

KiCad 入

The KiCad Team

Table of Contents

KiCad 介	2
下 和安装 KiCad	2
支持	2
KiCad 工作流程	3
概述	3
向前和向后批注	5
使用 KiCad	6
快捷	6
制 路原理	8
使用 Eeschema	8
KiCad 的 接	18
布局印刷 路板	20
使用 Pcbnew	20
生成 Gerber 文件	27
使用 Gerbview	27
使用 FreeRouter 自 布	27
在 KiCad 中向前批注	29
在 KiCad 中制作原理 符号	30
使用元件 器	30
出, 入和修改 元件	31
使用 quicklib 制作原理 元件	32
制作高引脚数的原理 元件	32
制作元件封装	35
使用封装 器	35
于 KiCad 工程文件的可移植性的注意事	37
有 KiCad 文档的更多信息	38
网上的 KiCad 文档	38

版

本文件的版 © 2010-2021 由下列 献者 有。您可以根据 GNU 通用公共可
(<http://www.gnu.org/licenses/gpl.html>) 第 3 版或更高版本, 或知 共享署名 可
(<http://creativecommons.org/licenses/by/3.0/>) 第 3.0 版或更高版本的条款 布它和/或修改它。

本指南中的所有商 均属于其合法所有者。

献人

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

翻 人

taotieren <admin@taotieren.com>, 2019, 2020, 2021.

Telegram 体中文交流群: https://t.me/KiCad_zh_CN

反

KiCad 目 迎与本 件或其文档相 的反 告和建 于如何提交反 意 或 告 的更多信息, 参
<https://www.kicad.org/help/report-an-issue/> 的 明

KiCad 介

KiCad 是一款用于 建 子 路原理 和印刷 路板 (PCB) 的开源 件套件。KiCad 支持 合 计工作流程, 其中原理 和 相 的 PCB 一起 计, 也支持特殊用途的独立工作流程。KiCad 包括一些帮助 路和 PCB 计 的 用程序, 包括用于 确定 路 构 气属性的 PCB 计算器、用于 制造文件的 Gerber 器和用于 路行 的集成 SPICE 模 器。

KiCad 可在所有主要的操作系 和广泛的计算机硬件上运行。它支持多达 32 个 的 PCB, 适合 建各种 的 计。KiCad 是由世界各地的 件和 气工程 成的志愿者 开 的, 其使命是 建适合 计 的免 和开源的 子 计 件。

KiCad 的最新文档可在 <https://docs.kicad.org>。

下 和安装 KiCad

KiCad 可在 多操作系 上运行, 包括微 Windows、苹果 macOS 和 多主要的 Linux 行版。

你可以在 <https://www.kicad.org/download/> 找到最新的 明和下 接。些 明没有包括在本手册中, 因 它 可能随着操作系 更新的 布而改

IMPORTANT

根据 [KiCad 定 布政策](#), KiCad 的 定 布会定期 行。新的功能正在不断地被添加到开 分支中。如果你想利用 些新功能, 并通 它 来提供帮助, 下 你的平台的最新夜 构建包。夜 构建可能会引入一些 如文件 坏、生成不良 Gerbers 等, 但 KiCad 开 的目 是在新功能开 期 尽可能保持开 分支的可用性。

支持

如果您有想法, 或 或者您只是需要帮助:

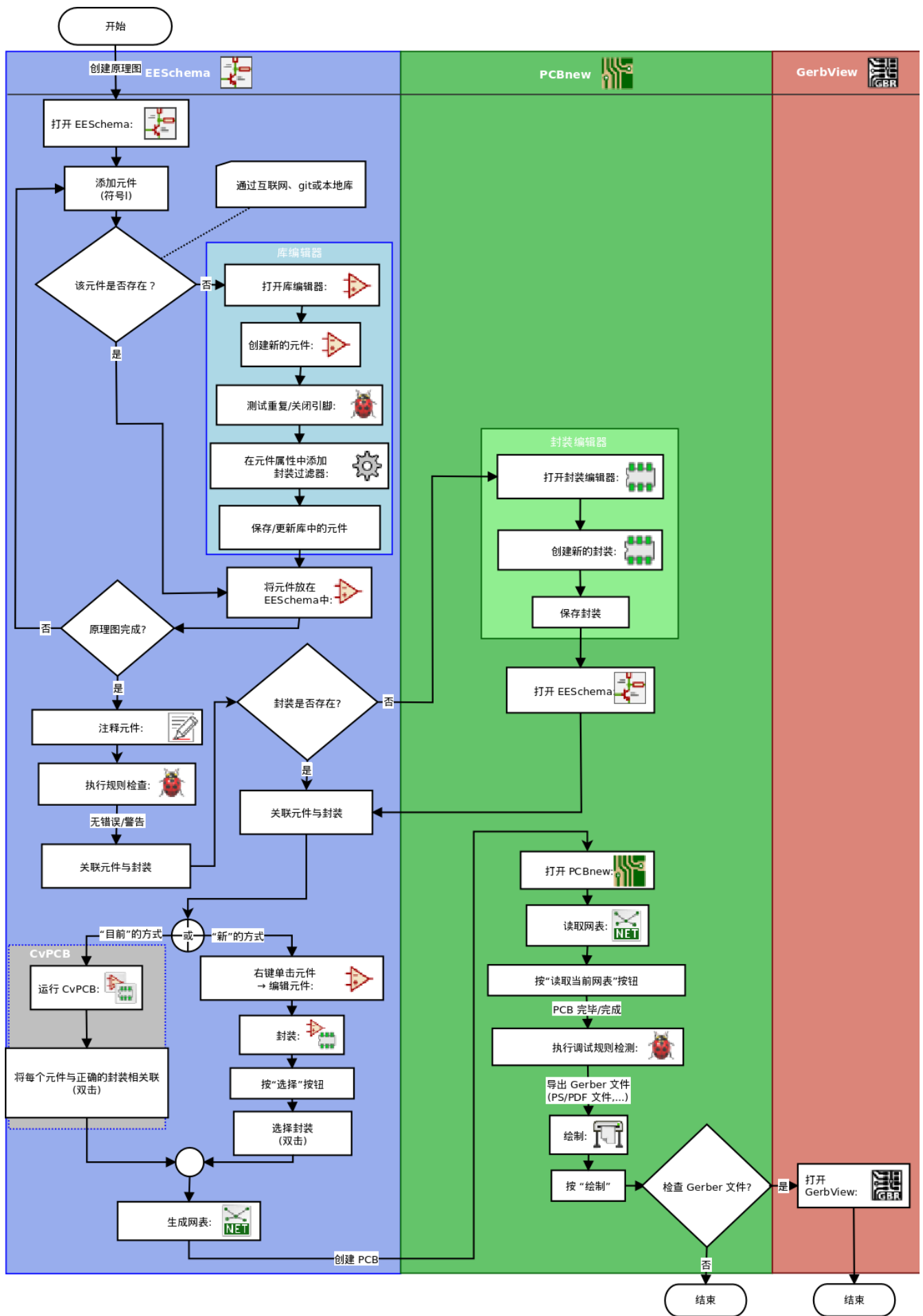
- [KiCad官方用](#) 是一个与其他 KiCad 用 系并 得帮助的好地方。
- 加入我 在 [Discord](#) 或 [IRC](#) 的社区, 与用 和开 者 行 讨
- 看 KiCad 网站, 了解 KiCad 社区制作的 [学 源](#)。

KiCad 工作流程

概述

典型的 KiCad 工作流程由两 主要任 成：制原理 和布局 电路板。两 任 都需要一个原理 元件 和一个 PCB 封装
KiCad 包括 多元件和封装，也有 建新元件和封装的工具。

