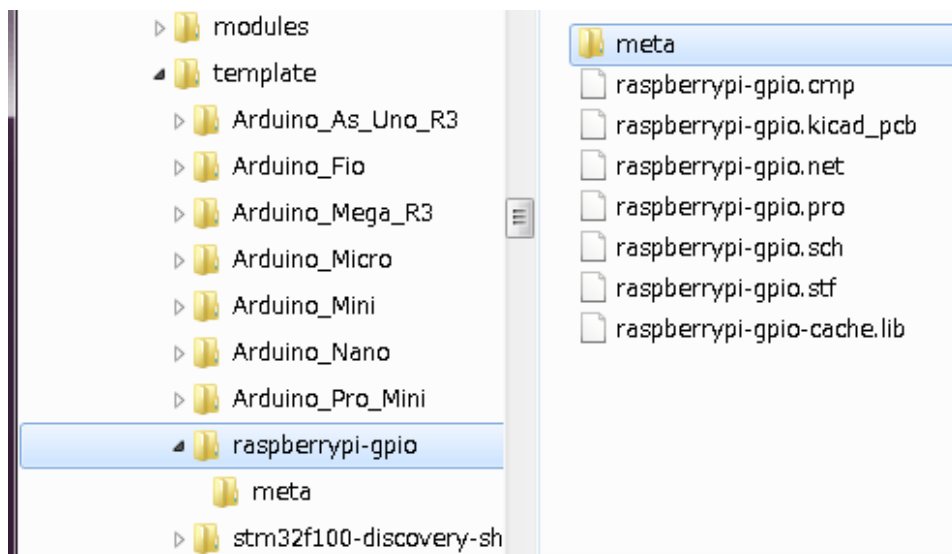
テンプレートの名前は、テンプレート・ファイルが保存されているディレクトリの名前です。meta という名前のサブ・ディレクトリ内にあるメタデータ・ディレクトリは、テンプレート情報ファイルを含んでいます。

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。

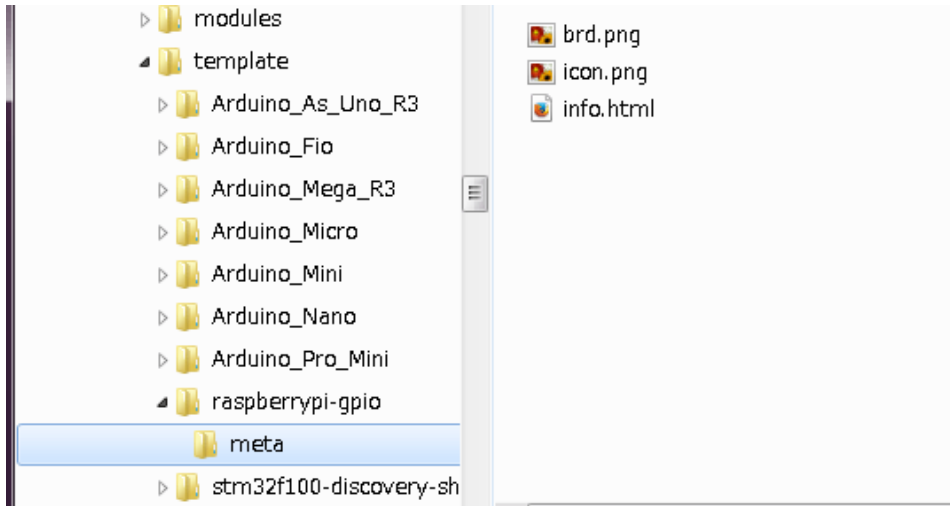
プロジェクトがテンプレートから作成された場合、テンプレートの名前で始まる全てのファイルとディレクトリは、拡張子部分を除いて、新規プロジェクトのファイル名で名前を置換されます。

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>
<META NAME="GENERATOR" CONTENT="LibreOffice 3.6 (Windows)">
<META NAME="CREATED" CONTENT="0;0">
<META NAME="CHANGED" CONTENT="20121015;19015295">
</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

Upgrading from Version 4 to Version 5

Changes were made to the behavior to KiCad during the version 5 development that can impact projects created with older versions of KiCad. This section serves as a guide to ensure the smoothest possible path when upgrading to version 5 of KiCad.

Chapter 5

Schematic Symbol Libraries

Schematic symbol libraries are no longer accessed using a symbol (referred to as components in version 4) look up list. Symbol libraries are now managed by a symbol library table that behaves similarly to the footprint library table. This change is a significant improvement, but some schematics may need manual intervention when being converted to version 5.

