KiCad ことはじめ

March 2, 2016

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

1.2 GNU/Linux の場合 2

1.3 Apple OS X の場合 3

1.4 Windows の場合 3

1.5 サポート 3

2 KiCad 作業の流れ **4**

2.1 KiCad 作業フローチャート 4

2.2 前方向と後方向のアノテーション 6

3 電子回路図の描画 **7**

3.1 Eeschema の使用 7

3.2 KiCad でのバス接続 21

4 プリント基板のレイアウト **23**

4.1 Pcbnew の使用 23

4.2 ガーバーファイルの生成 31

4.3 GerbView の使用 32

4.4 Freerouter による自動配線 32

5 KiCad の前方向アノテーション **34**

6 KiCad 回路図コンポーネントの作成 **36**

6.1 コンポーネント・ライブラリ・エディタの使用 36

6.2 コンポーネントのエクスポート、インポート、変更 39

6.3 quicklib による回路図コンポーネントの作成 39

6.4 大量ピンの回路図コンポーネントの作成 40

7 フットプリントの作成 **43**

7.1 フットプリント・エディタの使用 43

8 KiCad プロジェクトファイルの可搬性について **46**

9 KiCad ドキュメントの詳細 **48**

9.1 Web 上の KiCad ドキュメント 48

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 © 2010-2015 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 *

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

翻訳

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:

- 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 ソフトウェアの国際化について: https://github.com/KiCad/kicad-i18n/issues

発行日

2015 年 5 月 16 日

Chapter 1

KiCad のご紹介

KiCad は、電子回路図と PCB レイアウト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	拡張子
KiCad	プロジェクト・マネージャ	`*.pro`
Eeschema	回路図エディタ (回路図とコンポーネント)	`*.sch, *.lib, *.net`
CvPcb	フットプリント・セレクタ	`*.net`
Pcbnew	PCB レイアウト	`*.kicad_pcb`
GerbView	ガーバー・ビューア	全ての通常のガーバー
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、電気的安全間隔、カラーコード等のための計算機	なし
Pl Editor	図枠エディタ	`*.kicad_wks`

注意
拡張子の一覧は完全ではなく、KiCad が連携するファイルの一部のみを含んでおり、どのファイルがどの KiCad 固有のアプリケーションで使われるのかの基本的な理解に役立ちます。

KiCad は、複雑な電子基板の開発やメンテナンスに十分使えるほどに成熟していると考えることができます。

KiCad には基板サイズの制限がなく、最大 32 の導体レイヤ、最大 14 のテクニカルレイヤそして最大 4 の補助レイヤを容易に扱えます。KiCad は、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全てのファイルを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 のホームページ:

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 OS X、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を活用したりテストをするなら、あなたのプラットホームの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開発ブランチを可能な限り使用可能にしておくことを目標にしています。

1.2 GNU/Linux の場合

安定版： 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なければ、不安定版の指示に従って、最新の安定版を選択してインストールしてください。

不安定版（夜間の開発中ビルド）： 不安定版は最新のソースコードからビルドされます。不安定版は、ファイルを壊したり、悪いガーバーを生成したり等のバグがあることもありますが、一般的には安定しており、そして最新の機能を持っています。