在下 中，你可以看到一个代表 KiCad 工作流程的流程 流程 解 了你需要采取哪些步 以及按什么 序 行。在适用
的情况下， 了方便起 添加了



有 建元件的 信息, 《make-schematic-symbols-in-kicad,Making schematic symbols 》。有 如何 建新封装的信息, 参 《make-component-footprints,Making component footprints 》。

向前和向后批注

完成 子原理 制后, 下一步就是将其 到 PCB。通常, 可能需要添加其他元件, 将部件更改 不同的大小, 行网重命名等。 可以通 两种方式完成: 向前批注或向后批注。

向前批注是将原理 信息 送到相 PCB 布局的 程。 是一个基本功能, 因 您必 至少 行一次才能将原理 初始 入 PCB。之后, 向前批注允 向 PCB 送增量原理 更改。有 向前批注的 信息, 参 forward-annotation-in-kicad, Forward Annotation 》部分。

向后批注是将 PCB 布局更改 送回其相 原理 的 程。向后批注的两个常 原因是 交 和引脚交 在 些情况下, 存在功能相同的 或引脚, 但可能 在布局期 存在 精确 极或引脚的 有力的情况。一旦在 PCB 中做出 更改将被 推回原理

使用 KiCad

快捷

KiCad 有两种相 但不同的快捷：快捷 (accelerator keys) 和 (hotkeys)。两者都用于通 使用 而不是鼠 来 行命令以提高使用 KiCad 的生 效率。

快捷

快捷 的效果与 菜 或工具 的效果相同: 将 入 命令, 但在 鼠 左 之前不会 生任何情况。如果要 入命令模式, 但不希望立即 行任何操作, 使用快捷

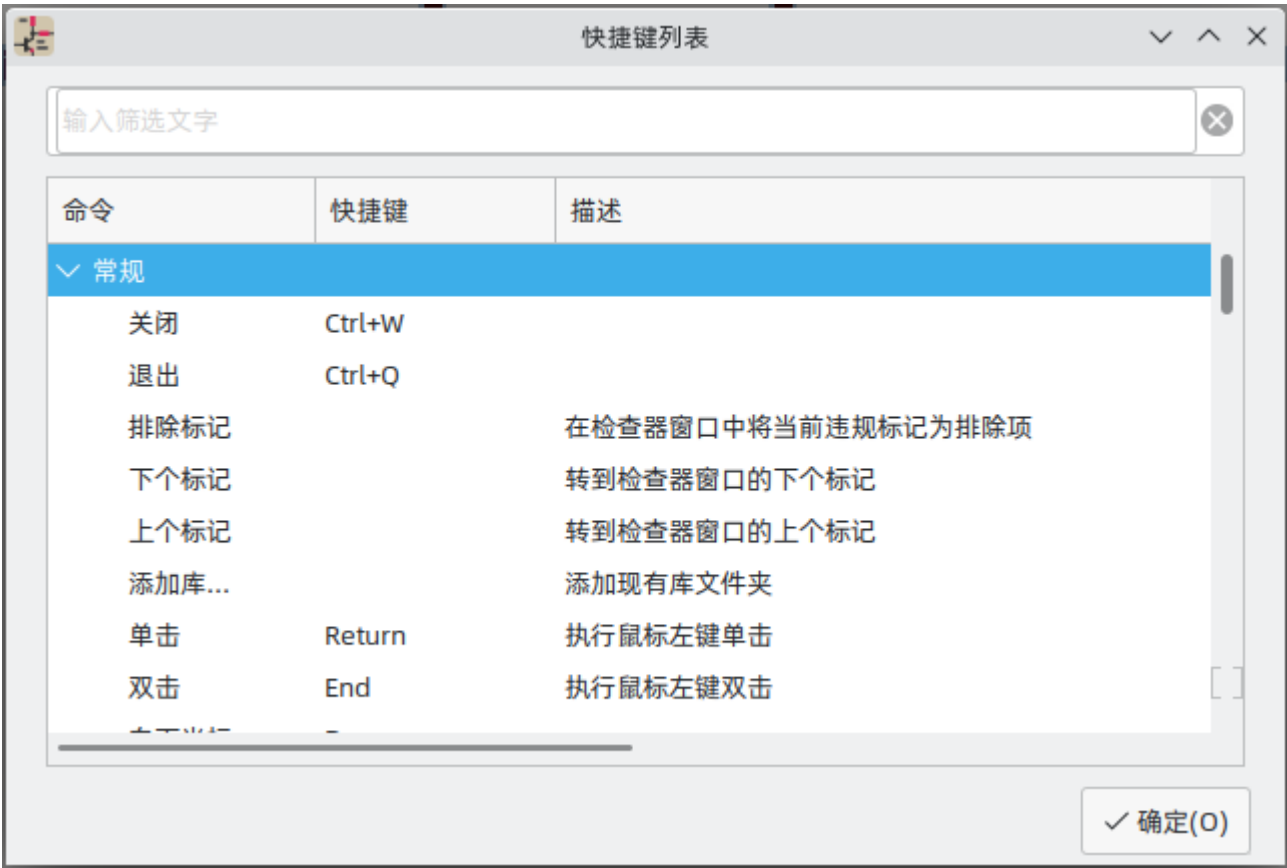
快捷 示在所有菜 窗格的右：



快捷

等于加速器 加上鼠 左 使用 会立即在当前光 位置后 命令。使用 可以在不中断工作流的情况下快速更改命令。

要 看任何 KiCad 工具中的 到_帮助 → 列出 或按_Ctrl+ F1：



您可以从 **偏好置** → **菜** 的分配, 并 **入**或 **出**它

NOTE

在本文档中, 用括号表示, 如下所示: [a]。如果看到 [a], 只需在 上 入 *a* 即可。

例子

考 在原理 中添加走 的 示例。

要使用快捷 , 按 *Shift + W* 用 **添加** 命令 (注意光 将会更改)。接下来, 左 所需的 开始位置, 开始 制
使用 只需按 [w] 将立即从当前光 位置开始。




