KiCad ことはじめ

December 22, 2020

Contents

1 KiCad のご紹介 **1**

1.1 KiCad のダウンロードとインストール 2

1.1.1 GNU/Linux の場合 2

1.1.2 Apple macOS の場合 3

1.1.3 Windows の場合 3

1.2 サポート 3

2 KiCad 作業の流れ **4**

2.1 概要 4

2.2 フォワード／バックアノテーション 6

3 KiCad の使用 **7**

3.1 ショートカットキー 7

3.1.1 アクセラレーターキー 7

3.1.2 ホットキー 7

3.1.3 例 8

4 電気回路図の作図 **9**

4.1 Eeschema の使用 9

4.2 KiCad でのバス接続 22

5 プリント基板のレイアウト **25**

5.1 Pcbnew の使用 25

5.2 ガーバーファイルの生成 32

5.3 GerbView の使用 32

5.4 FreeRouter による自動配線 32

6 KiCad のフォワードアノテーション **34**

7 KiCad 回路図シンボルの作成 **36**

7.1 コンポーネントライブラリエディターの使用 36

7.2 コンポーネントのエクスポート、インポート、変更 38

7.3 quicklib による回路図コンポーネントの作成 39

7.4 大量ピンの回路図コンポーネントの作成 40

8 フットプリントの作成 **42**

8.1 フットプリント・エディタの使用 42

9 KiCad プロジェクトファイルの可搬性について **44**

10 KiCad ドキュメントの詳細 **46**

10.1 Web 上の KiCad ドキュメント 46

洗練された電子プリント回路基板の開発を成功に導く *KiCad* をマスターするための必須で簡潔なガイド

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 Attribution 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

David Jahshan, Phil Hutchinson, Fabrizio Tappero, Christina Jarron, Melroy van den Berg.

翻訳

je6bmq <je6bmq AT gmail.com>, 2018. starfort <starfort AT nifty.com>, 2017-2019. kinichiro <kinichiro.inoguchi AT gmail.com>, 2015. silvermoon <silvermoon AT kicad.jp>, 2011-2015. yoneken <yoneken 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:

- About KiCad documentation: https://gitlab.com/kicad/services/kicad-doc/issues
- KiCad ソフトウェアについて: https://gitlab.com/kicad/code/kicad/issues
- KiCad ソフトウェアの国際化 (i18n) について: https://gitlab.com/kicad/code/kicad-i18n/issues

発行日

2015 年 5 月 16 日

Chapter 1

KiCad のご紹介

KiCad は、電気回路図と基板アートワークを作成するためのオープンソースのソフトウェアツールです。その一つの外観の下に、KiCad は、以下の独立したソフトウェアツール群のすばらしい組み合わせを実現しています:

プログラム名	説明	ファイル拡張子
KiCad	プロジェクトマネージャ	`*.pro`
Eeschema	回路図とコンポーネントのエディター	`*.sch, *.lib, *.net`
Pcbnew	基板とフットプリントのエディター	`*.kicad_pcb, *.kicad_mod`
GerbView	ガーバーとドリルファイルのビューア	`*.g\*, *.drl, etc.`
Bitmap2Component	ビットマップをコンポーネントやフットプリントに変換	`*.lib, *.kicad_mod, *.kicad_wks`
PCB Calculator	コンポーネント、線幅、電気的に安全な間隔、カラーコード等のための計算機	なし
Pl Editor	図枠エディター	`*.kicad_wks`

注意
ファイル拡張子リストは完全ではなく、KiCad が扱うファイルの一部しか含んでいません。しかしながら、どのファイルが各々の KiCad アプリケーションで使われるかを基本的に理解するのに役立ちます。

KiCad は、複雑な電子基板の開発やメンテナンスにも使えるほど十分にバージョンアップを重ねてきたと言えます。

KiCad には基板サイズの制限がなく、最大 32 の導体レイヤー、最大 14 のテクニカルレイヤーそして最大 4 の補助レイヤーを簡単に扱うことができます。KiCad は、フォトプロッター用のガーバーファイル、ドリルファイル、コンポーネントロケーションファイル等のプリント基板作成に必要なファイルを全て作ることができます。

オープンソース (GPL ライセンスに基づく) であるため、KiCad は、オープンソース志向の電子機器作成向けプロジェクトに理想的なツールです。

インターネット上の KiCad ホームページ:

http://www.kicad-pcb.org/

1.1 KiCad のダウンロードとインストール

KiCad は GNU/Linux、Apple macOS 、Windows で動作します。以下の URL で最新のインストール手順とダウンロードのリンクが見つかります:

http://www.kicad-pcb.org/download/

重要項目

KiCad 安定版は KiCad Stable Release Policy に基づいて定期的にリリースされます。新機能は継続的に開発ブランチに追加されています。もしあなたが新機能の利用を希望し、テストを手伝っていただけるのであれば、あなたのプラットホーム向けに作られた最新の夜間ビルドをダウンロードしてください。夜間ビルドにはファイルの破損や不正なガーバーの生成のようなバグがあるかもしれませんが、KiCad 開発チームは新機能の開発中もできるだけ開発ブランチを使用可能にしておくことを目標にしています。

1.1.1 GNU/Linux の場合

KiCad の安定版はほとんどのディストリビューションパッケージマネージャーで kicad と kicad-doc として見つけることができます。もしあなたがお使いのディストリビューションが最新の安定版を提供していなければ、安定していないビルド用の指示書に従い、最新の安定版を選択してインストールを行ってください。

Ubuntu で、KiCad の安定版ではない夜間ビルドをインストールする最も簡単な方法は *PPA* と *Aptitude* によるものです。端末で次のようにタイプします:

```
sudo add-apt-repository ppa:js-reynaud/ppa-kicad
sudo aptitude update && sudo aptitude safe-upgrade
sudo aptitude install kicad kicad-doc-en
```

Debian で、KiCad のバックポートされたビルドをインストールする最も簡単な方法は次の通りです:

```
# Set up Debian Backports
echo -e "
# stretch-backports
deb http://ftp.us.debian.org/debian/ stretch-backports main contrib non-free
deb-src http://ftp.us.debian.org/debian/ stretch-backports main contrib non-free
" | sudo tee -a /etc/apt/sources.list > /dev/null

# Run an Update & Install KiCad
sudo apt-get update
sudo apt-get install -t stretch-backports kicad
```

Fedora で、KiCad の安定版ではない夜間ビルドをインストールする最も簡単な方法は *copr* によるものです。KiCad を copr でインストールするには copr で次のようにタイプします:

```
sudo dnf copr enable @kicad/kicad
sudo dnf install kicad
```

あるいは、ダウンロードしてコンパイル済みバージョンの KiCad をインストールしたり、直接ソースコードをダウンロードしてコンパイルし、KiCad をインストールすることもできます。

1.1.2 Apple macOS の場合

macOS 用安定版の KiCad は、以下の URL で見つけることができます: http://downloads.kicad-pcb.org/osx/stable/

安定していない夜間の開発ビルドは、以下の URL で見つけられます: http://downloads.kicad-pcb.org/osx/

1.1.3 Windows の場合

Windows 用の安定版の KiCad は、以下の URL で見つけることができます: http://downloads.kicad-pcb.org/-windows/stable/

Windows 用の夜間の開発ビルドは、以下の URL で見つけられます: http://downloads.kicad-pcb.org/windows/

1.2 サポート

何か思いついたり、発言したいことがあったり、質問があったり、ヘルプが必要だったり…いずれの場合でも:

- Visit the forum
- 参加してみましょう。#kicad IRC channel on Freenode
- Watch tutorials

Chapter 2

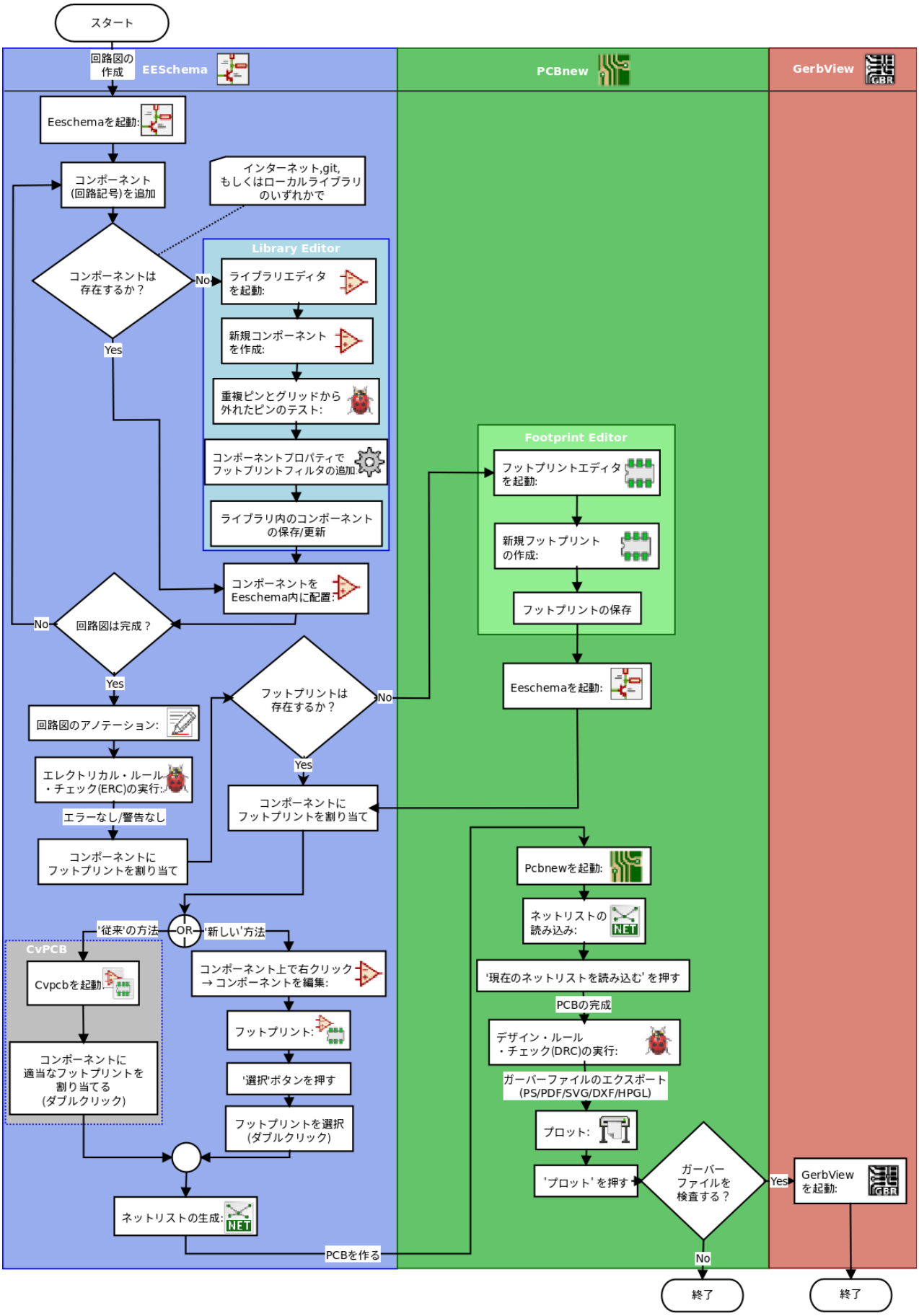
KiCad 作業の流れ

他の PCB 設計ツールとの類似性がある一方で、KiCad は、回路図のコンポーネントとフットプリントを別に持つ独自のワークフローを特徴としています。回路図を作成した後の時点でのみ、コンポーネントには実際のフットプリントが割り当てられます。

2.1 概要

KiCad での作業の流れは２つの主要なタスクで構成されています: 回路図の作成と基板のレイアウトです。コンポーネントライブラリとフットプリントライブラリの両方がこれら２つのタスクのために必要です。KiCad には多くのコンポーネントとフットプリントが含まれており、新しく作るためのツールも持っています。

下図は KiCad での作業の流れを示すフローチャートです。このフローチャートはあなたが取る必要のあるステップを、どんな順序で行うのかを説明しています。適用できる場合には便宜上アイコンを付け加えています。


スタート
回路図の作成
EESchema
PCBnew
GerbView
Eeschemaを起動:
コンポーネント(回路記号)を追加
インターネット,git,もしくはローカルライブラリのいずれかで
コンポーネントは存在するか？
No
Yes
Library Editor
ライブラリエディタを起動:
新規コンポーネントを作成:
重複ピンとグリッドから外れたピンのテスト:
コンポーネントプロパティでフットプリントフィルタの追加
ライブラリ内のコンポーネントの保存/更新
コンポーネントをEeschema内に配置:
回路図は完成？
No
Yes
回路図のアノテーション:
エレクトリカル・ルール・チェック(ERC)の実行:
エラーなし/警告なし
コンポーネントにフットプリントを割り当て
フットプリントは存在するか？
No
Yes
コンポーネントにフットプリントを割り当て
'従来'の方法
OR
'新しい'方法
CvPCB
Cvpcbを起動:
コンポーネントに適当なフットプリントを割り当てる(ダブルクリック)
コンポーネント上で右クリック→コンポーネントを編集:
フットプリント:
'選択'ボタンを押す
フットプリントを選択(ダブルクリック)
ネットリストの生成:
PCBを作る
Footprint Editor
フットプリントエディタを起動:
新規フットプリントの作成:
フットプリントの保存
Eeschemaを起動:
Pcbnewを起動:
ネットリストの読み込み:
'現在のネットリストを読み込む' を押す
PCBの完成
デザイン・ルール・チェック(DRC)の実行:
ガーバーファイルのエクスポート(PS/PDF/SVG/DXF/HPGL)
プロット:
'プロット' を押す
ガーバーファイルを検査する？
Yes
No
GerbView を起動:
終了
終了

コンポーネントの作成についての更なる情報は、「回路図シンボルの作成」をお読み下さい。また、新しいフットプリントの作成方法についての情報は、「フットプリントの作成」をご覧下さい。

Quicklib は、素早く KiCad のコンポーネントを作成することのできる Web ベースのインターフェイスを持ったツールです。Quicklib についての更なる情報は、「Quicklib による回路図コンポーネントの作成」を参照して下さい。

2.2 フォワード／バックアノテーション

電気回路図を完全に描き終わったら、次のステップはそれを基板に移すことです。しばしば、追加のコンポーネントを付け足す必要があったり、部品を違うサイズに変更したり、ネット名を変更するといったことが起こります。これは２つの方法で行うことができます: フォワードアノテーションとバックアノテーションです。

フォワードアノテーションとは、回路図の情報を対応する基板レイアウト図に送る機能です。少なくとも最初に一度は回路図から基板図へインポートしなければならないので、これは基本的な機能です。その後、回路図の変更はフォワードアノテーションによって基板図へと順次伝えられます。フォワードアノテーションについての詳細は、「KiCad のフォワード・アノテーション」の章で議論されています。

バックアノテーションとは、基板レイアウト図の変更を対応する回路図に送り戻す処理です。バックアノテーションを招く類似した２つの要因として、ゲートの入替とピンの入替が挙げられます。このような状況では、ゲートやピンは機能的に同一ですが、部品レイアウトの都合によって使われるピンやゲートが選択されて接続される信号が決定されます。基板図で選択が決定されたら、この変更は回路図へと差し戻されます。

Chapter 3

KiCad の使用

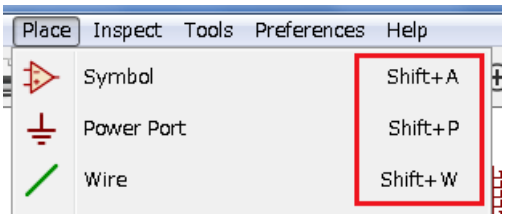
3.1 ショートカットキー

KiCad は関連はあっても異なる２種類のショートカットキーを持っています: アクセラレーターキーとホットキーです。マウスを使う代わりにキーボードをコマンド変更に使うことで、どちらも KiCad での作業の迅速化に使われます。

3.1.1 アクセラレーターキー

アクセラレーターキーは、メニューやツールバーアイコンをクリックするのと同様の効果を持ちます: コマンドは入力されますが、左マウスボタンがクリックされるまでは何も起きません。コマンドモードで入力したいが直ちに実行はしたくない時にはアクセラレーターキーを使用します。

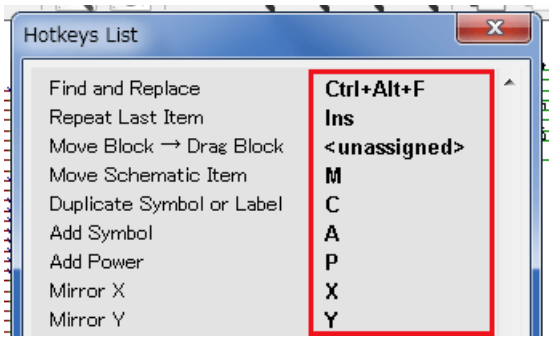
アクセラレーターキーは全てのメニューパネルで右端に表示されます:



3.1.2 ホットキー

ホットキーはアクセラレーターキーに左マウスクリックを足したものと同じです。ホットキーを使用すると現在のカーソル位置で直ちにコマンドを開始します。ワークフローを中断することなく直ちにコマンドを変更するにはホットキーを使用します。

KiCad のツールでホットキーを参照するには、ヘルプ → ホットキーリストと辿るか、Ctrl+F1 を押します:



設定 → ホットキーオプションメニューから、ホットキー割り当ての編集や、ホットキー設定のインポート、エクスポートが行えます。

注意
このドキュメントでは、ホットキーをこのような括弧で表現しています: [a] もし [a] を見たなら、キーボードの"a" キーを押して下さい。

3.1.3 例

回路図で線を引く簡単な例を考えてみましょう。

アクセラレーターキーを使う場合、" ワイヤを配置" コマンドを呼び出すよう"Shift + W" を押します (カーソルの形が変わります)。続いて、配線を開始するためには、線を引き始めたい場所でマウスを左クリックします。

ホットキーでは、単に [w] を押すと現在のカーソル位置から直ちに配線が始まります。

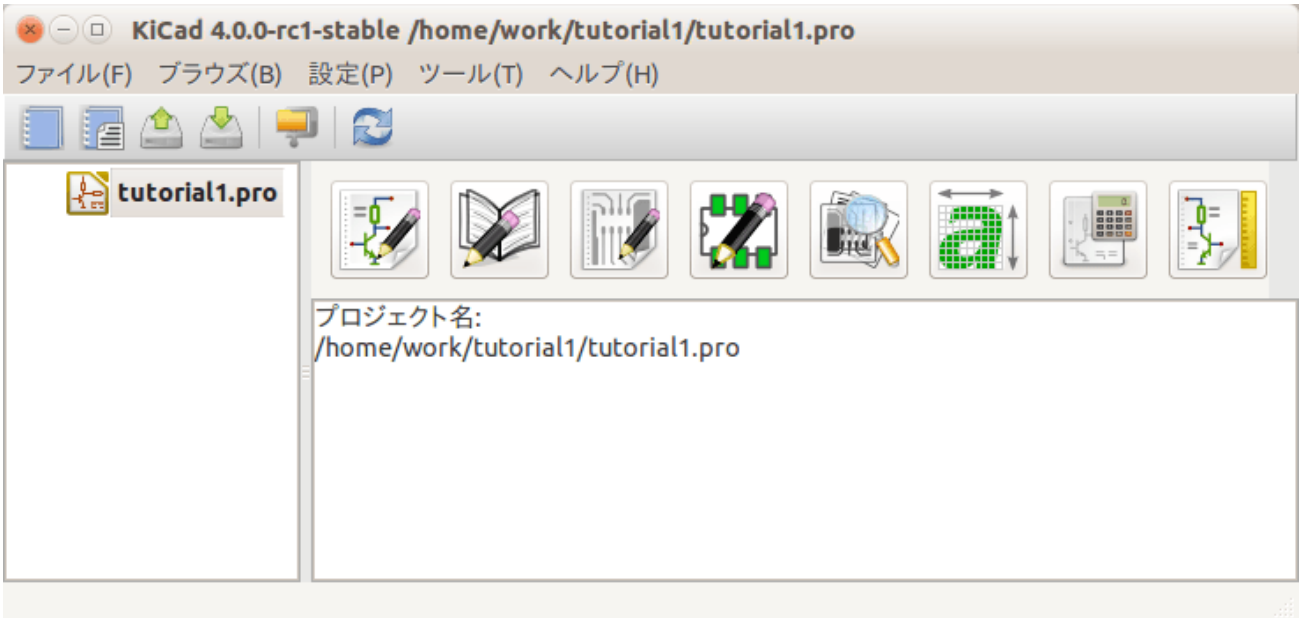
Chapter 4

電気回路図の作図

この章では KiCad を使って電気回路図を作図する方法を学びます。

4.1 Eeschema の使用

1. Windows の場合は kicad.exe を実行して下さい。Linux の場合は、端末で *kicad* とタイプして下さい。KiCad プロジェクトマネージャーのメインウィンドウが表示されることでしょう。ここから８つの独立したソフトウェアツールを呼び出すことができます: *Eeschema*, 回路図ライブラリーエディター, *Pcbnew*, *PCB* フットプリントエディター, *GerbView*, *Bitmap2Component*, *PCB Calulator* そして *Pl Editor* です。主要なツールがどのように使われるのかを理解するには KiCad 作業の流れの図を参照して下さい。



2. 新規プロジェクトを作成します: ファイル → 新規 → プロジェクトと辿って下さい。プロジェクトファイル名は *tutorial1* とします。プロジェクトファイルには拡張子".pro" が自動的に付加されるでしょう。ダイアログの外観は使用されているプラットフォームによって異なりますが、必ず新しいディレクトリを作るための

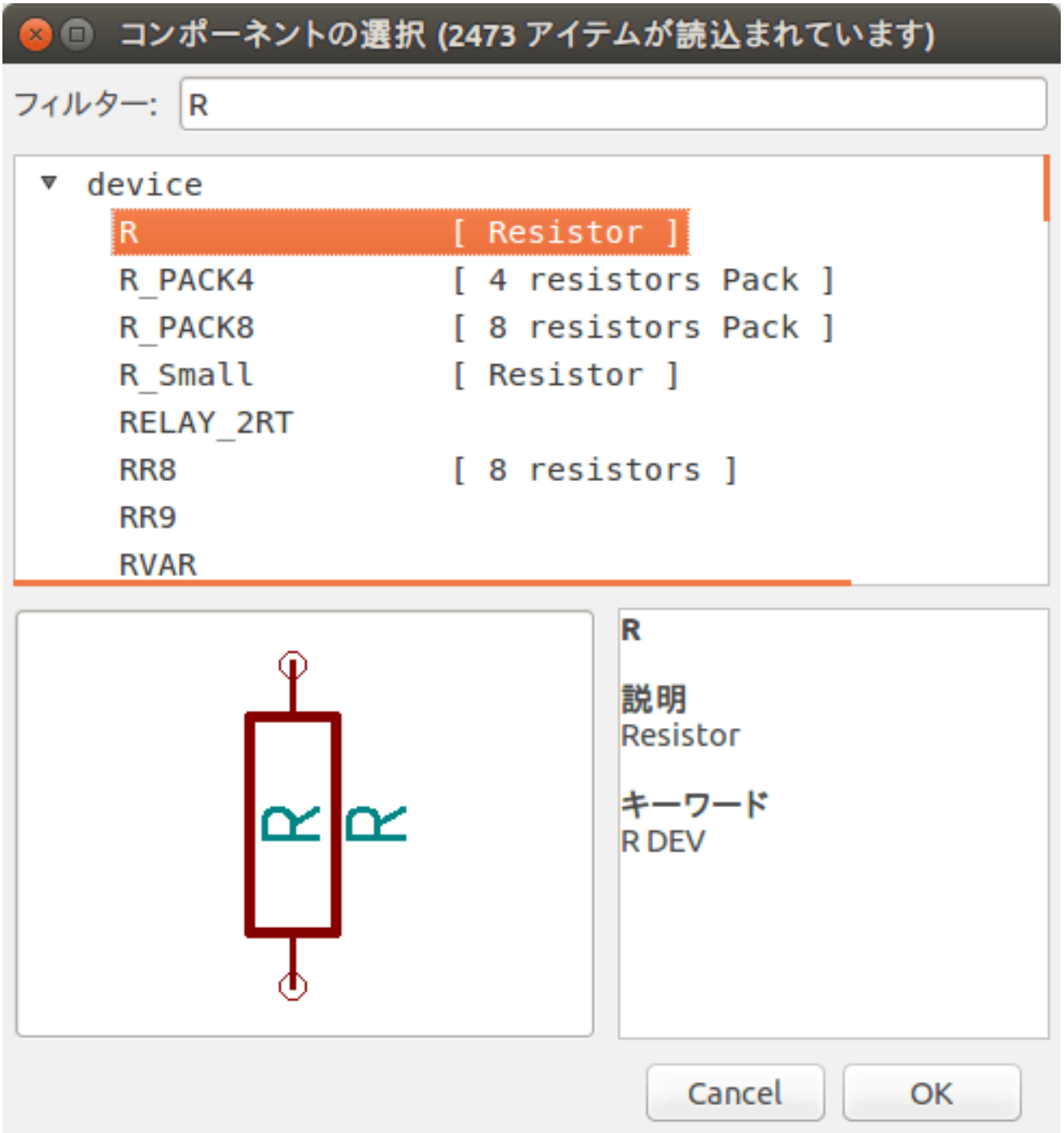
チェックボックスがあります。既にディレクトリを用意しているのでなければ、チェック状態のままにしておきます。全てのプロジェクトファイルはそこに保存されることになるでしょう。

3. では回路図の作成を始めましょう。回路図エディター *Eeschema* を起動します。左から 1 番目のボタンです。

4. 次に上部ツールバーにある ページ設定アイコン をクリックして下さい。適切な 用紙サイズ (*A4*,*8.5x11* etc.) をセットし、タイトルとして *Tutorial1* を入力します。. 表示されている他の情報は必要に応じて入力できます。OK をクリックします。この情報は回路図の下右隅に配置されます。ズームインするにはマウスホイールを使用してください。回路図全体を保存します: ファイル → 保存と辿って下さい。

5. 最初の部品を置いてみましょう。右ツールバーにある シンボルを配置アイコン をクリックして下さい。シンボルを追加ホットキー [a] を押して配置することもできます。

6. 回路図シートの中央でクリックして下さい。シンボルを選択ウィンドウが画面上に表示されます。抵抗器を配置してみます。検索欄に **R**esistor の *R* と入力し、*R* でフィルター検索をします。Resistor の上にある見出し *Device* にお気付きかも知れません。この見出し *Device* はそのコンポーネントが含まれる、使いやすいように同種のものを集めたライブラリーの名前です。



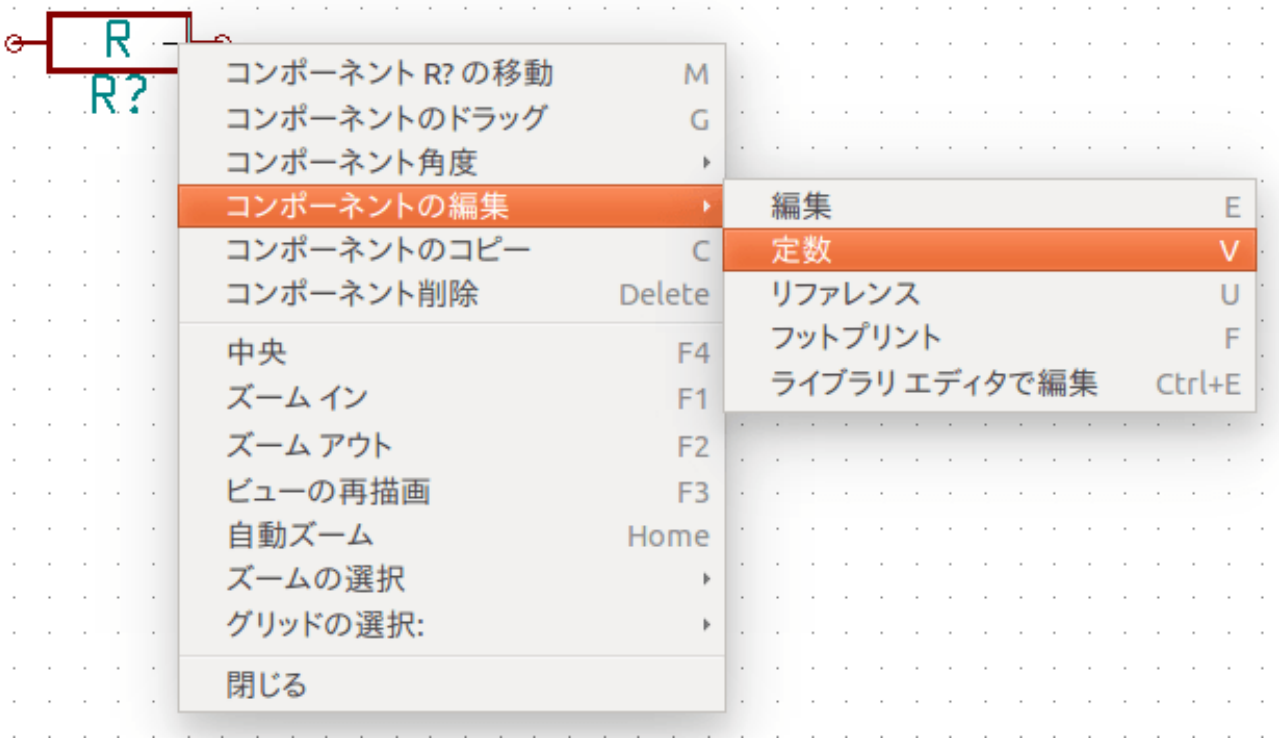
7. この上でダブルクリックしてみて下さい。これで シンボルを選択ウィンドウが閉じられます。配置したい場所でクリックすると、回路図シート内にコンポーネントが配置されます。

8. ズームインするにはコンポーネント上で拡大鏡をクリックします。あるいは、マウスホイールを使ってズームイン、ズームアウトができます。水平方向と垂直方向にパンするにはマウスのホイール (中央) ボタンを押します。

9. コンポーネント R の上にマウスカーソルを持っていって [r] を押してみてください。コンポーネントが回転します。回転させるためにコンポーネント上でクリックをする必要はありません。

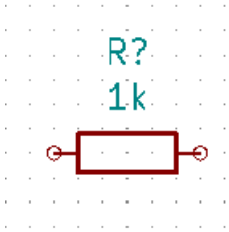
注意
何か別のものの上にマウスカーソルがあった場合、時にはメニューが現れるでしょう。明示的な選択メニューは KiCad ではしばしば見られ、互いに重なっているオブジェクトを選んで作業を可能にします。今回の場合、シンボル…*R*…でアクションを実行することを KiCad に伝えます。

10. コンポーネントの中央で右クリックして、プロパティ → 定数を編集を選択します。コンポーネントの上にマウスカーソルを持っていって、[v] を押すことでも同じ結果が得られます。代わりに [e] を押すと、もっと一般的なプロパティウィンドウが現れます。右クリックメニューには有効なアクションのホットキーが全て表示されていることに注目してください。