Ubuntu の場合、KiCad の不安定版の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
```

Fedora の場合、KiCad の不安定版の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端末で次のようにタイプします:

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

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

1.3 Apple OS X の場合

安定版： OS X 用の安定版の KiCad は以下の URL で見つけられます:
http://downloads.kicad-pcb.org/osx/stable/

不安定版（夜間の開発中ビルド）： 不安定版は最新のソースコードからビルドされます。不安定版は、ファイルを壊したり、悪いガーバーを生成したり等のバグがあることもありますが、一般的には安定しており、そして最新の機能を持っています。

不安定版の夜間ビルドは以下の URL で見つけられます:
http://downloads.kicad-pcb.org/osx/

1.4 Windows の場合

安定版： Windows 用の安定版の KiCad は以下の URL で見つけられます:
http://downloads.kicad-pcb.org/windows/stable/

不安定版（夜間の開発中ビルド）： 不安定版は最新のソースコードからビルドされます。不安定版は、ファイルを壊したり、悪いガーバーを生成したり等のバグがあることもありますが、一般的には安定しており、そして最新の機能を持っています。

Windows 用の夜間ビルドは以下の URL で見つけられます:
http://downloads.kicad-pcb.org/windows/

1.5 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:

- 訪れてみましょう。Forum
- 参加してみましょう。#kicad IRC channel on Freenode
- 熟読してみましょう。Tutorials

Chapter 2

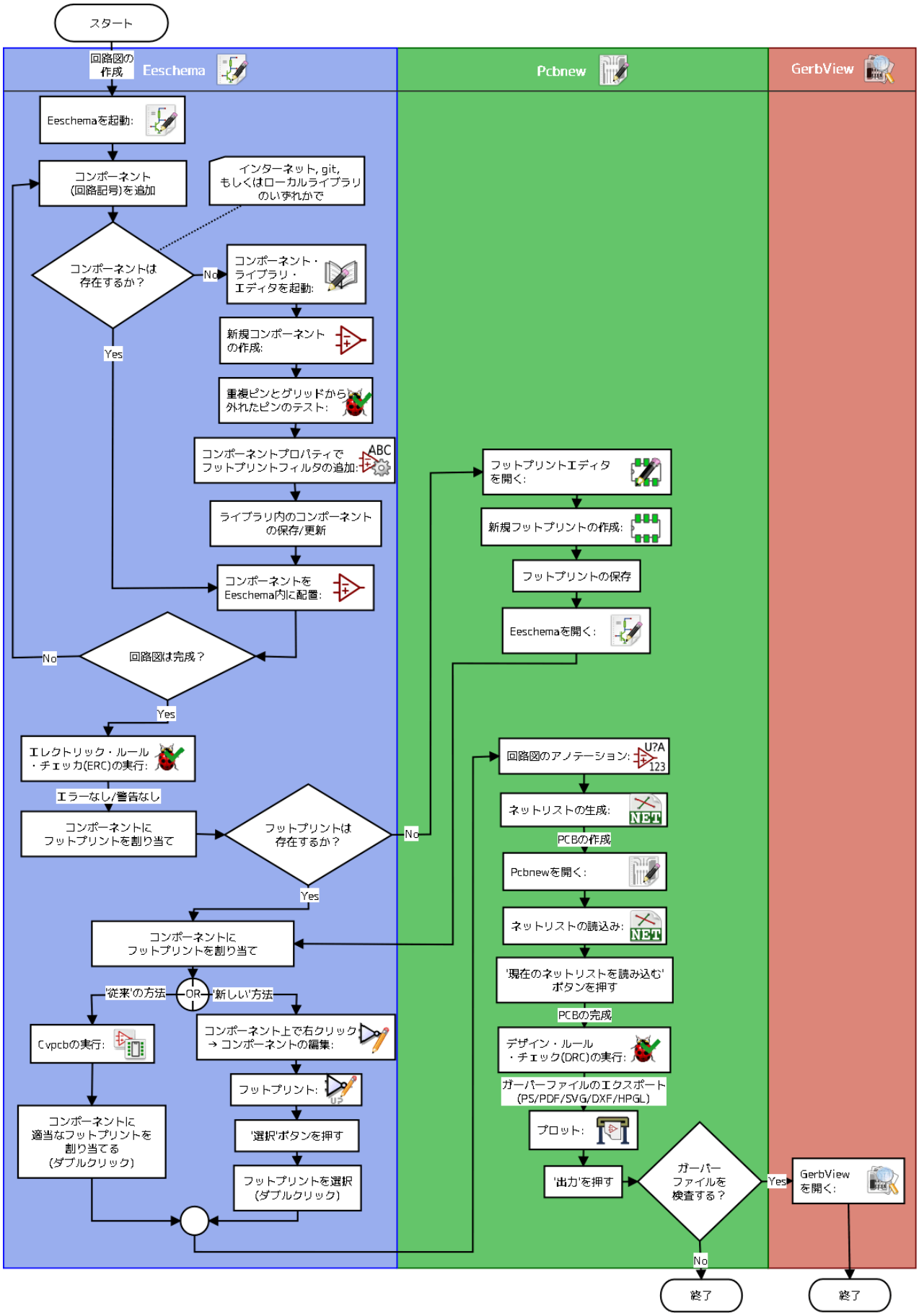
KiCad 作業の流れ

他の PCB ソフトウェア・ツールとの類似性にもかかわらず、KiCad は、回路図のコンポーネントとフットプリントを実際には別々の 2 つの実体とする、面白い作業の流れを特徴としています。このことはよくインターネットのフォーラムでの議論の対象となります。

2.1 KiCad 作業フローチャート

KiCad 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。それは回路図の作成と基板への配置です。これら 2 つのタスクのためにコンポーネントのライブラリとフットプリントのライブラリの両方が必要とされます。KiCad はその両方の多くを持っています。それでは不十分な場合、KiCad は、新しいそれらを作るのに必要なツールも持っています。

下の図は KiCad の作業の流れを示すフローチャートです。この図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
コンポーネント・ライブラリ・エディタを起動:
新規コンポーネントの作成:
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
エレクトリック・ルール・チェッカ(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の実行:
コンポーネント上で右クリック → コンポーネントの編集:
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)
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)
フットプリントエディタを開く:
新規フットプリントの作成:
フットプリントの保存
Eeschemaを開く:
回路図のアノテーション:
ネットリストの生成:
PCBの作成
Pcbnewを開く:
ネットリストの読込み:
'現在のネットリストを読み込む'ボタンを押す
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:
'出力'を押す
ガーバーファイルを検査する？
Yes
No
GerbView を開く:
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、この文書の 「KiCad 回路図コンポーネントの作成」 の章を参照して下さい。また、どのように新しいフットプリントを作成するかについての更なる情報は、この文書の 「フットプリントの作成」 の章を参照して下さい。

以下のサイト:

http://kicad.rohrbacher.net/quicklib.php

では、素早く KiCad のコンポーネントを作成することのできるツールの例を見つけることができるでしょう。quicklib についての更なる情報は、この文書の 「quicklib による回路図コンポーネントの作成」 の章を参照して下さい。

2.2 前方向と後方向のアノテーション

電子回路図を完全に描き終わったら、次のステップは KiCad の作業の流れに従い、それを PCB に移すことです。基板レイアウトの工程を部分的にあるいは完全に終え、追加のコンポーネントやネットが付け足される必要があるかもしれませんし、部品達を並び替えたり、色々あるでしょう。これは 2 つの方法で行うことができます: 後方向アノテーションと前方向アノテーション。

後方向アノテーションは、PCB レイアウトの変更をその対応する回路図に送り戻す処理です。この特別な機能を特に便利と思わない人もいるでしょう。

前方向アノテーションは回路図の変更を対応する PCB レイアウトに送る機能です。回路図に変更を加える度に PCB のレイアウト全体をやり直したくはないでしょうからこれは基本的な機能です。前方向アノテーションは 「KiCad の前方向アノテーション」 の章で議論されています。

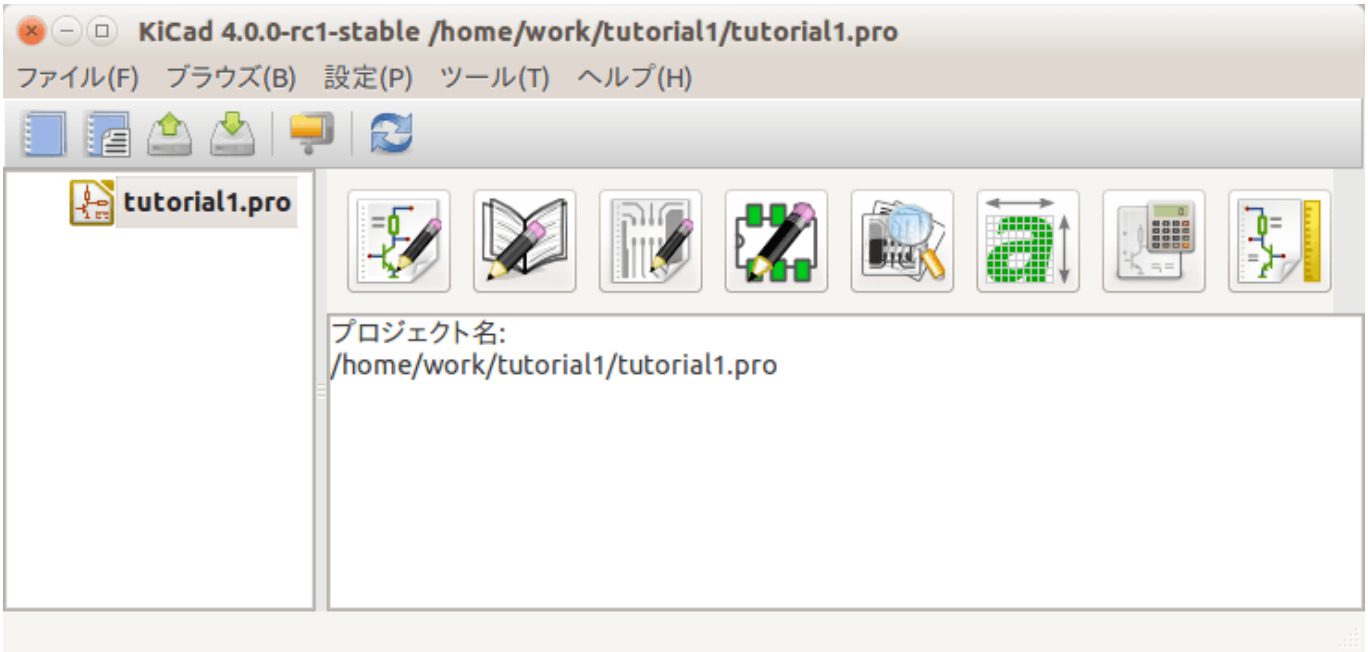
Chapter 3

電子回路図の描画

この章では KiCad を使ってどのように電子回路図を描画するかを学びます。

3.1 Eeschema の使用

1. Windows の場合、kicad.exe を実行して下さい。Linux の場合、端末で *kicad* とタイプして下さい。KiCad プロジェクト・マネージャのメイン画面が現れるでしょう。ここからあなたは次の 8 つの独立したソフトウェア・ツールを使うことができます: *Eeschema*, コンポーネント・ライブラリ・エディタ, *Pcbnew*, フットプリント・エディタ, *GerbView*, *Bitmap2Component*, *PCB Calulator* そして *Pl Editor* です。主要なツール群がどのように使われるのか理解するために 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 → 新規プロジェクト。プロジェクトファイルは *tutorial1* と名付けましょう。プロジェクトファイルの拡張子は自動的に".pro" となります。KiCad はこのプロジェクト専用のディレクトリを作るかを確認してくるので、" はい" をクリックしましょう。あなたのプロジェクトファイルは全てここに保存されます。

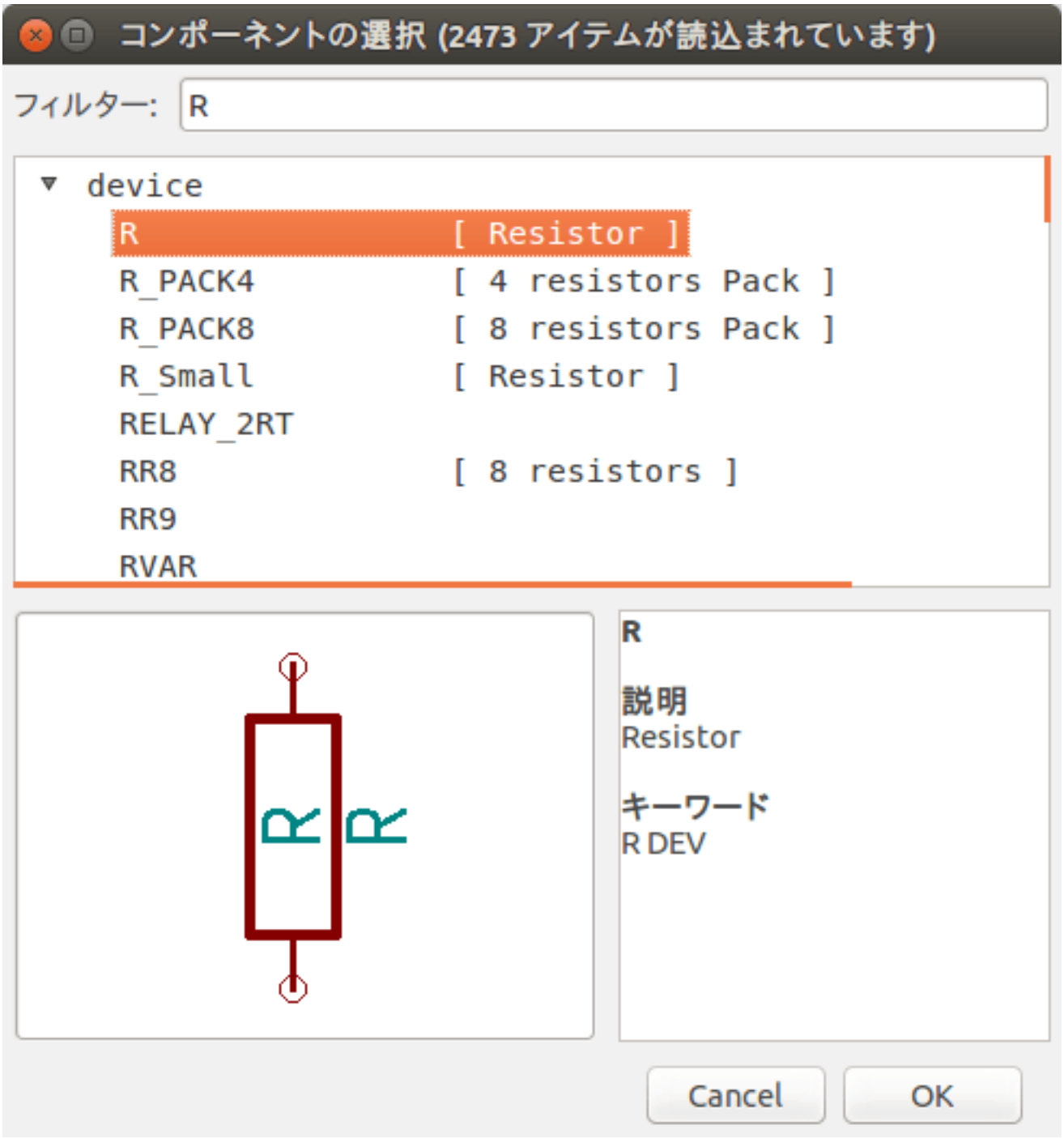
3. 回路図の作成から始めてみましょう。回路図エディタ *Eeschema* を起動します 。左から 1 番目のボタンです。

4. トップツールバーの ページの設定アイコン をクリックします。ページサイズを *A4* に選択し、タイトルを *Tutorial 1* と入力します。必要に応じて更に多くの情報をここで入力できることが分かるでしょう。OK をクリックします。この情報は回路図面の右下隅に取り込まれます。ズームインするにはマウスのホイールを使います。回路図プロジェクトを保存しましょう: ファイル → 回路図プロジェクトの保存

5. では最初のコンポーネントを置いてみます。右ツールバーの コンポーネントの配置アイコン をクリックします。同じ機能はショートカットの"a" キーによる コンポーネントの配置によっても行えます。

   注意
   "?" を押すと利用できるショートカットキーの一覧を見ることができます。

6. 回路図シートの中央あたりをクリックします。コンポーネントの選択ウィンドウが表示されます。抵抗器を配置してみましょう。フィルタに抵抗器 (**R**esitor) の *R* を入力して探します。抵抗器の上に *device* と見出しが出ることに気付いたと思います。この *device* という見出しは、そのコンポーネントが存在するライブラリの名前で、それはとても汎用的で便利なライブラリです。



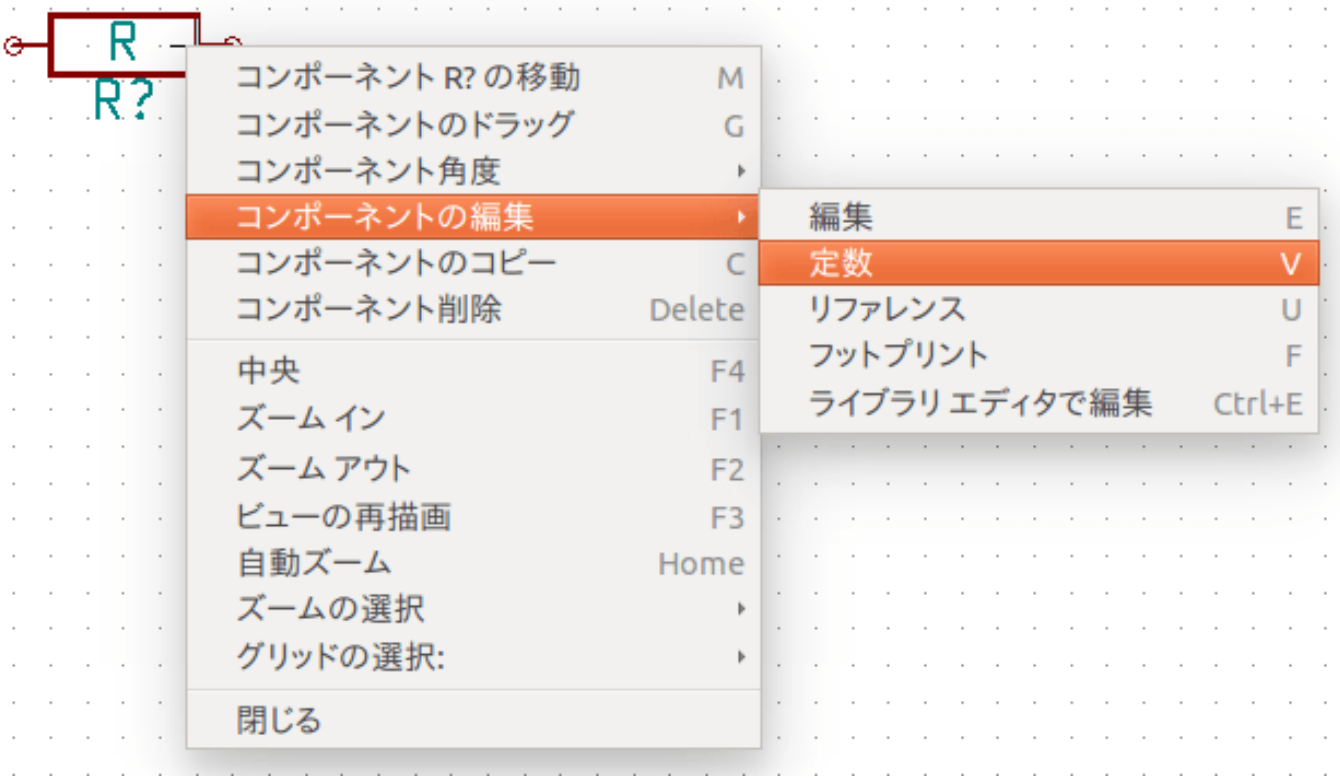
7. その上でダブルクリックします。これにより コンポーネントの選択ウィンドウが閉じられます。回路図シート上の置きたいと思う所をクリックすることでコンポーネントを配置します。

8. ズームインするためにコンポーネント上で拡大鏡をクリックします。あるいは、ズームインとズームアウトをするためにマウスのホイールを使います。水平方向と垂直方向にパンするにはマウスの（中央）ホイールを押します。

9. コンポーネント R の上にマウスカーソルを合わせて、”r” キーを押します。コンポーネントがどのように回転するか注目しましょう。

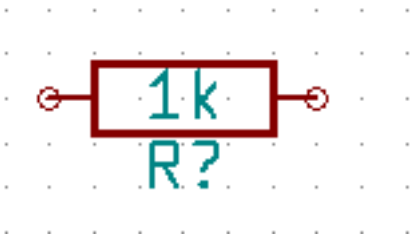
注意
コンポーネントを回転するのにクリックする必要はありません。