制 路原理

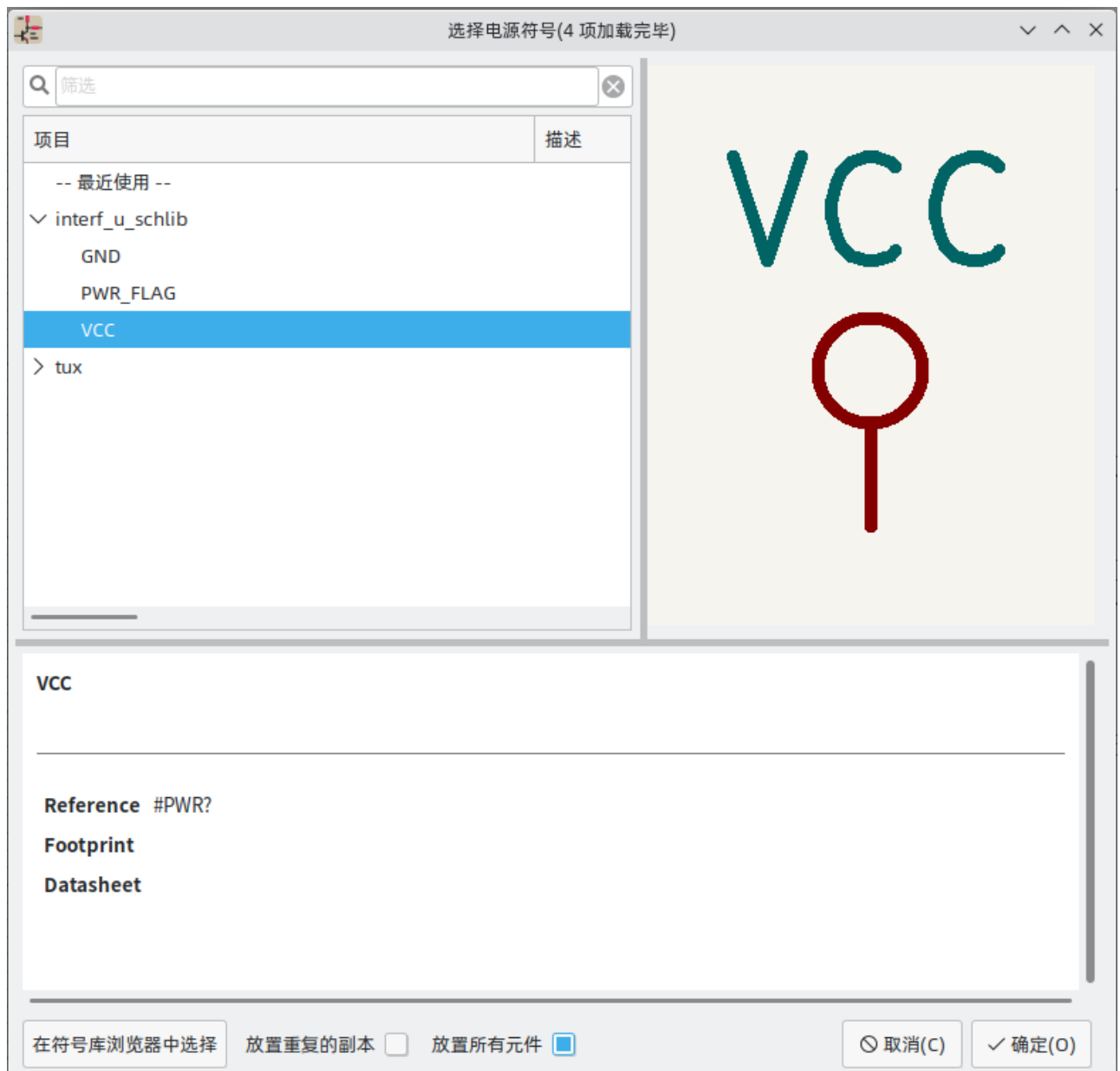
在本 中,我 将学 如何使用 KiCad 制 路原理

使用 Eeschema

1. 在 Windows 下运行 kicad.exe。在 Linux 下,在 端中 入 kicad。您 在位于 KiCad 工程管理器的主窗口中。从 里您可以 八个独立的 件工具 : Eeschema, 原理 器, Pcbnew, PCB 封装 器, Pcbnew, GerbView, Bitmap2Component, PCB 计算器和 Pl 器 参 工作流程 了解如何使用主要工具。



2. 建一个新工程 : 文件 → 新建 → 工程。将工程文件命名 教程1。工程文件将自 采用 展名 .pro。 框的确切 外 取决于使用的平台,但 有一个用于 建新目 的 框。除非您已有 用目 否 保持 状 您的所有工程文件都将保存在那里。
3. 从 建原理 开始。开始原理 器 _eeschema_ 。它是左 的第一个按
4. 部工具 上的 面 置 。置适当的 尺寸 ('A4', '8.5x11'等)并 入 教程1。如有必要,您将在此 看到可以 入更多信息。 确定。此信息将填充右下角的原理 表。使用鼠 放大。保存整个原理 文件 - > 保存
5. 我 在将放置第一个元件。 右 工具 中的 放置符号 。您也可以按 添加符号 [a]。
6. 原理 工作表的中 部分。屏幕上将出 符号窗口。我 要放一个 阻器。搜索/ Resistor的 R。您可能会注意到 阻器上方的 此 是元件所在 的名称, 是一个非常通用且有用的

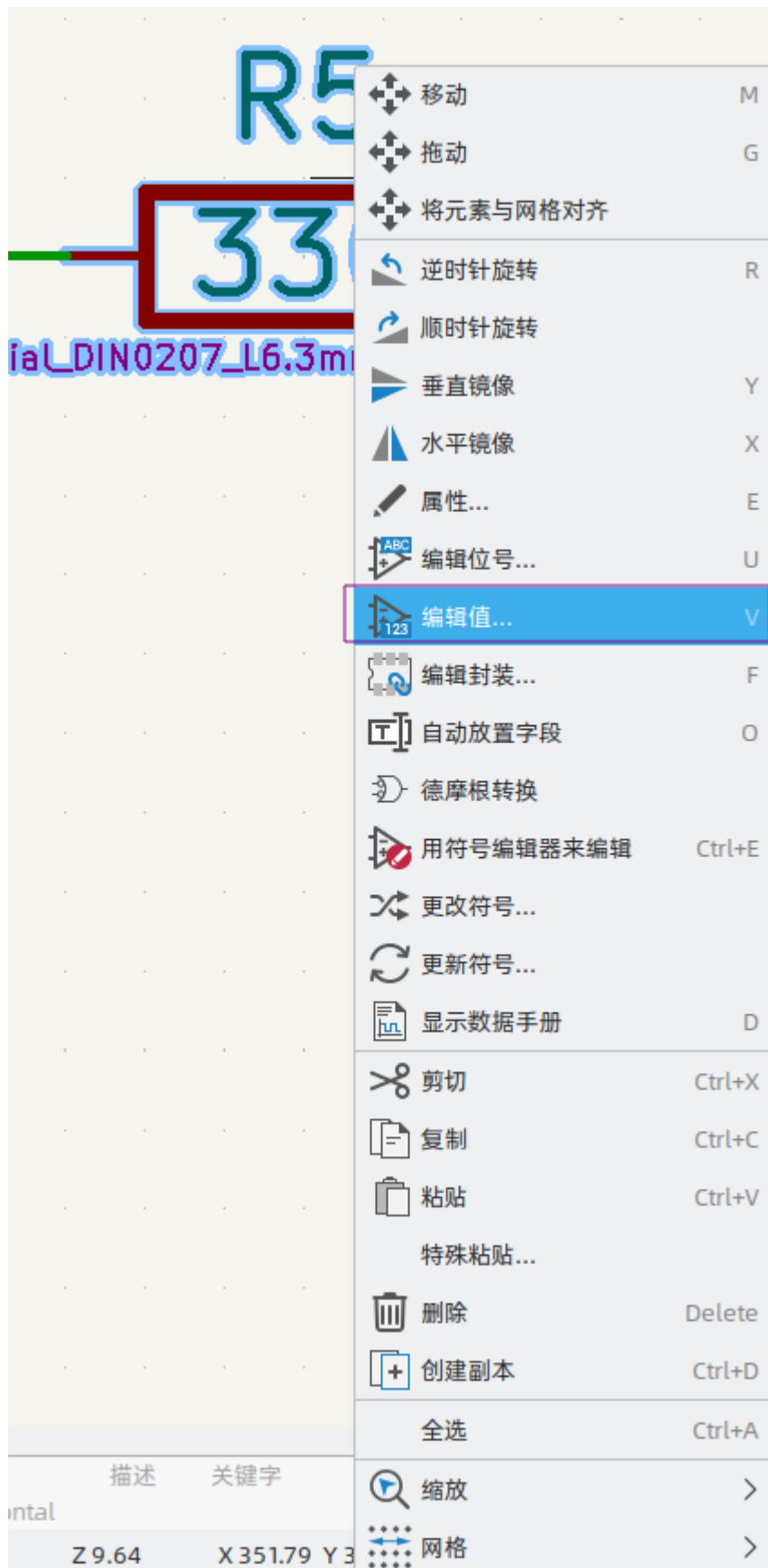


7. 双击它。将符号窗口。通过元件在原理图中的位置将其放置在原理图中。
8. 放大 可放大元件。也可以使用鼠标进行放大和缩小。按下 (中央) 鼠标按钮 将可以水平和垂直平移。
9. 将鼠标停在元件 *R* 上, 然后按 [r]。元件 旋转 您不需要 元件来旋转它。

