

# KiCad入门

KiCad团队

## 目录

- 1. KiCad简介
  - 1.1。下载并安装KiCad
  - 1.2。支持
- 2. KiCad工作流程
  - 2.1。概观
  - 2.2。前向和后向注释
- 3.使用KiCad
  - 3.1。快捷键
- 4.绘制电子原理图
  - 4.1。使用Eeschema
  - 4.2。KiCad的巴士连接
- 5.布局印刷电路板
  - 5.1。使用Pcbnew
  - 5.2。生成Gerber文件
  - 5.3。使用GerbView
  - 5.4。使用FreeRouter自动路由
- 6.在KiCad中转发注释
- 7.在KiCad中制作原理图符号
  - 7.1。使用组件库编辑器
  - 7.2。导出，导入和修改库组件
  - 7.3。使用quicklib制作原理图组件
  - 7.4。制作高引脚数的原理图元件
- 8.制作组件封装
  - 8.1。使用Footprint Editor
- 9.注意KiCad项目文件的可移植性
- 10.有关KiCad文档的更多信息
  - 10.1。网上的KiCad文档

掌握KiCad成功开发复杂电子印刷电路板的基本和简明指南。

## 版权

本文档由以下列出的贡献者版权所有©2010-2018。您可以根据GNU通用公共许可证（<http://www.gnu.org/licenses/gpl.html>），版本3或更高版本或Creative Commons Attribution License（[http](http://creativecommons.org/licenses/by/3.0/)）的条款进行分发和/或修改：[//creativecommons.org/licenses/by/3.0/](http://creativecommons.org/licenses/by/3.0/)），版本3.0或更高版本。

本指南中的所有商标均归其合法所有者所有。

## 贡献者

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

## 反馈

请将任何错误报告，建议或新版本指向此处：

- 关于KiCad文档：<https://github.com/KiCad/kicad-doc/issues>
- 关于KiCad软件：<https://bugs.launchpad.net/kicad>
- 关于KiCad软件国际化（i18n）：<https://github.com/KiCad/kicad-i18n/issues>

## 发布日期

2015年5月16日。

## 1. KiCad简介

KiCad是一个开源软件工具，用于创建电子原理图和PCB图形。在其独特的表面下，KiCad融合了以下独立软件工具的优雅集合：

程序名称	描述	文件扩展名
KiCad的	项目经理	*的.pro

程序名称	描述	文件扩展名
Eeschema	原理图和组件编辑器	* .sch, * .lib, * .net
Pcbnew	电路板和足迹编辑器	* .kicad_pcb, * .kicad_mod
GerbView	Gerber和钻取文件查看器	* .g \ *, * .drl等
Bitmap2Component	将位图图像转换为组件或封装	* .lib, * .kicad_mod, * .kicad_wks
PCB计算器	组件计算器，轨道宽度，电气间距，颜色代码等.....	没有
Pl编辑器	页面布局编辑器	* .kicad_wks

**注意** 文件扩展名列表不完整，仅包含KiCad支持的文件子集。它对于基本了解每个KiCad应用程序使用哪些文件非常有用。

KiCad可以被认为是足够成熟，可用于复杂电子板的成功开发和维护。

KiCad没有任何电路板尺寸限制，它可以轻松处理多达32个铜层，最多14个技术层和最多4个辅助层。KiCad可以创建构建印刷电路板所需的所有文件，用于照片绘图仪的Gerber文件，钻孔文件，组件位置文件等等。

作为开源（GPL许可），KiCad代表了面向创建具有开源风格的电子硬件的项目的理想工具。

在互联网上，KiCad的主页是：

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

### 1.1。下载并安装KiCad

KiCad运行在GNU / Linux，Apple macOS和Windows上。您可以在以下位置找到最新的说明和下载链接：

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

**重要** KiCad稳定版本根据[KiCad稳定版本政策](#)定期发布。新功能不断添加到开发分支中。如果您想利用这些新功能并通过测试帮助，请下载适用于您平台的最新每晚构建包。每夜构建可能会引入文件损坏，生成坏Gerbers等错误，但KiCad开发团队的目标是在新功能开发期间尽可能保持开发分支的可用性。

#### 1.1.1。在GNU / Linux下

KiCad的稳定版本可以在大多数发行版的软件包管理器中找到，如kicad和kicad-doc。如果您的发行版未提供最新的稳定版本，请按照不稳定版本的说明进行操作，并选择并安装最新的稳定版本。

在Ubuntu下，安装不稳定的夜间KiCad版本的最简单方法是通过PPA和Aptitude。在终端中键入以下内容：

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

在Fedora下安装不稳定的夜间构建的最简单方法是通过copr。要通过copr安装KiCad，请在copr中键入以下内容：

```
sudo dnf copr启用mangelajo / kicad
sudo dnf安装kicad
```

或者，您可以下载并安装KiCad的预编译版本，或直接下载源代码，编译和安装KiCad。

#### 1.1.2。在Apple macOS下

适用于macOS的KiCad的稳定版本可在以下[网址](#)找到：<http://downloads.kicad-pcb.org/osx/stable/>

不稳定的夜间开发版本可在以下[网址](#)找到：<http://downloads.kicad-pcb.org/osx/>

#### 1.1.3。在Windows下

适用于Windows的KiCad的稳定版本可在以下[网址](#)找到：<http://downloads.kicad-pcb.org/windows/stable/>

对于Windows，您可以在以下[网址](#)找到夜间开发版本：<http://downloads.kicad-pcb.org/windows/>

## 1.2。支持

如果您有想法，评论或问题，或者您需要帮助：

- [访问论坛](#)
- 加入Freenode上的 [#kicad IRC频道](#)
- [观看教程](#)

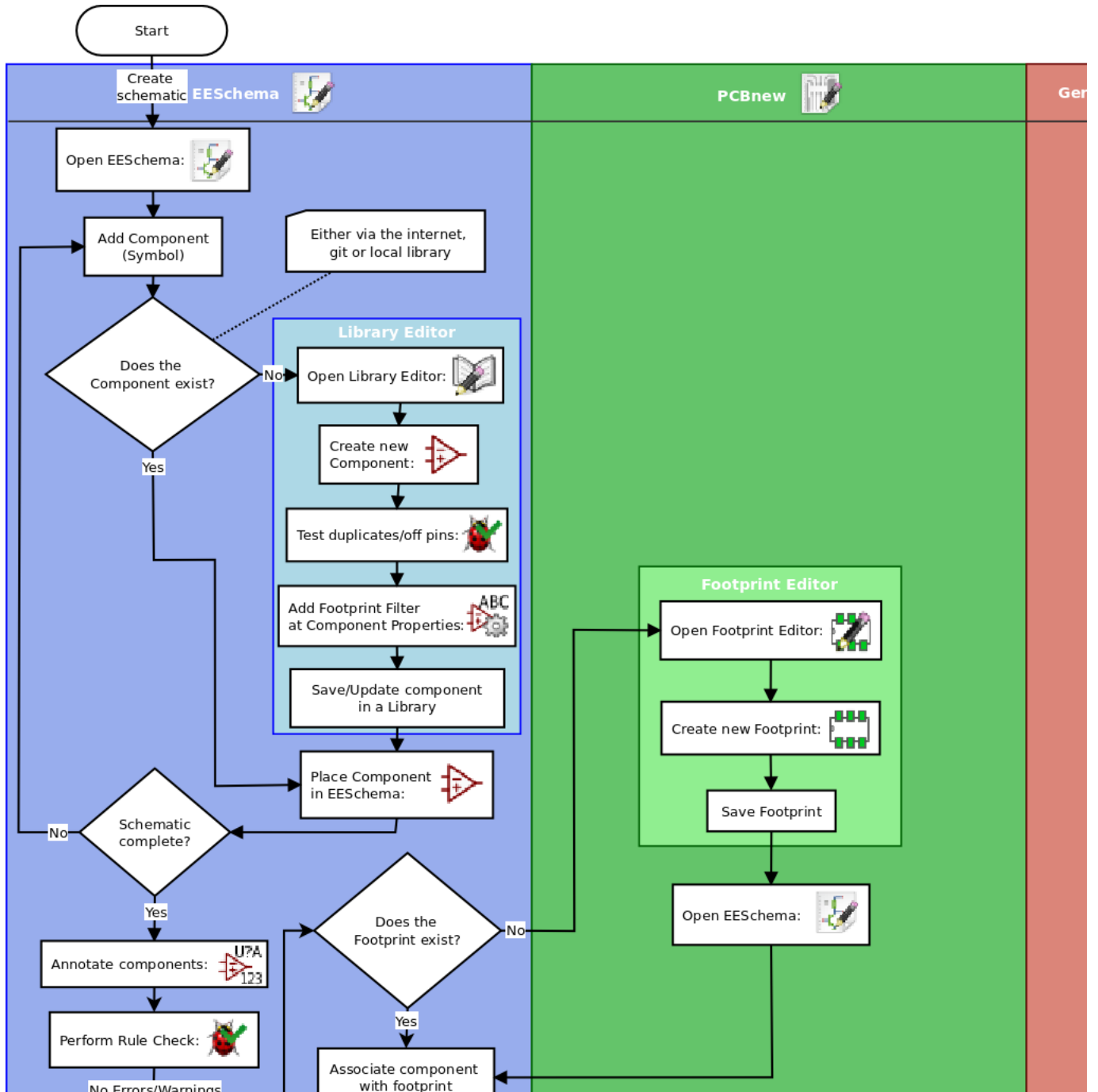
## 2. KiCad工作流程

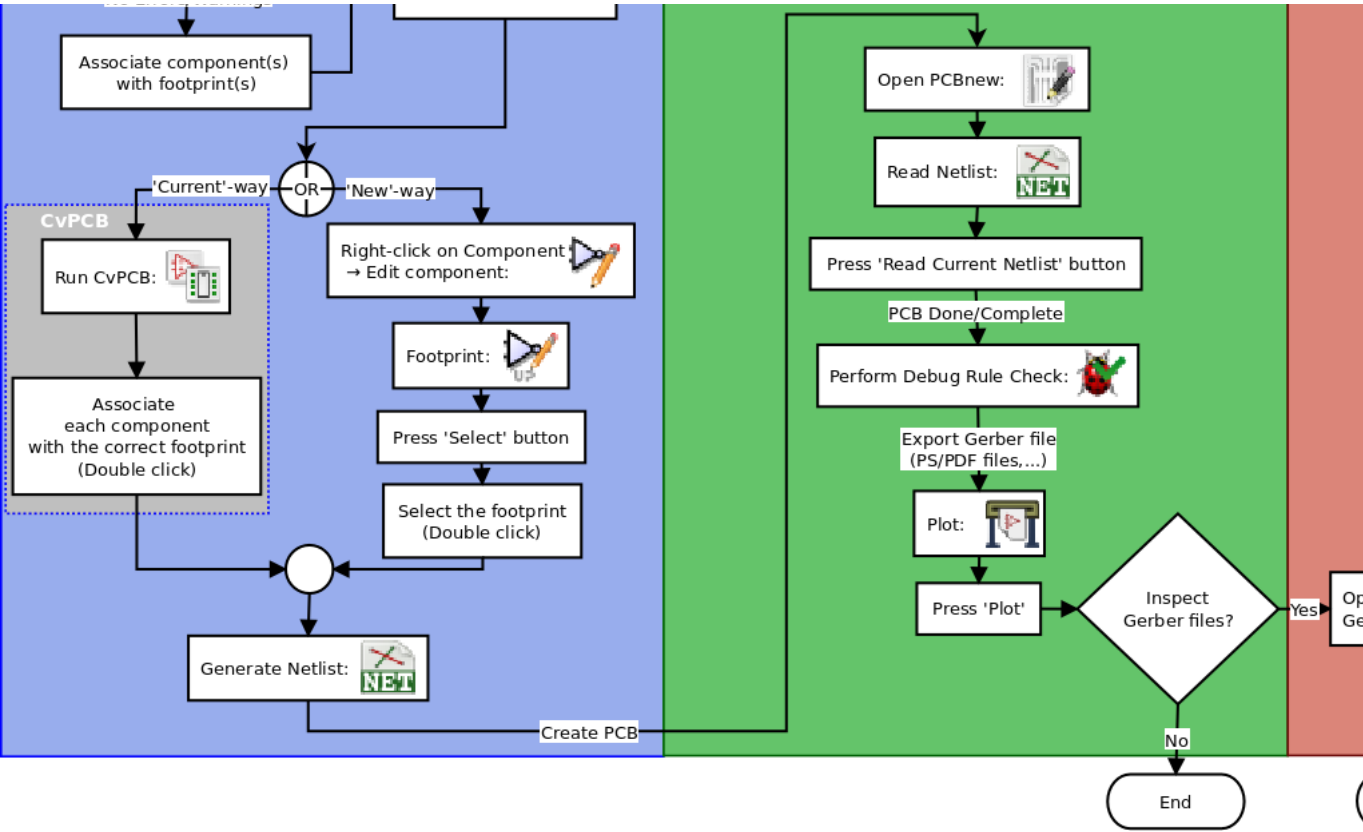
尽管与其他PCB设计软件具有相似性，但KiCad的特点是独特的工作流程，其中原理图元件和封装是分开的。仅在创建原理图之后才会为组件分配足迹。