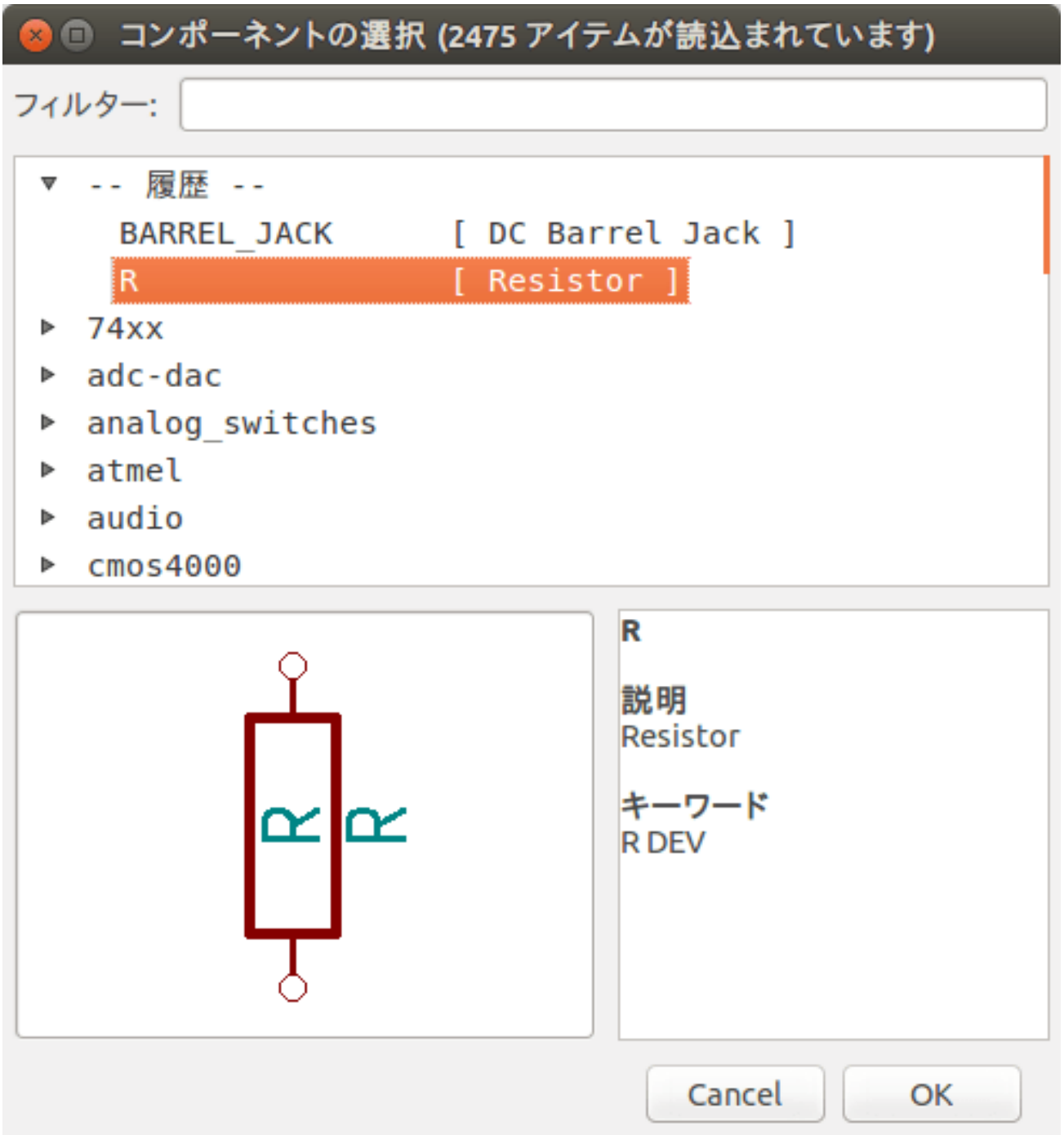
11. 定数フィールドを編集のウィンドウが表示されます。

注意
リファレンスフィールド (R?) を変更しないで下さい。これは後ほど自動的に行われます。これで抵抗器の定数は *1 k* になりました。



12. 別の抵抗器を配置するには、その抵抗器を置きたい場所で単にクリックだけして下さい。シンボルを選択のウィンドウが再び表示されます。

13. 以前に選んだ抵抗器が履歴リスト上に R として表示されています。OK をクリックしてコンポーネントを配置します。



14. 間違ったコンポーネントを削除したい場合、コンポーネント上で右クリックして コンポーネントを削除をクリックします。これでコンポーネントは回路図から取り除かれます。あるいは、削除したいコンポーネント上にマウスカーソルを持っていき、[Delete] を押すことでも削除できます。

15. 既に回路図シート上にあるコンポーネントにマウスカーソルを持っていき [c] を押すことで、複製することもできます。新しく複製したコンポーネントを置きたい場所でクリックして下さい。

16. ２つ目の抵抗器の上で右クリックします。’ドラッグ’を選択します。コンポーネントを移動させて左クリッ

クで確定します。コンポーネント上にマウスカーソルを持っていき [g] を押すことでも同じ機能を実行可能です。[r] はコンポーネントを回転させ、[x] と [y] は x 軸または y 軸に対して x コンポーネントを反転させます。

注意
右クリック → 移動または [m] も同様に様々なものを移動させるのに役立つオプションですが、コンポーネントのラベルと未接続コンポーネントにのみ使ったほうが良いでしょう。なぜそうなのかは後述します。

---

17. 2 つ目の抵抗器にマウスカーソルを持っていき [v] を押して編集します。*R* を *100* で置き換えます。"Ctrl+Z" によってどんな編集操作も「元に戻す」ことができます。

18. グリッドのサイズを変更します。回路図シート上では、全てのコンポーネントが広い間隔のグリッドにスナップされていることにお気付きのことでしょう。グリッドのサイズは 右クリック → グリッドで簡単に変更できます。概して回路図シートでは、*50.0* ミルグリッドの使用をお勧めします。

19. デフォルトのプロジェクトに設定されていないようなライブラリーからコンポーネントを追加してみましょう。メニューで、設定 → シンボルライブラリーを管理を選んで下さい。２つのタブ: グローバルライブラリーとプロジェクト固有のライブラリー、を持つシンボルライブラリーのウィンドウが表示されます。タブ各々が sym-lib-table ファイルを一つ持っています。有効なライブラリー (.lib file) は、これら sym-lib-table ファイルの一つに含まれていなければなりません。ファイルシステムにライブラリーファイルがあって未だ有効になっていなければ、sym-lib-table ファイルのどちらかに追加することができます。練習として、既に有効になっているライブラリーを追加してみます。

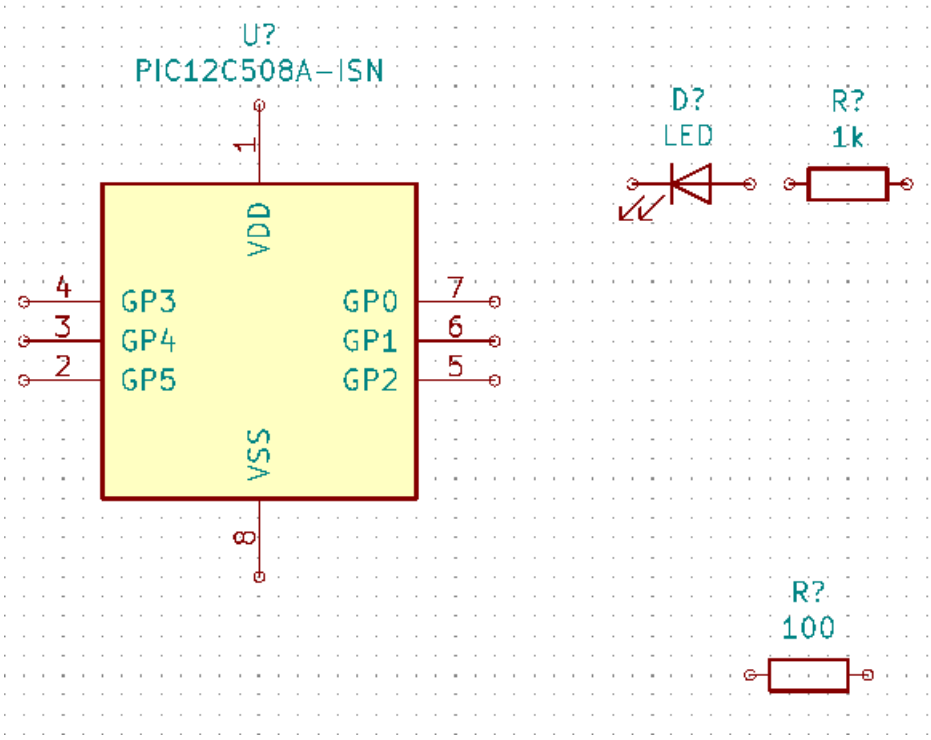
20. プロジェクト固有のテーブルを選択します。テーブルの下にあるライブラリーを参照ボタンをクリックします。あなたのコンピューターにインストールされている公式 KiCad ライブラリーの場所を見つける必要があります。多くの *.dcm* ファイルと *.lib* ファイルがある *library* ディレクトリを探しましょう。`C:\Program Files (x86)\KiCad\share\` (Windows) や `/usr/share/kicad/library/` (Linux) を見てみましょう。ディレクトリが見つかったら、*MCU_Microchip_PIC12.lib* ライブラリーを選んで追加し、ウィンドウを閉じます。選んだライブラリーがリストの最後に追加されます。次にその別名 (ニックネーム) をクリックして *microchip_pic12mcu* に変更します。OK を押してシンボルライブラリーのウィンドウを閉じて下さい。

21. コンポーネント追加のステップを繰り返しますが､今度は *Device* ライブラリーの代わりに *microchip_pic12mcu* ライブラリーを選択して *PIC12C508A-I/SN* コンポーネントを選びます。

22. マイクロコントローラーのコンポーネントにマウスカーソルを持っていきます。[x] と [y] がコンポーネントを反転させることに再び注目して下さい。G0 と G1 ピンが右を向くよう Y 軸に沿ってシンボルを反転させたままにします。

23. コンポーネント追加のステップを繰り返しますが、今度は *Device* ライブラリーを選択して *LED* コンポーネントを選びます。

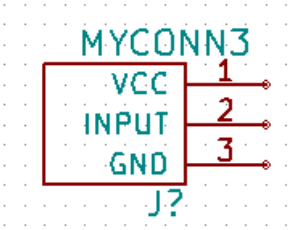
24. 回路図シート上の全てのコンポーネントを以下のように整理します。



25. 次に３ピンのコネクターとして *MYCONN3* という回路図コンポーネントを作る必要があります。「KiCad 回路図シンボルの作成」 の章にジャンプし、一からコンポーネントを作る方法を学んでから、基板での学習を続けるためにこの章に戻ってきて下さい。

26. さあ、新たに作成されたコンポーネントを配置することができるようになりました。[a] を押して *myLib* ライブラリーにある *MYCONN3* コンポーネントを選びます。

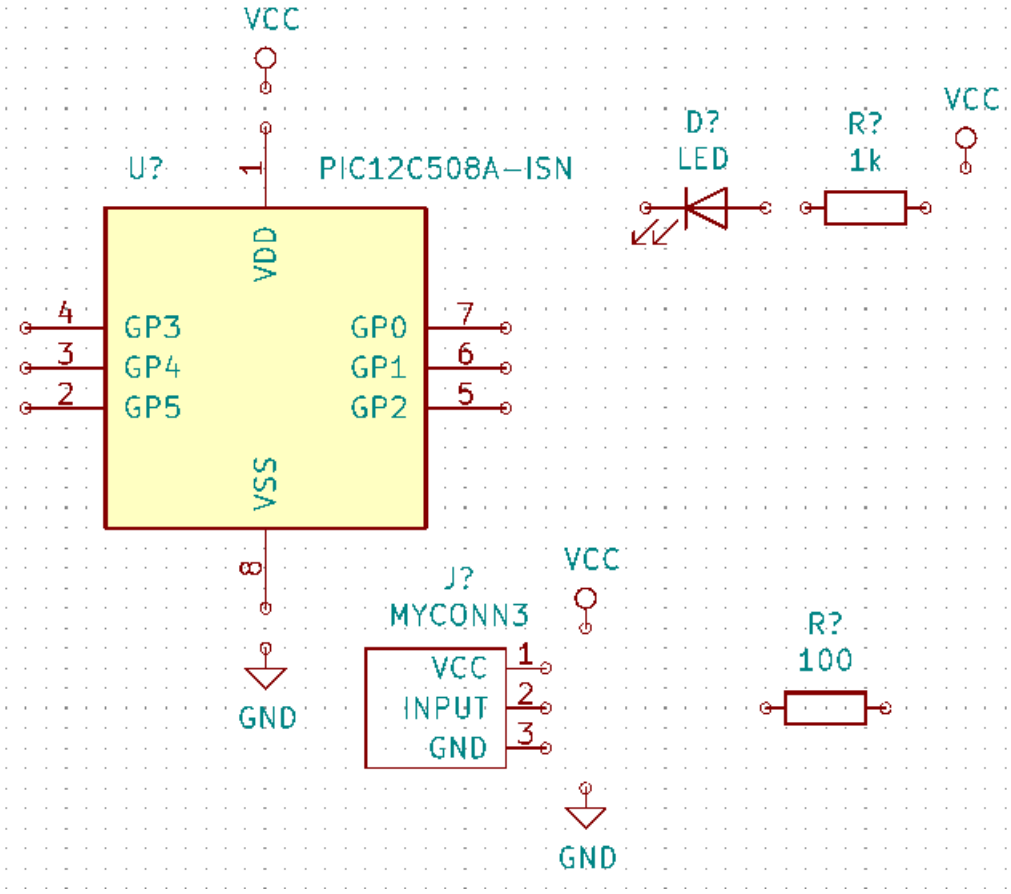
27. コンポーネントの識別子 *J?* が *MYCONN3* ラベルの下に現れるでしょう。その位置を変更したい場合は、*J?* の上で右クリックして リファレンスを移動 ([m] と同等) をクリックします。これを行う前か最中にズームインすると便利かもしれません。以下に示すようにコンポーネントの下に *J?* を再配置します。ラベルは何回でも好きなだけ動かすことができます。



28. 電源と GND のシンボルを配置します。右ツールバーにある 電源ポートを配置ボタン をクリックします。あるいは [p] を押します。コンポーネント選択のウィンドウで下方にスクロールし、*power* ライブラリーから *VCC* 選択します。OK をクリックして下さい。

29. 1 k の抵抗器のピン上方をクリックして VCC を配置します。マイクロコントローラー *VDD* の上方をクリックします。コンポーネント選択履歴から *VCC* を選んで、VDD ピンの隣に配置します。再び追加作業を繰り返し、VCC を *MYCONN3* の VCC ピンの上方に配置します。必要なら、リファレンスと定数を移動させておいて下さい。

30. ピン追加のステップを繰り返しますが、今度は GND を選択します。GND を *MYCONN3* GND ピンの下に配置します。別の GND シンボルをマイクロコントローラの VSS ピンの左に配置します。回路図はこのようになっているはずです。:



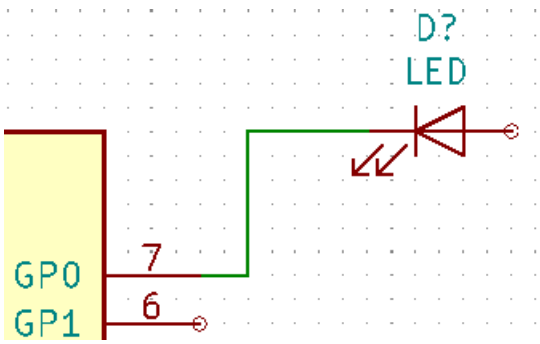
31. 次に、全てのコンポーネントを配線します。右ツールバーの ワイヤーを配置アイコン をクリックします。

注意
バスを配置を選ばないように注意しましょう。このボタンの直ぐ下にある太い線のアイコンです。「KiCad でのバス接続」 の節でバスの使い方を説明しています。

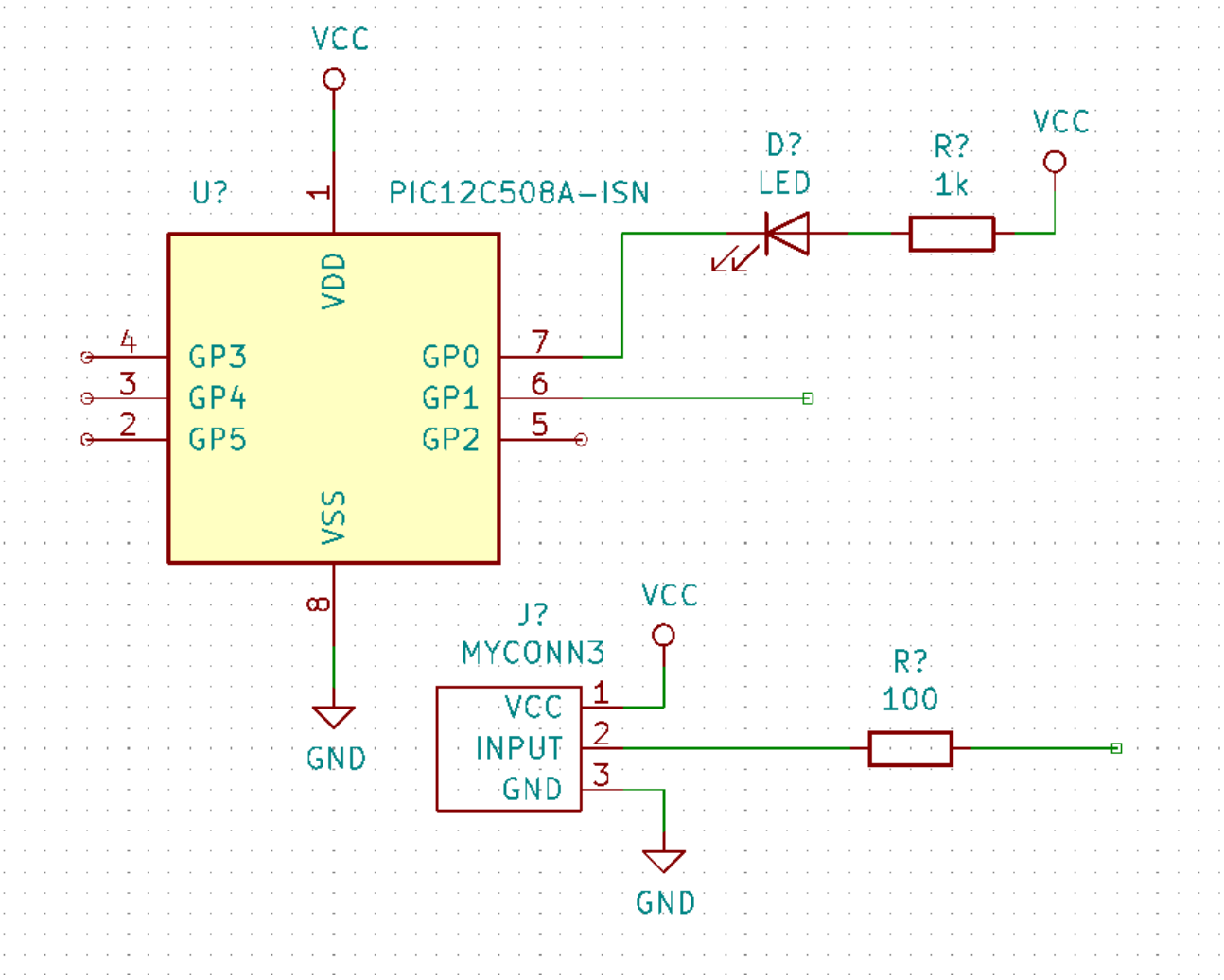
---

32. マイクロコントローラ 7 ピンの端にある小さな円をクリックし、それから LED 1 ピンの小さな円をクリックします。配線を曲げる時は一度クリックして下さい。接続しながらズームインすることができます。