10. コンポーネントの中央で右クリックし、コンポーネントの編集 → 定数を選択します。コンポーネントの上にマウスカーソルを合わせて"v" キーを押しても同じことができます。あるいは、"e" キーで更に全般的な編集ウィンドウを開けます。以下の右クリックのコンテキストメニューが、全ての可能なアクションに対する全ての利用可能なショートカットキーを見せてくれることに注目しましょう。



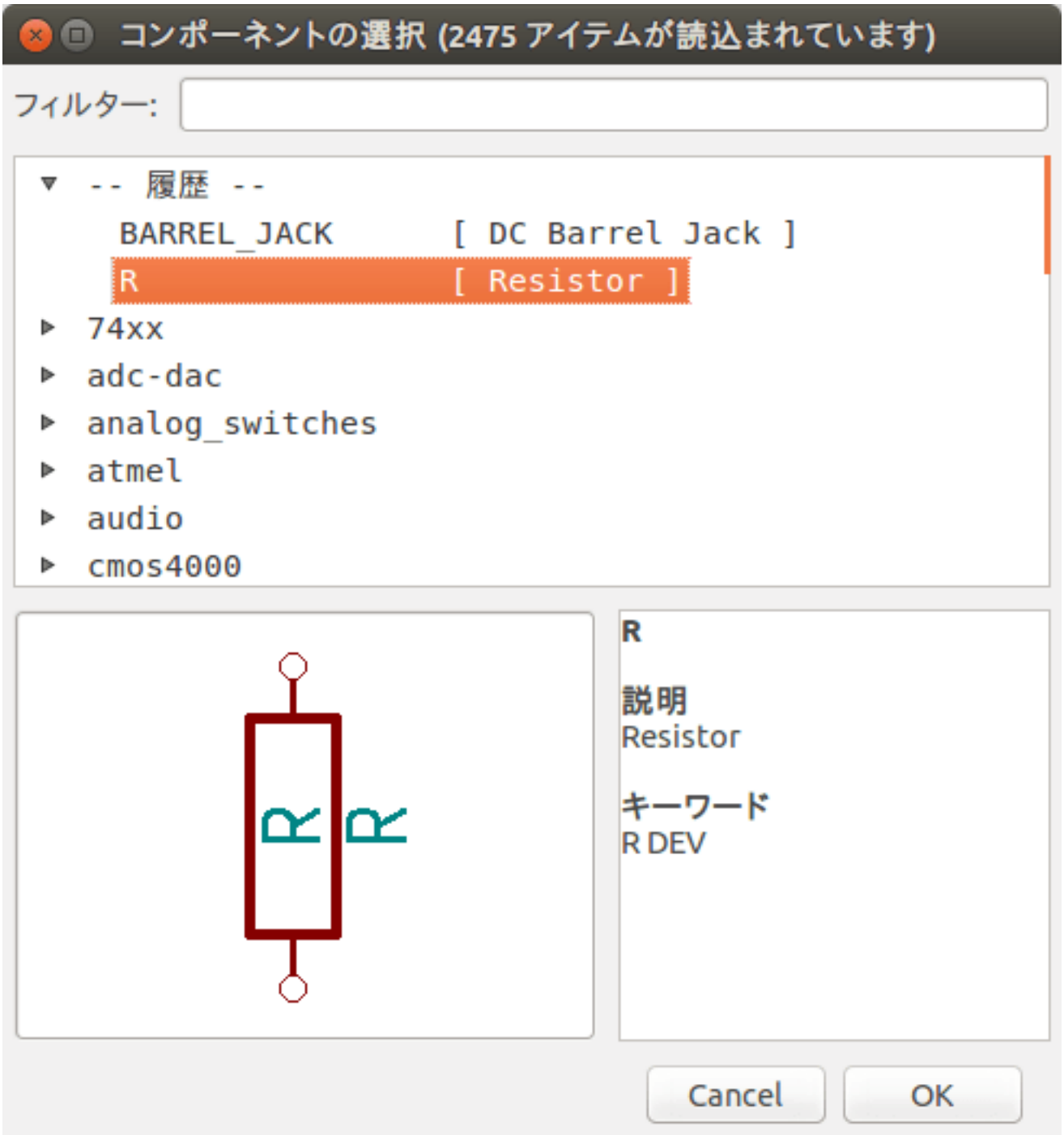
11. フィールド定数の編集のウィンドウが表示されます。現在の定数の R を $1k$ に置き換えます。OK をクリックします。

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。これで抵抗器の中の定数は $1k$ となりました。



12. 別の抵抗器を配置するには、抵抗器を置きたい場所を単にクリックします。コンポーネント選択のウィンドウが再び表示されます。

13. 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. 間違えたコンポーネントを削除したい場合は、コンポーネント上で右クリックして コンポーネント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置いて”Delete” キーを押すこともできます。

注意
デフォルトのショートカットキーは 設定 → ホットキー → ホットキーの編集で編集できます。変更は即座に保存されます。

---

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合わせて"c" キーを押すことにより、複製することができます。新しく複製したコンポーネントを置きたい場所をクリックして下さい。

16. 2 つ目の抵抗器を右クリックします。コンポーネントの移動を選択します。コンポーネントを再配置して左クリックで置きます。同じ機能が、コンポーネントにマウスカーソルを合わせて"g" キーを押すことでも可能です。回転させるためには"r" キーを使います。"x" キーと"y" キーはコンポーネントを反転させます。

注意
右クリック → コンポーネントの移動 ("m" キーと同等) もまた色んなものを移動する便利なオプションですが、コンポーネントのラベルや未接続のコンポーネントにのみ使うのが良いです。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置いて"v" キーを押して編集します。*R* を *100* で置き換えます。"Ctrl+Z" キーによってどんな編集操作も「元に戻す」ができます。

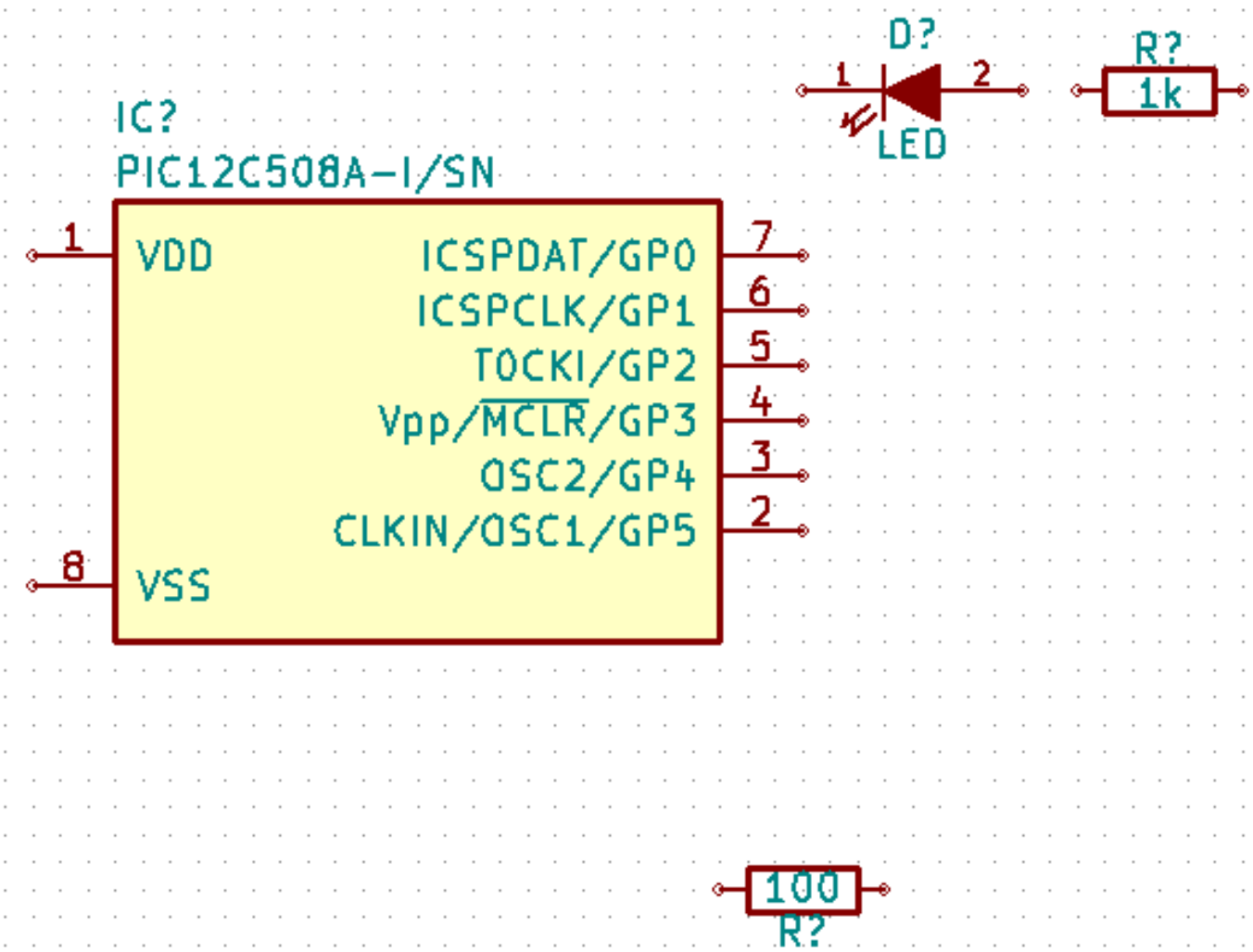
18. グリッドのサイズを変更します。回路図シート上では、全てのコンポーネントが大きなマス目のグリッドに沿っていることに気付いていると思います。グリッドのサイズは 右クリック → グリッドの選択で変更することができます。一般的に、回路図シートでは *50.0* ミルのグリッドをお勧めします。

19. コンポーネント追加のステップを続けますが、今度は *device* ライブラリの代わりに *microchip_pic12mcu* ライブラリを選択して、*R* ではなく *PIC12C508A-I/SN* コンポーネントを選びましょう。コンポーネントを配置する前に、設定 → コンポーネントライブラリから追加ボタンを押して、コンポーネントライブラリファイルに *microchip_pic12mcu* を追加しましょう。

20. マイクロコントローラのコンポーネントにマウスカーソルを置きます。"y" キーか"x" キーを押します。コンポーネントが x 軸、y 軸でどのように反転するかが分かります。キーをもう一度押すと元の方向に戻ります。

21. コンポーネント追加のステップを続けますが、今度は *device* ライブラリを選択して *LED* コンポーネントを選びましょう。

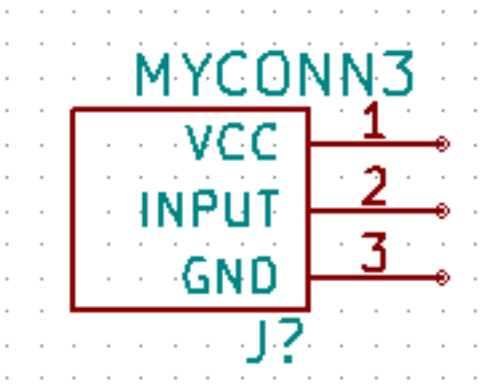
22. 回路図シート上の全てのコンポーネントを以下のように整理します。



23. 私達は 3 ピンのコネクタのために *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 回路図コンポーネントの作成」 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び、この章に戻ってきて下さい。

24. 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" キーを押して *List All* を選択します。*myLib* ライブラリを選択して *MYCONN3* コンポーネントを選びます。

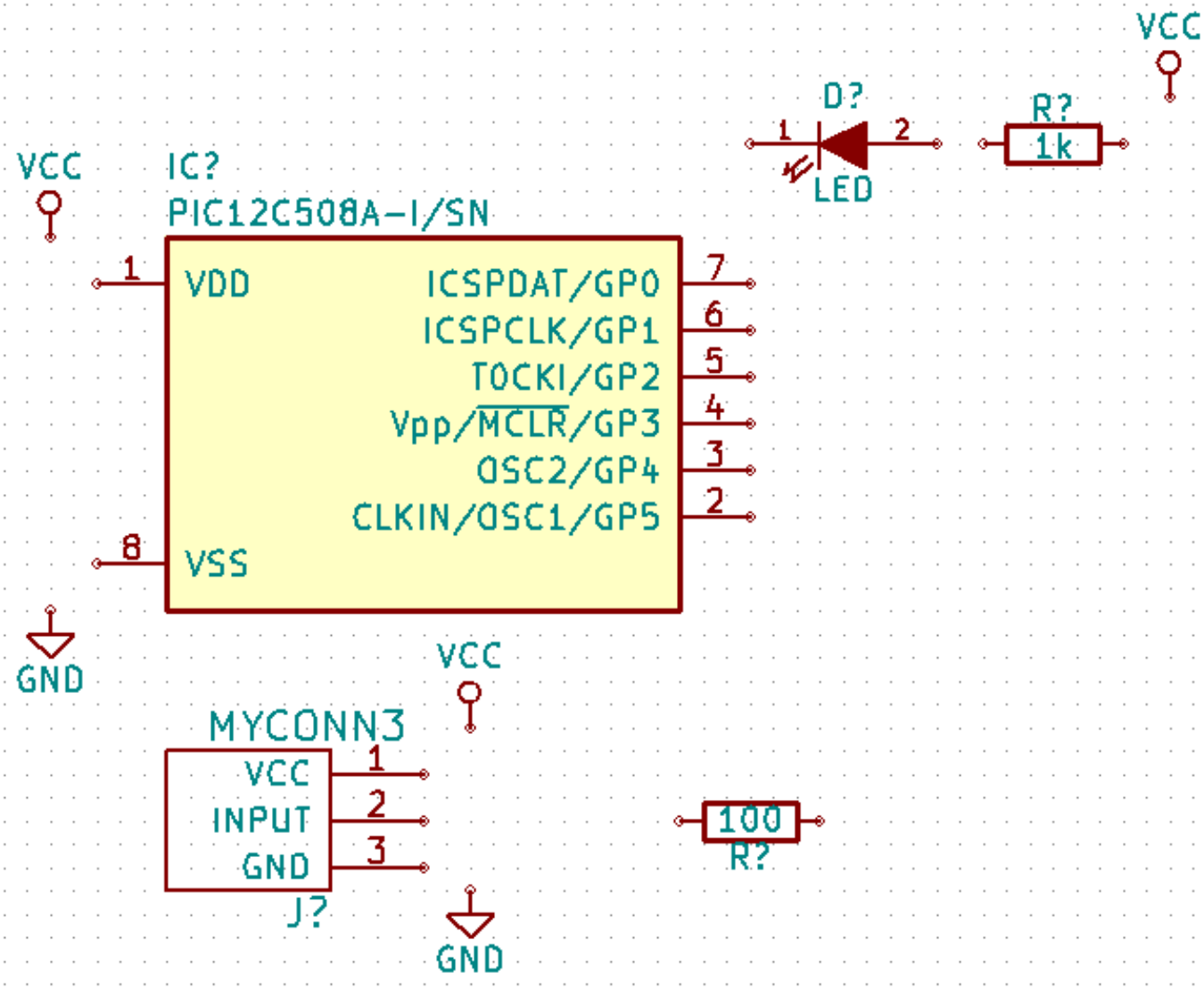
25. コンポーネントの識別子 *J?* が *MYCONN3* ラベルの下に見えます。位置を変えたいなら、*J?* の上で右クリックして リファレンスの移動 ("m" キーと同等) をクリックします。これを行う前か最中にズームインすると便利かもしれません。コンポーネントの *J?* を以下に示すように再配置します。ラベルは何回でも好きなだけ動かすことができます。



26. 電源と GND のシンボルを配置します。右ツールバーにある 電源ポートの配置ボタン をクリックします。あるいは、”p” キーを押します。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* ライブラリから *VCC* 選択します。OK をクリックします。

27. VCC 部品を配置するために、1k の抵抗器のピンの上方をクリックします。マイクロコントローラの *VDD* の上方をクリックします。コンポーネント選択履歴の中から *VCC* を選び、VDD ピンの隣に配置します。追加作業を繰り返して、VCC 部品を *MYCONN3* の VCC ピンの上方に配置します。

28. ピン追加のステップを続けますが、今度は GND 部品を選択します。GND 部品を *MYCONN3* の 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 ピンの右に配置します。回路図はこのようになっているはずです。



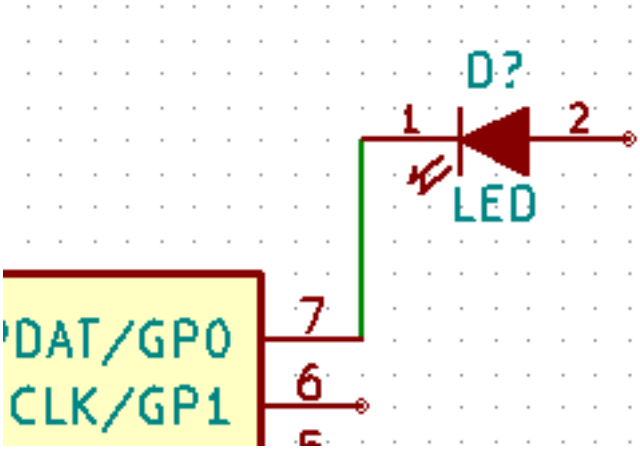
29. 次に、全てのコンポーネントの配線をします。右ツールバーの ワイヤの配置アイコン をクリックします。

注意
バスの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り線が太いです。「KiCad でのバス接続」 の節でバスをどのように使うか説明しています。

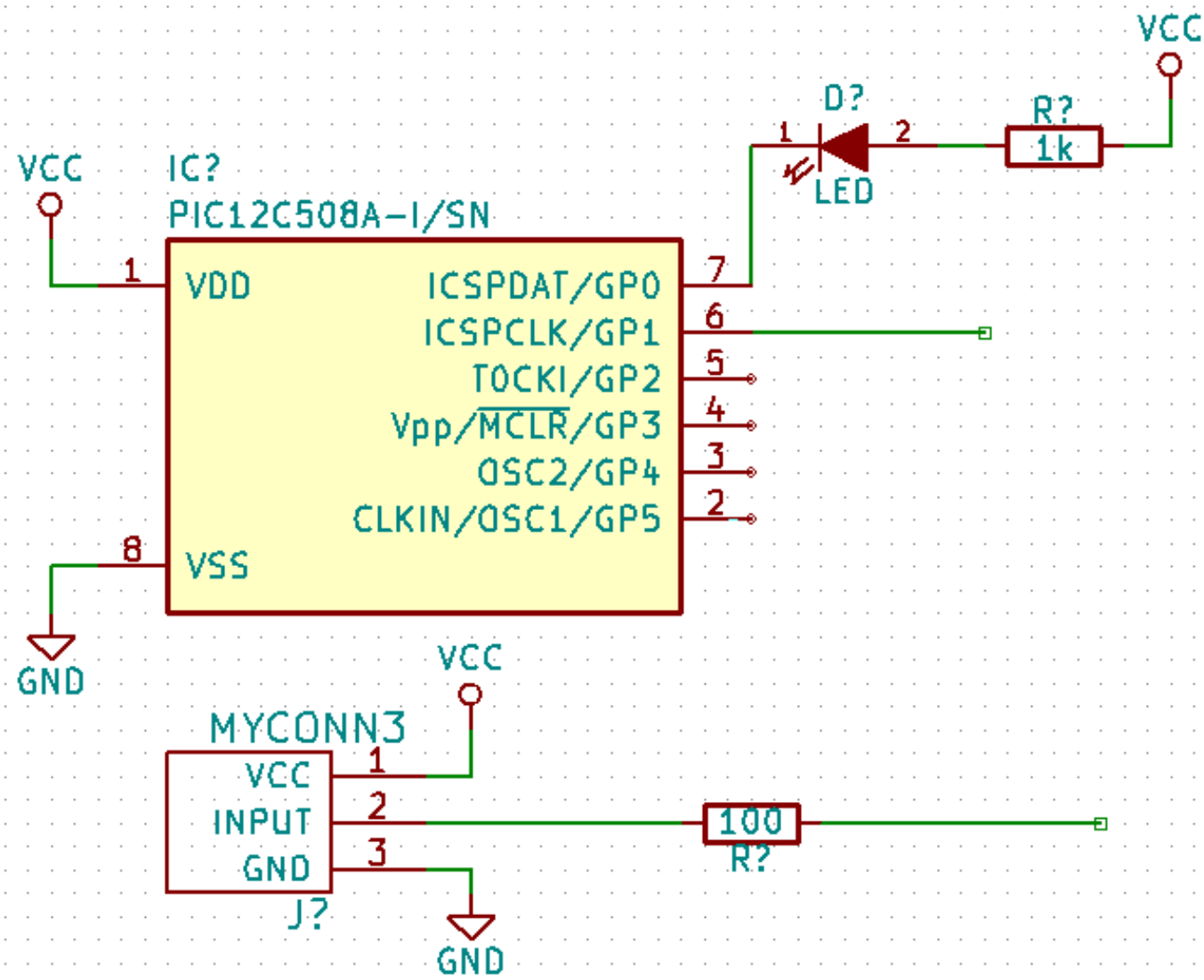
---

30. マイクロコントローラのピン 7 の端の小丸をクリックし、それから LED のピン 2 の小丸をクリックします。接続を配置する時に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" キー(掴む) を使うことが重要で、"m" キー(移動) は使うべきではありません。(掴む) を使うことでワイヤ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 16,25 を見直しましょう。



31. この処理を繰り返して以下の全てのコンポーネントの配線を済ませます。ワイヤを終端するにはダブルクリックをします。VCC と GND のシンボルへの配線は、VCC シンボルなら下部に、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ください。



32. 今度はラベルを使った別の接続方法を考えてみましょう。右ツールバーの ネット名の配置アイコン を

クリックしてネット名の配置ツールを選びます。”l” キーを使うこともできます。

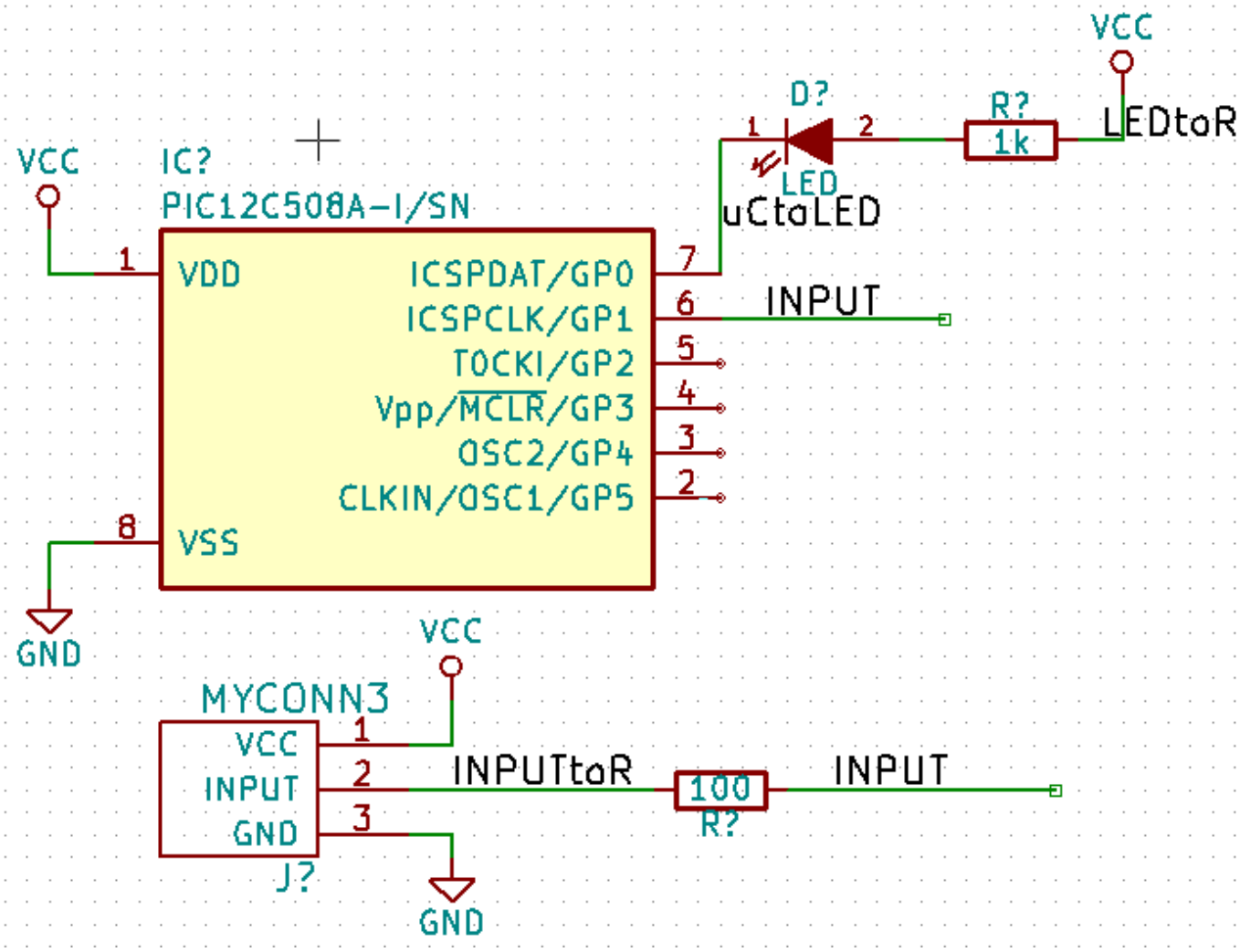
33. マイクロコントローラのピン 6 に接続されているワイヤの中程をクリックします。このラベルに *INPUT* と名前をつけます。

34. 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。2 つの同じ名前を持つラベルは、見えない接続を PIC のピン 6 と 100Ω 抵抗器の間に作ります。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ではなく、単純にピンにラベルを付けることができます。

35. ラベルはまた、単に配線に有用な名前をつける目的で使うこともできます。PIC のピン 7 にラベルを付けて *uCtoLED* と名前をつけたり、抵抗器と LED の間のワイヤに *LEDtoR* と名前をつけたり、*MYCONN3* と抵抗器の間のワイヤに *INPUTtoR* つけたり、です。

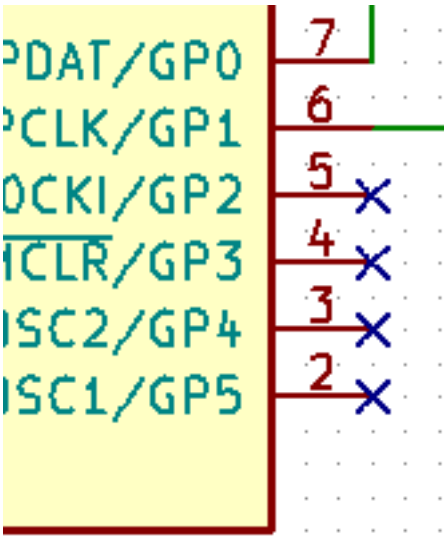
36. VCC と GND の線にラベルをつける必要はありません。そのラベルは、接続されている電源オブジェクトから暗黙的に定義されています。

37. 下図に最終的な結果がどのように見えるかを示します。



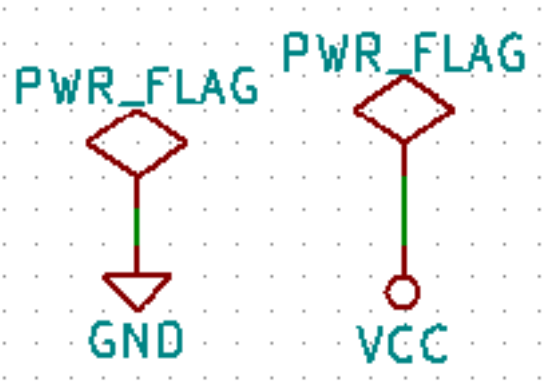
38. それでは接続されていないワイヤに対処しましょう。接続されていないピンやワイヤは KiCad にチェックされた時に警告されます。これらの警告を避けるため、ワイヤが接続されていないのが意図的であることをプログラムに指示することができます。

39. 右ツールバーにある 空き端子フラグを配置というアイコン をクリックします。2, 3, 4 と 5 の線の端にある小丸の上でクリックします。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。