### 2.1。概观

KiCad工作流程包括两个主要任务：绘制原理图并布置电路板。这两个任务都需要原理图元件库和PCB封装库。KiCad包含许多组件和脚印，并且还具有创建新组件和工具的工具。

在下图中，您将看到表示KiCad工作流程的流程图。该流程图说明了您需要采取的步骤以及顺序。适用时，会添加图标以方便使用。





有关创建组件的更多信息，请阅读 [制作原理图组件](#)。有关如何创建新封装的信息，请参阅 [制作组件封装](#)。  
[Quicklib](#)是一个工具，允许您使用基于Web的界面快速创建KiCad库组件。有关Quicklib的更多信息，请参阅 [使用Quicklib制作原理图组件](#)。

### 2.2。前向和后向注释

完成电子原理图后，下一步就是将其传输到PCB。通常，可能需要添加其他组件，将部件更改为不同的尺寸，进行网络重命名等。这可以通过两种方式完成：前向注释或后向注释。  
正向注释是将原理图信息发送到相应PCB布局的过程。这是一个基本功能，因为您必须至少执行一次才能将原理图导入PCB。然后，正向注释允许向PCB发送增量的原理图更改。有关正向注释的详细信息，请参阅“[前向注释](#)”一节。  
向后注释是将PCB布局更改发回其相应原理图的过程。向后注释的两个常见原因是门交换和引脚交换。在这些情况下，有功能相同的门或引脚，但只有在布局期间才有选择精确门或引脚的强有力的情况。一旦在PCB中做出选择，则将此更改推回原理图。

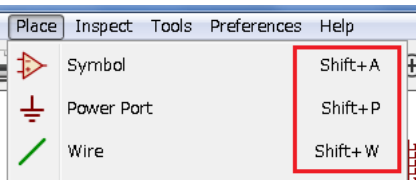
## 3.使用KiCad

### 3.1。快捷键

KiCad有两种相关但不同的快捷键：加速键和热键。两者都用于通过使用键盘而不是鼠标来更改命令来加速KiCad的工作。

#### 3.1.1。加速键

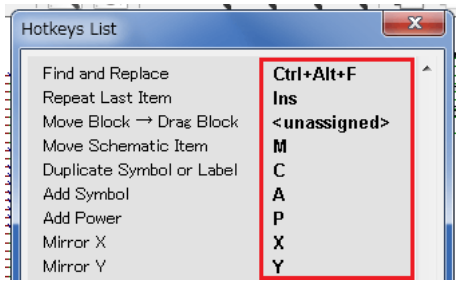
加速键与单击菜单或工具栏图标具有相同的效果：将输入命令，但在单击鼠标左键之前不会发生任何操作。如果要进入命令模式但不想立即采取任何操作，请使用加速键。  
加速键显示在所有菜单窗格的右侧：



#### 3.1.2。热键

热键等于加速键和鼠标左键。使用热键立即在当前光标位置启动命令。使用热键快速更改命令，而不会中断工作流程。

要查看任何KiCad工具中的热键，请转到 **帮助→列出热键** 或按Ctrl + F1：



您可以从 **首选项→热键选项** 菜单中编辑热键的分配，然后导入或导出它们。

**注意** | 在本文档中，热键用括号表示，如下所示：[a]。如果看到[a]，只需在键盘上键入“a”键即可。

### 3.1.3。例

考虑在原理图中添加导线的简单示例。

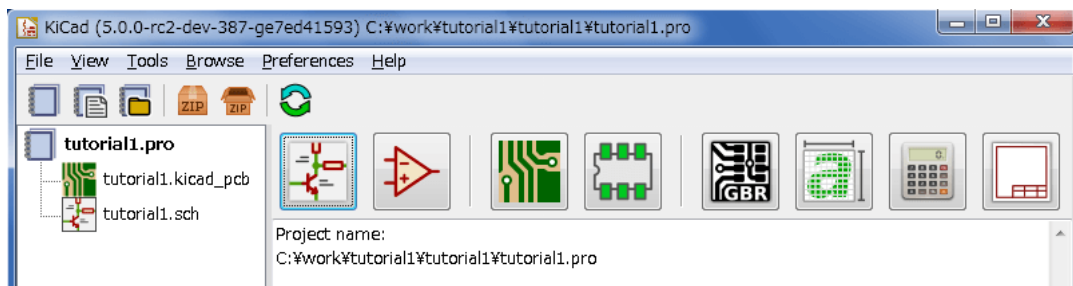
要使用加速键，请按“Shift + W”调用“添加线”命令（注意光标将改变）。接下来，左键单击所需的电线起始位置以开始绘制电线。使用热键，只需按[w]，电线将立即从当前光标位置开始。

## 4.绘制电子原理图

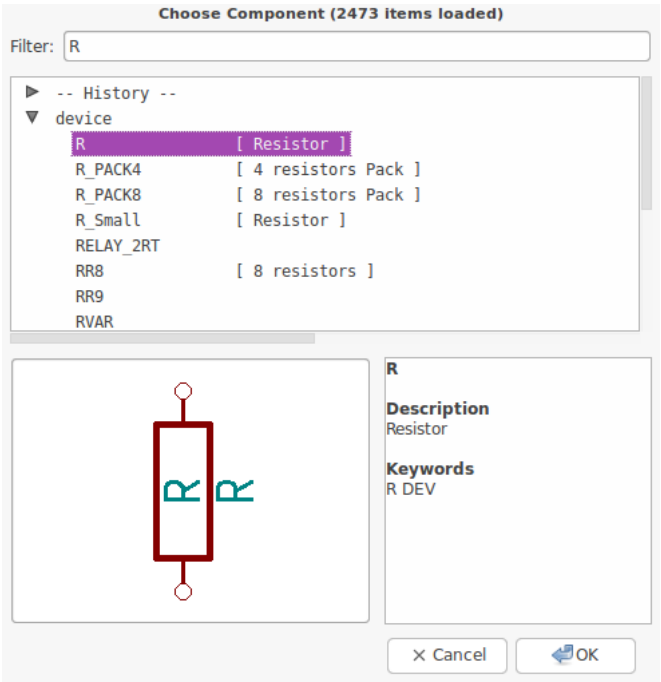
在本节中，我们将学习如何使用KiCad绘制电子原理图。

### 4.1。使用Eeschema

- 在Windows下运行kicad.exe。在您的终端下的Linux类型*kicad*。您现在位于KiCad项目经理的主窗口中。从这里您可以访问八个独立的软件工具：*Eeschema*，*原理图库编辑器*，*Pcbnew*，*PCB封装编辑器*，*GerbView*，*Bitmap2Component*，*PCB计算器*和*PI编辑器*。请参阅工作流程图，了解如何使用主要工具。



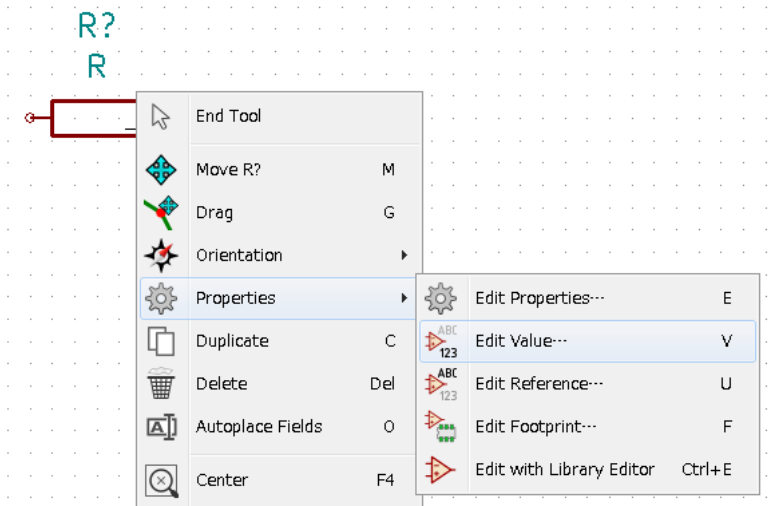
- 创建一个新项目：文件 → 新建 → 项目。将项目文件命名为*tutorial1*。项目文件将自动采用扩展名“.pro”。对话框的确切外观取决于使用的平台，但应该有一个用于创建新目录的复选框。除非您已有专用目录，否则请保持检查状态。您的所有项目文件都将保存在那里。
- 让我们从创建原理图开始。启动原理图编辑器 *Eeschema*，这是左边的第一个按钮。
- 单击顶部工具栏上的“**页面设置**”图标。设置适当的**纸张尺寸**（A4，8.5x11等），并输入标题**教程1**。如有必要，您将在此处看到可以输入更多信息。单击确定。此信息将填充右下角的原理图表。使用鼠标滚轮放大。保存整个原理图：文件 → 保存
- 我们现在将放置第一个组件。单击 右侧工具栏中的“**放置符号**”图标。您也可以按**添加符号**热键[a]。
- 单击原理图工作表的中间部分。一个**选择符号**窗口将出现在屏幕上。我们要放一个电阻器。在搜索/过滤器/R的[R esistor。您可能会注意到电阻器上方的**器件**标题。此 **设备**标题是组件所在库的名称，这是一个非常通用且有用的库。



- 7. 双击它。这将关闭“选择符号”窗口。将组件放在原理图表中，方法是单击所需的位置。
- 8. 单击放大图图标以放大组件。或者，使用鼠标滚轮放大和缩小。按下滚轮（中央）鼠标按钮可水平和垂直平移。
- 9. 尝试将鼠标悬停在组件R上并按[r]。组件应该旋转。您无需实际单击组件即可旋转它。

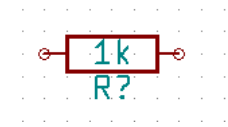
**注意** 如果您的鼠标也在Field Reference (R) 或Field Value (R?) 上，则会出现一个菜单。您将在KiCad中经常看到Clarify Selection菜单; 它允许处理彼此重叠的对象。在这种情况下，告诉KiCad你想对Symbol ... R...执行操作。

- 10. 右键单击组件中间，然后选择“属性”→“编辑值”。将鼠标悬停在组件上并按[v]可以获得相同的结果。或者，[c]将带您进入更一般的“属性”窗口。请注意下面的右键单击菜单如何显示所有可用操作的热键。

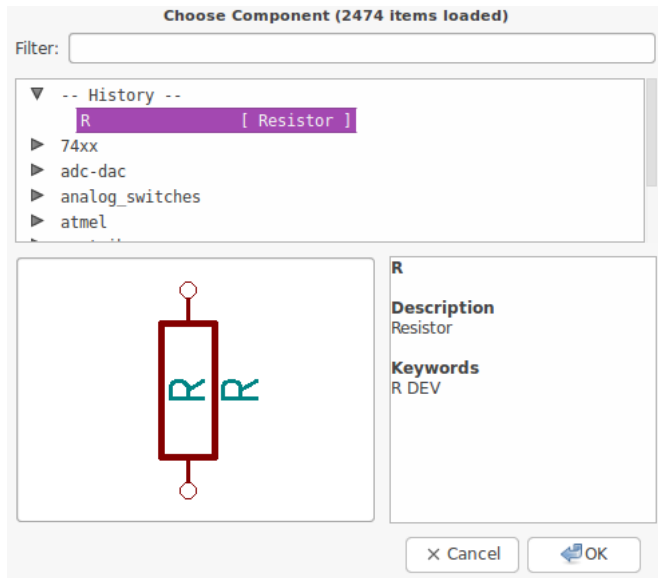


- 11. 将出现“编辑值字段”窗口。替换当前值 - /R用 的1k。单击确定。

**注意** 不要更改参考字段 (R?) ，稍后会自动完成。电阻内部的值现在应为1 k。



- 12. 要放置另一个电阻，只需单击要显示电阻的位置。符号选择窗口将再次出现。
- 13. 您之前选择的电阻器目前正处于历史列表，显示为/R。单击“确定”并放置组件。