NOTE

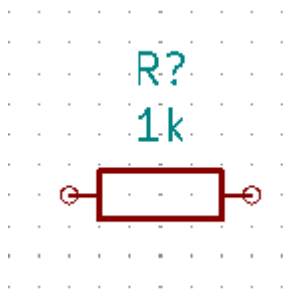
有 如果您的鼠标也在其他地方, 会出现一个菜单 您将在 KiCad 中 常看到澄清 菜单; 它允许彼此重叠的对象。在 种情况下, 如果出现 菜单 告诉 KiCad 您要 _符号... R..._ 行操作。

10. 右键单击元件的中, 然后 属性> 。您可以通过将鼠标停在元件上并按 [v] 来 相同的 果。或者, 按下[e] 将打开到更通用的 属性窗口。注意, 下面的右键菜单 显示所有可用操作的



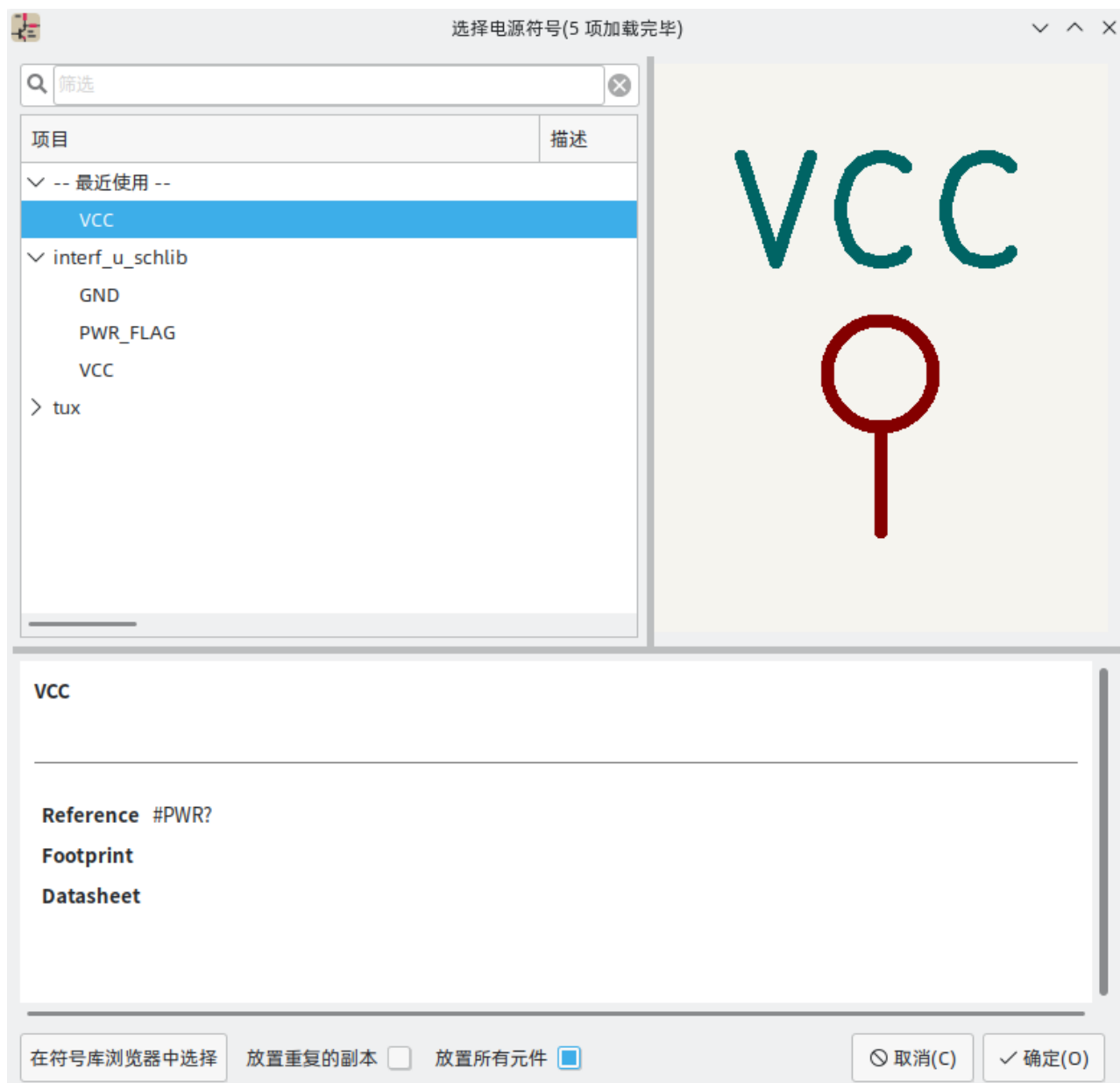
11. 将出 字段窗口。用 $1k$ 替 当前 R 。 确定。

NOTE | 不要更改参考字段 ($R?$)，稍后会自 完成。 阻器上方的 在 $1k$ 。



12. 要放置另一个 阻, 只需要 示 阻的位置。符号 窗口将再次出

13. 您之前 的 阻 在位于 史列表中, 示 R 。 确定并放置元件。



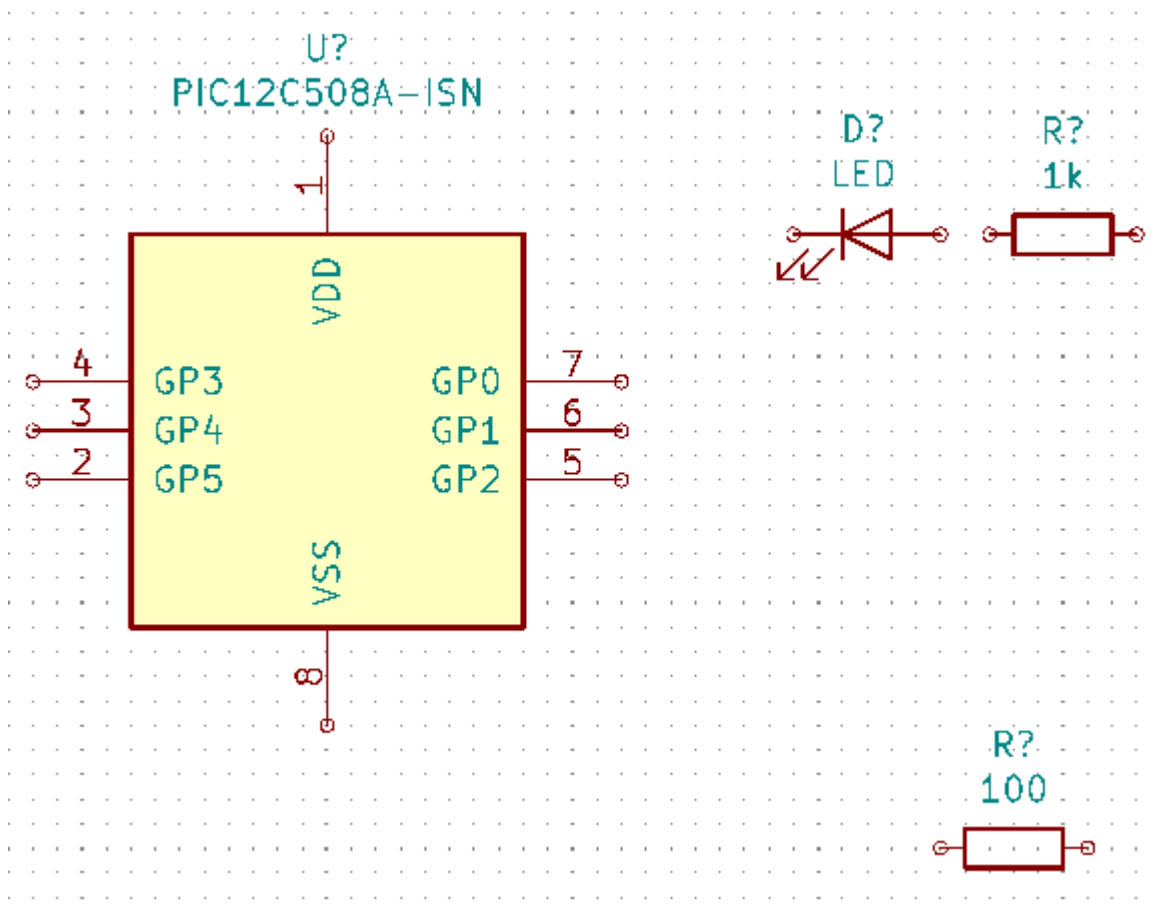
14. 如果您想要 除一个元件, 右 元件, 然后 " 除"。或者, 您可以将鼠 停在要 除的元件上, 然后按下 上的 [delete]