注意
配線されたコンポーネントを再配置したい場合、[g] (掴む) を使うことが重要で、[m] (移動) を使うべきではありません。(掴む) を使うことでワイヤーの接続を維持できます。コンポーネントの移動方法を忘れたならステップ 24 を見直しましょう。



33. この処理を繰り返し、下のように他のコンポーネントの配線を全て済ませます。ワイヤーを終端するにはダブルクリックします。VCC と GND シンボルへの配線は、VCC シンボルなら下部、GND シンボルなら上部中央にタッチします。下のスクリーンショットを見て下さい。



34. 今度はラベルを使って接続する別の方法を検討してみましょう。右ツールバーの ネットラベルを配置アイコン をクリックし、ネットラベルツールを選びます。[1] を使うこともできます。

35. マイクロコントローラーの 6 ピンに接続されているワイヤーの中程をクリックします。このラベルに INPUT と名前をつけます。ラベルはまだ独立したアイテムで、例えば、動かしたり、回転させたり、削除することが

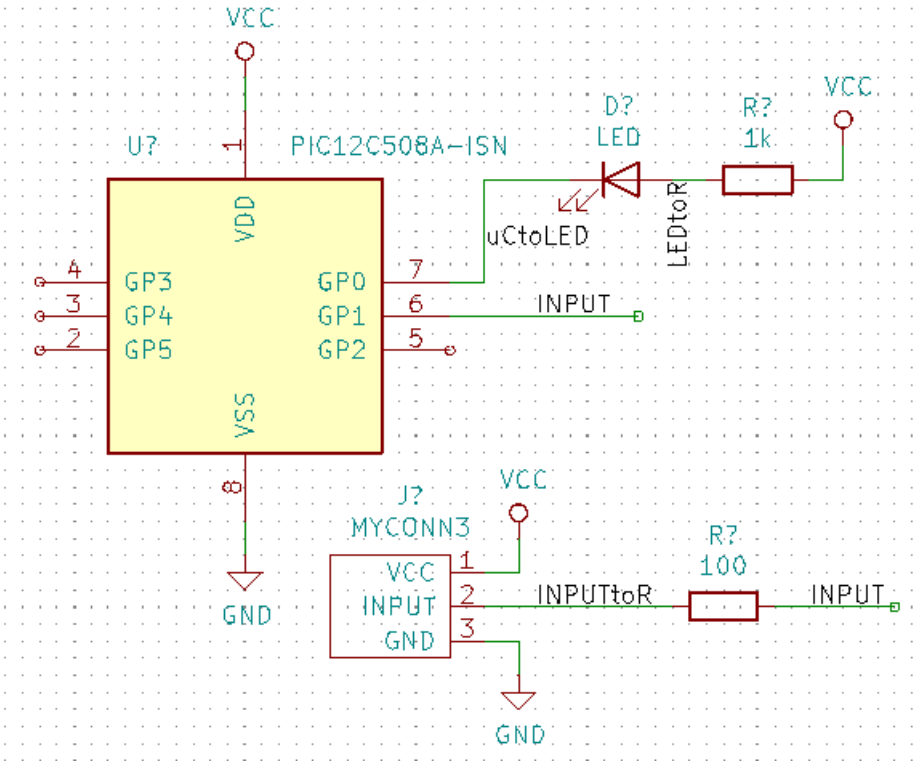
できます。ラベルの効果を生じさせるには、ラベルの小さな四角形のアンカーがワイヤーまたはピンの上に正確に存在しなければなりません。

36. 同じ手順に従い、100Ω の抵抗器の右にある配線に別のラベルを配置します。同様に *INPUT* と名前を付けます。同じ名前を持つ２つのラベルは、PIC のピン６と 100Ω 抵抗器の間に見えない接続を作ります。これは線が混雑した複雑なデザインの場合に便利な配線テクニックです。ラベルを配置するのに必ずしもワイヤは必要なく、単にピンにラベルを付けることもできます。

37. ラベルはまた、単に配線に分かりやすい名前をつける目的で使うこともできます。例えば、PIC のピン７にラベルを付けて *uCtoLED* と名前を入力します。抵抗器と LED 間のワイヤーに *LEDtoR* と名前を付けたり、*MYCONN3* と抵抗器の間のワイヤーに *INPUTtoR* と名前を付けるような使い方です。

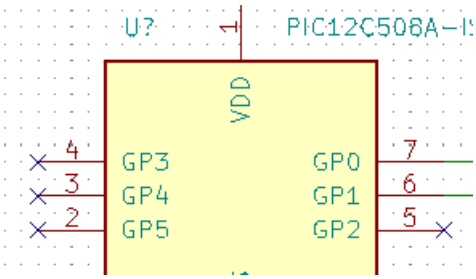
38. 接続されている電源オブジェクトからラベルが暗黙的に定義されているので、VCC と GND の線にラベルをつける必要はありません。

39. 下図に最終的な結果がどのように見えるかを示します。



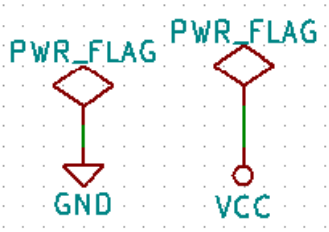
40. それでは接続されていないワイヤーに対処しましょう。接続されていないピンやワイヤーは KiCad のチェック時に警告されます。これらの警告を避けるため、ワイヤーが意図的に接続されていないことをプログラムに指示できます。

41. 右ツールバーにある 空き端子フラグを配置アイコン をクリックします。ピン２、３、４、５をクリックします。X は未接続が意図的であることを示すために表示されます。



42. 見えない電源ピンを持つコンポーネントがあります。左ツールバーの 非表示ピンを表示アイコン をクリックすることでそれらを見えるようにできます。VCC と GND の名前は重要で、隠れた電源ピンは自動的に接続されます。一般的には、隠れた電源ピンを作らないように努力すべきです。

43. ここで、どこかから供給される電源を KiCad に表示するための 電源フラグを追加する必要があります。[a] を押して *power* ライブラリーにある *PWR_FLAG* を検索して下さい。それらを 2 つ配置します。以下のように GND ピンと VCC に接続します。



注意
これにより、この典型的な回路図チェックの警告を避けることができます: ピンは他のピント接続されていますが、このピンを起動するピンがありません

---

44. 時には、いろいろな場所でコメントを書いておいたほうがいいこともあります。回路図にコメントを追加するには、右ツールバーにある テキストを配置アイコン T を使います。

45. ここで全てのコンポーネントにユニークな識別子を与える必要があります。実際、私達のコンポーネントの多くは未だ *R?* や *J?* と名付けられています。識別子の割り当ては、上部ツールバーにある 回路図シンボルをアノテーションのアイコン をクリックすることで自動的に行われます。

46. 回路図をアノテーションウィンドウでは、全ての回路図を使用を選択し、アノテーションボタンをクリックします。次に確認メッセージで OK をクリックして、閉じるをクリックします。コンポーネントの全ての *?* が数字に置き換えられたことに注目してください。これで各識別子はユニークになりました。私達の例では、*R1* 、*R2* 、*U1* 、*D1* そして *J1* と名付けられました。

47. 次に回路図のエラーをチェックします。上部ツールバーの エレクトリカルルールのチェックを実行アイコン をクリックします。実行ボタンをクリックします。接続されていないワイヤー等のエラーや警告のレポートが通知されます。エラー 0、警告 0 にしましょう。エラーや警告がある場合には、小さな緑色の矢印が回路図上のエラーや警告のある場所に表示されます。よりエラーの情報を得るため *ERC* レポートファイルを生成をチェックし、実行ボタンを再度押します。

注意
もし" デフォルトのエディタが指定されていません。エディタを指定して下さい。" の警告が出たなら、`C:\Windows\notepad.exe` (Windows) か `/usr/bin/gedit` (Linux) を設定してみてください。

---

48. ついに回路図が完成しました。これでコンポーネントに対応するフットプリントを付け足して、ネットリストのファイルを作成することができます。上部ツールバーにある ネットリストを生成アイコン をクリックして下さい。生成をクリックし、デフォルトのファイル名のままで 保存をクリックします。

注意
KiCad の以前のバージョンでは、ネットリストが必須でした。最近のバージョンでは、そうではなく、代わりに ツール → 回路図から基板を更新を使うことができます。これを実行する場合は、最初にフットプリントをシンボルに割り当てる必要があります。

---

49. ネットリストファイルを生成したら、上部ツールバーにある *Cvpcb* を実行アイコン をクリックします。' ファイルが存在しません' のエラーが表示されたら、無視して OK をクリックします。

注意
シンボルにフットプリントを追加するには、もっと多くの方法があります。

- シンボル上で右クリックし、→ プロパティ → フットプリントを編集シンボル上でダブルクリック、あるいは右クリックし、→ プロパティ → プロパティを編集 → フットプリント
- ツール → シンボルフィールドを編集回路図エディター (Eeschema) 一般オプションにある、シンボルの選択中にフットプリントをプレビューをチェックし、配置する新しいシンボルの選択時にフットプリントを選択

---

50. *Cvpcb* は、回路図中の全てのコンポーネントに KiCad ライブラリーにあるフットプリントをリンクできます。中央のペインには、回路図で使用されている全てのコンポーネントが表示されます。ここで *D1* を選択してみましょう。右側のペインには利用可能なフットプリントが全て現れるので、ここで下スクロールして *LED_THT:LED-D5.0mm* の上でダブルクリックしてみて下さい。

51. 右側のペインには、利用可能なフットプリントから選択されたサブグループだけを表示することもできます。これは適切なフットプリントだけを KiCad に提示させるための手段です。これらのフィルターを有効、あるいは無効に切り替えるには、アイコン, and をクリックします。

52. *U1* には、フットプリント *Package_DIP:DIP-8_W7.62mm* を選択します。*J1* には、フットプリント *Connector:Banana_Jack_3Pin* を選択します。*R1* と *R2* には、フットプリント *Resistor_THT:R_Axial_DIN0207_L6.3mm_I* を選択します。

53. 選択したフットプリントがどのように見えるのか知りたければ、選択したフットプリントを表示アイコン をクリックして、現在のフットプリントをプレビュー表示させることが出来ます。

54. 割り当てが終わりました。ここで、ファイル → 回路図を保存のクリックまたはボタン 適用して、回路図の保存＆続行で、回路図を保存することができます。

55. *Cvpcb* を閉じて *Eeschema* 回路図エディターに戻ることが出来ます。Cvpcb で保存していなければ、ここでファイル → 保存をクリックし保存して下さい。再びネットリストを作成します。このネットリストは全てのフットプリントが更新されています。フットプリントが割り当てられていないデバイスがないように注意して下さい。ご自身でフットプリントを作ることになるでしょう。これについては、本ドキュメントの後の章で説明しています。

注意
この状態では全てのシンボルがフットプリントを割り当てられています。ネットリストと続く２つのステップに代わり、ツール → 回路図から基板を更新を使用できます。実行すると、Pcbnew が 回路図から基板を更新ダイアログとともに開きます。ここからは、このチュートリアルにある Pcbnew の章での指示に従って進めることができます。

---

56. KiCad プロジェクトマネージャーへと切り替えます。ファイルリストの中にネットリストファイルを見つけることができます。

57. ネットリストファイルには、全てのコンポーネントとそのそれぞれのピン接続が記述されています。ネットリストファイルは、実際には容易に検査、編集や記載することのできるテキストファイルです。

注意
ライブラリーファイル (*.lib) もテキストファイルで、同様に容易に編集したり記載することができます。

---

58. 部品表 (BOM) を作成するには、*Eeschema* 回路図エディターを起動して上部ツールバーにある 部品表を生成アイコン をクリックします。デフォルトでは、プラグインは全て無効になっています。プラグインを追加ボタンをクリックして一つ追加します。使いたい *.xsl ファイルを選択します。ここでは *bom2csv.xsl* を選択します。

注意

Linux:

もし xsltproc が見つからない場合、ダウンロードして次のようにインストールします:

```
sudo apt-get install xsltproc
```

これは Ubuntu のような Debian 由来のディストリビューション用です。もしくは

```
sudo yum install xsltproc
```

これは RedHat 由来のディストリビューション用です。もし２種類のディストリビューション以外をお使いの場合は、お使いのディストリビューションのパッケージマネージャーコマンドを使って xsltproc パッケージをインストールして下さい。

xsl ファイルが置かれる場所: */usr/lib/kicad/plugins/*.

Apple OS X:

もし xsltproc が見つからない場合、これを保存している Apple のサイトから Apple Xcode ツールをインストールするか、ダウンロードして次のようにインストールします:

```
brew install libxslt
```

xsl ファイルが置かれる場所: */Library/Application Support/kicad/plugins/*.

Windows:

xsltproc.exe と含まれる xsl ファイルは、それぞれ < *KiCad install directory* > *bin* と < *KiCad install directory* > *bin**scripting**plugins* に置かれます。

全てのプラットホーム:

最新の bom2csv.xsl はここから入手できます:

https://gitlab.com/kicad/code/kicad/raw/master/eeschema/plugins/xsl_scripts/bom2csv.xsl

KiCad は自動的にコマンドを生成します、例えば:

```
xsltproc -o "%O" "/home/<user>/kicad/eeschema/plugins/bom2csv.xsl" "%I"
```

拡張子を与えたい場合は、このコマンドラインを変更します:

```
xsltproc -o "%O.csv" "/home/<user>/kicad/eeschema/plugins/bom2csv.xsl" "%I"
```

更に情報を得るには、ヘルプボタンを押して下さい。

59. では 生成を押してみましょう。ファイル (プロジェクトと同じ名前です) はプロジェクトフォルダの中にあります。***.csv** ファイルを LibreOffice Calc か Excel で開いてみましょう。インポートのウィンドウが表示されたら、OK を押します。

これであなたは次の章” プリント基板のレイアウト” に進む準備ができました。ですが先に進む前に、コンポーネントのピンをバスで接続する方法をざっと見ておきましょう。

4.2 KiCad でのバス接続

時には、コンポーネント A の連続した幾つかのピンと、別のコンポーネント B の同様に連続したピンを接続しなければならないことがあります。この場合には２つの選択肢があります: 私達が既に見てきたラベルによる方法、

あるいはバス接続の使用です。どのようにするのか見てみましょう。

1. ４ピンのコネクターが３つあり、互いのピン同士を接続したいと仮定しましょう。コネクター P4 のピン４にラベルを付けるためにはラベルオプション ([l] を押す) を使います。このラベルを *a1* と名付けます。自動で同様にピン４の下のピン (ピン３) にレベルを追加するには [Insert] を押します。ラベルが自動的に *a2* とリネームされることに注目しましょう。

2. [Insert] を更に 2 回押して下さい。このキーは 最後のアイテムをリピート機能に対応しており、あなたの仕事を楽にしてくれるとても便利なコマンドです。

3. ラベル付け作業を他の２つのコネクター CONN_2 と CONN_3 にも繰り返して終了します。このまま進めて基板を作成すると、３つのコネクターが互いに接続されているはずです。ここまで説明してきた結果を図２に示します。美しく見せるため、図 3 に示すように、アイコン を使った一連の ワイヤ-バスエントリーを配置とアイコン を使ったバスラインを付け加えることも可能です。基板には何も影響がないので付加しても特に差し支えありません。