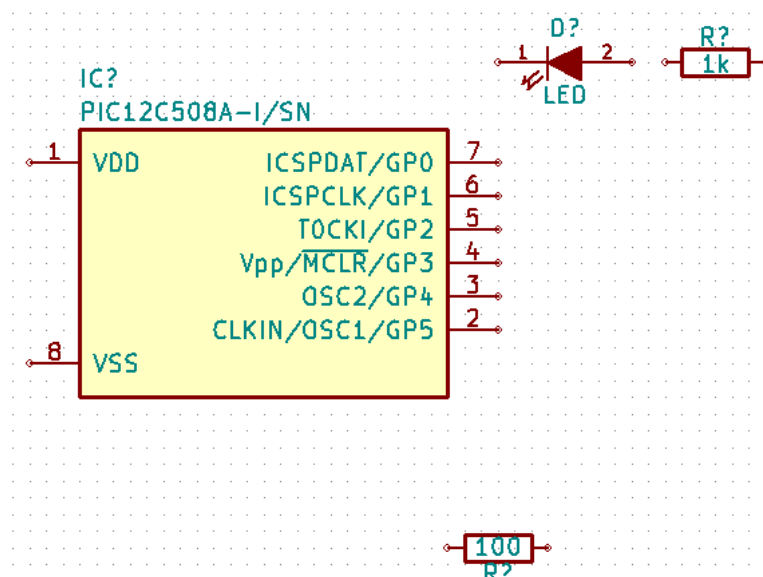


14. 如果您输入错误并想要删除组件，请右键单击该组件，然后单击“删除”。这将从原理图中删除组件。或者，您可以将鼠标悬停在要删除的组件上，然后按[删除]。
15. 您还可以通过将原型上的组件悬停在其上并按[c]来复制原理图纸上已有的组件。单击要放置新重复组件的位置。
16. 右键单击第二个电阻。选择拖动。重新定位组件并左键单击以放下。将鼠标悬停在组件上并按[g]可以实现相同的功能。[r]将旋转组件，而[x]和[y]将围绕其x轴或y轴翻转。

#### 注意

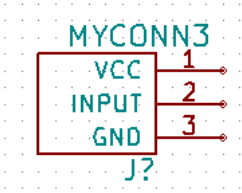
右键单击 → 移动或[m]也是移动任何物体的有用选项，但最好仅将其用于尚未连接的组件标签和组件。我们稍后会看到为什么会这样。

17. 通过将第二个电阻悬停在其上并按[v]编辑第二个电阻。将R替换为100。您可以使用Ctrl + Z撤消任何编辑操作。
18. 更改网格大小。您可能已经注意到，在原理图表上，所有组件都被捕捉到大间距网格上。您可以通过右键单击 → 网格轻松更改网格的大小。通常，建议使用50.0密耳的网格作为原理图表。
19. 我们从未在默认项目中配置的库中添加组件。在菜单中，选择“首选项” → “管理符号库”。在Symbol Libraries窗口中，您可以看到两个选项卡：Global Libraries和Project Specific Libraries。每个都有一个sym-lib-table文件。要使库（.lib文件）可用，它必须位于其中一个sym-lib-table文件中。如果文件系统中没有库文件但尚未提供，则可以将其添加到带有浏览库的sym-lib-table文件之一。为了练习，我们现在将添加一个已经可用的库。
20. 您需要找到计算机上安装官方KiCad库的位置。查找包含数百个.dcm和.lib文件的库目录。尝试C: \ Program Files (x86) \ KiCad \ share \ (Windows) 和 / usr / share / kicad / library / (Linux)。找到目录后，选择并添加MCU\_Microchip\_PIC12.lib库并关闭窗口。您将收到一个警告，该名称已存在于列表中；无论如何添加它。它将添加到列表的末尾。现在单击其名称并将其更改为microchip\_pic12mcu。单击确定关闭符号库窗口。
21. 重复添加组件步骤，但这次选择 microchip\_pic12mcu库而不是Device库并选择 PIC12C508A-1SN组件。
22. 将鼠标悬停在微控制器组件上。请注意，[x]和[y]再次翻转组件。将组件返回到其原始方向。
23. 重复添加组件步骤，这次选择设备库并从中选择LED组件。
24. 组织原理图纸上的所有组件，如下所示。

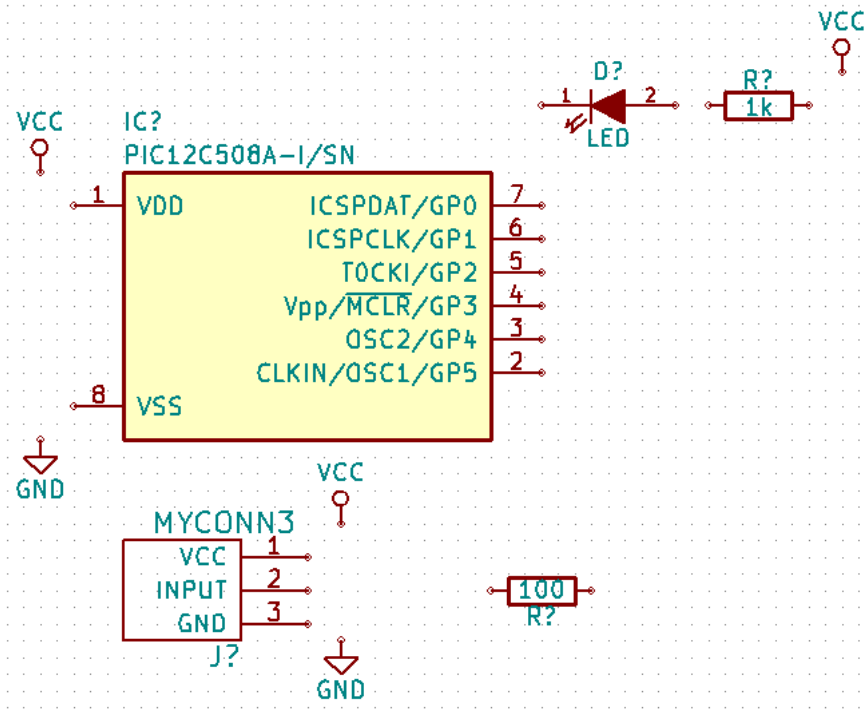




25. 我们现在需要为我们的3针连接器创建原理图组件MYCONN3。您可以跳转到[KiCad中](#)标题为[Make Schematic Components](#)的部分，了解如何从头开始制作该组件，然后返回本节继续使用该板。
26. 您现在可以放置新制作的组件。按[a]并选择myLib库中的MYCONN3组件。
27. 组件标识符J? 将出现在MYCONN3标签下。如果要更改其位置，请右键单击J? 然后单击 *Move Field*（相当于[m]）。在执行此操作之前/之后放大可能会有所帮助。重新定位J? 在如下所示的组件下。标签可以随意移动多次。



28. 是时候放置电源和接地符号了。单击右侧工具栏上的“放置电源端口”按钮。或者，按[p]。在组件选择窗口中，向下滚动并从电源库中选择VCC。单击确定。
29. 单击1 k电阻的引脚上方以放置VCC部件。单击微控制器VDD上方的区域。在组件选择历史记录部分中，选择VCC并将其放在VDD引脚旁边。再次重复添加过程，并在MYCONN3的VCC引脚上方放置一个VCC部分。
30. 重复添加引脚步骤，但这次选择GND部分。将GND部分放在MYCONN3的GND引脚下。在微控制器的VSS引脚左侧放置另一个GND符号。您的原理图现在应该如下所示：



31. 接下来，我们将连接所有组件。单击右侧工具栏上的“放置导线”图标。

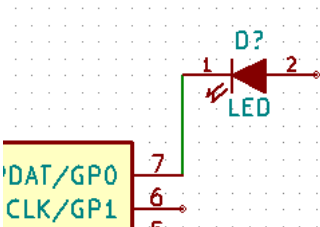
注意

小心不要选择Place bus，它直接显示在此按钮下方，但线条较粗。[KiCad中的总线连接](#)部分 将说明如何使用总线部分。

32. 单击微控制器引脚7末端的小圆圈，然后单击LED引脚1上的小圆圈。您可以在放置连接时放大。

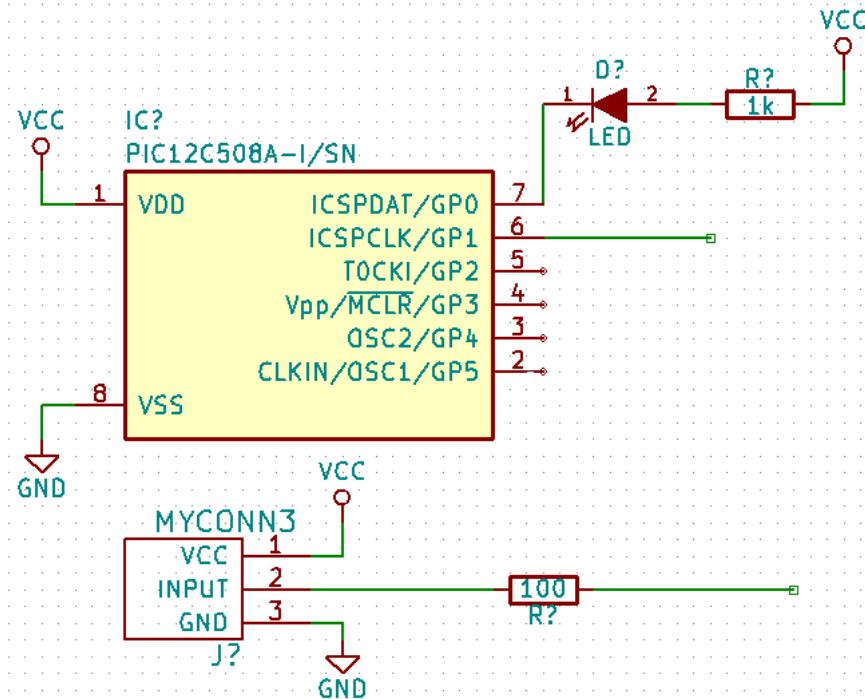
注意

如果要重新定位有线组件，重要的是使用[g]（抓取）而不是[m]（移动）。使用抓取将保持电线连接。如果您忘记了如何移动组件，请查看步骤24。

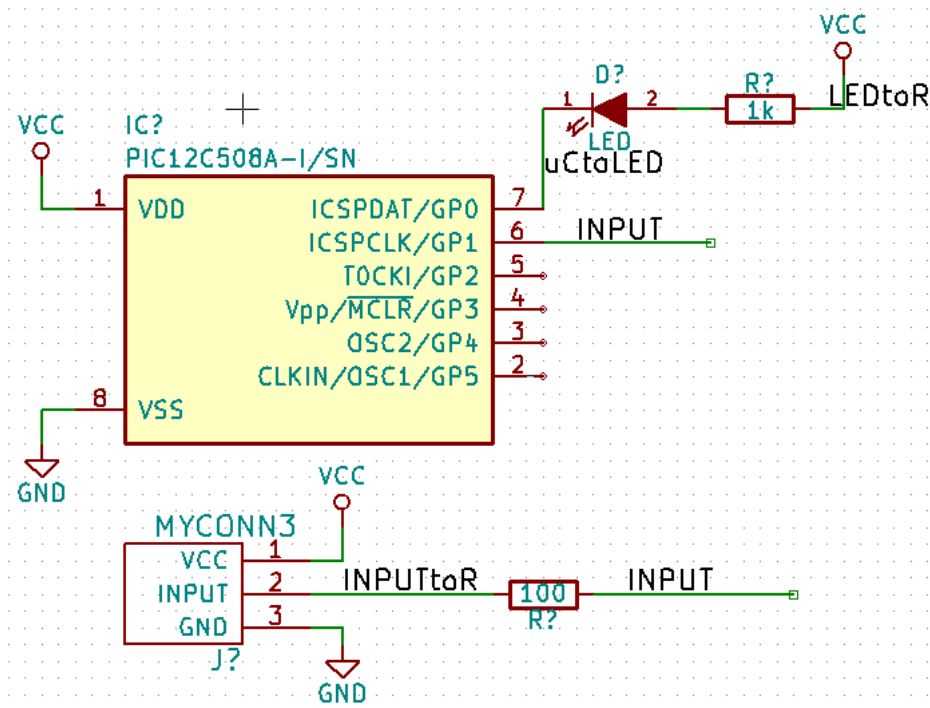




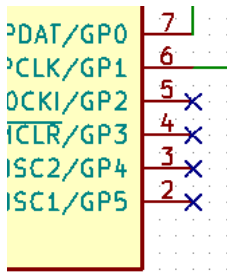
33. 重复此过程并连接所有其他组件，如下所示。要双击终止电线。当连接VCC和GND符号时，导线应接触VCC符号的底部和GND符号的中间顶部。请参见下面的屏幕截图