40. 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 非表示ピンを表示のアイコン 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 の名前付けが尊重される場合、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。

41. KiCad に、どこから電源が来るのかを示すために、電源フラグを追加する必要があります。"a" キーを押して *List All* を選択し、*power* ライブラリの上でダブルクリックを行い、*PWR_FLAG* を探します。それを 2 つ配置します。以下に示すように GND と VCC に接続しましょう。



注意
これにより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: 警告 power_in ピンは駆動されていません (Net xx)

---

42. あちこちにコメントを書いておくのが良いこともあります。回路図にコメントを追加するには、右ツールバーの テキストの配置のアイコン を使います。

43. 全てのコンポーネントはユニークな識別子を持つ必要があります。実際には私達のコンポーネントの多くはまだ *R?* や *J?* と名付けられています。識別子の割り当ては、回路図の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。

44. 回路図のアノテーションウィンドウでは、全ての回路図、階層を使用を選択し、アノテーションボタンをクリックします。確認のメッセージに OK をクリックし、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がわかります。各識別子はユニークです。私達の例では、*R1* 、*R2* 、*U1* 、*D1* そして *J1* と名付けられました。

45. ここで回路図のエラーをチェックします。*ERC* の実行アイコン 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* レポートファイルの生成をチェックし、実行ボタンを再度押します。

46. 回路図は完成しました。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す、ネットリストのファイルを作成することができるようになりました。トップツールバーにある ネットリストの生成のアイコン

をクリックします。生成をクリックし、デフォルトのファイル名で 保存をクリックします。

47. ネットリストを生成したら、トップツールバーにある *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。ファイルが存在しませんのエラーが表示されたら OK をクリックします。

48. *CvPcb* により、回路図の全てのコンポーネントに、KiCad ライブラリ内のフットプリントをリンクすることができます。左側のペインは回路図で使われている全てのコンポーネントを表示しています。まず *D1* を選択します。右側のペインは全ての利用可能なフットプリントを表示しています。ここでは *LEDs:LED-5MM* までスクロールダウンして、それをダブルクリックします。

49. 右側のペインに、利用可能なフットプリントの内、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 KiCad が適切なフットプリントだけをあなたに提示しようとするためです。このフィルタを有効/無効するにはこれらのアイコン をクリックします。

50. *IC1* にはフットプリント *Housings_DIP:DIP-8_W7.62mm* を選択します。*J1* にはフットプリント *Connect:Banana_Jack_3Pin* を選択します。*R1* と *R2* にはフットプリント *Discret:R1* footprint を選択します。

51. 選択したフットプリントがどんな風に見えるのかを知りたいなら 2 つの選択肢があります。現在のフットプリントのプレビューには 選択したフットプリントを見るのアイコン をクリックします。あるいは フットプリントのドキュメントを表示のアイコン をクリックすると、全ての利用可能なフットプリントの PDF 文書が得られます。印刷して部品の寸法が適合することを確認できます。

52. 割り当てができました。これで全ての関連付けたフットプリントでもってネットリストのファイルを更新することができます。ファイル → 編集を保存をクリックします。デフォルトの *tutorial1.net* のままで保存をクリックします。あるいはアイコン も使えます。これでネットリストのファイルは全てフットプリントで更新されました。フットプリントのない部品がある場合、独自にフットプリントを作る必要があることに注意して下さい。このことについてはこの文書の後の章で説明されます。

53. *CvPcb* を閉じて *Eeschema* 回路図エディタに戻ります。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。

54. KiCad プロジェクト・マネージャに切り替えます。

55. ネットリストには、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。ネットリストはテキストファイルなので、容易に検査したり編集したり書いたりすることができます。

    注意
    ライブラリファイル (**.lib*) もテキストファイルなので、同様に容易に編集したり記述することができます。

56. 部品表 (BOM) を作成するには *Eeschema* 回路図エディタを起動してトップツールバーにある 部品表の生成のアイコン

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。* プラグインの追加 * ボタンをクリックすることで追加することができます。使いたい *.xsl ファイルを選択します。この場合は *bom2csv.xsl* を選択します。

    注意
    *.xsl ファイルは、KiCad をインストールしたディレクトリの *plugins* にあり、それは次の場所に位置します: /usr/lib/kicad/plugins/
    または次のようにしてファイルを入手します:

    ```
    wget https://raw.githubusercontent.com/KiCad/kicad-source-mirror/master/eeschema/ ↩
        plugins/bom2csv.xsl
    ```

    KiCad は自動的にコマンドを生成します、例えばこのように:

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

    拡張子を与えたい場合、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"
    ```

    ヘルプボタンを押すことで更に情報が得られます。

57. さあ 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。

あなたはこれで次の章の PCB レイアウトのパー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さっと見ておきましょう。

3.2 KiCad でのバス接続

コンポーネント A の連続したピンと、別のコンポーネント B の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 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とバス接続です。どうするのか見てみましょう。

1. 4 ピンのコネクタが 3 つあり、ピン同士を接続したいのだと仮定しましょう。ラベルを使って (”l” キーを押す) コネクタ P4 のピン 4 に *a1* という名前のラベルを付けます。Ins キーを押して、ピン 4 の下のピン 3 に自動的に同様の処理を行います。ラベルは自動的に *a2* とリネームされます。

2. ”Insert” キーを更に 2 回押してください。”Insert” キーは *Repeat last item* 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 CONN_2 と CONN_3 にも繰り返して終えました。このまま進めて PCB を作成すると、3 つのコネクタがお互い接続されていることでしょう。図 2 に説明したことが表されています。美しく見せるために、図 3 に示すよう、アイコン を使った一連の ワイヤ -バスエントリを配置と、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。PCB にはなんの影響もありませんが。

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、ラベルはピンに直接付けることができるのでした。

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、そのラベル付けは少し異なっています (b1, b2, b3, b4)。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。ピンにラベル付けする方法 (も可能ですが) は使わずに、代わりに、バス毎に 1 ラベルで、バスラインにラベル付けする方法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” キーを押して) *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。