4. 図２のピンに接触している短いワイヤは厳密には不要であることを指摘しておきます。実際には、ラベルはピンに直接付けることができます。

5. もう一歩進んで、４つ目のコネクタ CONN_4 を仮定します。何らかの理由により、そのラベル付け (b1, b2, b3, b4) は少し異なっています。私達は *Bus a* と *Bus b* をピンとピンの方法で接続したいのです。バスラインへのラベル付けを、ピンにラベル付けする方法 (も可能ですが) を使わず、代わりにバス毎に１つのラベルで実現したいわけです。

6. 前に説明したラベル付けの方法を使って、CONN_4 に接続してラベルを付けます。ピンには b1, b2, b3 そして b4 と名前を付けます。アイコン を使った一連の ワイヤ-バスエントリーを配置と、アイコン を使ったバスラインに、ピンを接続します。図４を見て下さい。

7. CONN_4 のバスにラベルを付けて ([l] キーを押して) *b[1..4]* と名付けます。

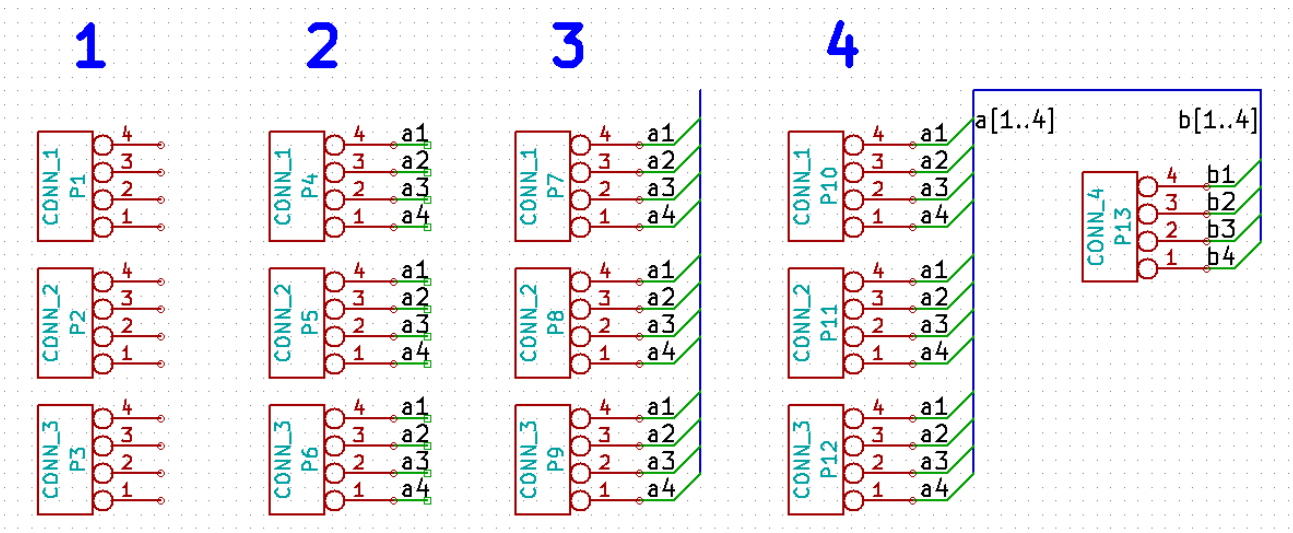
8. 以前のバスにラベルを付けて ([l] を押して) *a[1..4]* と名付けます。

9. これでボタン を使って、バス a[1..4] とバス b[1..4] を接続することでできるようになりました。

10. ２つのバスを接続することで、ピン a1 は自動的にピン b1 に接続され、a2 は b2 に接続され、以下同様となります。図４は最終結果がどのように見えるか示しています。

    注意
    [Insert] による 最後のアイテムをリピート機能は、繰り返される作業に上手く使うことができます。例えば、短いワイヤを全てのピンに接続する図２、図３、図４での作業はこの機能で実現できました。

11. [Insert] による 最後のアイテムをリピート機能はまた、アイコン を使った、多くの連続した ワイヤ-バスエントリーを配置にも適用できます。


1
2
3
4
CONN_1 P1
CONN_2 P2
CONN_3 P3
CONN_1 P4
CONN_2 P5
CONN_3 P6
CONN_1 P7
CONN_2 P8
CONN_3 P9
CONN_1 P10
CONN_2 P11
CONN_3 P12
CONN_4 P13
a[1..4]
b[1..4]
a1
a2
a3
a4
b1
b2
b3
b4

Chapter 5

プリント基板のレイアウト

生成したネットリストのファイルを使って PCB のレイアウトを行う時が来ました。これは *Pcbnew* ツールで行います。

注意

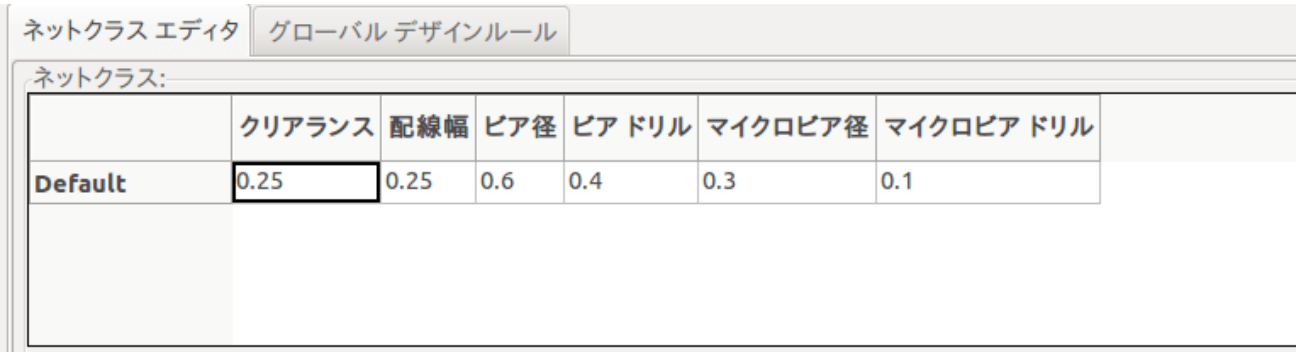
Eeschema で 回路図から基板を更新を使ったならば、ネットリストとステップ５は必要ありません。ステップ６と７のように基板にフットプリントをドロップすることができ、それから図面情報とステップ２〜４でのデザインルールを入力します。

5.1 Pcbnew の使用

1. KiCad プロジェクトマネージャーで、基板レイアウトエディター (Pcbnew) アイコン をクリックします。*Eeschema* から対応するツールバーのボタンを使うこともできます。*Pcbnew* のウィンドウが開きます。「ボード **.kicad_pcb* は存在しません。新規作成しますか？」というメッセージが出たら、はいをクリックします。

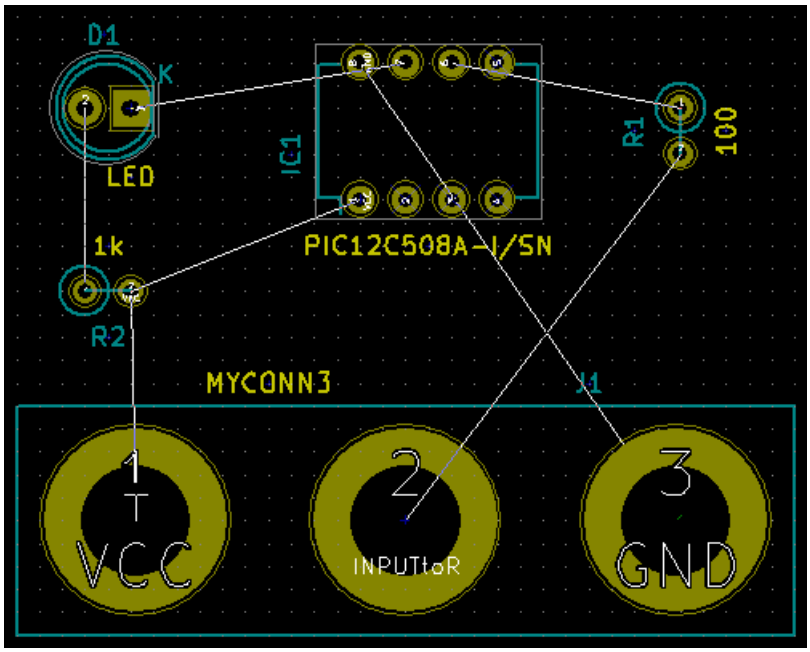
2. いくつかの回路図情報を入力することから始めます。上部ツールバーの ページ設定アイコン をクリックします。適切な ページサイズ (*A4*、*8.5x11* など) に設定して、タイトルを *Tutorial 1* と設定します。

3. 基板メーカーが要求する クリアランスと 最少配線幅に設定することから始めるのは良い考えです。一般的にクリアランスを *0.25*、最少配線幅を *0.25* に設定します。セットアップ → デザインルールメニューをクリックします。もし何も表示されていなければ ネットクラスエディターのタブをクリックします。以下のように、ウィンドウ上部にある クリアランスのフィールドを *0.25* に、配線幅のフィールドを *0.25* に変更します。ここでの寸法は mm です。



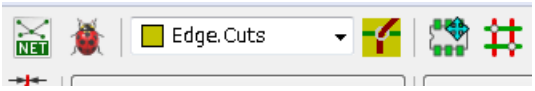
	クリアランス	配線幅	ビア径	ビア ドリル	マイクロビア径	マイクロビア ドリル
Default	0.25	0.25	0.6	0.4	0.3	0.1

4. グローバルデザインルールタブをクリックし、最小配線幅を *0.25* に設定します。変更を確定するには OK ボタンをクリックし、デザインルールエディターのウィンドウを閉じます。
5. さあネットリストファイルをインポートしましょう。上部ツールバーにある ネットリストをロードアイコン をクリックします。Eeschema で作られたのであれば、ネットリストファイルフィールドにネットリストファイル *tutorial1.net* が選択されていなければなりません。現在のネットリストを読み込むをクリックします。その後、閉じるボタンをクリックします。
6. 全てのコンポーネントが表示されるはずです。それらは選択されており、マウスカーソルと一緒に移動します。
7. 基板の中央にコンポーネントを移動します。必要であれば、コンポーネントの移動中にズームインやズームアウトすることができます。マウスの左ボタンをクリックします。
8. 全てのコンポーネントは ラッツネストと呼ばれる細いワイヤの組で接続されています。基板のラッツネストを非表示 *(*非表示*)* ボタン が押されていることを確認します。これで全てのコンポーネントを接続しているラッツネストを見ることができます。
9. コンポーネントにマウスカーソルを持っていき、[m] を押すことで移動できます。配置したい所でクリックして下さい。あるいはクリックしてコンポーネントを選んでからドラッグすることもできます。コンポーネントを回転させるには [r] を押して下さい。配線の交差数が最少になるまで、全てのコンポーネントを移動します。



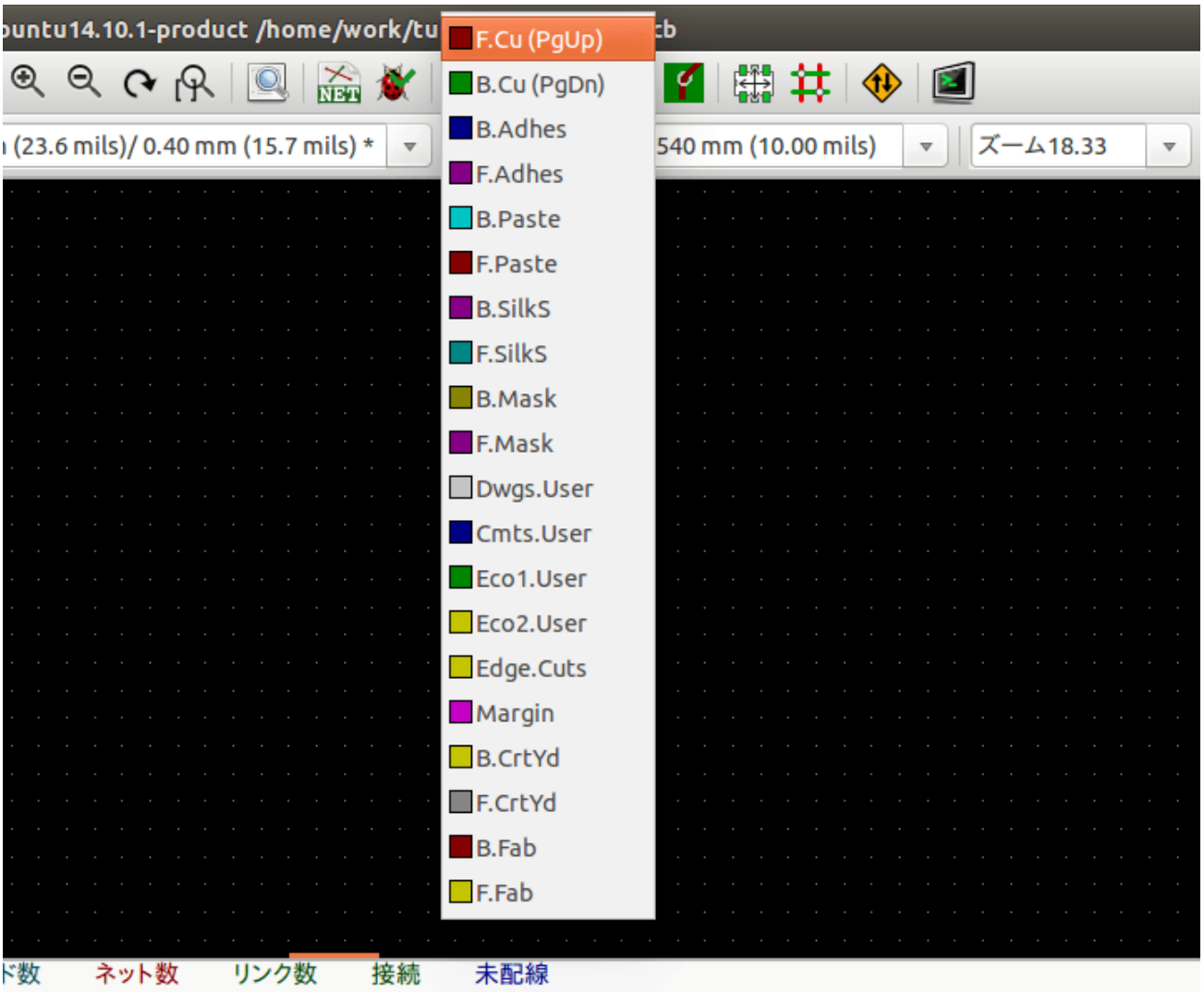
10. 100Ω 抵抗器の一つのピンが PIC コンポーネントのピン６にどのように接続されているかに注目しましょう。これがピンを接続するのに使ったラベル付けによる方法の結果です。回路図の煩雑さを軽減できるので、しばしばラベルは実際にワイヤーで接続するよりも好まれます。

11. 基板の外形を定義しましょう。上部ツールバーのドロップダウンメニューから *Edge.Cuts* レイヤーを選択します。右ツールバーにある 図形ラインを追加アイコン をクリックします。基板の各コーナーをクリックして基板外形をトレースします。緑色のレジスト端と基板端の間には少し隙間を残すことを覚えておきましょう。