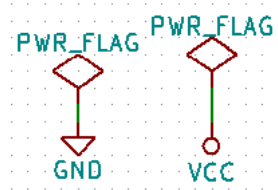
34. 我们现在将考虑使用标签建立连接的另一种方法。单击 右侧工具栏上的“[放置网络标签](#)”图标，选择一个网络标签工具。你也可以用[]。 **A**
35. 单击连接到微控制器引脚6的导线中间。将此标签命名为INPUT。
36. 按照相同的步骤在100欧姆电阻的右侧放置另一个标签。也将其命名为INPUT。这两个标签具有相同的名称，在PIC的引脚6和100欧姆的电阻之间产生不可见的连接。当在复杂设计中连接导线时，这是一种有用的技术，其中绘制线条会使整个原理图变得更加混乱。要放置标签，您不一定需要电线，您只需将标签贴在别针上即可。
37. 标签也可用于简单地标记电线以用于提供信息。在PIC的引脚7上放置一个标签。输入名称 *uCtoLED*。将电阻器和LED之间的导线命名为 *LEDtoR*。将MYCONN3和电阻之间的导线命名为 *INPUTtoR*。
38. 您不必标记VCC和GND线，因为标签是从它们所连接的电源对象中隐含的。
39. 您可以在下面看到最终结果应该是什么样子。



40. 我们现在处理未连接的电线。当KiCad检查时，任何未连接的引脚或电线都会产生警告。为了避免这些警告，您可以指示程序故意连接未连接的电线，或者手动将每个未连接的电线或引脚标记为未连接。
41. 单击右侧工具栏上的“[放置无连接标志](#)”图标。单击引脚2,3,4和5.X似乎表示缺少有线连接是故意的。 **X**



42. 某些组件具有不可见的电源引脚。您可以通过单击左侧工具栏上的“**显示隐藏的图钉**”图标 使其可见。如果遵守VCC和GND命名，则隐藏的电源引脚会自动连接。一般来说，你应该尽量不要制作隐藏的电源引脚。
43. 现在需要添加一个**Power Flag**来向KiCad表明电源是从某个地方进来的。按[A]和搜索 **PWR\_FLAG**这是在 **电力库**。放置其中两个。将它们连接到GND引脚和VCC，如下所示。

**注意**

这将避免经典的原理图检查警告：警告引脚power\_in未驱动（Net xx）

44. 有时候在这里和那里写评论是件好事。要在原理图上添加注释，请使用右侧工具栏上的“**放置文本**”图标。
45. 现在，所有组件都需要具有唯一标识符。事实上，我们的许多组件仍被命名为R? 还是J?。通过单击顶部工具栏上的**Annotate schematic symbols**图标，可以自动完成标识符分配。
46. 在Annotate Schematic窗口中，选择**Use the entire schematic**并单击**Annotate**按钮。在确认消息中单击“确定”，然后单击“**关闭**”。请注意所有的 ? 已被数字取代。每个标识符现在都是唯一的 在我们的示例中，它们被命名为R1, R2, U1, D1和J1。
47. 我们现在将检查原理图的错误。单击顶部工具栏上的**执行电气规则检查**图标。单击“**运行**”按钮。生成一个报告，通知您任何错误或警告，例如断开的电线。你应该有0个错误和0个警告。如果出现错误或警告，原理图中将出现一个绿色小箭头，表示错误或警告所在的位置。选中“**创建ERC文件**”报告，然后再次按“**运行**”按钮以接收有关错误的更多信息。

**注意**

如果您收到“未找到默认编辑器，则必须选择它”的警告，请尝试将路径设置为c: \ windows \ notepad.exe（windows）或 / usr / bin / gedit（Linux）。

48. 原理图现已完成。我们现在可以创建一个Netlist文件，我们将添加每个组件的占用空间。单击顶部工具栏上的 **Generate netlist**图标。单击**Generate Netlist**按钮并保存在默认文件名下。
49. 生成Netlist文件后，单击顶部工具栏上的“**运行Cvpcb**”图标。如果弹出丢失的文件错误窗口，请忽略它并单击“确定”。
50. **Cvpcb**允许您将原理图中的所有组件与KiCad库中的封装链接起来。中心的窗格显示原理图中使用的所有组件。在这里选择D1。在右侧窗格中，您可以看到所有可用的占用空间，此处向下滚动到**LED\_THT: LED-D5.0mm**并双击它。
51. 右侧窗格可能仅显示可用轮廓线的选定子组。这是因为KiCad正试图向您推荐一些合适的足迹。单击图标，然后 启用或禁用这些过滤器。
52. 对于U1，选择**Package\_DIP: DIP-8\_W7.62mm**足迹。对于J1，选择**Connector: Banana\_Jack\_3Pin**足迹。对于R1和R2，选择**Resistor\_THT: R\_Axial\_DIN0207\_L6.3mm\_D2.5mm\_P2.54mm\_Vertical**足迹。
53. 如果您有兴趣知道您选择的足迹是什么样的，您可以单击 **查看所选足迹**图标 以预览当前足迹。
54. 你完成了。您可以通过单击**文件 → 保存原理图**或使用**应用, 保存原理图和继续**按钮立即 **保存原理图**。
55. 您可以关闭**Cvpcb**并返回**Eeschema**原理图编辑器。如果您没有将其保存在Cvpcb中，请单击**文件 → 保存立即保存**。再次创建网表。您的网表文件现已更新，包含所有足迹。请注意，如果您缺少任何设备的占地面积，则需要制作自己的封装。这将在本文档的后续部分中解释。
56. 切换到KiCad项目经理。
57. 网表文件描述了所有组件及其各自的引脚连接。网表文件实际上是一个文本文件，您可以轻松地检查，编辑或编写脚本。

**注意**

库文件（\*.lib）也是文本文件，它们也可以轻松编辑或编写脚本。

58. 要创建物料清单（BOM），请转到Eeschema原理图编辑器，然后单击顶部工具栏上的*生成物料清单*图标。默认情况下，没有活动的插件。您可以通过单击“**添加插件**”按钮添加一个。选择要使用的\*.xsl文件，在这种情况下，我们选择**bom2csv.xsl**。

### 注意

**Linux的：**

如果缺少xsltproc，您可以下载并安装它：

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

对于像Ubuntu这样的Debian派生发行版，或者

```
sudo yum install xsltproc
```

对于RedHat派生的发行版。如果您不使用这两种发行版，请使用您的发行版软件包管理器命令来安装xsltproc软件包。

xsl文件位于：`/usr/lib/kicad/plugins/`。

**Apple OS X：**

如果缺少xsltproc，您可以从应该包含它的Apple站点安装Apple Xcode工具，或者下载并安装它：

```
brew install libxslt
```

xsl文件位于：`/Library/Application Support/kicad/plugins/`。

**视窗：**

xsltproc.exe和包含的xsl文件将分别位于 `<KiCad安装目录>\bin`和 `<KiCad安装目录>\bin\scripting\plugins`。

**所有平台：**

您可以通过以下方式获取最新的bom2csv.xsl：

[https://raw.githubusercontent.com/KiCad/kicad-source-mirror/master/eeschema/plugins/xsl\\_scripts/bom2csv.xsl](https://raw.githubusercontent.com/KiCad/kicad-source-mirror/master/eeschema/plugins/xsl_scripts/bom2csv.xsl)

**KiCad自动生成命令，例如：**

```
xsltproc -o"%0"/ home / <user>/kicad/eeschema/plugins/bom2csv.xsl"%I"
```

您可能想要添加扩展名，因此请将此命令行更改为：

```
xsltproc -o"%0.csv"/ home / <>> / kicad / eeschema /plugins / bom2csv.xsl"%I"
```

按“帮助”按钮获取更多信息。

59. 现在按*Generate*。该文件（与项目同名）位于项目文件夹中。使用LibreOffice Calc或Excel 打开\*.csv文件。将出现导入窗口，按OK。

您现在可以转到PCB布局部分，这将在下一节中介绍。但是，在继续之前，让我们快速了解如何使用总线连接组件引脚。

## 4.2. KiCad的巴士连接

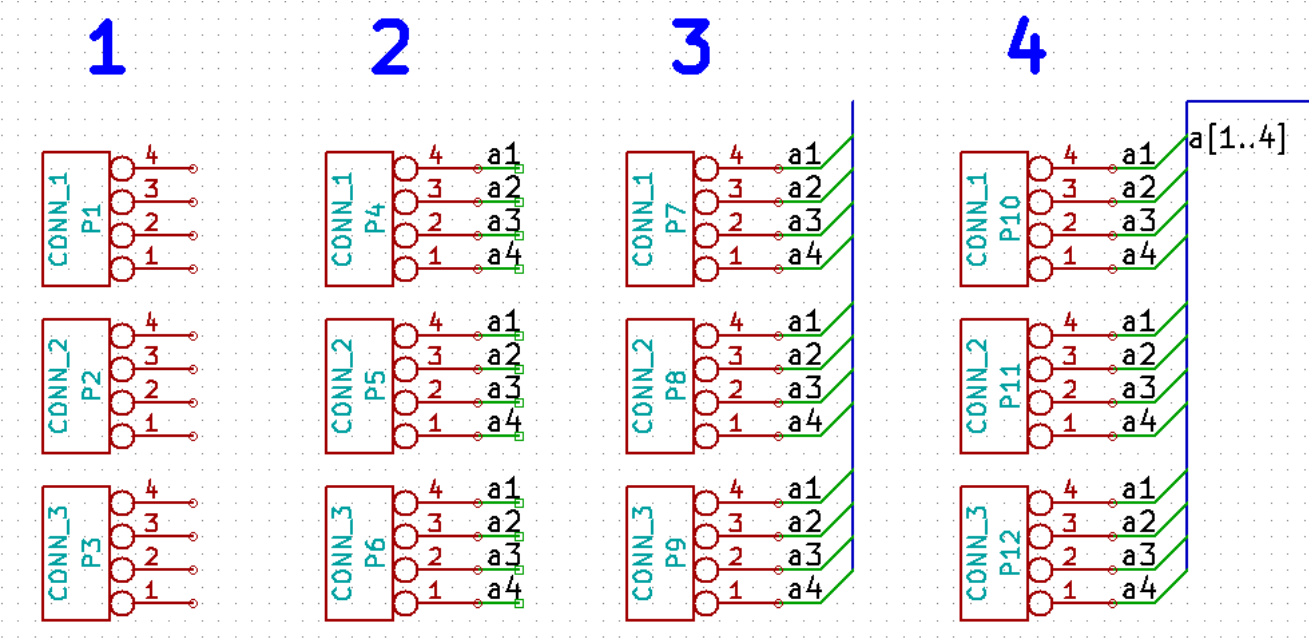
有时您可能需要将组件A的多个顺序引脚与组件B的其他顺序引脚连接。在这种情况下，您有两个选项：我们已经看到的标记方法或使用总线连接。让我们看看如何做到这一点。