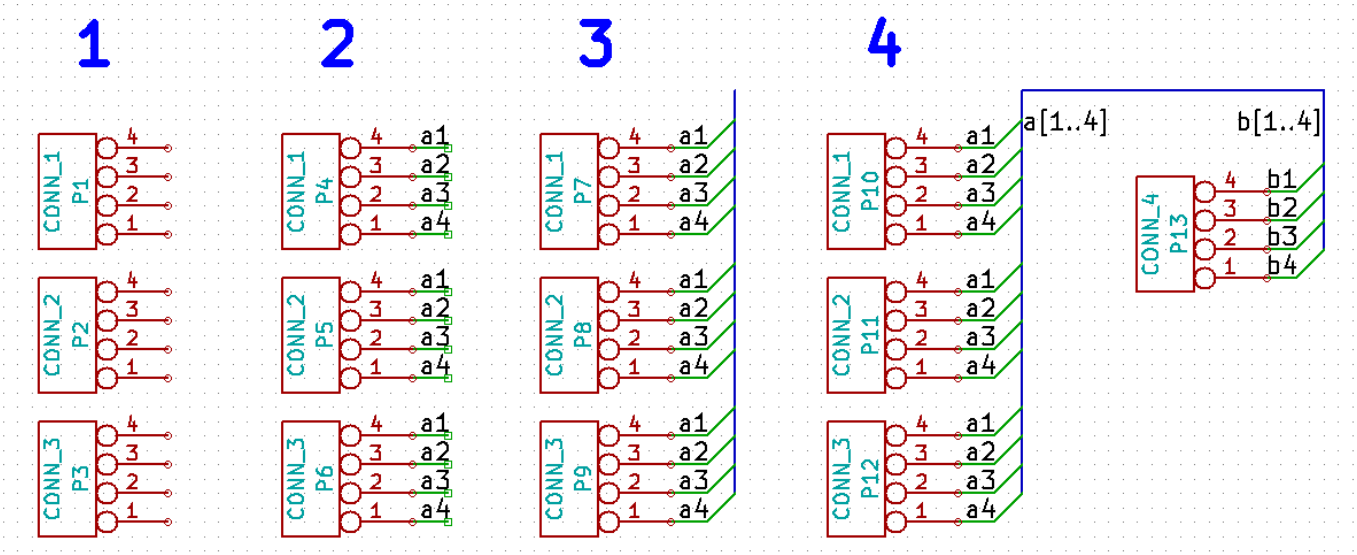
10. 2 つのバスを接続することで、ピン a1 は自動的にピン b1 に接続され、a2 は b2 に接続され、以下同様となります。図 4 は最終結果がどのように見えるか示しています。

注意

”Insert” キーによる *Repeat last item* 機能は、繰り返される周期的な作業にうま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できました。

---

11. ”Insert” キーによる *Repeat last item* 機能は、アイコン を使っての、たくさんの連続した ワイヤ -バス エントリを配置にも適用できます。



Chapter 4

プリント基板のレイアウト

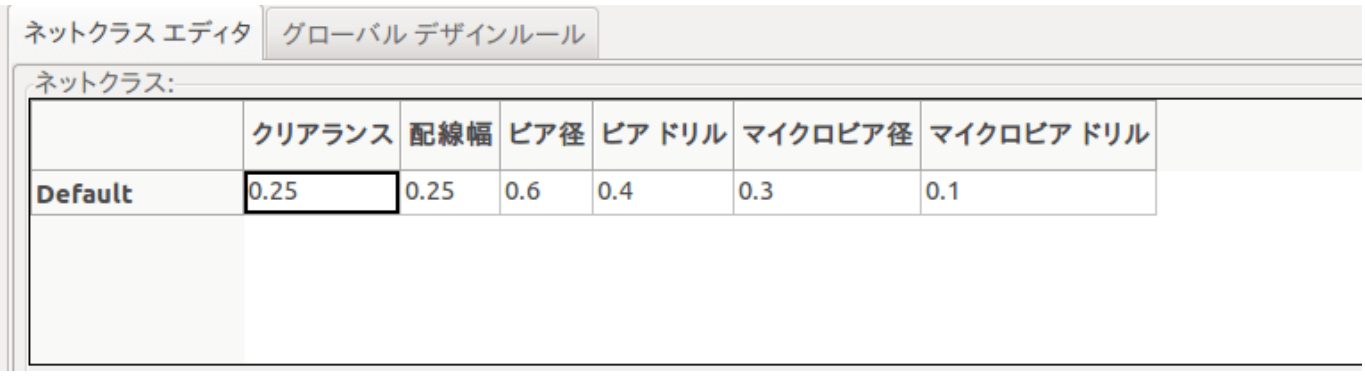
生成したネットリストのファイルを使って PCB のレイアウトを行う時が来ました。これは *Pcbnew* ツールで行います。

4.1 Pcbnew の使用

1. 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* 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？」というエラーメッセージが出たら はい をクリックします。

2. いくつか回路図情報を入力することから始めます。トップツールバーの ページ設定のアイコン をクリックします。ページサイズを *A4* に設定して、タイトルを *Tutorial 1* と設定します。

3. PCB 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*** に、最少配線幅を ***0.25*** に設定します。メニューの * デザインルール → デザインルールをクリックします。もしまだ表示されていなければ ネットクラスエディタのタブをクリックします。以下に示すように、ウィンドウの上部にある 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。



4. グローバルデザインルールタブ上でクリックし、最小の配線幅' を '*0.25* に設定します。変更を確定するため OK ボタンをクリックし、デザインルールエディタのウィンドウを閉じます。

5. さあネットリストのファイルをインポートしましょう。トップツールバーにある ネットリストの読込みアイコン をクリックします。ネットリストファイルを参照するボタンをクリックし、選択ダイアログの中で *tutorial1.net* 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はページ上の左上隅に表示されるはずです。見えない場合にはスクロールアップしましょう。

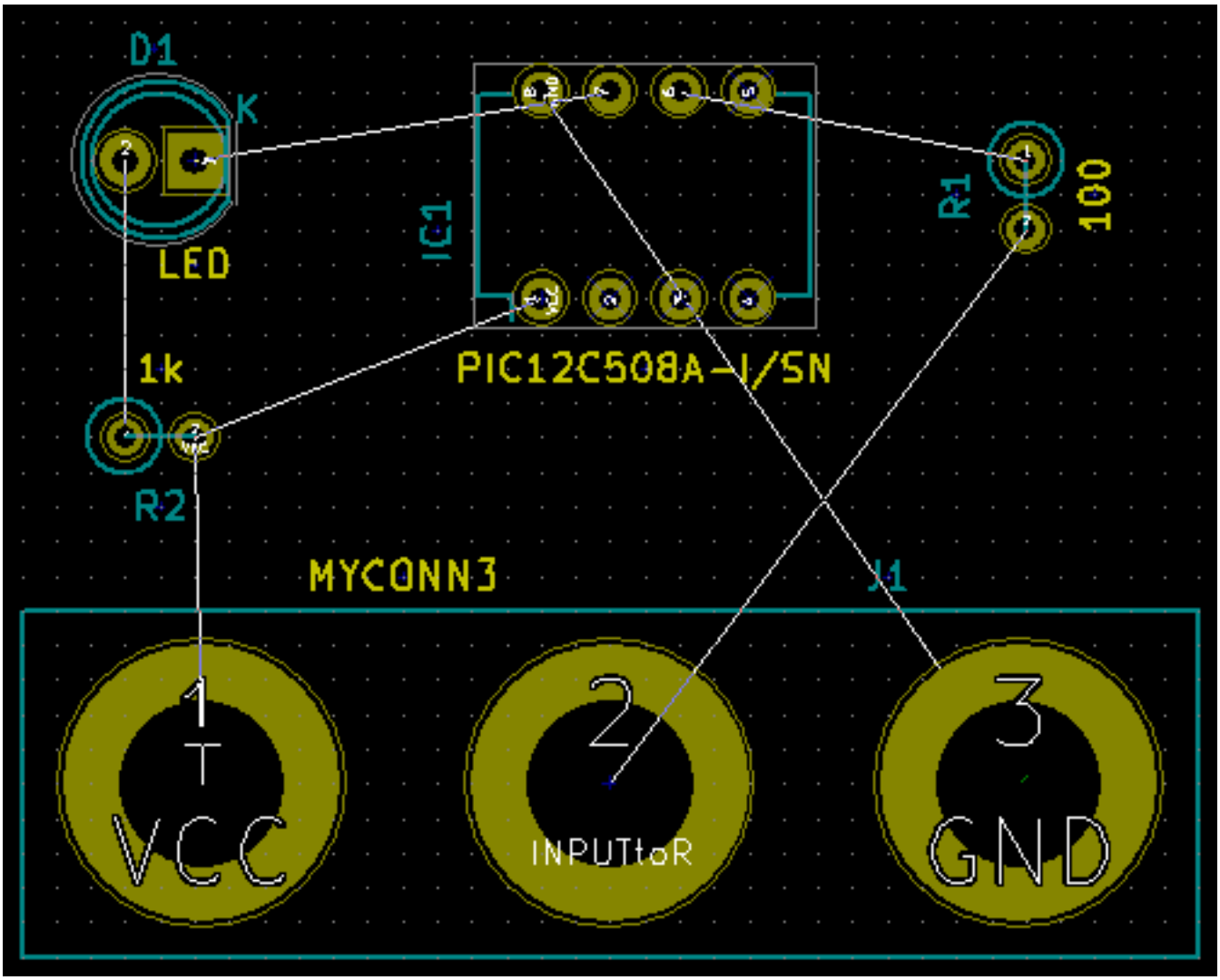
7. マウスで全てのコンポーネントを選択して基板の中央に移動します。必要であれば、コンポーネントの移動中にズームインやズームアウトすることができます。

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で接続されています。ボードのラッツネストを非表示のボタン が押されていることを確認します。これでラッツネストが全てのコンポーネントをつないでいるのが見えます。

   注意
   ツールチップは逆を示します; このボタンを押すことでラッツネストを表示します。

9. コンポーネントにマウスカーソルを置いて"g" キーを押すことで移動することができます。配置したい所でクリックします。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。

   注意
   コンポーネントを掴む ("g" キーで) 代わりに"m" キーを使って移動すると接続が失われることに注意しましょう (同じことが Eeschema でもおきます)。要するに、いつでも"g" キーを使いましょう。



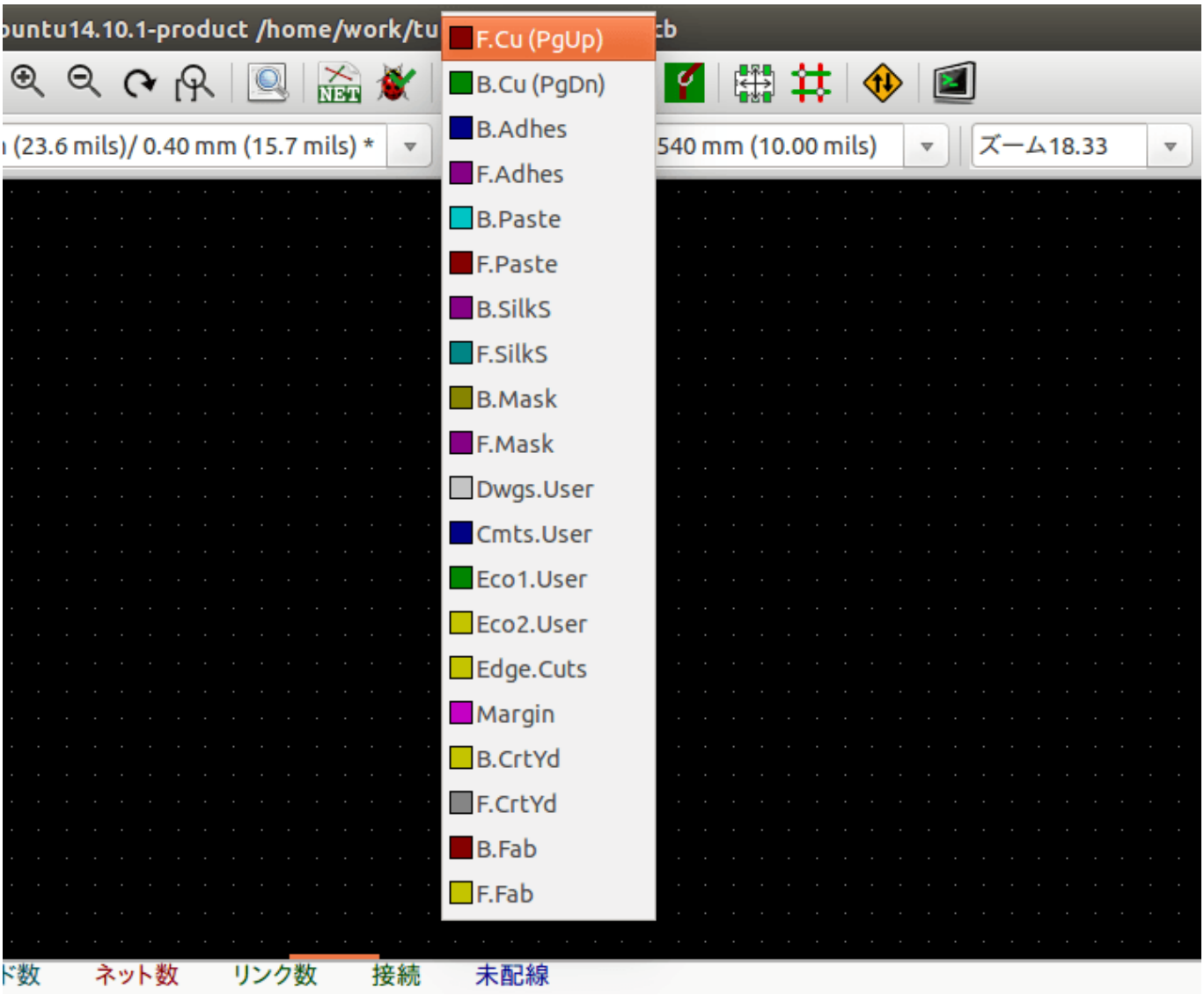
10. ラッツネストが消えたり画面が乱れてきた場合には、右クリックして ビューの再描画をクリックします。どのように 100Ω の抵抗器のピンが PIC コンポーネントのピン 6 に接続されているかに注意しましょう。これがピンを接続するのに使ったラベル付けの結果です。ラベルは実際のワイヤより好まれます。実際のワイヤは回路図を乱雑にするからです。

11. PCB の外形を定義しましょう。トップツールバーのドロップダウンメニューから *Edge.Cuts* を選択します。

    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。緑色端部と PCB の端部の間には小さな隙間を残すことを覚えておきましょう。

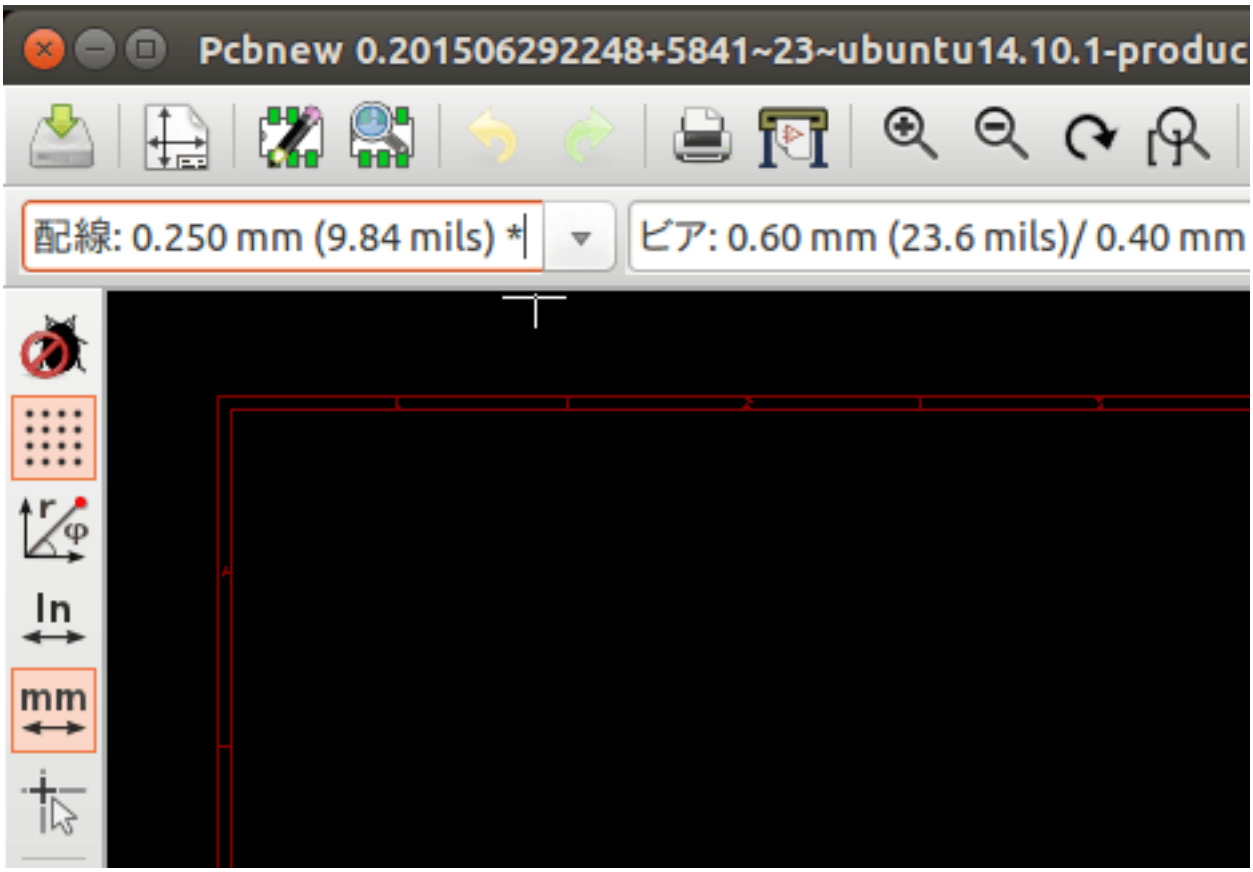
12. 次に GND を除く全てのワイヤを接続してしまいましょう。実際の所、全ての GND の接続は、基板の底部の銅箔面（ *B.Cu* と呼ばれます）に配置されたグランドプレーンを使って一気に行います。

13. 私達はどの銅のレイヤで作業するかを選ばねばなりません。トップツールバーにあるドロップダウンメニューの *F.Cu (”PgUp”* キー*)* を選択します。



14. 例えば 4 層 PCB で行くと決定したならば、デザインルール → レイヤのセットアップを開いて 銅のレイヤ数を 4 に変更します。レイヤの表ではレイヤに名前を付けられ、それらを何に使うのかを決定することができます。プリセットレイヤのグループメニューで、便利なプリセットが選択できることを覚えておきましょう。

15. 右ツールバーにある 配線とビアの追加' のアイコン をクリックします。'*J1* のピン 1 をクリックしパッド *R2* への配線を引きます。配線の終点を設定するため、ダブルクリックします。この配線の幅はデフォルトで 0.250mm です。トップ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。デフォルトではただ一つの配線幅が利用できることに気付いてください。



16. もっと配線幅を追加したい場合: デザインルール → デザインルール → グローバルデザインルールタブを表示し、このウィンドウの右下で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の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) をご覧下さい。

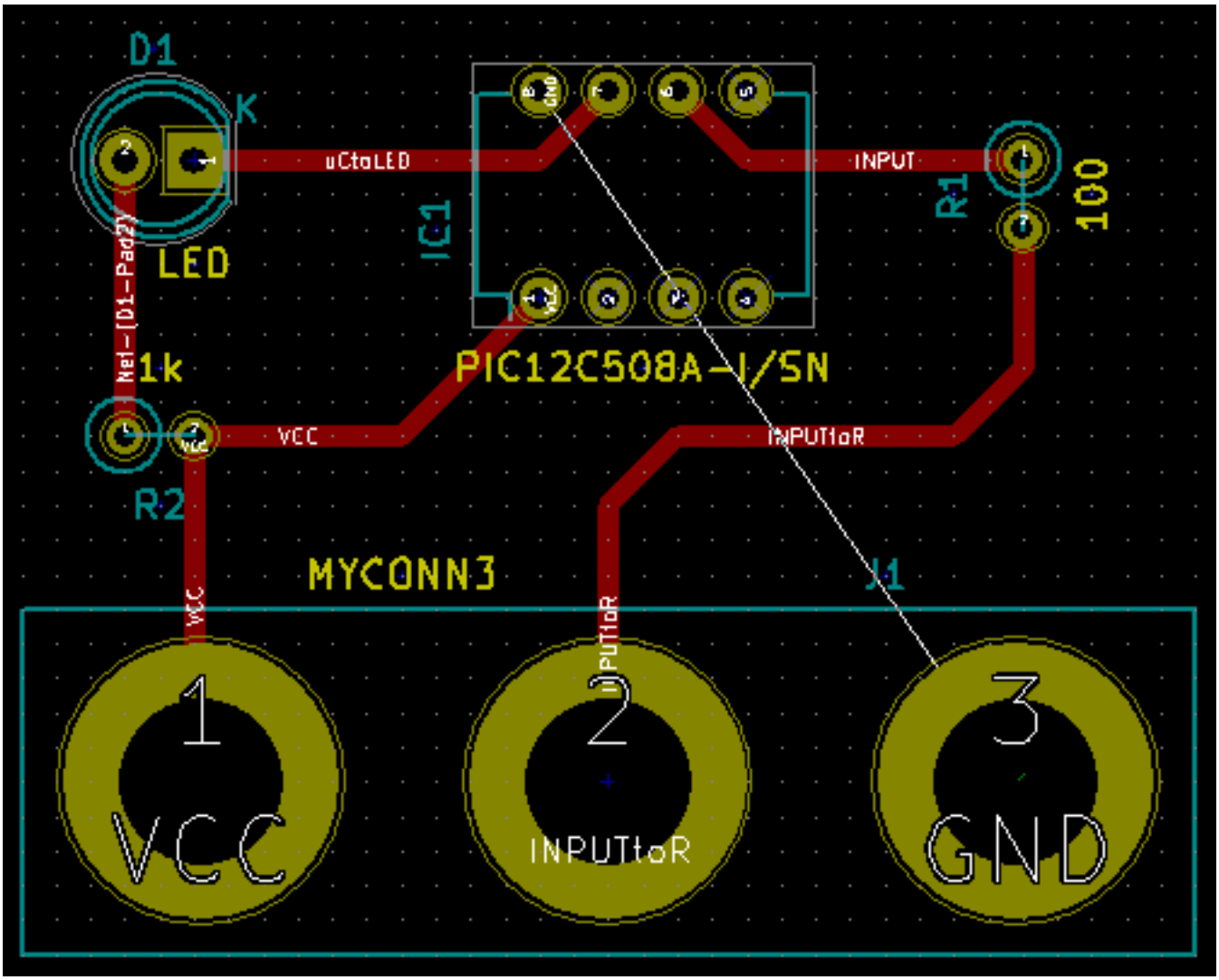
	幅
配線 1	0.01
配線 2	0.02
配線 3	0.05
配線 4	0.08
配線 5	0.10
配線 6	0.15
配線 7	0.20

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。デザインルール → デザインルール → ネットクラスエディタを表示して *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* を選び、右で *power* を選び、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 → グリッドの選択です。コンポーネントを配置して配線で接続する前か後に、最適なグリッドサイズを選択して下さい。

19. J1 のピン 3 を除く全ての配線が接続されるまで、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。



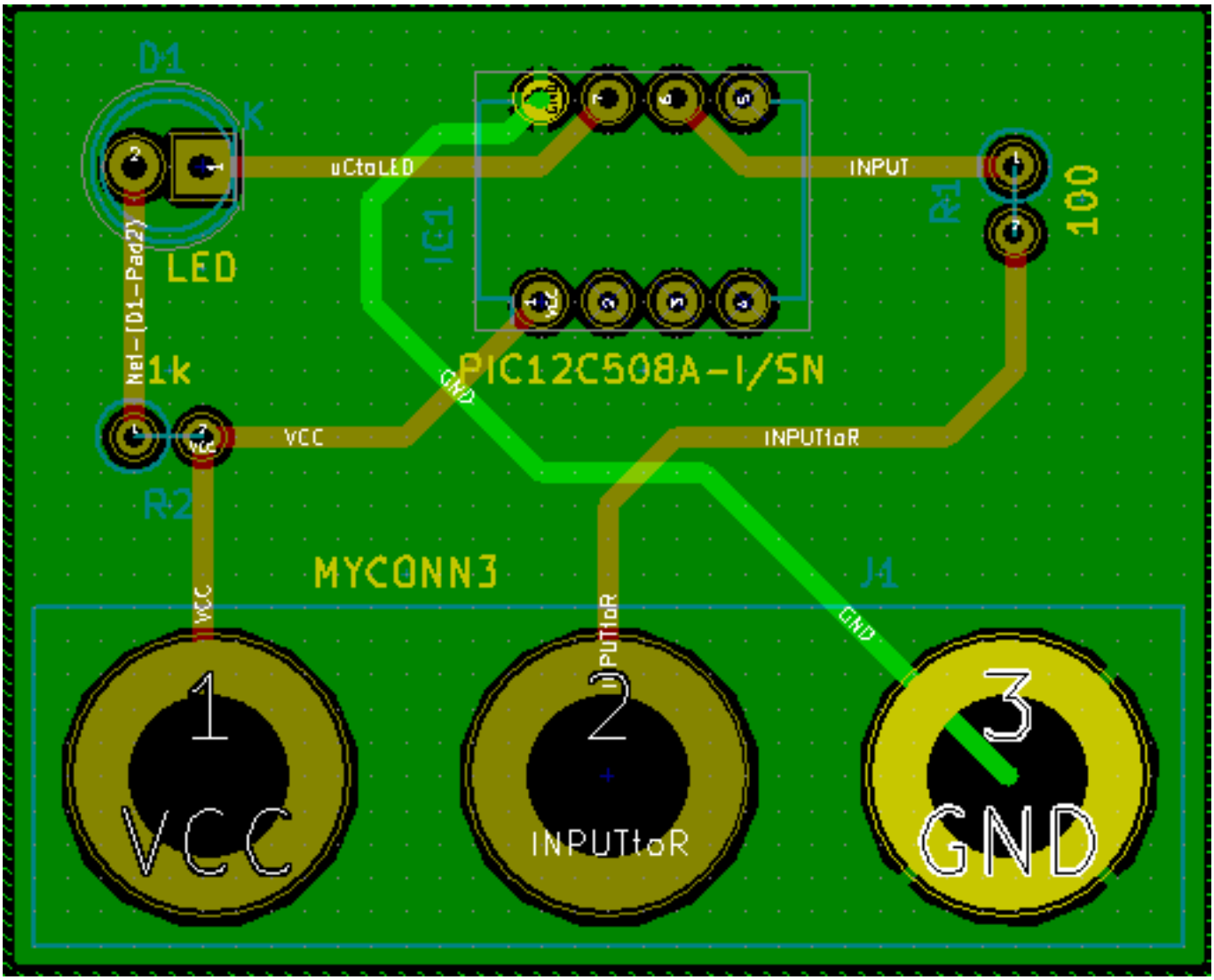
20. PCB の他の銅箔面側での配線を行いましょう。トップツールバーにあるドロップダウンメニューで *B.Cu("PgDn"* キー*)* を選択します。' 配線とビアの追加' のアイコン をクリックします。J1 のピン 3 と U1 のピン 8 の間の配線を行います。実際の所、グラウンドプレーンですませるのでこの配線は必要ではありません。配線の色が変わっていることに注意しましょう。

21. レイヤの変更によってピン **A** からピン **B** に行きましょう。 ビアを配置することで、配線をしている銅プレーンを変更することが可能です。上面の銅プレーンに配線している時に、右クリックして 貫通ビアの配置を選択するか単に"v" キーを押します。これにより配線を終えた所で底面のレイヤに行くことができます。

22. 特定の接続を検査したい場合には右ツールバーにある ネットをハイライトのアイコン をクリックします。J1 のピン 3 をクリックします。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。

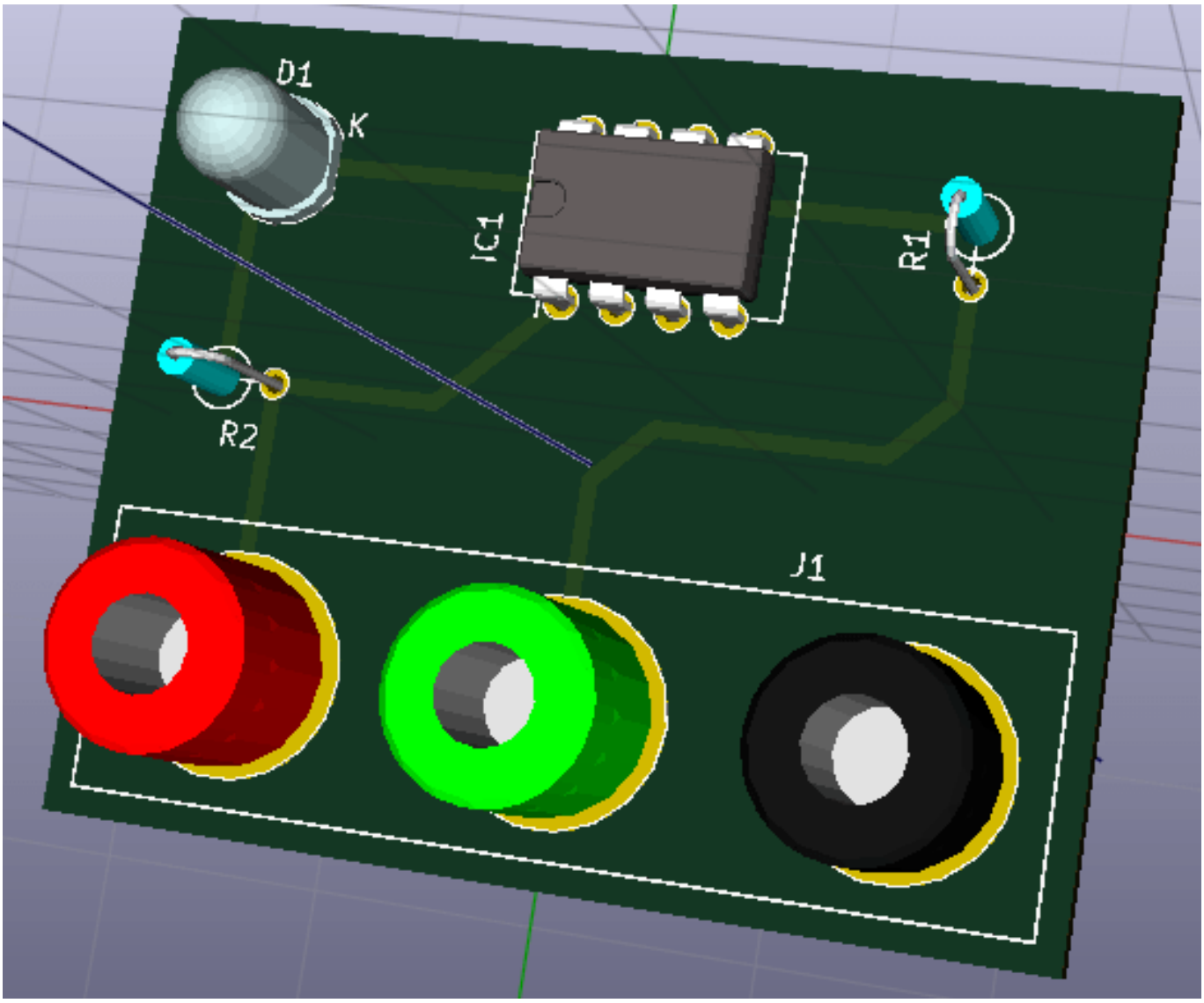
23. 全ての GND ピンに接続されるグランドプレーンを作りましょう。右ツールバーにある 塗りつぶしゾーンの追加’のアイコン をクリックします。基板の周囲に長方形を描くつもりで、想定するコーナーの 1 つをクリックします。表示されたダイアログの中で、’パッド接続を サーマルリリーフに、外枠の角度を 任意角度に設定して OK をクリックします。

24. 順番にそれぞれのコーナーでクリックして基板の外形を描きます。ダブルクリックで長方形を完成させます。描いた領域の内側で右クリックします。全てのゾーンを塗りつぶすをクリックすると、基板は緑色に塗りつぶされ、このように見えるでしょう。