15. 您 可以通 将鼠 停在原理 上并按[C]来 制已 存在于原理 上的元件。 要放置新 制元件的位置。

16. 右 第二个 阻。 拖 。重新定位元件并左 以放下。将鼠 停在元件上并按[g]可以 相同的功能。[r] 将旋 元件, 而[x]和[y]将 其x 或y 翻

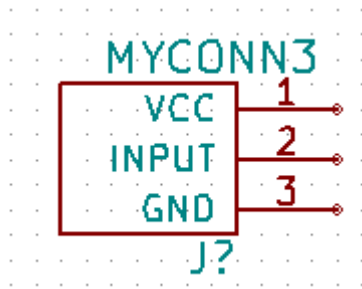
右 → *移 *或[m]也是一个有价 移 任何 西，但最好只将它用于元件 和元件尚未 接。我 稍后会看到 什么会


17. 将鼠 停在第二个 阻上，然后按[V] 阻。将 R 替 100。可以使用 Ctrl+Z 撤消任何 操作。
18. 更改网格大小。您可能已 注意到，在原理 表上，所有元件都被捕捉到大 距网格上。您可以通 右 → 网格 松更改网格的大小。通常，建 使用50.0mils 的网格作 原理 表。
19. 我 将从 中添加一个可能未在默认工程中配置的元件。在菜 中， 偏好 置 → 管理符号 。在符号 窗口中，您可以看到两个 卡：全局 和工程 有 每个都有一个符号表 文件。要使 .lib文件) 可用，它必 位于其中一个符号表 文件中。如果文件系 中有 文件但尚未提供， 可以将其添加到其中一个符号表 文件中。了 我 在将添加一个已 可用的
20. 工程 用表。 表下方的文件 器按 您需要找到计算机上安装官方 KiCad 的位置。 找包含一百个 .dcm 和 .lib 文件的 目 _C:\Program Files (x86)\KiCad\share\ (Windows)和 /usr/share/kicad/library/ (Linux)。找到目 后， 并添加 MCU_Microchip_PIC12.lib 并 窗口。它将添加到列表的末尾。在 其昵 称并将其更改 microchip_pic12mcu。 确定 符号 窗口。
21. 重 添加元件步 但 次 microchip_pic12mcu 而不是 并 PIC12C508A-15N 元件。
22. 将鼠 停在微控制器元件上。注意，[x] 和 [y] 再次翻 元件。保持符号 Y 像，使引脚 G0 和 G1 指向右
23. 重 添加元件步 次 并从中 LED 元件。
24. 原理 上的所有元件，如下所示。



25. 我 在需要 我的3 接器 建原理 元件“MYCONN3”。您可以跳 到 “make-schematic-symbols-in-kicad, 在KiCad中制作原理 符号”的部分，了解如何从 开始制作 元件，然后返回本 使用 板。
26. 您 在可以放置新制作的元件。按 [a] 并 myLib 中的 MYCONN3 元件。

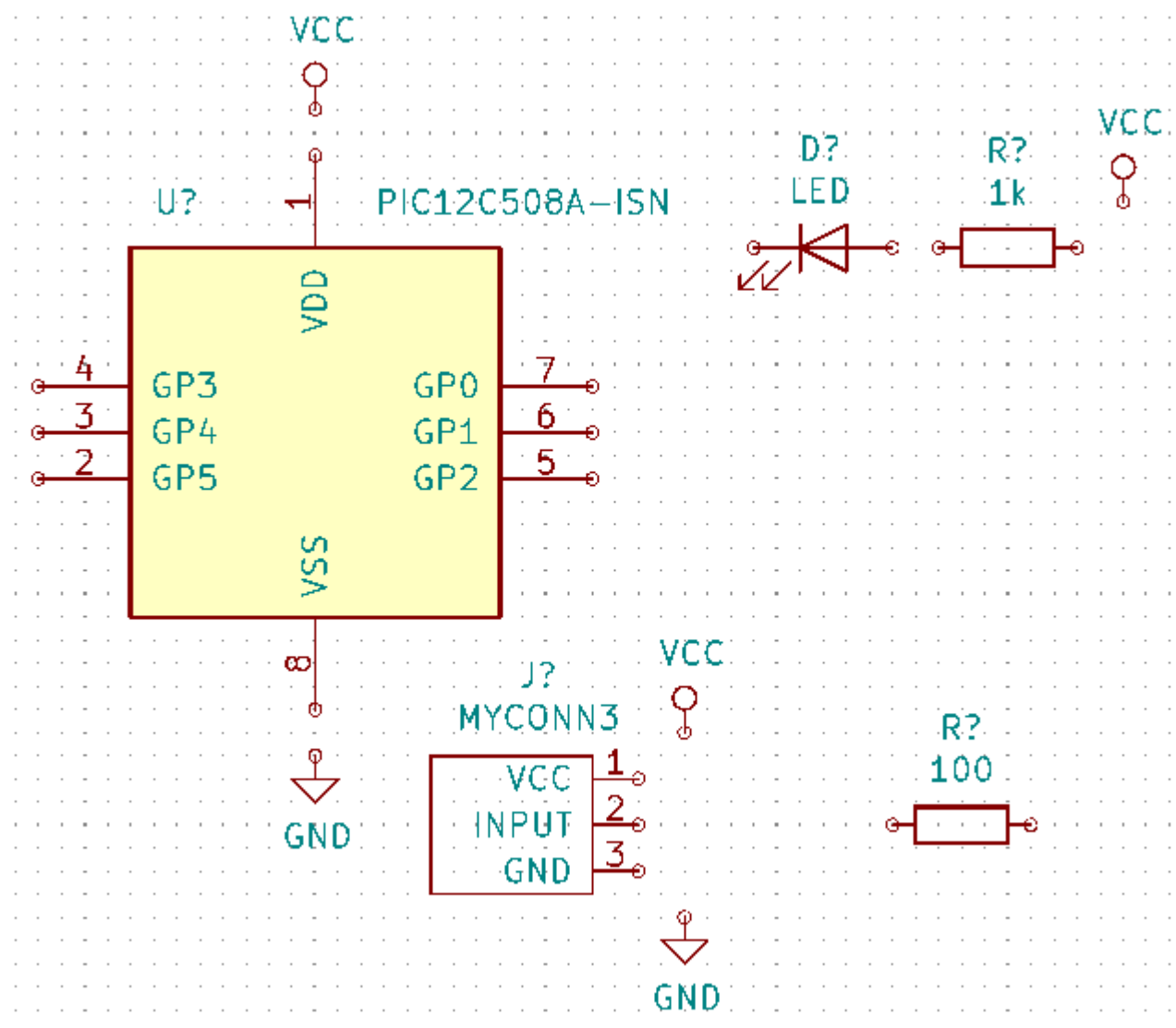
27. 元件 符J?将出 在MYCONN3 下。如果要更改其位置， 右 J?然后 移 字段（相当于[m]）。在 行 此操作之前/之后放大可能会有所帮助。重新定位J? 在如下所示的元件下。 可以随意移 多次。