- 让我们假设您有三个4针连接器，您想要将引脚连接在一起。使用标签选项（按`[L]`）标记P4部件的引脚4。将此标签命名为a1。现在按`[Insert]`将相同的项目自动添加到引脚4（PIN 3）下方的引脚上。注意标签是如何自动重命名为a2的。
- 再按`[插入]`两次。这个键对应于动作*重复最后一项*，它是一个无限有用的命令，可以让你的生活更轻松。
- 在另外两个连接器CONN\_2和CONN\_3上重复相同的标签操作，您就完成了。如果继续制作PCB，您将看到三个连接器相互连接。图2显示了我们描述的结果。出于美观目的，还可以使用图标使用图标和总线将一系列Place线添加到总线入口，如图3所示。但是，请注意，对PCB没有影响。
- 应该指出的是，连接到图2中的引脚的短导线不是严格必需的。实际上，标签可以直接应用于引脚。
- 让我们更进一步，假设您有一个名为CONN\_4的第四个连接器，无论出于何种原因，它的标签恰好有点不同（b1，b2，b3，b4）。现在我们想要以引脚到引脚的方式将总线a与总线b连接起来。我们希望不使用引脚标记（这也是可能的），而是使用总线上的标签，每个总线一个标签。
- 使用之前说明的标记方法连接并标记CONN\_4。将引脚命名为b1，b2，b3和b4。使用图标将引脚连接到一系列Wire to bus entry，并使用图标连接到总线。见图4。
- 在CONN\_4的总线上放一个标签（按`[L]`）并命名为b[1..4]。

- 8. 在前一个总线上放一个标签（按`[I]`）并将其命名为`a[1..4]`。
- 9. 我们现在可以做的是使用带按钮的总线将总线a `[1..4]`与总线b `[1..4]`连接起来。
- 10. 通过将两条总线连接在一起，引脚a1将自动连接到引脚b1，a2将连接到b2，依此类推。图4显示了最终结果的样子。

**注意** 通过`[插入]`可访问的 **重复最后一项**选项可以成功用于重复期间项目插入。例如，连接到图2，图3和图4中所有引脚的短导线都已放置此选项。

- 11. 通过`[Insert]`访问 的 *Repeat last item*选项也被广泛用于 使用图标将多个Wire系列放入总线入口。

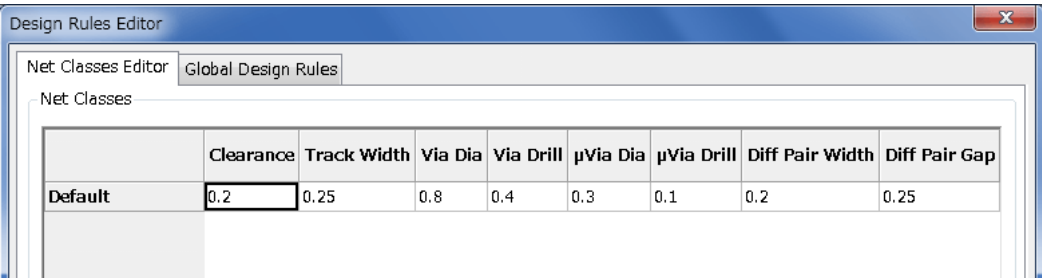


## 5. 布局印刷电路板


现在是时候使用您生成的网表文件来布局PCB了。这是通过Pcbnew工具完成的。

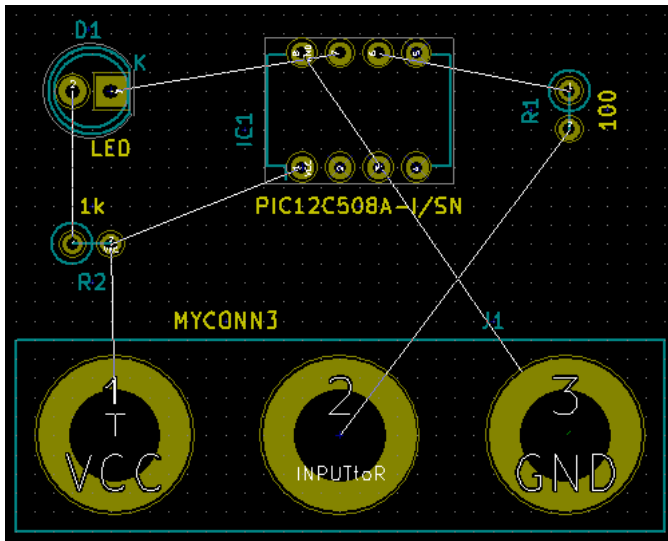
### 5.1. 使用Pcbnew


- 1. 从KiCad项目经理处，单击Pcb布局编辑器图标。该Pcbnew窗口将打开。如果您收到错误消息，指出\*.kicad\_pcb文件不存在并询问您是否要创建它，只需单击是。
- 2. 首先输入一些原理图信息。单击顶部工具栏上的“ 页面设置”图标。设置适当的纸张尺寸（A4，8.5x11等），标题为教程1。
- 3. 最好将间隙和 最小轨道宽度设置为PCB制造商要求的宽度。通常，您可以将间隙设置为0.25，将最小轨道宽度设置为0.25。单击设置 → 设计规则菜单。如果它尚未显示，请单击“ 网络类编辑器”选项卡。将窗口顶部的“ 清除”字段更改为0.25，将“ 轨道宽度”字段更改为0.25，如下所示。这里的测量单位是mm。

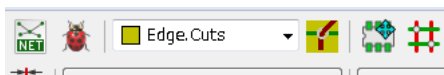


- 4. 单击“ 全局设计规则”选项卡，将“ 最小轨道宽度”设置为 0.25。单击“确定”按钮以提交更改并关闭“设计规则编辑器”窗口。
- 5. 现在我们将导入网表文件。单击顶部工具栏上的“ 读取网表”图标。如果是从Eeschema创建的，则应在Netlist文件字段中选择网表文件tutorial1.net。单击“ 读取当前网表”。然后单击“ 关闭”按钮。
- 6. 现在应该可以看到所有组件。选择它们并按照鼠标光标。
- 7. 将组件移动到板的中间。如有必要，您可以在移动组件时放大和缩小。单击鼠标左键。

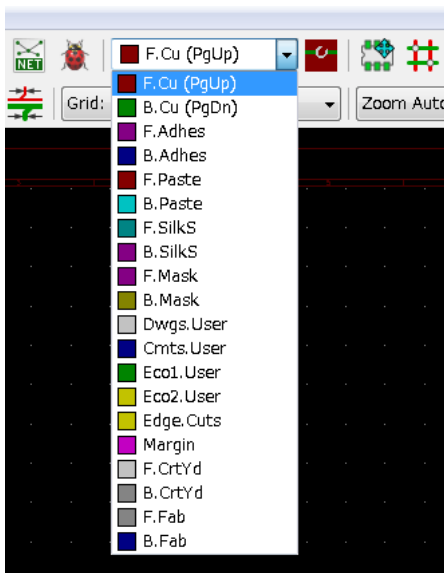
8. 所有组件都通过称为*ratsnest*的一组细线连接。确保按下 **显示/隐藏板鼠标按钮**。通过这种方式，您可以看到*ratsnest*链接所有组件。
9. 您可以通过将每个组件悬停在其上并按[m]来移动它们。单击要放置它们的位置。或者，您可以通过单击选择组件然后拖动它。按[r]旋转组件。移动所有组件，直到最小化电线交叉的数量。




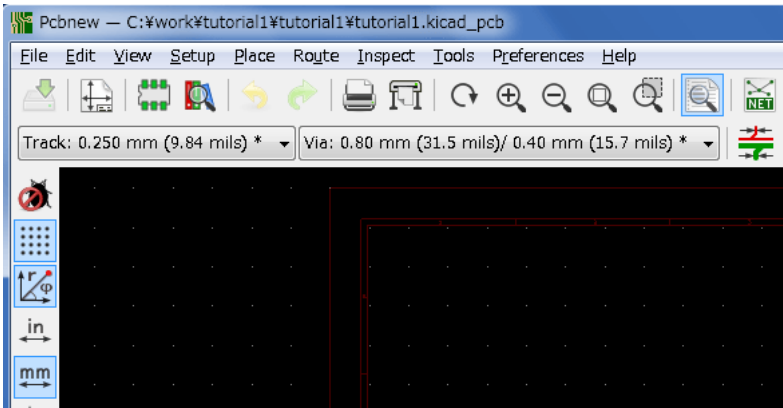
10. 注意100欧姆电阻的一个引脚如何连接到PIC组件的引脚6。这是用于连接引脚的标记方法的结果。标签通常比实际的电线更受欢迎，因为它们使原理图更加杂乱。
11. 现在我们将定义PCB的边缘。从顶部工具栏的下拉菜单中选择*Edge.Cuts*图层。单击右侧工具栏上的“**添加图形线**”图标。跟踪电路板边缘，点击每个角落，并记住在绿色边缘和PCB边缘之间留一个小间隙。



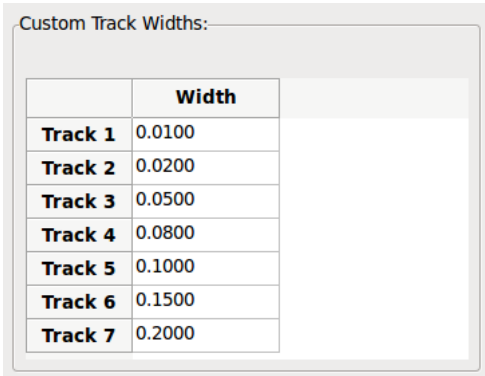
12. 接下来，连接除GND之外的所有电线。事实上，我们将使用放置在电路板底部铜线（称为*B.Cu*）上的接地层一次连接所有GND连接。
13. 现在我们必须选择我们想要处理的铜层。在顶部工具栏的下拉菜单中选择 *F.Cu (PgUp)*。这是前顶部铜层。



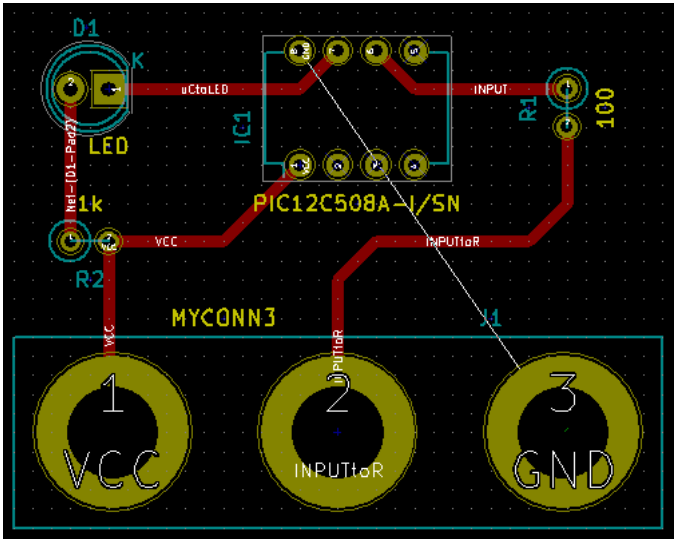
14. 例如，如果您决定使用4层PCB，请转到 **设置** → **图层设置**并将 **铜图**层更改为4.在 **图层表**中，您可以命名图层并确定它们的用途。请注意，可以通过“**预设图层分组**”菜单选择非常有益的预设。
15. 单击右侧工具栏上的 **Route tracks**图标。单击J1的第1针并运行轨道以填充 R2。双击以设置轨道结束的点。此轨道的宽度将默认为0.250 mm。您可以从顶部工具栏的下拉菜单中更改轨道宽度。请注意，默认情况下，您只有一个可用的轨道宽度。



16. 如果要添加更多轨道宽度，请转到：设置 → 设计规则 → 全局设计规则选项卡，然后在此窗口的右下角添加您希望可用的任何其他宽度。然后，您可以在布置电路板时从下拉菜单中选择轨道的宽度。请参见下面的示例（英寸）。