25. トップツールバーにある デザインルールチェックの実行のアイコン 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* の起動をクリックします。エラーはないはずです。未結線情報の一覧をクリックします。未接続はないはずです。OK をクリックして DRC のウィンドウを閉じます。

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。

4.2 ガーバーファイルの生成

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

1. KiCad から *Pcbnew* ツールを開き、アイコン をクリックして基板のファイルを読み込みます。
2. ファイル → プロットをクリックします。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. 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	**KiCad** レイヤ名	旧 **KiCad** レイヤ名	デフォルトの拡張子	**"Protel** の拡張子を使用する" がチェックされている時
裏面導体レイヤ	B.Cu	Copper	.GBR	.GBL
表面導体レイヤ	F.Cu	Component	.GBR	.GTL
表面シルクスクリーン	F.SilkS	SilkS_Cmp	.GBR	.GTO
裏面ソルダマスク	B.Mask	Mask_Cop	.GBR	.GBS
表面ソルダマスク	F.Mask	Mask_Cmp	.GBR	.GTS
基板外形	Edge.Cuts	Edges_Pcb	.GBR	.GM1

4.3 GerbView の使用

1. 全てのガーバーファイルを見るためには KiCad プロジェクト・マネージャ上で *GerbView* アイコンをクリックします。ドロップダウンメニューで *Layer 1* を選択します。ファイル → ガーバーファイルを読み込むをクリックするか、アイコン をクリックします。生成された全てのガーバーファイルを一つずつ読み込みます。それら全てが積み重なって表示されることに注意しましょう。
2. どのレイヤを表示するかの選択/非選択を右側のメニューで行います。メーカーに送る前に注意深く全てのレイヤを検査しましょう。
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。

4.4 Freerouter による自動配線

基板の手配線は手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。最初に重要な配線を手で行ってから、退屈な仕事をさせるために自動配線をセットするということを覚えておきましょう。その仕事は未配線についてのみ応えてくれます。ここで私達が使う自動配線は *freerouting.net* の Freerouter です。

注意

Freerouter はオープンソースの java アプリケーションで、KiCad と使うためには自分でビルドする必要があります。Freerouter のソースコードはこのサイトにあります: https://github.com/nikropht/FreeRouting

---

1. *Pcbnew* から ファイル → エクスポート → **Specctra DSN** ファイルをクリックするか、ツール → **FreeRoute** → 現在のボードを" **Specctra DSN**" ファイルへエクスポートをクリックするかして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。

注意
ツール → **FreeRoute** ウィンドウには素晴らしいヘルプボタンがあり、ファイルビューアで **Freerouter Guidelines** という小さなドキュメントを開いてくれます。Freerouter を効果的に使うためにこのガイドラインに従いましょう。

---

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、いつでも必要なら停止することができます。

3. トップバーにある *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がありません。

6. *Pcbnew* に戻りましょう。ツール → **FreeRoute** をクリックし、スペクトラ・セッションファイル *(.ses)* のバックインポートボタンをクリックして *.ses* ファイルを選択し、新たに配線された基板をインポートします。

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” キーと、右ツールバーにある’ 配線とビアの追加’ のアイコン の配線ツールを使いましょう。

Chapter 5

KiCad の前方向アノテーション

電子回路図、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したら、基板を現実のものとするために、PCB メーカーに全てを送る準備ができました。

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、トップツールバーにある 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。トップツールバーにある *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。トップツールバーにある 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。上述した前方向アノテーションとは別に、後方向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、後方向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 6

KiCad 回路図コンポーネントの作成

あなたが回路図に配置したいコンポーネントが 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、新しい回路図コンポーネントが 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 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。例えば、ここからです:

http://per.launay.free.fr/kicad/kicad_php/composant.php

KiCad では、コンポーネントは *DEF* で始まり *ENDDEF* で終わるテキストの断片です。普通は一つ以上のコンポーネントが拡張子 *.lib* のライブラリファイルに含まれています。コンポーネントをライブラリファイルに追加したいなら、あなたはただカット＆ペーストを使えばよいのです。

6.1 コンポーネント・ライブラリ・エディタの使用

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、デフォルトの参照記号を *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。

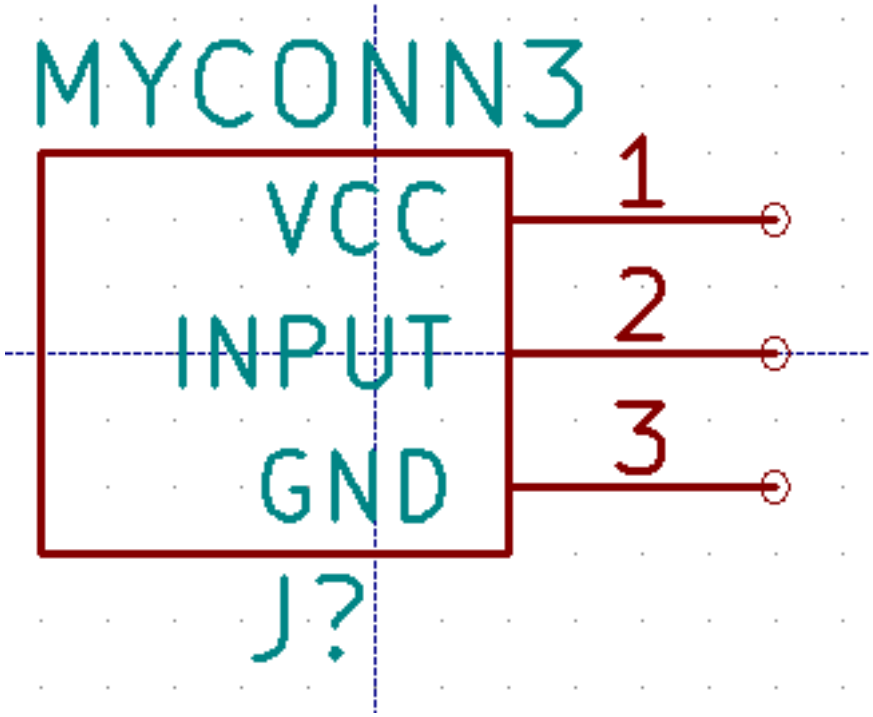
3. 現れた ピンのプロパティ ウィンドウで、ピン名を *VCC* 、ピン番号を *1* 、エレクトリックタイプを パッシブ と設定して OK をクリックします。

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。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. 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。

9. 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。

10. 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。

11. トップツールバーにある 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。トップツールバーにある 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。

12. 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。

13. 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。

6.2 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。

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"キーを押して *MY_RELAY_2RT* にリネームします。

7. トップツールバーにある 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。トップツールバーにある 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。

6.3 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。

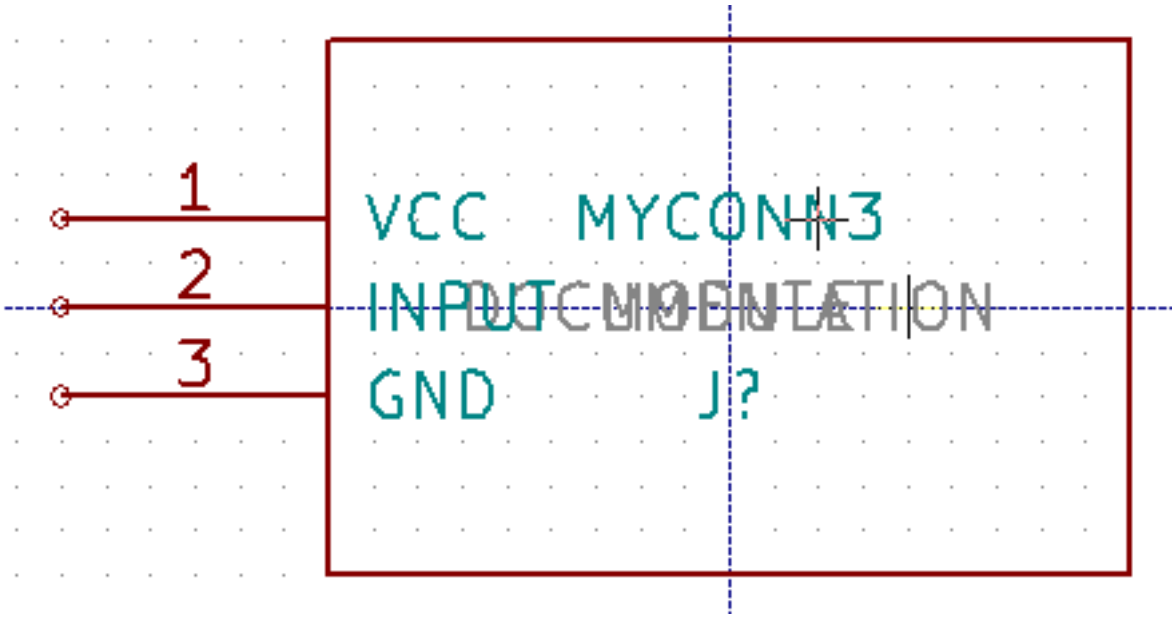
1. *quicklib* のウェブページに行きましょう: http://kicad.rohrbacher.net/quicklib.php

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。

6.4 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
```