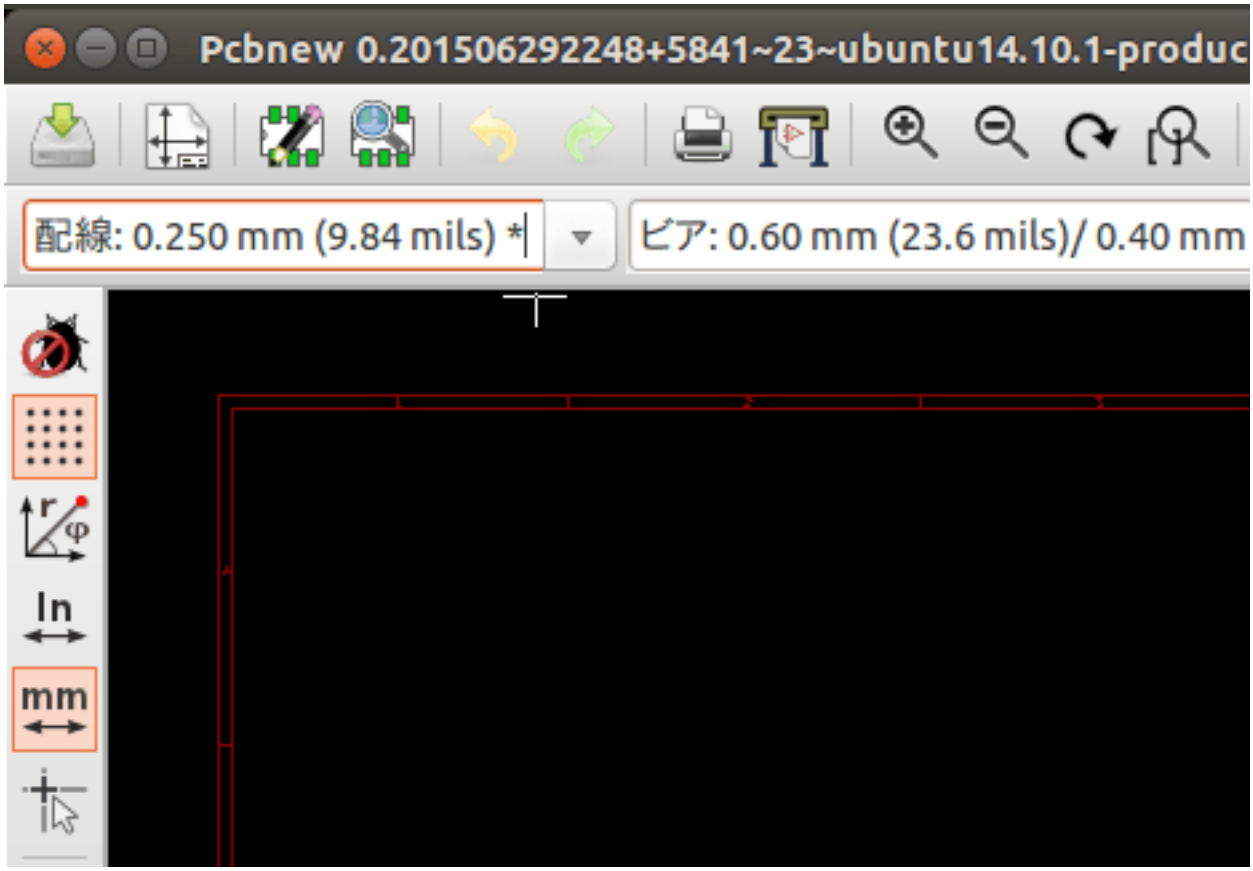
12. 次に GND を除く全てのワイヤーを接続します。実際のところ、基板裏側の銅箔 (*B.Cu* と呼ばれます) に配置されたグラウンド面を使うことで、全ての GND 接続は纏めて接続されます。

13. ここで作業する導体レイヤーを選択しなければなりません。上部ツールバーのドロップダウンメニューにある *F.Cu (PgUp)* を選択します。これは表面の導体レイヤーです。



14. 例えば、代わりに４層基板にすると決めたなら、セットアップ → レイヤーセットアップへ行って 導体レイヤーを４に変更します。レイヤーの表では、レイヤーに名前を付けて、用途を決めることができます。プリセットレイヤーグループメニューで選択可能な、便利なプリセットがあることを覚えておきましょう。

15. 右ツールバーにある 配線アイコン をクリックします。*J1* のピン１をクリックしてパッド *R2* への配線を引きます。配線の終点を設定するには、ダブルクリックして下さい。この配線幅は、デフォルトの 0.250 mm になっています。上部ツールバーのドロップダウンメニューから配線幅を変更できます。デフォルトでは一つの配線幅しか利用できないことに注意して下さい。



16. 更に配線幅を追加したい場合は、セットアップ → デザインルール → グローバルデザインルールタブと辿って、このウィンドウの右下にある表に利用したい配線幅を追加します。これで基板のレイアウト時にドロップダウンメニューから配線幅を選ぶことができるようになります。以下の例 (インチ単位) を参照して下さい。

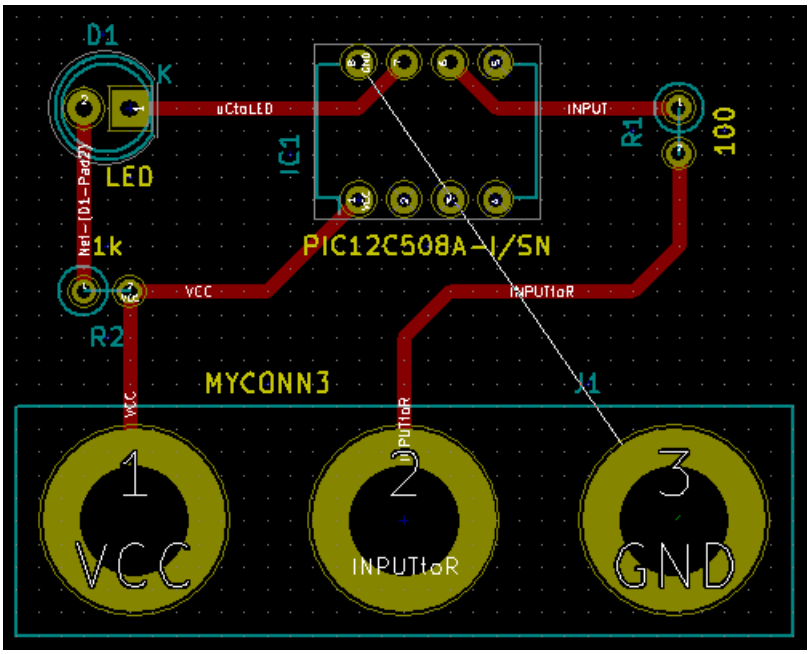
カスタム配線幅:

	幅
配線 1	0.01
配線 2	0.02
配線 3	0.05
配線 4	0.08
配線 5	0.10
配線 6	0.15
配線 7	0.20

17. あるいは、指定したオプションのセットの中でネットクラスを追加することができます。セットアップ → デザインルール → ネットクラスエディターを表示して *power* という新しいクラスを追加します。配線幅を 8 ミル (0.0080 と表示) から 24 ミル (0.0240 と表示) に変更します。次に *power* クラスに GND 以外の全てを追加します。(左で *default* を選択、右で *power* を選択、矢印を使います)

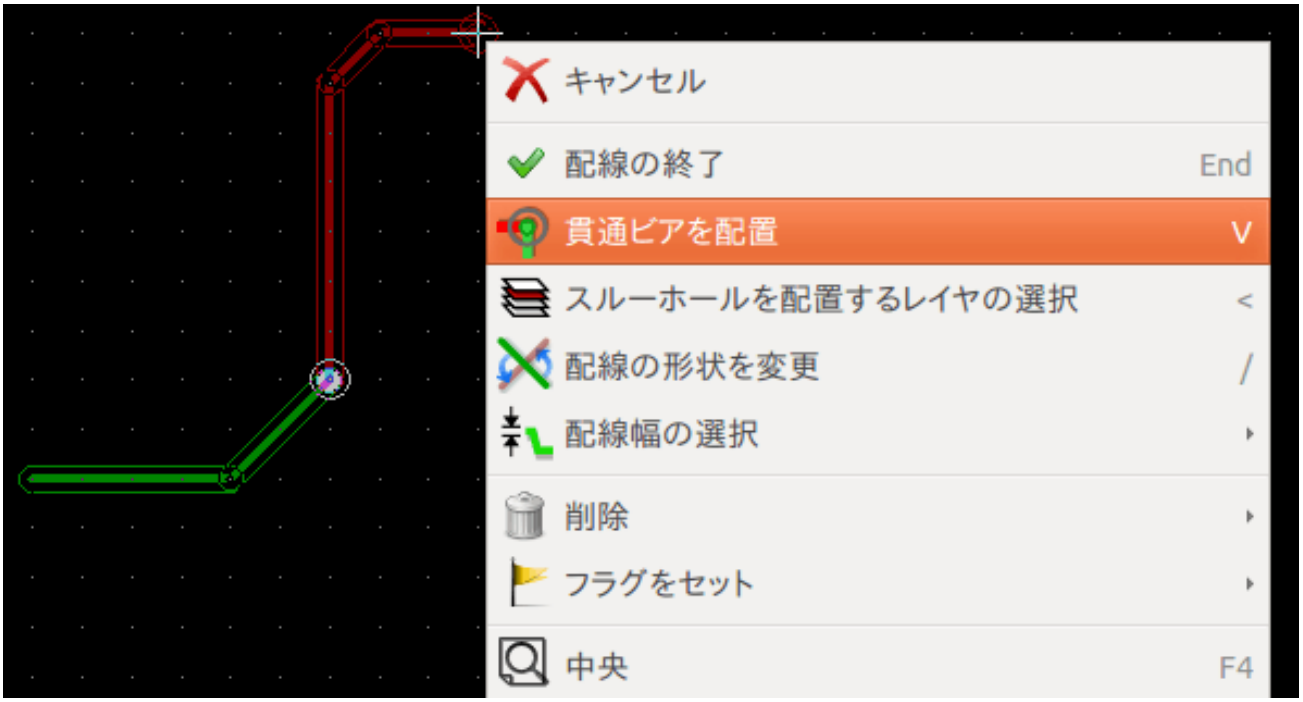
18. グリッドサイズを変更したい場合は 右クリック → グリッドと辿ります。コンポーネントを配置したり、配線で接続する前、或いは後に、状況に応じて適したグリッドサイズを選択するようにして下さい。

19. J1 のピン 3 を除く全ての配線が終わるまで、この作業を繰り返します。あなたの基板は以下の例のようになるでしょう。



20. 基板の他の銅箔面で配線を行いましょう。上部ツールバーのドロップダウンメニューで *B.Cu* を選択します。配線アイコン をクリックします。J1 のピン３と U1 のピン８の間の配線を行って下さい。グラウンド面で接続することができるので、この配線は実際には必要ありません。配線の色が変わっていることに注目しましょう。

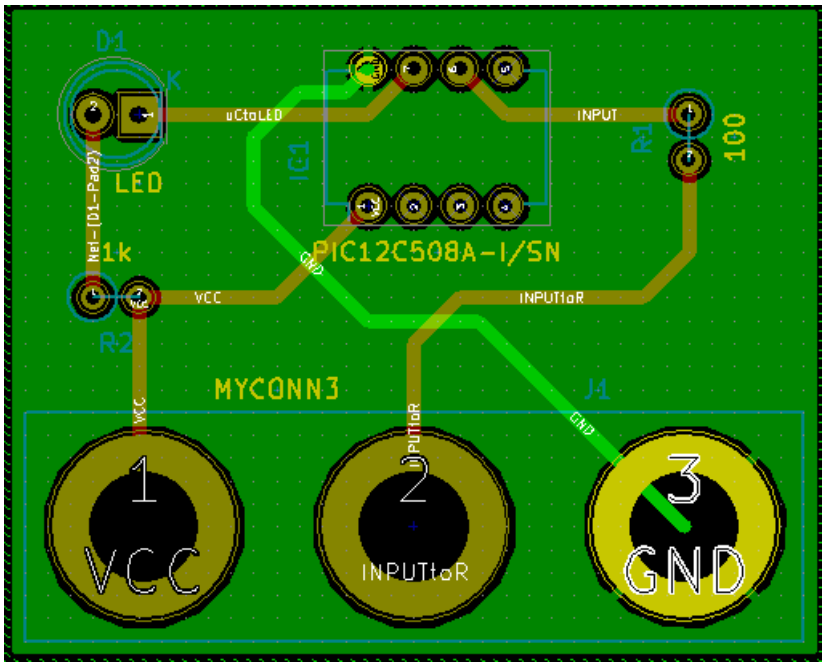
21. レイヤーを変更して、ピン **A** からピン **B** に行きましょう。ビアを配置することで、配線している銅箔面を変更できます。表面の銅箔面に配線している時に、右クリックして ビアを配置を選ぶか単に [v] を押して下さい。これにより、配線を終えたところで裏面レイヤーに移ります。



22. 特定の接続を検査したい場合は、右ツールバーにある ネットをハイライトアイコン をクリックします。J1 のピン３をクリックして下さい。配線そのものとそれに接続されている全てのパッドがハイライトされるでしょう。

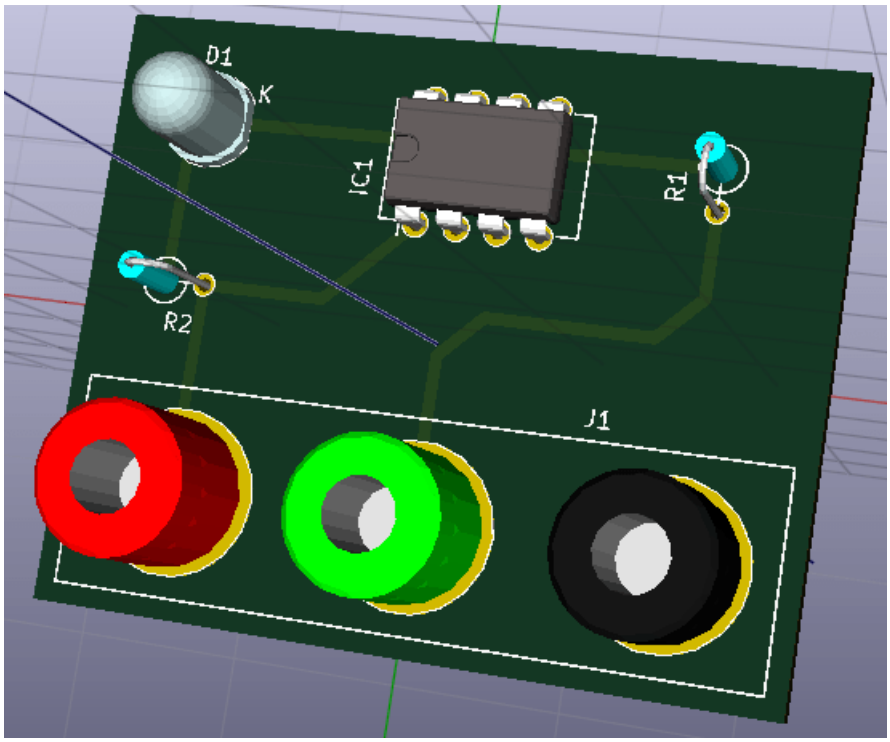
23. 全ての GND ピンに接続されるグラウンド面を作っていきます。右ツールバーにある 塗り潰しゾーンを追加アイコン をクリックして下さい。基板の周囲に長方形を描くつもりで、角にしたいところに来たらクリックして下さい。表示されたダイアログでは、デフォルトのパッド接続を サーマルリリーフに、外枠の角度を *0, 45, 90* 度に設定して、OK をクリックします。

24. 順番にそれぞれの角でクリックして基板外形を描いていきます。最初の角に戻ったら再度クリックして長方形を完成させます。今、描いた領域の内側で右クリックして下さい。ゾーン →' 全て塗り潰し' をクリックします。基板は緑色に塗り潰され、このように見えるでしょう。:



25. 上部ツールバーにある デザインルールチェックを実行アイコン をクリックして、デザインルールチェックを実行します。*DRC* を起動をクリックして下さい。エラーなしのはずです。未結線情報リストをクリックします。未接続のアイテムはないはずです。DRC ダイアログを閉じるには、OK をクリックします。

26. ファイル → 保存をクリックしてファイルを保存します。3D で基板を閲覧するためには 表示 → **3D** ビューアをクリックします。



27. PCB の周囲でマウスをドラッグして PCB を回転させることができます。

28. 基板はこれで完成です。メーカーに送るためにはガーバーファイルを生成する必要があります。

5.2 ガーバーファイルの生成

基板が完成したら、全てのレイヤーのガーバーファイルを生成して、基板を製造してくれる、お好みの基板メーカーに送ることができます。

1. KiCad から、*Pcbnew* 基板エディターを開きます。
2. ファイル → プロットをクリックして下さい。出力フォーマットとして ガーバーを選択し、ガーバーファイルを保存するフォルダを選択します。製造ファイル出力ボタンを押して出力します。
3. ドリルファイルを生成するには、*Pcbnew* から ファイル → プロットを再び行います。デフォルトの設定でよいでしょう。
4. これらは典型的な 2 層 PCB を製造するために選択する必要があるレイヤです:

レイヤー	**KiCad** レイヤー名	デフォルトのガーバー拡張子	**"Protel** の拡張子を使用" が有効な場合
裏面導体レイヤー	B.Cu	.GBR	.GBL
表面導体レイヤー	F.Cu	.GBR	.GTL
表面シルク面	F.SilkS	.GBR	.GTO
裏面ハンダマスク	B.Mask	.GBR	.GBS
表面ハンダマスク	F.Mask	.GBR	.GTS
基板外形	Edge.Cuts	.GBR	.GM1

5.3 GerbView の使用

1. 全てのガーバーファイルを見るには、KiCad プロジェクトマネージャーに行って、*GerbView* アイコンをクリックします。ドロップダウンメニューまたはレイヤーマネージャーで、グラフィックレイヤー *1* を選択します。ファイル → ガーバーファイルを開くをクリックするか、アイコン をクリックします。選択して、生成された全てのガーバーファイルを開きます。それら全てが積み重なって表示されることに着目して下さい。
2. ファイル → **Excellon** ドリルファイルを開くと辿って、ドリルファイルを開きます。
3. 表示するレイヤーを選択/非選択にするには、右側にあるレイヤーマネージャーを使用します。メーカーに送る前に各レイヤーを注意深く検査しましょう。
4. 表示は Pcbnew と同様です。グリッドを変更するには、表示領域内で右クリックして グリッドをクリックして下さい。