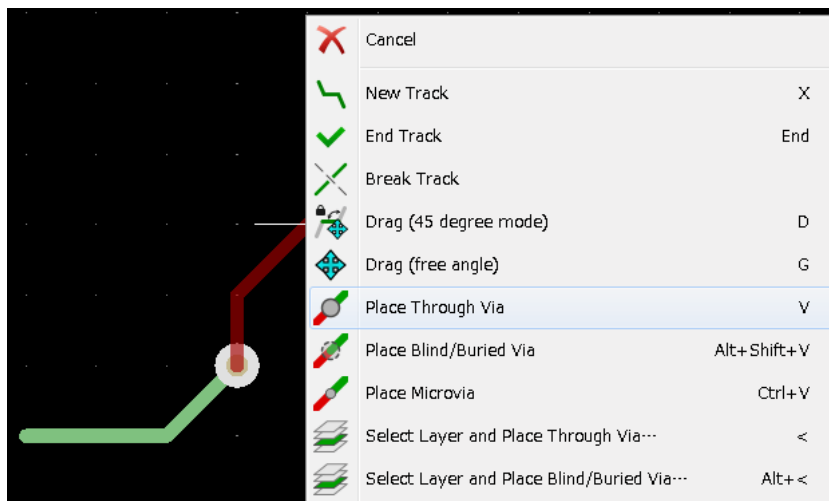




17. 或者，您可以添加一个Net Class，在其中指定一组选项。转到设置 → 设计规则 → 网络类编辑器，然后添加一个名为power的新类。将轨道宽度从8密耳（表示为0.0080）更改为24密耳（表示为0.0240）。接下来，将除地面之外的所有内容添加到功率等级（在左侧选择默认值，在右侧选择电源并使用箭头）。
18. 如果要更改网格大小，请右键单击 → 网格。在放下组件并将它们与轨道连接在一起之前或之后，请务必选择合适的网格尺寸。
19. 重复此过程，直到连接除J1的引脚3之外的所有电线。您的电路板应如下所示。

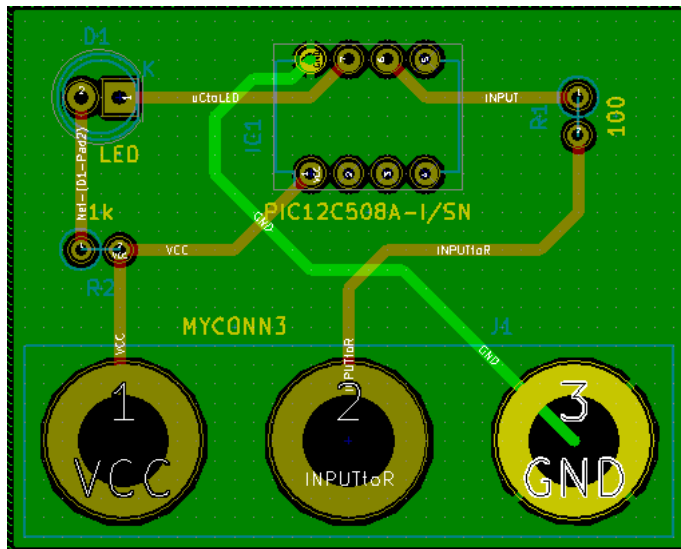


20. 现在让我们在PCB的另一个铜侧上运行轨道。在顶部工具栏的下拉菜单中选择 *B.Cu*。单击Route tracks 图标。在J1的引脚3和U1的引脚8之间画一条轨道。这实际上没有必要，因为我们可以用地平面做到这一点。注意轨道的颜色是如何变化的。
21. 从销甲去通过改变层到管脚乙。通过放置通道可以在运行轨道时更改铜平面。当您在上部铜平面上运行轨道时，右键单击并选择“放置Via”或只需按[V]。这会将您带到底层，您可以在其中完成曲目。



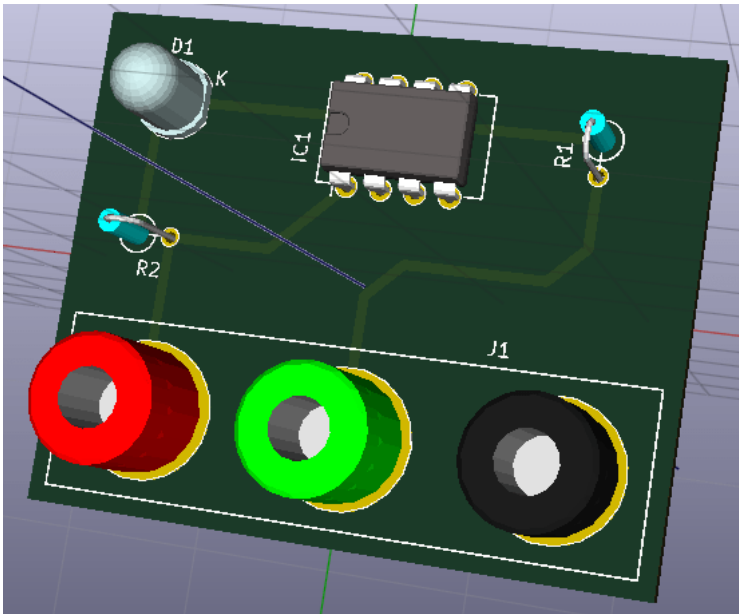


22. 如果要检查特定连接，可以单击右侧工具栏上的“**突出显示**”网络图标。单击J1的第3针。轨道本身和连接到它的所有打击垫都应突出显示。
23. 现在我们将制作一个连接到所有GND引脚的接地层。单击右侧工具栏上的“**添加填充区域**”图标。我们将在棋盘周围跟踪一个矩形，因此单击您想要其中一个角的位置。在出现的对话框中，将**默认打击垫连接**设置为“**热释放**”并将“**轮廓坡度**”设置为**仅限H，V和45度**，然后单击“确定”。
24. 通过单击旋转中的每个角来跟踪板的轮廓。通过第二次单击第一个角完成矩形。右键单击您刚刚跟踪的区域。单击**区域**→**填充或重新填充所有区域**。董事会应填写绿色，看起来像这样：



25. 运行通过单击检查器的设计规则进行设计规则检查图标顶部工具栏上。单击Start DRC。应该没有错误。单击List Unconnected列表。应该没有未连接的项目。单击“确定”关闭“DRC控制”对话框。
26. 单击文件 → 保存以保存文件。要以3D方式欣赏您的电路板，请单击视图 → 3D查看器。





- 27. 您可以拖动鼠标来旋转PCB。
- 28. 你的董事会是完整的。要将其发送给制造商，您需要生成所有Gerber文件。

5.2。生成Gerber文件

完成PCB后，您可以为每个层生成Gerber文件，并将它们发送给您最喜欢的PCB制造商，他们将为您制作电路板。

- 1. 从KiCad开始，打开Pcbnew板编辑器。
- 2. 单击文件 → 绘图。选择Gerber作为Plot格式，然后选择放置所有Gerber文件的文件夹。单击“绘图”按钮继续。
- 3. 要生成钻取文件，请从Pcbnew再次转到文件 → 绘图选项。默认设置应该没问题。
- 4. 这些是您制作典型的2层PCB时需要选择的层：

层	KiCad图层名称	默认的Gerber扩展	启用“使用Protel文件扩展名”
底层	B.Cu	.GBR	.GBL
顶层	F.Cu	.GBR	.GTL
顶部叠加	F.SilkS	.GBR	.GTO
底部焊料抗蚀剂	B.Mask	.GBR	.GBS
顶级焊料抗蚀剂	F.Mask	.GBR	.GTS
边缘	Edge.Cuts	.GBR	.GM1

5.3。使用GerbView

- 1. 要查看所有Gerber文件，请转到KiCad项目管理器并单击GerbView图标。在下拉菜单或图层管理器中，选择图形图层1。单击文件 → 打开Gerber文件或单击图标。选择并打开所有生成的Gerber文件。注意它们如何一个在另一个上面显示。
- 2. 使用File → Open Excellon Drill File打开钻取文件。
- 3. 使用右侧的图层管理器选择/取消选择要显示的图层。在发送生产之前，请仔细检查每一层。
- 4. 该视图与Pcbnew类似。在视图内右键单击，然后单击“网格”以更改网格。

5.4。使用FreeRouter自动路由

手动布线板既快速又有趣，但对于具有大量组件的电路板，您可能希望使用自动布线器。请记住，您应首先手动路由关键迹线，然后设置自动布线器以执行无聊位。它的工作只会考虑到未路由的痕迹。我们在这里使用的自动布线器是FreeRouting。

注意

FreeRouting是一个开源的Java应用程序。目前FreeRouting存在于几个或多或少相同的副本中，您可以通过互联网搜索“freerouting”找到它们。它可以在源代码形式或预编译的Java包中找到。




1. 从Pcbnew单击文件 → 导出 → Specctra DSN 并在本地保存文件。启动FreeRouter并单击Open Your Own Design 按钮，浏览dsn文件并加载它。
2. FreeRouter具有KiCad目前不具备的一些功能，包括手动路由和自动路由。FreeRouter主要有两个步骤：首先，对电路板进行布线，然后对其进行优化。完全优化可能需要很长时间，但您可以随时停止它。
3. 您可以通过单击顶部栏上的“自动布线器”按钮来启动自动路由。底栏为您提供有关正在进行的路由过程的信息。如果通过计数超过30，则您的主板可能无法使用此路由器自动执行。更多地展开您的组件或更好地旋转它们并再试一次。零件的旋转和位置的目标是降低老鼠的交叉航空公司的数量。
4. 左键单击鼠标可以停止自动路由并自动启动优化过程。另一次左键单击将停止优化过程。除非你真的需要停下来，否则最好让FreeRouter完成它的工作。
5. 单击文件 → 导出Specctra会话文件菜单，并使用.ses扩展名保存板文件。您实际上不需要保存FreeRouter规则文件。
6. 回到Pcbnew。您可以通过单击文件 → 导入 → 光谱会话并选择.ses文件来导入新布线的电路板。

如果有任何您不喜欢的路由跟踪，可以使用[删除]和路由工具（右侧工具栏上的路径跟踪图标）将其删除并重新路由。 

## 6.在KiCad中转发注释

完成电子原理图，封装布局，电路板布局和生成Gerber文件后，您就可以将所有内容发送给PCB制造商，以便您的电路板成为现实。

通常，这种线性工作流程并非如此单向。例如，当您必须修改/扩展您或其他人已完成此工作流程的板时，您可能需要移动组件，替换其他组件，更改占用空间等等。在此修改过程中，您不想做的是从头开始重新布线整个电路板。相反，这就是你如何做到这一点：

1. 假设您想要用CON2替换假设的连接器CON1。
2. 您已经拥有完整的原理图和完全布线的PCB。
3. 从KiCad开始，启动Eeschema，删除CON1并添加CON2进行修改。使用图标保存原理图项目，然后单击顶部工具栏上的  生成图标。
4. 点击Netlist然后保存。保存为默认文件名。你必须重写旧的。
5. 现在为CON2分配足迹。单击顶部工具栏上的“运行Cypcb”图标 。将足迹分配给新设备CON2。其余组件仍然具有分配给它们的前轮廓线。关闭Cypcb。
6. 返回原理图编辑器，单击File → Save Whole Schematic Project保存项目。关闭原理图编辑器。
7. 从KiCad项目经理处，单击Pcbnew图标。该Pcbnew窗口将打开。
8. 旧的，已经布线的电路板应该会自动打开。让我们导入新的网表文件。单击顶部工具栏上的“读取网表”图标 。
9. 单击Browse Netlist Files按钮，在文件选择对话框中选择网表文件，然后单击Read Current Netlist。然后单击“关闭”按钮。
10. 此时，您应该能够看到已布置所有先前组件的布局。在左上角，您应该看到所有未布线的组件，在我们的例子中是CON2。用鼠标选择CON2。将组件移动到板的中间。
11. 放置CON2并路由它。完成后，保存并继续照常生成Gerber文件。


此处描述的过程可以根据需要轻松重复多次。除了上面描述的前向注释方法之外，还有另一种称为后向注释的方法。此方法允许您从Pcbnew修改已经布线的PCB，并在原理图和网表文件中更新这些修改。然而，Backward Annotation方法没有那么有用，因此这里不再描述。