In previous versions, KiCad used a list of library files to search when locating symbols in the Eeschema file. When locating a symbol, each path would be searched and the first library that held the symbol name would be used.

From v5, KiCad symbol names are prefixed with a nickname, and a [lookup table matching nicknames to library paths](#) is used to locate the library which holds the symbol. The table is called the *symbol library table* and built from configuration files stored in the user's KiCad configuration directory and the currently loaded project directory.

To upgrade a KiCad project from v4 to v5, nicknames for all of the library files need to be created and then schematic symbol names need to be prefixed with the correct nickname.

5.1 Global Symbol Library Table.

Eeschema v5 will automatically create a global symbol table when first started. You will be given a chance to skip this and create your own global symbol table by hand. You only need to do this if don't use KiCad symbol libraries at all. Otherwise it is easier to modify the automatically generated global symbol table.

注意

If you track the [symbol library repository](#), changes made to the default global symbol library table are not tracked by KiCad. You will have to manually keep the global symbol library table up to date.

5.2 Symbol Library Table Mapping

Automatic remapping of symbols will be executed whenever a schematic is opened that has not been remapped. There are a few steps you should take ahead of time in order for the remapping to be the most effective.

注意

If you have been using a development build of KiCad, copy the full default global symbol library table file (sym-lib-table) from the template folder installed with the KiCad libraries or from the [KiCad library repo](#) to your KiCad user configuration folder. This will replace the empty one (most likely) created by Eeschema. If you do not do this, you will most likely end up with a bunch of broken symbol links.



警告

Remapped schematics will not be compatible with older versions of KiCad. The Remap Symbols dialog will make a backup of your schematic files and you should do the same if you remap manually.

1. If possible, keep version 4 of KiCad installed on your system unless you have never used any of the symbol libraries distributed with KiCad.
 2. If you get warning about missing libraries when you start version 4 of Eeschema, make sure to fix the missing libraries if they contain symbols that are in the schematic before you attempt to remap your schematic. Otherwise, the correct symbol will not be found and you will end up with broken symbol links in your schematic. You can test this by left clicking on a symbol in the schematic and verifying that the symbol is not being loaded from the cache library. If a symbol is being loaded from the cache library, Eeschema cannot find your part in the system or project symbol libraries. If you need a cached part to be available to other projects on your system, you will need to integrate it into a system or project library manually.
 3. If symbol recovery is required during the remapping process, do not dismiss it. Failure to recover symbols will result in broken symbol links or the wrong symbol being linked in the schematic.
 4. During the remapping process, symbol libraries not found in the global symbol library table will be used to create a project specific symbol library table. You can move them manually to the global symbol library table if that is your preference.
 5. For the most accurate remapping, create a project library by copying the project cache file (project-name-cache.lib) to a different file and add it to the top of the symbol library list. You must use a version of KiCad prior to the symbol library table implementation in order to do this.
-

注意

A tool has been provided to attempt to fix remapping issues. If there are missing symbol library links in a schematic, they can be fixed by opening the "Tools→Edit Symbol Library References..." menu entry and clicking on the "Map Orphans" button.

5.3 Remapping Search Order

When remapping symbols, KiCad proceeds in the following order to assign the library to a symbol:

1. Global Symbol Library Table: Symbols are preferentially mapped to the global symbol library table, if one exists.
-

2. Project specific libraries: Libraries listed in the project library list that are not in the global symbol library table are searched next.
3. Project cache file: If a symbol doesn't exist in the listed libraries above, it is first rescued — a copy is made from the cache and placed in the *proj-rescue.lib* — before the symbol is mapped to this new, rescue library.

5.4 Symbol Names and Symbol Library Nickname Limitations

Symbol names may not contain <SPACE>, ':', '/'.

Library nicknames may not contain <SPACE>, ':', '/'.

Existing symbol names with these characters must be renamed by manually editing the relevant schematic and library files.

Chapter 6

Symbol Cache Library Availability

The cache library is no longer shown in either the symbol library viewer or the symbol library editor. The cache should never be edited because any changes are overwritten by the next schematic save.

Chapter 7

Board File Format Changes

Several new features have been added to Pcbnew which impact the board file format. Using these new features in board designs will prevent them from being opened with previous versions of Pcbnew.

- Rounded rectangle footprint pads.
- Custom shape footprint pads.
- Footprint pad names longer than four characters.
- Keep out zones on more than a single layer.
- 3D models offset saved as millimeters instead of inches.
- Footprint text locking.

7.1 Global Footprint Library Table.

If you track the [footprint library repository](#), changes made to the default global footprint library table are not tracked by KiCad. You will have to manually keep the global footprint library table up to date.