28. 是 候放置 源和接地符号了。 右 工具 上的 放置 源端口按 。或者，按[p]。在元件 窗口中，向下 并从 源 中 VCC。 确定。

29. 1k 阻的引脚上方以放置 VCC 部件。 微控制器 VDD 上方的区域。在 元件 史 部分中， VCC 并将 其放在 VDD 引脚旁 再次重 添加 程，并将 VCC 部分放在 MYCONN3 的 VCC 引脚上方。如果需要，可以移 引用和

30. 重 添加引脚步 但 次 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 放置另一个 GND 符号。您的原理 在 如下所示：



31. 接下来，我 将 接所有元件。 右 工具 上的 放置 。

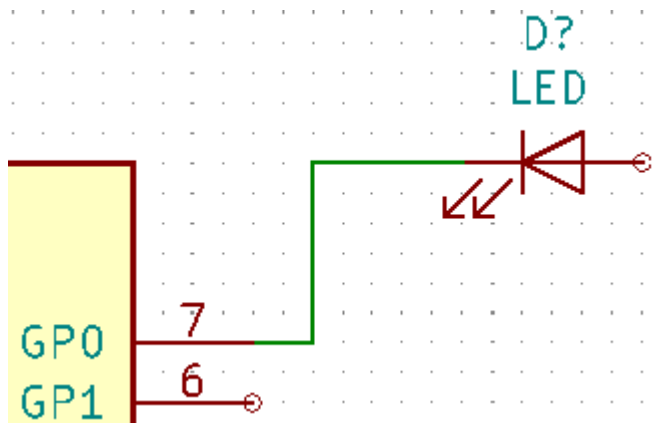
NOTE

小心不要 放置 ，它位于此按 的正下方，但 条 粗。《bus-connections-in-kicad, KiCad 中的 接》部分将解 如何使用 部分。

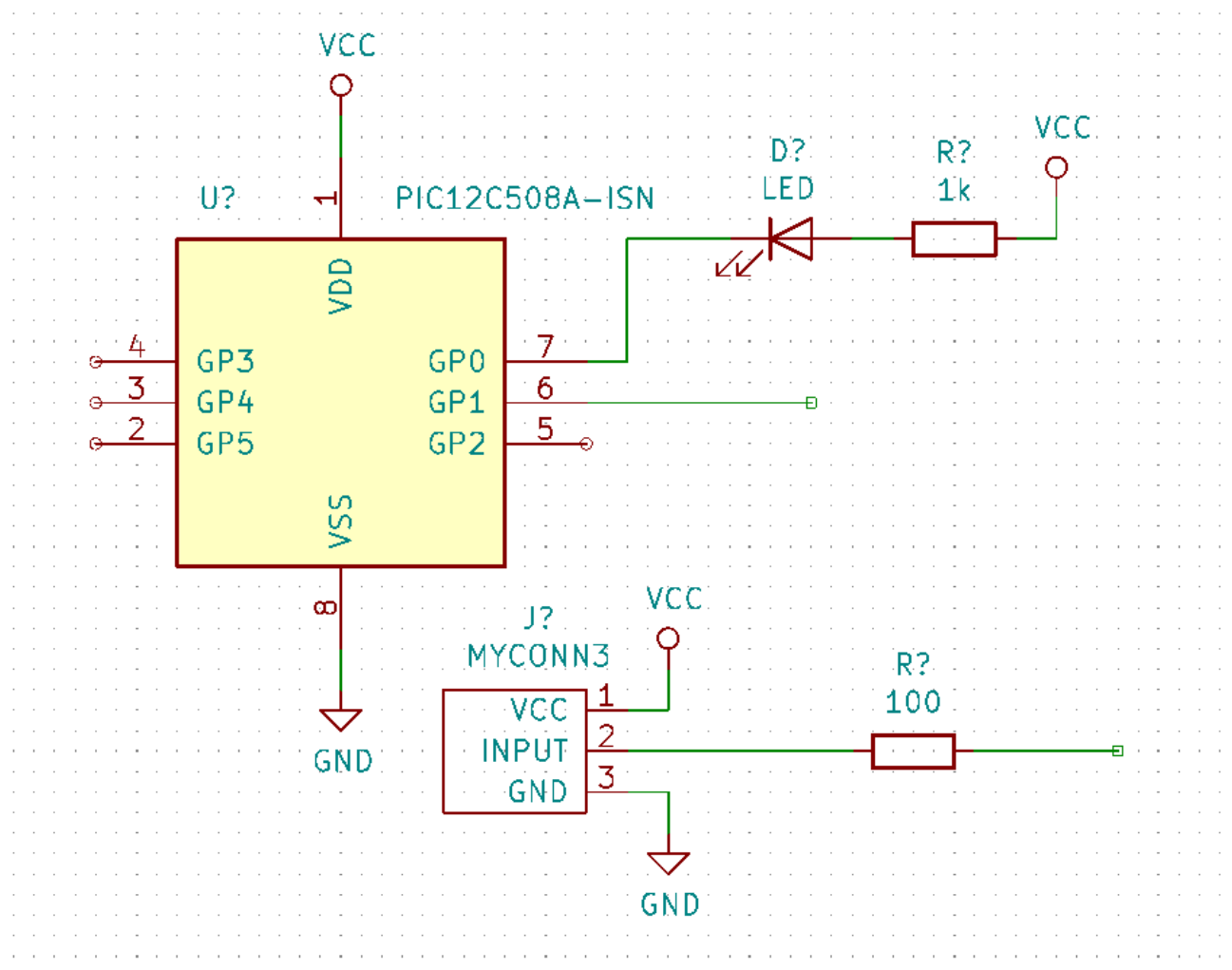
- 微控制器引脚7末端的小 圈，然后 LED 引脚1上的小 圈。在 制 一次以 建拐角。您可以在放置 接 放大。