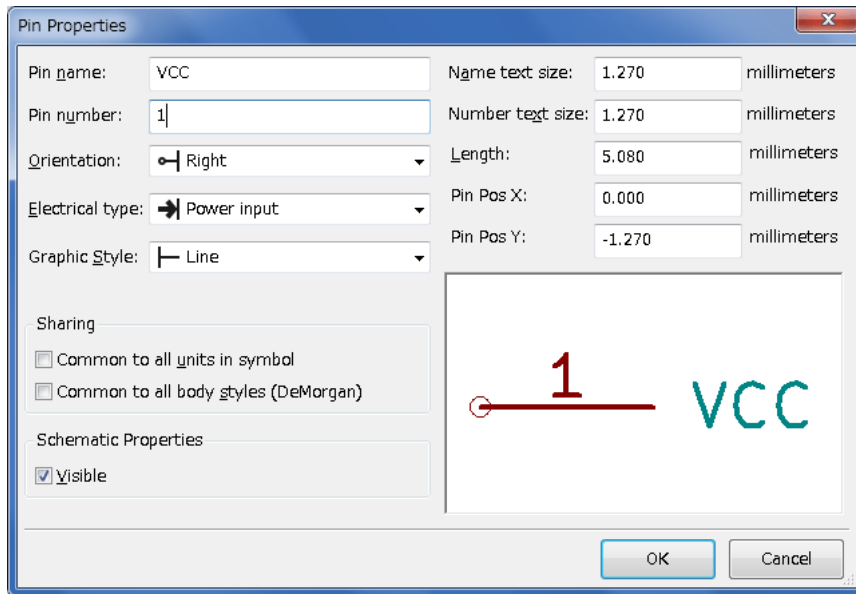
## 7.在KiCad中制作原理图符号

有时，您想要放置在原理图中的符号不在KiCad库中。这很正常，没有理由担心。在本节中，我们将了解如何使用KiCad快速创建新的原理图符号。不过，请记住，您始终可以在Internet上找到KiCad组件。

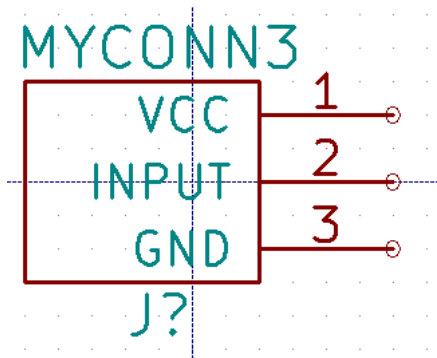
在KiCad中，符号是一段以DEF开头并以ENDDEF结尾的文本。一个或多个符号通常放在扩展名为.lib的库文件中。如果要将符号添加到库文件，只需使用文本编辑器的剪切和粘贴命令即可。

### 7.1. 使用组件库编辑器

1. 我们可以使用 **组件库编辑器**（Eeschema的一部分）来创建新组件。在我们的项目文件夹tutorial1中，让我们创建一个名为library的文件夹。在我们创建新组件后，我们将在内部放置新的库文件 myLib.lib。
2. 现在我们可以开始创建新组件了。从KiCad启动 Eeschema，单击Library Editor图标，然后单击New component图标。将出现“组件属性”窗口。将新组件命名为MYCONN3，将默认参考标识符设置为J，将每个包的单位数设置为1。单击确定。如果出现警告，请单击“是”。此时，组件仅由其标签组成。让我们添加一些引脚。单击“添加引脚”图标  在右侧工具栏上。要放置销钉，请在MYCONN3标签正下方的零件编辑器工作表中间单击鼠标左键。
3. 在出现的Pin Properties窗口中，将引脚名称设置为 VCC，将引脚编号设置为1，将Electrical类型设置为 Power输入，然后单击OK。



4. 在MYCONN3标签下方点击您想要的位置放置引脚。
5. 重复放置引脚步骤，此时 引脚名称应为 INPUT， 引脚编号应为2， 电气类型应为无源。
6. 重复放置引脚步骤，此时 引脚名称应为GND， 引脚编号应为3， 电气类型应为无源。将销钉一个放在另一个上面。组件标签 MYCONN3应位于页面的中心（蓝线交叉的位置）。
7. 接下来，绘制组件的轮廓。单击“添加矩形”图标。我们想在引脚旁边画一个矩形，如下图所示。要执行此操作，请单击矩形左上角的位置（不要按住鼠标按钮）。再次单击矩形右下角的位置。



8. 如果要用黄色填充矩形，请在“首选项”→“选择颜色方案”中将填充颜色设置为黄色4，然后使用[c]在编辑屏幕中选择矩形，选择“填充背景”。
9. 将组件保存在库myLib.lib中。单击 New Library图标，导航到tutorial1 / library /文件夹并保存名为myLib.lib的新库文件。
10. 去首 → 组件库，并添加两个教程1 /库/在 用户定义的搜索路径，并在MYLIB.LIB 组件库文件。
11. 单击“选择工作库”图标。在“选择库”窗口中，单击myLib，然后单击“确定”。注意窗口的标题如何表示当前正在使用的库，现在应该是myLib。
12. 单击顶部工具栏中的“更新当前库中的当前组件”图标。单击顶部工具栏中的“保存当前加载的磁盘库”图标，保存所有更改。在出现的任何确认消息中单击“是”。新的原理图组件现已完成，并在窗口标题栏中指示的库中可用。
13. 您现在可以关闭“组件库”编辑器窗口。您将返回到原理图编辑器窗口。您的新组件现在可以从库myLib中使用。
14. 您可以通过将任何库file.lib文件添加到库路径来使其可用。从Eeschema，转到 首选项 → 库，并在用户定义的搜索路径和组件库文件中的file.lib中添加路径。

## 7.2。导出，导入和修改库组件

而不是从头开始创建库组件，有时更容易从已经制作并修改它。在本节中，我们将了解如何将组件从KiCad标准库 设备导出 到您自己的库myOwnLib.lib，然后进行修改。

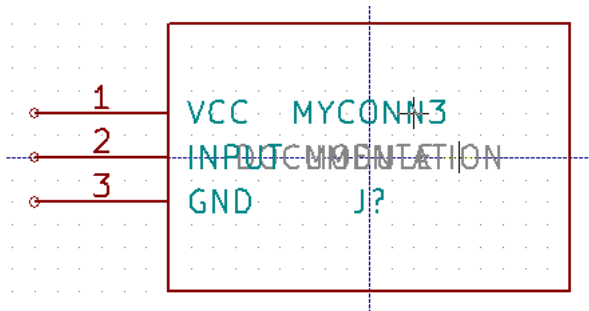
1. 从KiCad启动Eeschema，单击Library Editor图标，单击Select working library图标并选择库 设备。单击Load组件以从当前lib图标 进行编辑，然后导入RELAY\_2RT。
2. 单击Export组件图标，导航到库/ 文件夹并保存名为myOwnLib.lib的新库文件。

- 您可以通过将此组件和整个库`myOwnLib.lib`添加到库路径来使其可用。从 *Eeschema*，去首选项 → 组件库，并添加两个 库/在用户定义的搜索路径和`myOwnLib.lib`的 组件库文件。关上窗户。
- 单击“选择工作库”图标。在Select Library窗口中，单击`myOwnLib`并单击OK。注意窗口的标题如何表示当前正在使用的库，它应该是 `myOwnLib`。
- 单击Load组件以从当前lib图标 进行编辑，然后导入RELAY\_2RT。
- 您现在可以根据需要修改组件。将鼠标悬停在标签 `RELAY_2RT`上，按[e]并将其重命名为`MY_RELAY_2RT`。
- 单击顶部工具栏中的“更新当前库中的当前组件”图标。单击顶部工具栏中的“保存当前加载的磁盘库”图标，保存所有更改。

## 7.3。使用quicklib制作原理图组件

本节介绍创建MYCONN3原理图元件的另一种方法（见MYCONN3使用因特网工具上方）`quicklib`。

- 前往`quicklib`网页：<http://kicad.rohrbacher.net/quicklib.php>
- 使用以下信息填写页面：组件名称：MYCONN3参考前缀：J引脚布局样式：SIL引脚数，N：3
- 单击Assign Pins图标。使用以下信息填写页面：引脚1：VCC引脚2：输入引脚3：GND。类型：所有3个引脚都被动。
- 单击“预览”图标，如果满意，请单击“构建库组件”。下载该文件并将其重命名为 `tutorial1 / library / myQuickLib.lib`。你完成了！
- 使用KiCad查看它。从KiCad项目经理，启动 *Eeschema*，单击Library Editor图标，单击Import Component 图标，导航到 `tutorial1 / library /` 并选择`myQuickLib.lib`。



- 您可以将此组件和整个库`myQuickLib.lib`添加到KiCad库路径中。从 *Eeschema*，去首选项 → 组件库和添加库在 用户定义的搜索路径和`myQuickLib.lib`在 组件库文件。

正如您可能猜到的，当您想要创建具有大引脚数的组件时，这种创建库组件的方法非常有效。

## 7.4。制作高引脚数的原理图元件

在`quicklib`的标题为Make Schematic Components的部分中，我们了解了如何使用基于web的`quicklib`工具制作原理图元件。但是，您偶尔会发现需要创建一个具有大量引脚（几百个引脚）的原理图元件。在KiCad中，这不是一项非常复杂的任务。

- 假设您要为具有50个引脚的器件创建原理图元件。通常的做法是使用多个低引脚数的图纸绘制它，例如两个图纸，每个图纸有25个引脚。该组件表示允许简单的引脚连接。
- 创建组件的最佳方法是使用`quicklib`分别生成两个25引脚组件，使用Python脚本重新编号它们的引脚，最后通过使用复制和粘贴将它们合并为一个单独的DEF和ENDDEF组件。
- 您将在下面找到一个简单的Python脚本示例，它可以与`in.txt`文件和`out.txt`文件一起使用，以重新编号该行：X PIN1 1 -750 600 300 R 50 50 1 1 I into X PIN26 26 -750 600 300 R 50 50 1 1 I这是针对文件`in.txt`中的所有行完成的。

简单的脚本

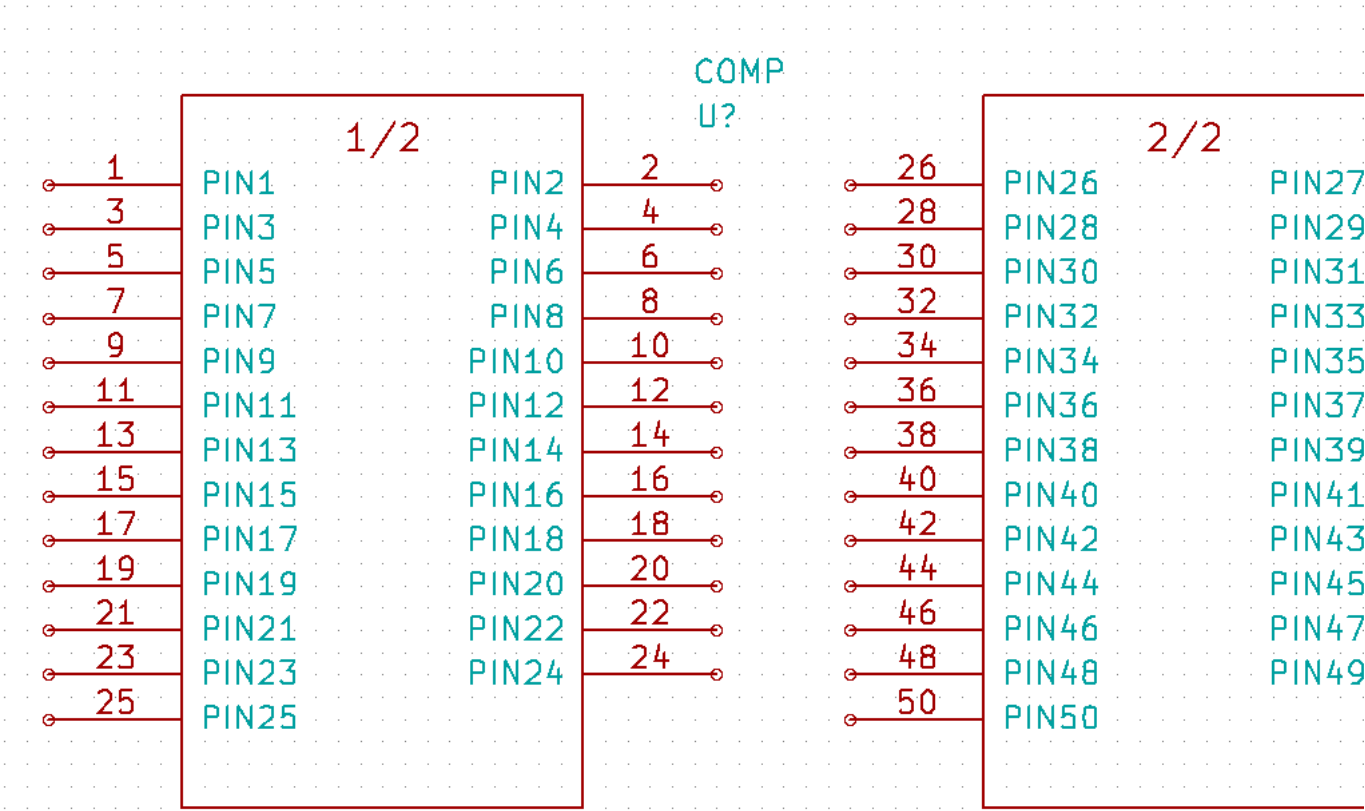
```
! #在 /usr / 斌 / 包膜蟒蛇
''' 简单的脚本来操纵KiCad的元件引脚编号'''
import sys, re
尝试:
    鳍 = 开 ( sys。 ARGV [ 1 ], 'R' )
    FOUT = 开放 ( sys。 argv [ 2 ], 'w' )
除了:
    打印 "哦，错误使用这个应用程序，尝试："， sys。 argv [ 0 ], "in.txt out.txt"
    sys。退出 ( )
for ln in fin。readlines ( ):
    obj = re。搜索 ( "(X PIN) (\\d *) (\\S) (\\d *) (\\S *)。", LN)
如果 OBJ:
    NUM = INT ( OBJ。组 ( 2 ) ) + 25
```

```
LN = OBJ.group ( 1 ) + str ( num ) + obj.组 ( 3 ) + str ( num ) + obj.组 ( 5 ) + '\n'fout.写 ( ln )
fin.close ( ) ; fout.关闭 ( ) #
#
约正则表达式语法和KiCad的成分生成的详细信息:
#http://gskinner.com/RegExr/
#http://kicad.rohrbacher.net/quickLib.php
```