5.4 FreeRouter による自動配線

基板の手配線は早くて楽しいですが、大量のコンポーネントを含む基板の場合には自動配線を使いたくなるかもしれません。最初に手配線で重要な配線を行ってから、うんざりするような仕事を自動配線にセットすべきだということを覚えておいて下さい。自動配線は未配線にだけ働きます。ここで私達が使う自動配線ツールは、FreeRouting です。

注意

FreeRouting はオープンソースの java アプリケーションです。今のところ FreeRouting は、"freerouting" でインターネットを検索すると見つかる、幾つかのほとんど同一のコピーとして存在しています。ソースコードのみ、あるいはプリコンパイルされた java パッケージとして見つけられるでしょう。

---

1. *Pcbnew* で ファイル → エクスポート → **Specctra DSN** をクリックし、ローカルにファイルを保存します。FreeRouter を起動して、*Open Your Own Design* ボタンをクリックし、*dsn* ファイルを選んで読み込ませます。
2. FreeRouter は、手配線と自動配線の両方で、KiCad が現時点で持っていない特徴を持っています。FreeRouter は２つの主要なステップで操作します: 最初に基板に配線を行い、次にそれを最適化します。完全最適化には時間がかかるかもしれませんが、必要ならいつでも停止することができます。
3. 上部ツールバーにある *Autorouter* ボタンをクリックすることで自動配線を開始できます。下部のバーは実行中の配線作業の情報を知らせてくれます。もしも *Pass* カウントが 30 以上になったら、あなたの基板はおそらくこのルーターでは自動配線できないでしょう。コンポーネント間隔をもっと広げたり良い向きに回転させてから再挑戦しましょう。部品の回転と位置決めのゴールは、ラッツネストの空中交差の数を少なくすることです。
4. マウスの左クリックにより、自動配線を停止して最適化プロセスを自動的に開始させることができます。もう一度左クリックすると、最適化プロセスが停止します。停止する必要がないのであれば、FreeRouter が仕事を終えるのに任せるのが良いでしょう。
5. **File** → **Export Specctra Session File** メニューをクリックして基板ファイルを *.ses* 拡張子で保存します。FreeRouter ルールファイルは保存する必要ありません。
6. *Pcbnew* に戻りましょう。ファイル → インポート → **Spectra Session** をクリックして *.ses* ファイルを選択すると、新たに配線された基板をインポートすることができます。

もし気に入らない配線があれば、配線を削除して再配線できます。[Delete] と、右ツールバーにある 配線アイコン の配線ツールを使いましょう。

Chapter 6

KiCad のフォワードアノテーション

電気回路図、フットプリントの割り当て、基板レイアウト、ガーバーファイルの生成を完了すると、基板を実際に作る基板メーカーに全てを送る準備が整います。

しばしば、この線形の作業の流れは一方向ではなくなります。例えば、既にこの作業の流れを完了していたはずの基板を変更/拡張しなければならない時ですが、あなたは、部品を動かしたり、他のものと入れ替えたり、部品を変更したり、等々する必要が起こりえます。この変更の作業の中で、あなたがやりたくないことは基板全体を一から配線し直すことでしょう。代わりにこのようにすべきです:

1. 仮想的なコネクタの CON1 を CON2 と入れ替えたいと想定しましょう。
2. あなたは既に完成した回路図と配線を終えた PCB を持っています。
3. KiCad から *Eeschema* を起動して、CON1 を削除し CON2 を追加して変更を行います。アイコン で回路図プロジェクトを保存して、上部ツールバーにある ネットリストの生成のアイコン をクリックします。
4. ネットリストをクリックし 保存します。デフォルトのファイル名で保存しましょう。古いファイルを上書きします。
5. CON2 にフットプリントを割り当てます。上部ツールバーにある *CvPcb(*コンポーネントとフットプリントの関連付け*)* を実行のアイコン をクリックします。新しいデバイスの CON2 にフットプリントを割り当てます。残りのコンポーネントは以前に割り当てたフットプリントのままです。*CvPcb* を閉じます。
6. Eeschema に戻り、ファイル → 回路図プロジェクトの保存をクリックしてプロジェクトを保存します。Eeschema を閉じます。
7. KiCad プロジェクト・マネージャから *Pcbnew* のアイコンをクリックします。*Pcbnew* のウィンドウが開きます。
8. 古い、配線済みの基板が自動的に開きます。新しいネットリストファイルをインポートしましょう。上部ツールバーにある ネットリストの読込みのアイコン をクリックします。
9. ネットリストファイルを参照するボタンをクリックして、ネットリストファイルを選択し、現在のネットリストを読み込むをクリックします。そして 閉じるボタンをクリックします。

10. この時点で、以前の全ての部品が配線済みのレイアウトが見られます。左上の隅に全ての未配線の部品、今回の例では CON2 が見えるでしょう。マウスで CON2 を選択します。基板の中央まで移動しましょう。
11. CON2 を配置して配線します。完了したらいつものように保存してガーバーファイルの生成をしましょう。

ここに書いた手順は必要に応じて何度でも容易に繰り返すことが可能です。上述したフォワード・アノテーションとは別に、バック・アノテーションとして知られる別の方法があります。この方法は Pcbnew で配線済みにした基盤の変更を、回路図とネットリストに反映することを可能にします。が、バック・アノテーションはそんなに便利ではないのでここでは書きません。

Chapter 7

KiCad 回路図シンボルの作成

時には、回路図に配置したいシンボルが KiCad のライブラリの中にないことがあります。これは当たり前のことで嘆くことではありません。この章では、新しい回路図シンボルが KiCad でいかに素早く作れるのかを見ていきましょう。しかし、KiCad のコンポーネントはいつでもインターネット上で見つけられることも覚えておきましょう。

KiCad では、シンボルは DEF で始まり ENDDEF で終わるテキストの断片です。通常は一つ以上のシンボルが拡張子 *.lib* のライブラリーファイルに含まれています。シンボルをライブラリーファイルに追加したければ、テキストエディターのカット＆ペーストコマンドを使うだけです。

7.1 コンポーネントライブラリエディターの使用

1. 新しいコンポーネントの作成には コンポーネント・ライブラリ・エディタ (*Eeschema* の一部) を使います。プロジェクトフォルダ *tutorial1* 内に *library* というフォルダを作りましょう。新コンポーネントを作ったら、そこに新しいライブラリファイル *myLib.lib* を置きます。

2. さあ新しいコンポーネントの作成を始めましょう。KiCad から *Eeschema* を起動して、コンポーネント・ライブラリ・エディタのアイコン をクリックし、新規コンポーネント作成のアイコン をクリックします。コンポーネント・プロパティのウィンドウが現れます。新しいコンポーネントを *MYCONN3* と名付け、デフォルトのリファレンス記号を *J* に、パッケージ内のユニット数を *1* に設定します。OK をクリックします。警告が出たら yes をクリックしておきます。この時点ではコンポーネントは、そのラベルだけで構成されています。ピンをいくつか足してみましょう。右ツールバーにある コンポーネントにピンを追加のアイコン をクリックします。ピンを配置するには、シートの *MYCONN3* ラベルの下あたりを左クリックします。

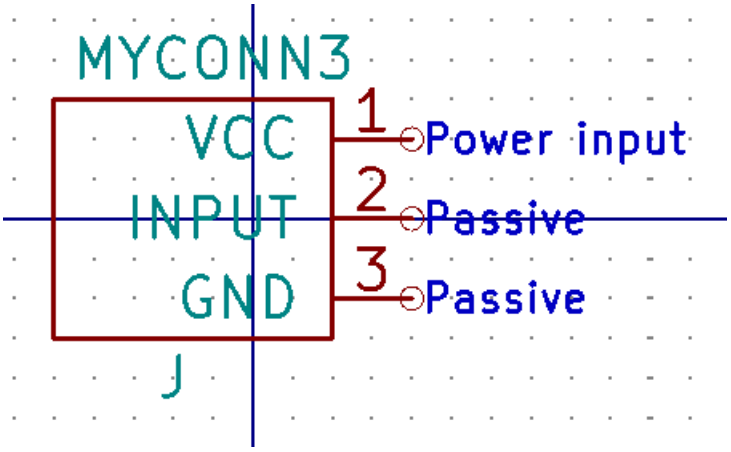
3. 現れた ピンのプロパティウィンドウで、ピン名を *VCC* 、ピン番号を *1* 、エレクトリック・タイプを 電源入力に設定して OK をクリックします。

4. 適当な場所、*MYCONN3* ラベルの右下あたり、をクリックしてピンを配置します。

5. ピン配置のステップを繰り返します。今度は ピン名は *INPUT* で、ピン番号は *2* で、エレクトリックタイプは パッシブとします。

6. ピン配置のステップを繰り返します。今度は ピン名は *GND* で、ピン番号は *3* で、エレクトリックタイプはパッシブとします。ピンを順に重ねて配置します。コンポーネントのラベルの *MYCONN3* をページの中心 (青いラインが交差する所) にします。

7. 次にコンポーネントの輪郭を描きましょう。コンポーネントのボディに矩形を入力のアイコン をクリックします。以下に示すようにピンに隣接して長方形を描きます。まず、長方形の左上の角にしたい所をクリックします（マウスボタンを押したままにしません)。そして、長方形の右下の角にしたい所をクリックします。



8. 矩形を黄色で塗りつぶしたい場合、まず 設定 → 色の設定で" ボディ背景色" を" 黄色 4" に設定します。そし

て、編集画面上で塗りつぶしたい矩形の枠上にマウスカーソルを置いて [e] を押して 図形のプロパティウィンドウを表示し、背景色で塗りつぶしを選択します。

9. コンポーネントをあなたのライブラリ *myLib.lib* に保存しましょう。新しいライブラリへ現在のコンポーネントを保存のアイコン をクリックして、フォルダ *tutorial1/library/* を選び、新しいライブラリファイルを *myLib.lib* という名前で保存します。

10. 設定 → コンポーネントライブラリで、*tutorial1/library/* を ユーザ定義の検索パスに追加し、*myLib.lib* を コンポーネントライブラリファイルに追加します。

11. 作業ライブラリの選択のアイコン をクリックします。ライブラリの選択ウィンドウ内で *myLib* を選択して OK をクリックします。ウィンドウの上部が現在使用中のライブラリを示しており、それが *myLib* であることに注意しましょう。

12. 上部ツールバーにある 現在のライブラリ内の現在のコンポーネントを更新のアイコン をクリックします。上部ツールバーにある ディスクに現在のライブラリを保存のアイコン をクリックして変更を全て保存します。どの確認メッセージにも はいをクリックして下さい。新しい回路図コンポーネントは完成して、ウィンドウのタイトルバーに示されているライブラリから使えます。

13. コンポーネント・ライブラリ・エディタのウィンドウを閉じます。Eeschema のウィンドウに戻ります。あなたの新しいコンポーネントはライブラリ *myLib* から利用できます。

14. ライブラリパスに追加することによって、ライブラリの *file.lib* ファイルを利用できるようになります。*Eeschema* から 設定 → コンポーネントライブラリとして、それへのパスを ユーザ定義の検索パスに追加し、*file.lib* を コンポーネントライブラリファイルに追加します。

7.2 コンポーネントのエクスポート、インポート、変更

一からコンポーネントを作るよりも、既に作られたものから始めてそれを変更する方が簡単なこともあります。この章では、KiCad の標準ライブラリ *device* から、あなた独自のライブラリ *myOwnLib.lib* にコンポーネントをエクスポートして、それを変更する方法を見ていきましょう。

1. KiCad から *Eeschema* を起動して、コンポーネント・ライブラリ・エディタのアイコン をクリックして、作業ライブラリの選択のアイコン をクリックし、*device* ライブラリを選びます。現在のライブラリからエディタへコンポーネントを読み込むのアイコン をクリックして *RELAY_2RT* をインポートします。

2. コンポーネントのエクスポートのアイコン をクリックして、フォルダ *library/* へ行き、新しいライブラリファイルを *myOwnLib.lib* という名前で保存します。

3. ライブラリパスに追加することで、このコンポーネントとライブラリ全体 *myOwnLib.lib* が利用できるようになります。*Eeschema* から 設定 → コンポーネントライブラリとして、*library/* を ユーザ定義の検索パスに追加し、*myOwnLib.lib* を コンポーネントライブラリファイルに追加します。追加したら"OK" ボタンを押してウィンドウを閉じます。

4. 作業ライブラリの選択のアイコン をクリックします。ライブラリの選択ウィンドウ内で *myOwnLib* を選択して OK をクリックします。ウィンドウの上部が現在使用中のライブラリを示しており、それが *myOwnLib* であることに注意しましょう。

5. 現在のライブラリからエディタへコンポーネントを読み込むのアイコン をクリックして *RELAY_2RT* をインポートします。

6. これでコンポーネントを好きなように変更できます。マウスカーソルをラベル *RELAY_2RT* に重ねて [e] を押して *MY_RELAY_2RT* にリネームします。

7. 上部ツールバーにある 現在のライブラリ内の現在のコンポーネントを更新のアイコン をクリックします。上部ツールバーにある ディスクに現在のライブラリを保存のアイコン をクリックして変更を全て保存します。

7.3 quicklib による回路図コンポーネントの作成

この章では、インターネットのツール *quicklib* を使った、回路図コンポーネント MYCONN3 (前の「MYCONN3」参照) の別の作成方法を紹介します。

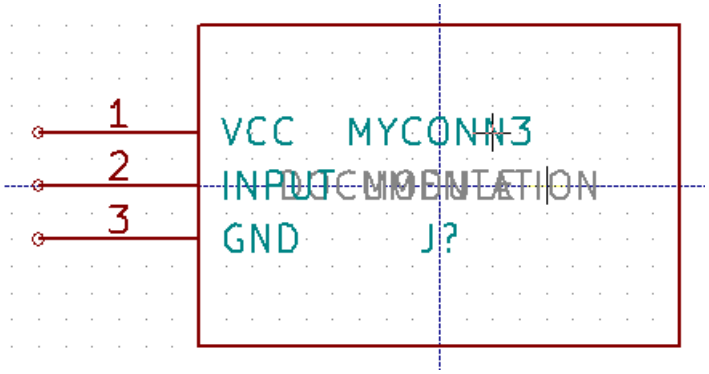
1. *quicklib* のウェブページに行きましょう: http://kicad.rohrbacher.net/quicklib.php

2. ページに次の情報を入力しましょう: Component name: MYCONN3、Reference Prefix: J、Pin Layout Style: SIL、Pin Count, N: 3

3. *Assign Pins* をクリックします。ページに次の情報を入力しましょう: Pin 1: VCC、Pin 2: input、Pin 3: GND 。Type は 3 つのピンとも Passive を選択します。

4. *Preview* をクリックします。満足なら *Build Library Component* をクリックします。ファイルをダウンロードして *tutorial1/library/myQuickLib.lib* と名付けます。できました！

5. それを KiCad を使って見てみましょう。KiCad プロジェクト・マネージャから *Eeschema* を起動して、コンポーネント・ライブラリ・エディタのアイコン をクリックし、コンポーネントのインポートのアイコン をクリックし、*tutorial1/library/* へ行き *myQuickLib.lib* を選択します。



6. ライブラリパスに追加することで、このコンポーネントとライブラリ全体 *myQuickLib.lib* が利用できるようになります。*Eeschema* から 設定 → コンポーネントライブラリとして、*library/* を ユーザ定義の検索パスに追加し、*myQuickLib.lib* を コンポーネントライブラリファイルに追加します。

このコンポーネント作成方法は、たくさんのピンを持つコンポーネントを作りたい時に非常に効果的であると想像できるでしょう。

7.4 大量ピンの回路図コンポーネントの作成

quicklib による回路図コンポーネントの作成の章で、ウェブベースのツール *quicklib* を使った回路図コンポーネントの作成方法を見ました。しかし、時に、大量ピン (数百のピン) を持つ回路図コンポーネントを作る必要があることに気付くでしょう。KiCad では、これはそんなにややこしい仕事ではありません。