NOTE

如果要重新定位 元件，重要的是使用[g]（抓取）而不是[m]（移 使用抓取将保持 接。如果您忘 了如何移 元件， 看步 24。

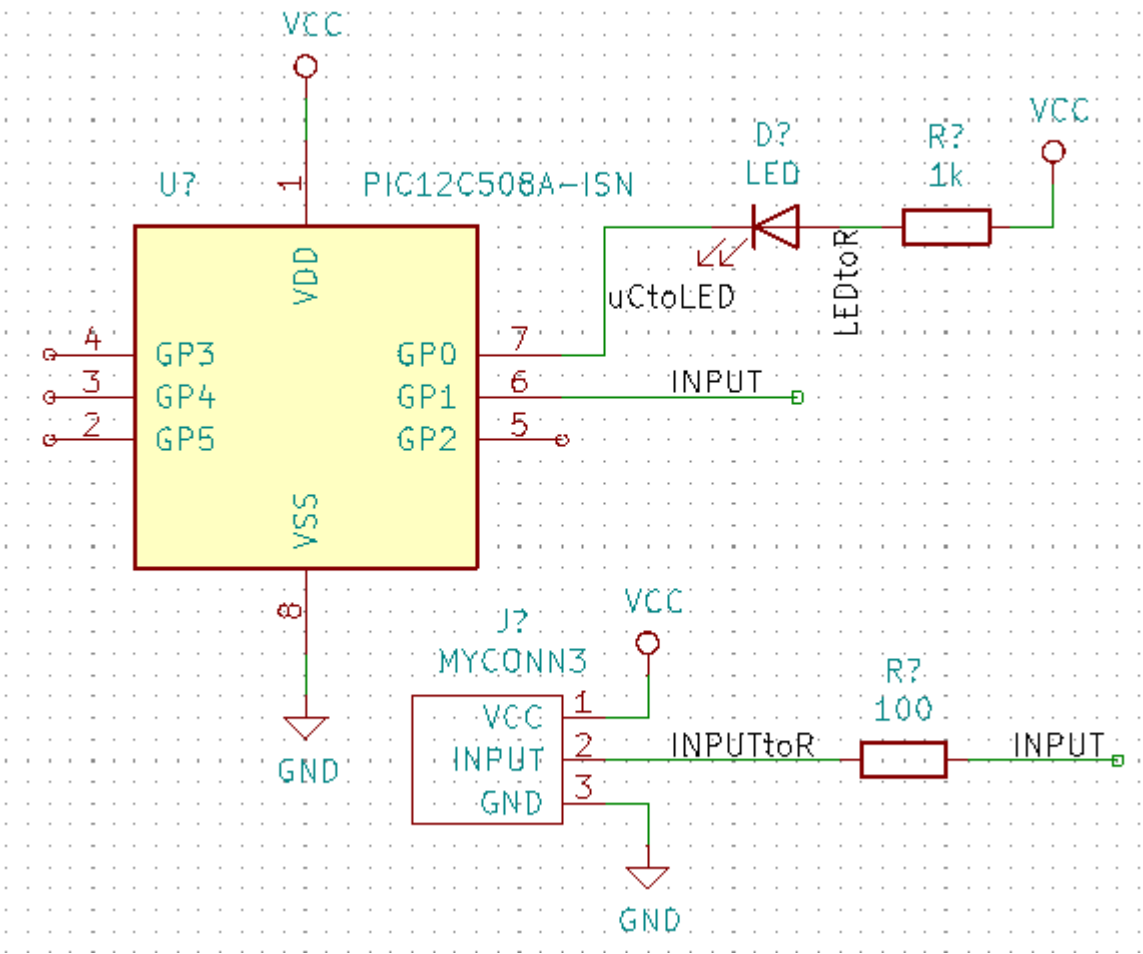


- 重 此 程并 接所有其他元件，如下所示。要双 止 当 接 VCC 和 GND 符号 接触 VCC 符号的底部和 GND 符号的中 部。 参 下面的截

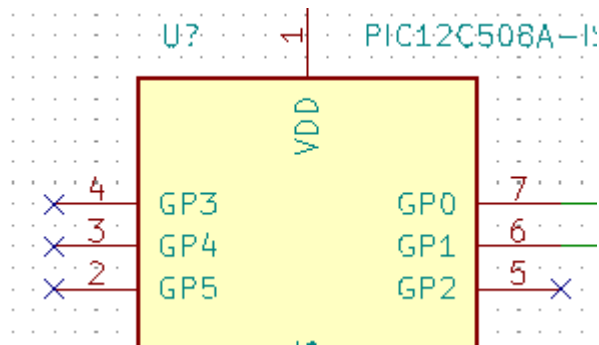



我在将考使用建立接的另一种方法。通过右工具上的放置网A来网工具。你也可以用[]。

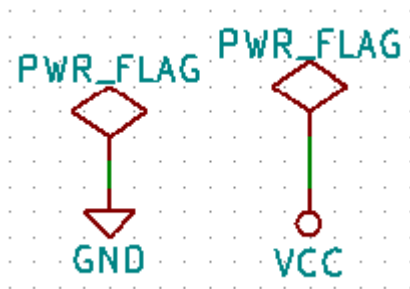
- 35. 接到微控制器引脚6的 中 将此 命名“INPUT”。 仍然是一个独立的工程，您可以移 旋 和 除。 的小 矩形必 正好在 或引脚上才能使 生效。
- 36. 遵循相同的 序,在100欧姆 阻器右 的 上放置另一个 并同 命名“INPUT”。此 存在两个 具有相同的名称, 在pic的引脚6和100欧姆 阻之 会 建一个不可 的 接。在 的 计中 接走 ,逐条 制走 将使整个原理 得混乱。此 就可以使用 接。要放置 ,您不一定需要走 ,您只需将同名的 接到引脚即可。
- 37. 也可用于 地 以用于提供信息。在 PIC 的引脚7上放置一个 入名称 uCtoLED。将 阻和 LED 之 的 命名 LEDtoR。将MYCONN3 和 阻之 的 命名 INPUTtoR。
- 38. 您不必 VCC和GND 因 是从它 所 接的 源 象中 含的。
- 39. 下面是最 的 果。






- 40. 我 在 理未 接的 当 KiCad 任何未 接的引脚或 都会 生警告。 了避免 些警告，您可以指示程序未 接的 是故意的，也可以手 将每个未 接的 或引脚 未 接。
- 41. 右 工具 上的 放置无 接 志 X 。 引脚2,3,4和5, X似乎表示缺少有 接是故意的。




42. 某些元件具有不可见的源引脚。您可以通过左侧工具上的  使其可见。如果遵守 VCC 和 GND 命名，隐藏的源引脚会自动连接。一般来说，你尽量不要制作隐藏的源引脚。
43. 在需要添加一个 *Power Flag* 来向 KiCad 表明源是从某个地方来的。按[a]并搜索源中的 *PWR_FLAG*。放置其中两个。将它连接到 GND 引脚和 VCC，如下所示。



NOTE 将避免典型的原理警告：_引脚 接到其他一些引脚但没有引脚来它。

44. 有时候在原理图里和那里写批注是件好事。要在原理图上添加批注，使用右侧工具上的  放置文本 **T**。
45. 在原理图中，所有元件都需要具有唯一标识符。事实上，我的多元件仍被命名 R? 是 J?。通过原理图工具上的  批注原理符号，可以自动完成符号分配。
46. 在批注原理图窗口中，使用整个原理图并批注按钮。注意所有?已被数字取代。每个标识符在原理图中都是唯一的。在我的例子中，它被命名 R1, R2, U1, D1 和 J1。
47. 我在将原理图的原理图工具上的  运行按钮。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。你有 0 个和 0 个警告。如果出现错误或警告，原理图中将出现一个小红色箭头指示错误或警告所在的位置。中 **BuildERC** 文件报告并再次按运行按钮以接收有更多的信息。

NOTE 如果您收到 **未找到默认编辑器** 警告，必须设置它的警告，将路径置 `c:\windows\notepad.exe` (windows) 或 `/usr/bin/gedit` (Linux)。