1. 在将两个组件合并为一个组件时，有必要使用Eeschema的库编辑器移动第一个组件，以便第二个组件不会在它上面移动。您将在下面找到最终的.lib文件及其在Eeschema中的表示。

\* .lib文件的内容

```
EESchema-LIBRARY版本2.3
#encoding utf-8
#COMP
DEF COMP U 0 40 YY 1 FN
F0"U"-1800 -100 50 HVC CNN
F1"COMP"-1800 100 50 HVC CNN
画
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
ENDDDEF
#End图书馆
```



1. 这里介绍的Python脚本是一个非常强大的工具，用于处理引脚号和引脚标签。但是，请注意，它的所有功能都来自于神秘但却非常有用的正则表达式语法：<http://gskinner.com/RegExr/>。

## 8.制作组件封装

与其他EDA软件工具不同，其中一种类型的库包含原理图符号和足迹变化，KiCad .lib文件包含原理图符号，而.kicad\_mod文件包含足迹。Cypcb用于将脚印映射到符号。

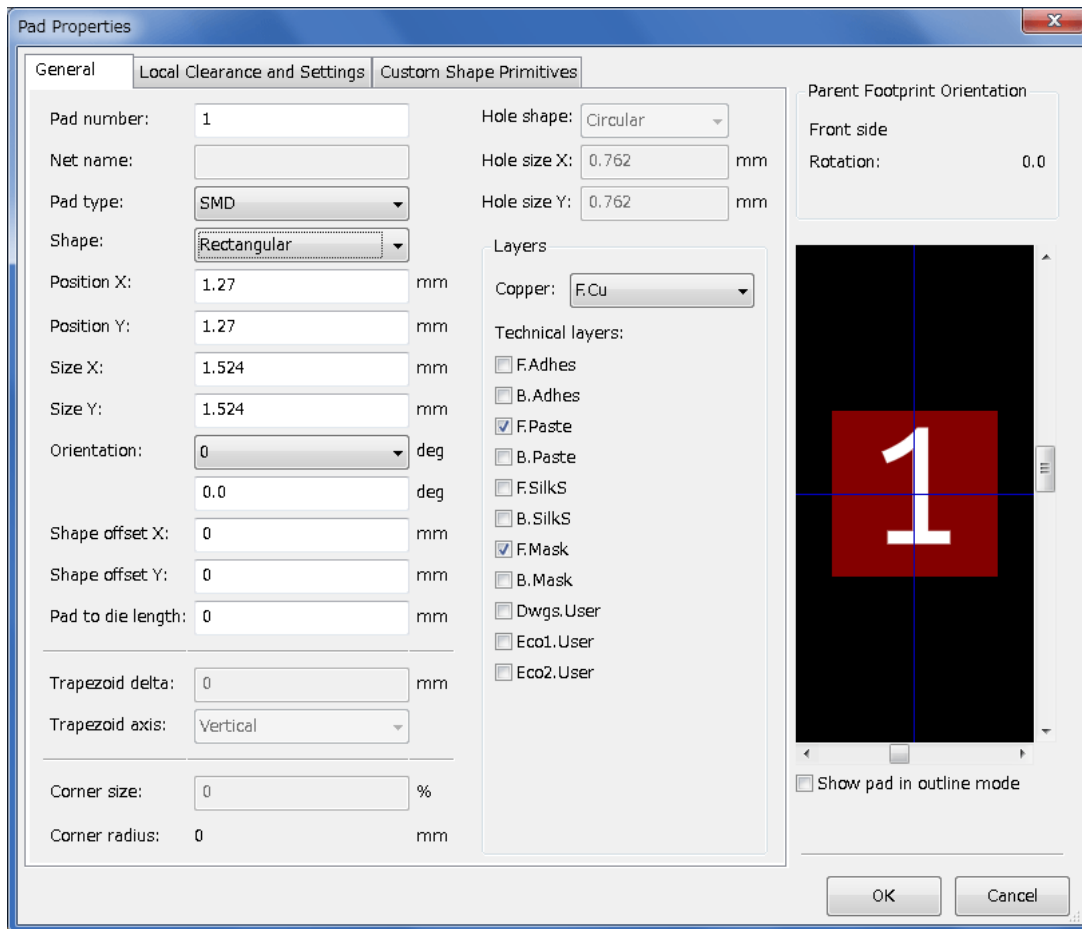


至于`.lib`文件，`.kicad_mod`库文件是可以包含从一个到多个部分的任何文本文件。

有一个广泛的足迹库与KiCad，但有时您可能会发现您需要的足迹不在KiCad库中。以下是在KiCad中创建新PCB封装的步骤：

## 8.1. 使用Footprint Editor

1. 从KiCad项目经理开始使用`Pcbnew`工具。单击顶部工具栏上的“**打开封装编辑器**”图标。这将打开`Footprint Editor`。
2. 我们要保存新的足迹`MYCONN3`在新的封装库`myfootprint`。在`tutorial1 / project`文件夹中创建一个新文件夹`myfootprint.pretty`。单击首选项 → 足迹库管理器，然后按**附加库**按钮。在表格中，输入“myfootprint”作为昵称，输入“\${KIPRJMOD}/myfootprint.pretty”作为库路径，并输入“KiCad”作为插件类型。按“确定”关闭“PCB库表”窗口。单击顶部工具栏上的“**选择活动库**”图标。选择`myfootprint`库。
3. 单击顶部工具栏上的“**新建足迹**”图标。键入`MYCONN3`的足迹名。在屏幕中间将出现`MYCONN3`标签。在标签下你可以看到`REF *` 标签。右键单击`MYCONN3`并将其移至`REF *`上方。右键单击`REF _ *`，选择“**编辑文本**”并将其重命名为`SMD`。将显示值设置为不可见。
4. 选择右侧工具栏上的“**添加垫**”图标。单击工作表以放置垫。右键单击新打击垫，然后单击“**编辑打击垫**”。你也可以用[c]。



5. 将`Pad Num`设置为1，将`Pad Shape`设置为`Rect`，将`Pad Type`设置为`SMD`，将`Shape Size X`设置为0.4，将`Shape Size Y`设置为0.8。单击确定。再次单击“**添加垫**”并再放置两个垫。
6. 如果要更改网格大小，请右键单击 → **网格选择**。在放下组件之前，请务必选择合适的网格尺寸。
7. 将`MYCONN3`标签和`SMD`标签**移开**，使其看起来像上图所示。
8. 放置垫时，通常需要测量相对距离。将光标放在您想要相对坐标点  $(0,0)$  的位置，然后按空格键。移动光标时，您将看到光标在页面底部位置的相对指示。可以随时按空格键以设置新原点。
9. 现在添加一个足迹轮廓。单击右侧工具栏中的“**添加图形线**”或“**多边形**”按钮。绘制组件周围连接器的轮廓。
10. 单击顶部工具栏上的“**活动库中的保存封装**”图标，使用默认名称`MYCONN3`。

## 9. 注意KiCad项目文件的可移植性

您需要将哪些文件发送给某人才能完全加载和使用您的KiCad项目？

当您有一个KiCad项目与某人共享时，重要的是原理图文件`.sch`，板文件`.kicad_pcb`，项目文件`.pro`和网表文件`.net`与原理图零件文件`.lib`一起发送。和脚印文件`.kicad_mod`。只有这样，人们才能完全自由地修改原理图和电路板。

使用KiCad原理图，人们需要包含符号的`.lib`文件。需要在`Eeschema` 首选项中加载这些库文件。另一方面，使用板（`.kicad_pcb`文件），脚印可以存储在`.kicad_pcb`文件中。您可以向某人发送`.kicad_pcb`文件，而不是其他任何内容，他们仍然可以查看和编辑该板。但是，当他们想要从网表加载组件时，脚本库（`.kicad_mod` 文件）需要存在并加载到`Pcbnew`首选项中，就像原理图一样。此外，有必要加载`.kicad_mod`中的首选项文件`Pcbnew`为了让这些足迹出现在`Cupcb`中。

如果有人向您发送了一个带有足迹的`.kicad_pcb`文件，您可以在另一个电路板中使用，您可以打开足迹编辑器，从当前电路板加载足迹，并将其保存或导出到另一个足迹库中。您也可以从导出所有的脚印`.kicad_pcb`文件一次通过**Pcbnew** → 文件 → 存档 → 脚印 → 创建足迹存档，这将创建一个新的`.kicad_mod`文件有所有主板的脚印。

最重要的是，如果PCB是你想要分发的唯一东西，那么电路板文件`.kicad_pcb`就足够了。但是，如果您想让人们完全能够使用和修改原理图，其组件和PCB，强烈建议您压缩并发送以下项目目录：

```
教程1 /
| - tutorial1.pro
| - tutorial1.sch
| - tutorial1.kicad_pcb
| - tutorial1.net
| - 图书馆/
| | - myLib.lib
| | - myOwnLib.lib
| \ - myQuickLib.lib
|
| - myfootprint.pretty /
| \ - MYCONN3.kicad_mod
|
\ - gerber /
  | - .....
  \ - .....
```

## 10.有关KiCad文档的更多信息

这是KiCad中大多数功能的快速指南。有关更详细的说明，请参阅可通过每个KiCad模块访问的帮助文件。单击**帮助** → **手动**。

KiCad为其所有四个软件组件提供了一套非常好的多语言手册。

所有KiCad手册的英文版都随KiCad一起发布。

除了手册之外，KiCad还随本教程一起发布，该教程已被翻译成其他语言。本教程的所有不同版本均可免费分发所有最新版本的KiCad。本教程以及手册应与您的KiCad版本一起打包在您指定的平台上。

例如，在Linux上，典型位置位于以下目录中，具体取决于您的确切分布：

在/ **usr** /共享/ **DOC** /中的kicad /帮助/ **EN** /  
在/ **usr** /本地/共享/ **DOC** /中的kicad /帮助/ **EN**

在Windows上它位于：

<安装目录> / **share** / **doc** / **kicad** / **help** / **en**

在OS X上：

/ **Library** / **Application Support** / **kicad** / **help** / **en**

### 10.1。网上的KiCad文档

最新的KiCad文档可在Web上以多种语言提供。

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

---

上次更新时间2018-07-18 16:45:41 UTC