1. 50 ピンのデバイスのための回路図コンポーネントを作成したいとします。複数の少量ピンの図形、例えば 25 ピンの図形を 2 つ、を使ってそれを描くのが一般的な方法です。このコンポーネント表現はピン接続が容易でしょう。
2. このコンポーネントを作成する最良の方法は、*quicklib* を使って別々に 25 ピンのコンポーネントを生成して、Python スクリプトでそれらのピン番号を振り直し、最終的に 2 つを統合して一組の DEF と ENDDEF の中にコピー＆ペーストすることです。
3. 以下に *in.txt* ファイルと *out.txt* ファイルと連動して使えるシンプルな Python スクリプトの例を示します。これは *in.txt* ファイルの中の全ての行に対して、`X PIN1 1 -750 600 300 R 50 50 1 1 I` を `X PIN26 26 -750 600 300 R 50 50 1 1 I` のように数字の振り替えをします。

シンプルなスクリプト

```
#!/usr/bin/env python
''' simple script to manipulate KiCad component pins numbering'''
import sys, re
try:
    fin=open(sys.argv[1],'r')
    fout=open(sys.argv[2],'w')
except:
    print "oh, wrong use of this app, try:", sys.argv[0], "in.txt out.txt"
    sys.exit()
for ln in fin.readlines():
    obj=re.search("(X PIN)(\d*)(\s)(\d*)(\s.*)",ln)
if obj:
    num = int(obj.group(2))+25
    ln=obj.group(1) + str(num) + obj.group(3) + str(num) + obj.group(5) +'\n'
    fout.write(ln)
fin.close(); fout.close()
#
# for more info about regular expression syntax and KiCad component generation:
# http://gskinner.com/RegExr/
# http://kicad.rohrbacher.net/quicklib.php
```

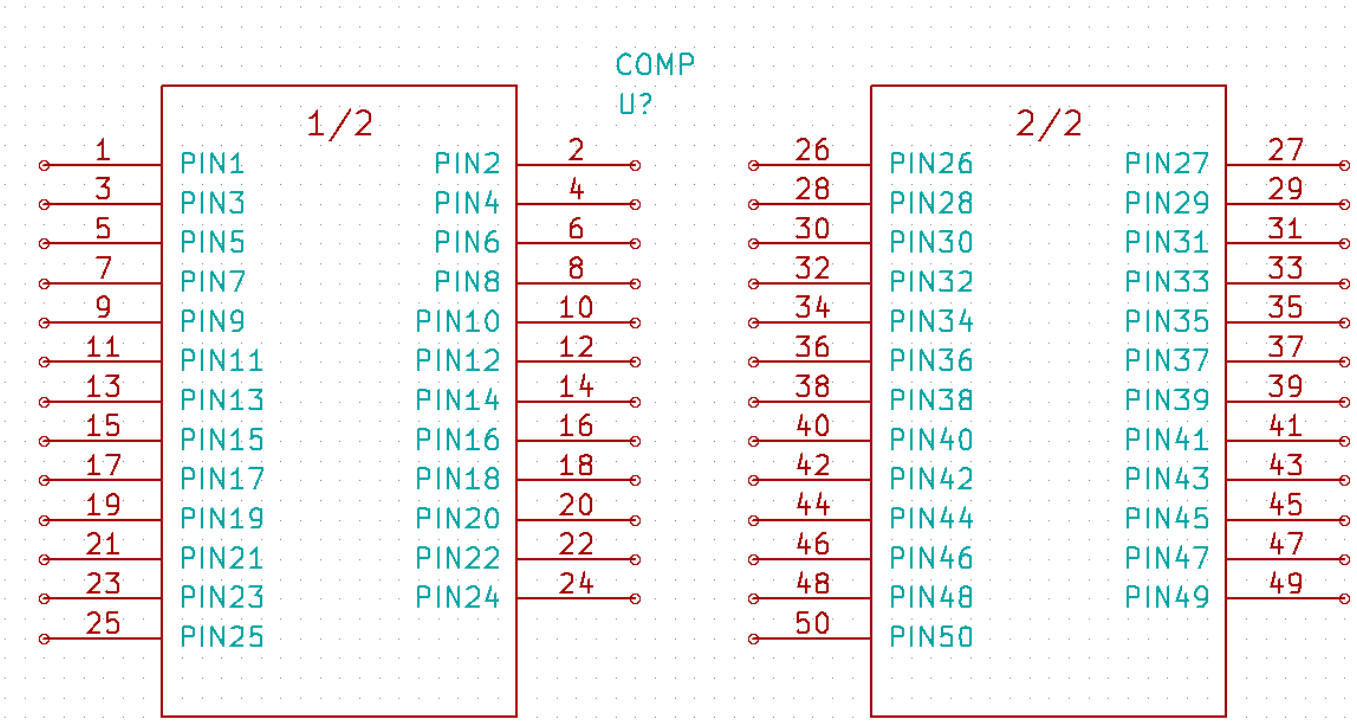
1. 2 つのコンポーネントを一つに統合するために、Eeschema からコンポーネント・ライブラリ・エディタを使って、1 番目のコンポーネントを移動し、2 番目に重ならないようにしてやる必要があるでしょう。以下に最終的な.lib ファイルとその *Eeschema* での表現を示します。

***.lib ファイルの内容**

```
Eeschema-LIBRARY Version 2.3
#encoding utf-8
# COMP
DEF COMP U 0 40 Y Y 1 F N
F0 "U" -1800 -100 50 H V C CNN
F1 "COMP" -1800 100 50 H V C CNN
DRAW
S -2250 -800 -1350 800 0 0 0 N
S -450 -800 450 800 0 0 0 N
X PIN1 1 -2550 600 300 R 50 50 1 1 I

...

X PIN49 49 750 -500 300 L 50 50 1 1 I
ENDDRAW
ENDDEF
#End Library
```



1. ここで紹介している Python スクリプトは、ピン番号とピンラベルを操作するとても強力なツールです。その威力は、難解ではありますが驚くほどに便利な正規表現構文による、ということを覚えておいて下さい: http://gskinner.com/RegExr/

Chapter 8

フットプリントの作成

他の、一種類のライブラリに回路図記号とフットプリントの両方を含む EDA ソフトウェア・ツールと異なり、KiCad の *.lib* ファイルは回路図記号を、*.kicad_mod* ファイルはフットプリントを含んでいます。*CvPcb* は回路図記号にフットプリントを割り当てるのに使われます。

.lib と同様に、*.kicad_mod* ファイルはテキストファイルで、一つのフットプリントを含むことができます。

KiCad には豊富なフットプリントのライブラリがありますが、あなたが必要とするフットプリントが KiCad のライブラリ内にないこともあるでしょう。ここでは KiCad で新しい PCB フットプリントを作成する手順を示します:

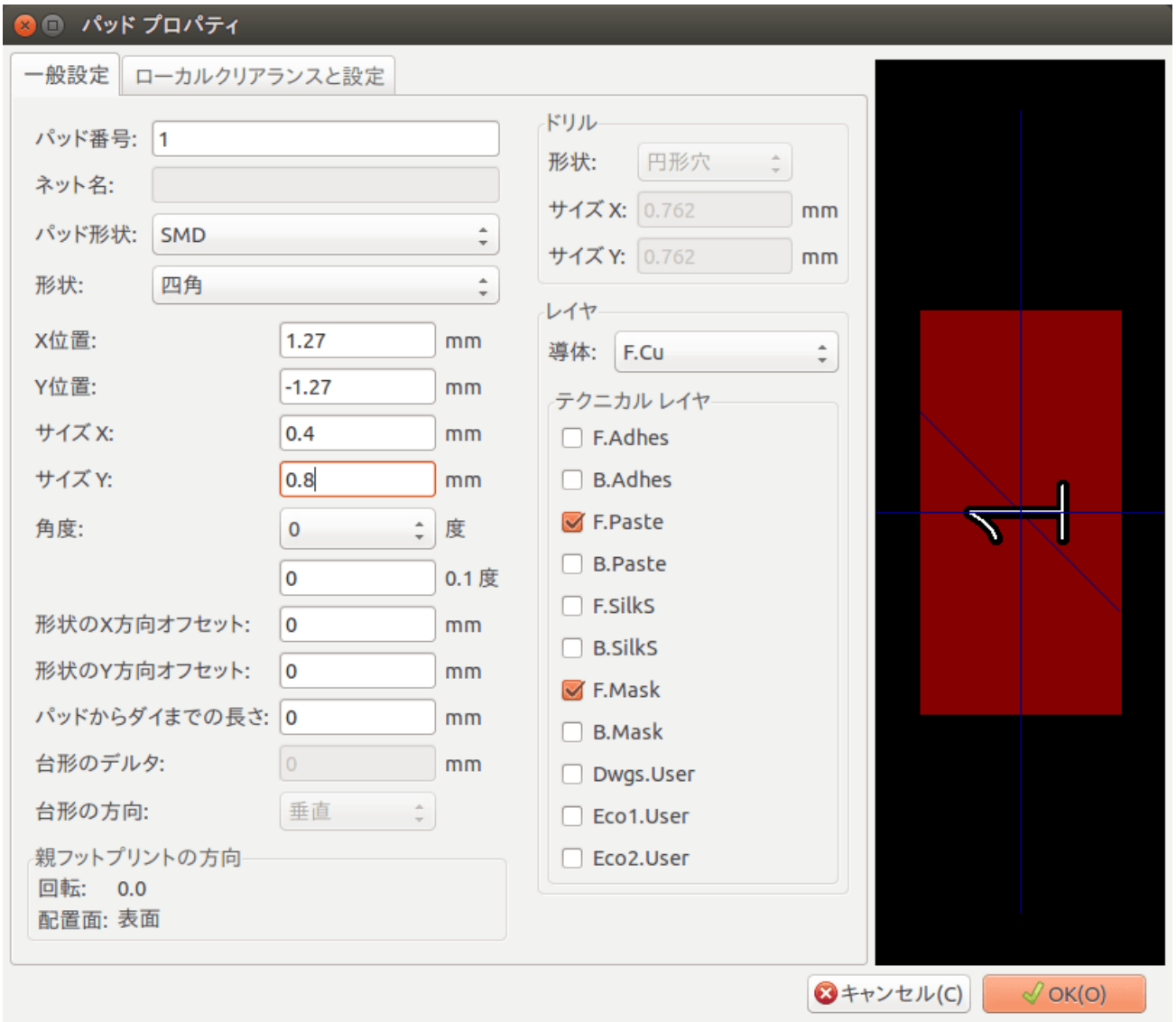
8.1 フットプリント・エディタの使用

1. KiCad プロジェクト・マネージャから *Pcbnew* ツールを起動します。上部ツールバーにある フットプリントエディタを開くのアイコン をクリックします。フットプリント・エディタが開きます。

2. 新しいフットプリントライブラリー *myfootprint* に新規フットプリント *MYCONN3* を保存します。*tutorial1/* プロジェクトフォルダーに新しいフォルダー *myfootprint.pretty/* を作成して下さい。設定 → フットプリントライブラリーを管理をクリックし、ライブラリーを追加ボタンを押します。テーブルで、別名 (ニックネーム) には"myfootprint" 、ライブラリーのパスには"${KIPRJMOD}/myfootprint.pretty"、プラグインの種類には"KiCad" を入力します。ライブラリーテーブルのウィンドウを閉じるには OK を押して下さい。上部ツールバーの アクティブなライブラリーを選択アイコン をクリックします。*myfootprint* ライブラリーを選択して下さい。

3. 上部ツールバーにある 新規フットプリントのアイコン をクリックします。フットプリント名に *MYCONN3* を入力します。画面中央に *MYCONN3* ラベルが表示されます。そのラベルの下に *REF** があります。MYCONN3 の上で右クリックし、REF の上方に移動します。REF* 上で右クリックして、テキストの編集を選び、*SMD* にリネームします。テキストプロパティの 表示の項目で 非表示を選択します。

4. 右ツールバーにある パッドを追加のアイコン を選択します。パッドを配置するため画面をクリックします。新しいパッド上で右クリックし パッドを編集をクリックします。代わりに [e] も使えます。



5. パッド番号を *1* に、形状を 四角に、パッド形状を *SMD* に、サイズ *X* を *0.4* に、サイズ *Y* を *0.8* に設定し、OK をクリックします。再度 パッド入力をクリックし、もう 2 つパッドを配置します。
6. グリッドサイズを変更したいなら 右クリック → グリッドの選択とします。配置する前に適切なグリッドサイズを選択しましょう。
7. 上の図のように *MYCONN3* と *SMD* のラベルを外側に移動します。
8. パッドを配置する時には、相対的な距離を測る必要があります。相対座標軸の *(0,0)* としたい所にカーソルを置き、スペースキーを押します。カーソルを動かすと、カーソル位置の相対的な指標がウィンドウの下方に見えるでしょう。新しい原点を設定するにはスペースキーを押しましょう。
9. フットプリントの輪郭を加えましょう。右ツールバーにある 図形ライン（またはポリゴン）を入力のボタン をクリックします。パーツの周囲にコネクタの外形を描きます。
10. 上部ツールバーにある アクティブなライブラリへフットプリントを保存のアイコン をクリックし、デフォルト名の MYCONN3 で保存します。

Chapter 9

KiCad プロジェクトファイルの可搬性について

あなたの KiCad プロジェクトを、他の人が完全に取り込んで使えるために、どのファイルを送る必要があるでしょうか？

誰かと KiCad のプロジェクトを共有する時、回路図ファイル *.sch* 、基板ファイル *.kicad_pcb* 、プロジェクトファイル *.pro* 、ネットリストファイル *.net* を、コンポーネントライブラリファイル *.lib* 及びフットプリントライブラリファイル *.kicad_mod* と一緒に送ることが重要です。この方法でのみ完全に自由に回路図と基板を変更することができます。

KiCad の回路図では、回路図記号を含む *.lib* ファイルが必要です。それらのライブラリファイルは *Eeschema* の設定で読み込まれる必要があります。一方、基板 (*.kicad_pcb* ファイル) では、フットプリントは *.kicad_pcb* ファイルの中に保存させることができます。誰かに *.kicad_pcb* ファイルだけを送っても、彼はその基板を見たり編集したりできるわけです。しかし、もし彼がネットリストからコンポーネントを読み込みたいのであれば、フットプリントライブラリ (*.kicad_mod* ファイル) が存在し、回路図の場合と同様に、*Pcbnew* の設定で読み込まれている必要があります。同様に、*CvPcb* でモジュールが表示されるようにするためには、*Pcbnew* の設定で *.kicad_mod* ファイルが読み込まれていることが必要です。

もし誰かがフットプリント付きの *.kicad_pcb* ファイルを送ってくれたら、あなたはそれを他の基板でも利用したいでしょう。フットプリント・エディタを開き、現在の基板からフットプリントを読み込み、保存するか別のフットプリント・ライブラリ・ファイルにエクスポートします。*.kicad_pcb* ファイルから全てのフットプリントを一度にエクスポートすることもできます。**Pcbnew** → ファイル → フットプリントのアーカイブ → フットプリントアーカイブの作成で、基板上の全てのフットプリントの *.kicad_mod* ファイルが生成されます。

結論として、PCB だけを配布したいのなら、基板ファイル *.kicad_pcb* だけで十分です。しかし、あなたが他の人に、回路図と部品と基板の、利用と変更の完全な自由を与えたいのならば、以下のプロジェクトディレクトリを zip して送ることをお勧めいたします。

```
tutorial1/
|-- tutorial1.pro
|-- tutorial1.sch
|-- tutorial1.kicad_pcb
|-- tutorial1.net
```

```
|-- library/
|   |-- myLib.lib
|   |-- myOwnLib.lib
|   \-- myQuickLib.lib
|
|-- myfootprint.pretty/
|   \-- MYCONN3.kicad_mod
|
\-- gerber/
    |-- ...
    \-- ...
```

Chapter 10

KiCad ドキュメントの詳細

この文書は KiCad の大部分の機能のクイックガイドでした。より詳細な手順については、それぞれの KiCad ツールから参照できるヘルプファイルを参照して下さい。ヘルプ → マニュアルをクリックです。

KiCad には、その 4 つのソフトウェアコンポーネント全てに、とても良い多言語のマニュアルが付いてきます。

全 KiCad マニュアルの英語版は KiCad と一緒に配布されます。

マニュアルに加えて、KiCad はこの、他の言語にも翻訳されたチュートリアルと一緒に配布されます。このチュートリアルは無料で最新版の KiCad と共に配布されます。このチュートリアルとマニュアルはご利用の KiCad と共にパッケージされています。

Linux の場合、ディストリビューションにもよりますが、一般的に次のディレクトリでしょう:

```
/usr/share/doc/kicad/help/en/
/usr/local/share/doc/kicad/help/en
```

Windows の場合:

```
<installation directory>/share/doc/kicad/help/en
```

OS X の場合:

```
/Library/Application Support/kicad/help/en
```

10.1 Web 上の KiCad ドキュメント

最新の KiCad のドキュメンテーションは、複数の言語に翻訳されて Web 上にあります。