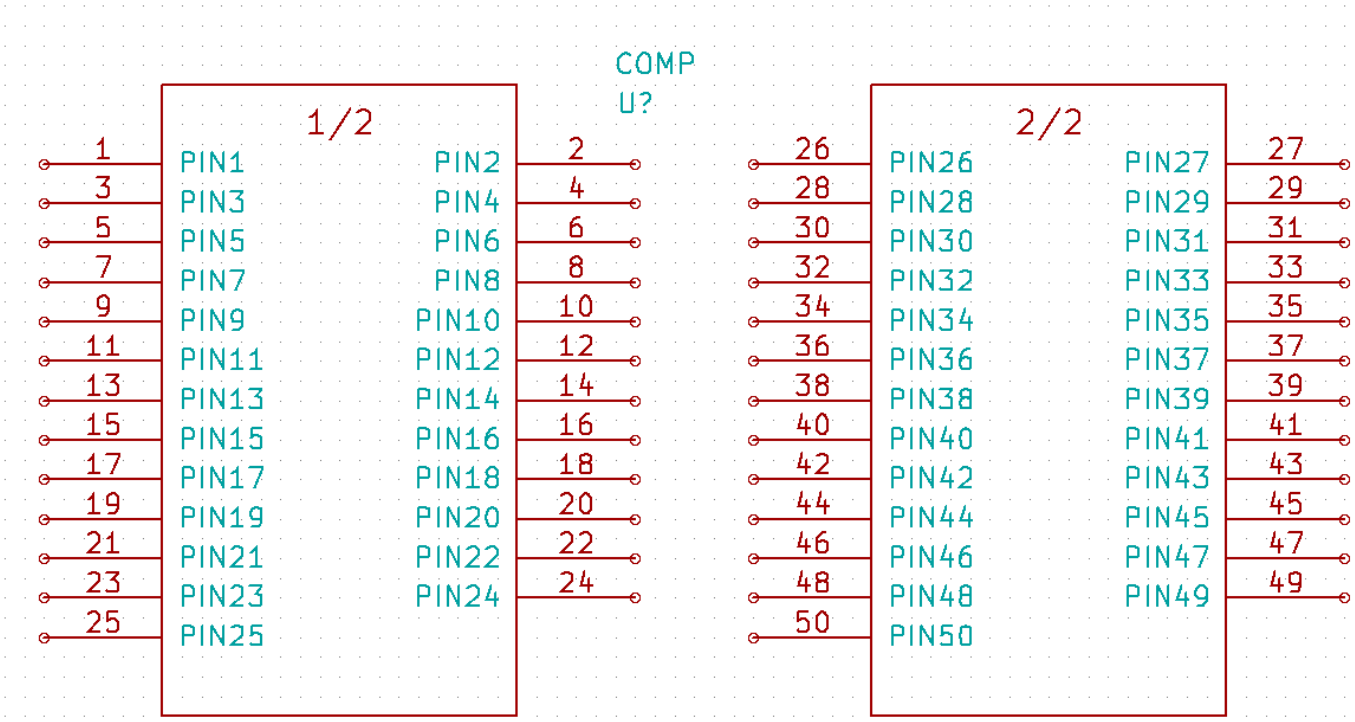
1. 2 つのコンポーネントを一つに統合するために、Eeschema からコンポーネント・ライブラリ・エディタを使って、1 番目のコンポーネントを移動し、2 番目に重ならないようにしてやる必要があるでしょう。以下に最終的な.lib ファイルとその *Eeschema* での表現を示します。

***.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 7

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* ファイルはコンポーネントを、*.kicad_mod* ファイルはフットプリントを含んでいます。*CvPcb* はコンポーネント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:

7.1 フットプリント・エディタの使用

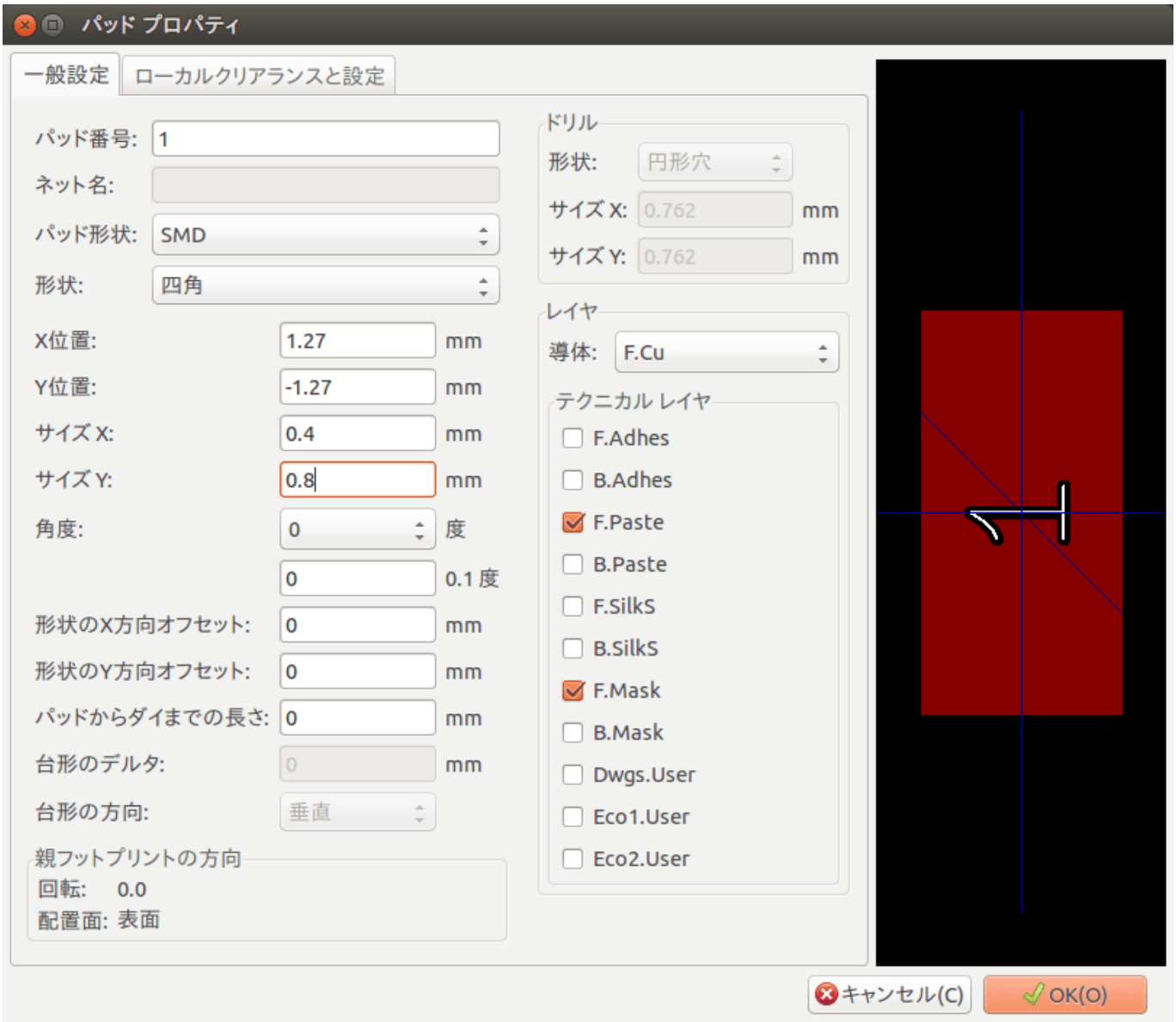
1. KiCad プロジェクト・マネージャから *Pcbnew* ツールを起動します。トップツールバーにある フットプリントエディタを開くのアイコン をクリックします。フットプリント・エディタが開きます。

2. 新しいフットプリント *MYCONN3* を新しいフットプリントライブラリ *myfootprint* に保存します。プロジェクトのフォルダ *tutorial1/* 内に新しく *myfootprint.pretty/* フォルダを作成して下さい。設定 → フットプリントライブラリの管理をクリックして ライブラリの追加' ボタンを押します。テーブルに次のように入力します。別名（ニックネーム）には*"myfootprint"* 、ライブラリのパスには*"${KIPRJMOD}/myfootprint.pretty"* 、プラグインの種類には*"KiCad"*。入力したら *OK* を押して、*PCB* ライブラリ一覧のダイアログを閉じます。トップツールバーにある' アクティブなライブラリを選択のアイコン をクリックします。そして *myfootprint* ライブラリを選択しましょう。

3. トップツールバーにある 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 パッド入力のアイコン 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 パッドの編集をクリックします。代わりに”e” キーのショートカットも使えます。



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. トップツールバーにある 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 8

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 9

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
```

9.1 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。

http://kicad-pcb.org/help/documentation/