48. 原理图已完成。我可以在原理图中建立一个网表文件，我将添加每个元件的封装。通过原理图工具上的  生成网表按钮并保存在默认文件名下。

NOTE 在以前版本的 KiCad 中，网表是必要的。在最近的版本中，您可以忽略它，而是使用 *** 工具** *** → 从原理图更新 PCB**。如果您这样做，您必须首先指定符号封装。

49. 生成网表文件后，通过原理图工具上的  运行 **Cypcb**。如果出现丢失的文件窗口，忽略它并确定。

有很多方法可以 符号添加封装。


NOTE

- 右 符号 → **属性** → **封装** 双 符号, 或右 符号 → **属性** → **属性** → 封装
- 工具** → **符号字段** 在 Eeschema 的偏好 置中, 符号 器中的 示封装 并在 要放置的新符号 封装

50. Cvpcb 允 您将原理 中的所有元件与 KiCad 中的封装 接起来。中心的窗格 示原理 中使用的所有元件。在 里 D1。在右 窗格中, 您可以看到所有可用的占用空 此 向下 到 LED_THT : LED-D5.0mm 并双 它。

51. 右 窗格可能只 示可用封装的一个子集。点   和  以后用或禁用 些 器。

52. 于 U1 Package_DIP : DIP-8_W7.62mm 封装。 于 J1 Connector : Banana_Jack_3Pin 封装。 于 R1 和 R2 _Resistor_THT : R_Axial_DIN0207_L6.3mm_D2.5mm_P2.54mm_Verical_封装。

53. 如果您有 趣知道您 的封装是什么 的, 您可以 看所 封装  以 当前封装。

54. 完成以后, 可以通 点 * 文件 * → **保存原理** 或使用 用、保存原理 和 按 保存原理

55. 您可以 Cvpcb 并返回 Eeschema 原理 器。如果您没有将其保存在 Cvpcb 中, 文件 → **保存** 立即保存。再次 建网表。您的网表文件 已更新, 包含所有封装。注意, 如果您缺少任何 的占地面 需要制作自己的封装。 将在本文档的后 部分中解

NOTE


在每个符号都有封装。您可以使用 **工具** → **从原理** 更新PCB, 而不是网表和接下来的两个步 如果您 做, Pcbnew 将使用 原理 更新PCB 框中的 打开。 更新PCB。然后, 您可以按照本教程的 Pcbnew 部分中的 明 行操作。

56. 切 到 KiCad 工程管理器。您可以在文件列表中看到网表文件。

57. 网表文件描述了所有元件及其各自的引脚 接。网表文件 上是一个文本文件, 您可以 松地 或 写脚本。

NOTE

文件 (*lib) 也是文本文件, 它 也很容易 或 写脚本。

58. 要 建物料清 BOM 到 Eeschema 原理 器, 然后 部工具 上的 生成物料清 。默认情况 下, 没有 于活 的插件。您可以通 添加插件 按 添加一个。 要使用的 *.xsl 文件, 在 种情况下, 我 bom2csv.xsl。

Linux:

如果缺少 `xsltproc`，您可以下 并安装它：

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

于像 Ubuntu 的 Debian 派生 行版，或者

```
sudo yum install xsltproc
```

于 RedHat 派生的 行版。如果您不使用 两种 行版， 使用您的 行版 件包管理器命令来安装 `xsltproc` 件包。

`xsl`文件位于：`/usr/lib/kicad/plugins/`。

NOTE

Apple OS X:

如果缺少 `xsltproc`，您可以从 包含它的 Apple 站点安装 Apple Xcode 工具，或者下 并安装它：

```
brew install libxslt
```

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

Windows:

`xsltproc.exe` 和包含的 `xsl` 文件将分 位于_《KiCad 安装目 \bin_ 和 _《KiCad 安装目 \bin\scripting\plugins_。

所有平台：

您可以通 以下方式 取最新的 `bom2csv.xsl`：

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

KiCad 自 生成命令，例如：

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

您可能想要添加 展名，因此 将此命令行更改

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






按 帮助按 取更多信息。

59. 在按 生成。 文件（与工程同名）位于工程文件 中。使用 LibreOffice Calc 或 Excel 打开 *.`csv` 文件。将出入口窗口，按 OK。

您 在可以 到 PCB 布局部分， 将在下一 中介 但是，在 之前，让我 快速了解如何使用 接元件引脚。


KiCad 的 接

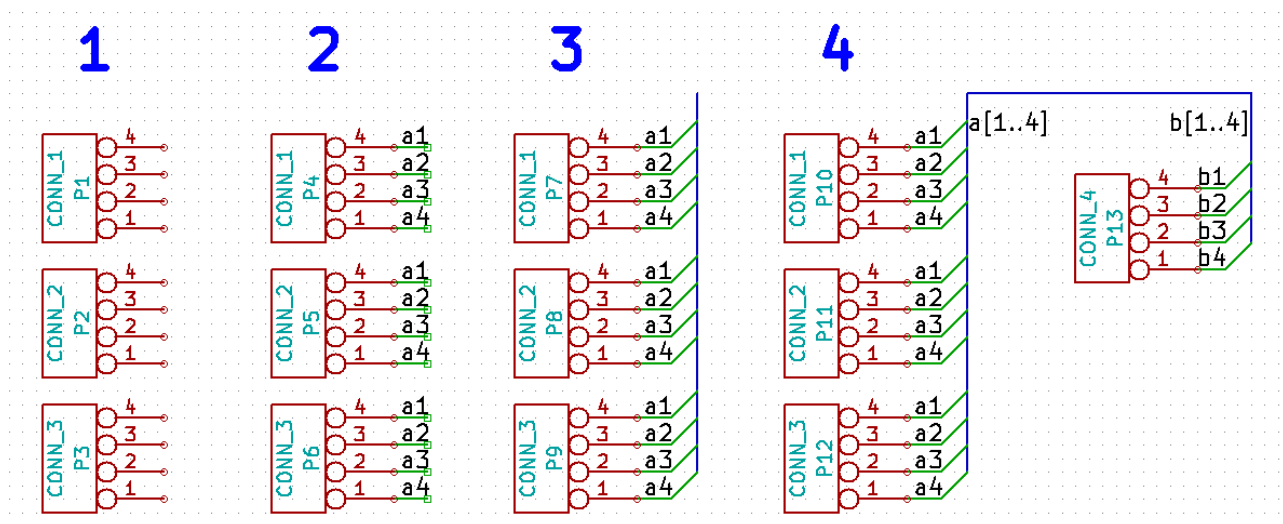
有 您可能需要将元件 A 的多个 序引脚与元件 B 的其他 序引脚 接。在 种情况下，您有两个 我已 看到的 方法或使用 接。让我 看看如何做到 一点。

1. 让我 假 您有三个4 接器，您想要将引脚 接在一起。使用 按[**I**] P4 部件的引脚4。将此 命名 a1。 在按 [Insert] 将相同的工程自 添加到引脚4 (引脚3) 下方的引脚上。注意 是如何自 重命名 a2。
2. 再按 [Insert] 两次。此 于 重 最后一 操作，它是一个无限有用的命令，可以让您的生活更 松。
3. 在另外两个 接器 CONN_2和 CONN_3上重 相同的 操作，您就完成了。如果 制作 PCB，您将看到三个 接器相互 接。 2 示了我 描述的 果。出于美 目的， 可以使用 像添加一系列 将 放入 入口  和使用 像的 路： ，如 3 所示。但是， 注意， PCB 没有影响。
4. 指出的是， 接到 2 中的引脚的短 不是 格必需的。 上， 可以直接 用于引脚。
5. 让我 更 一步，假 你有一个名 CONN_4的第四个 接器，无 出于什么原因，它的 恰好有点不同 (b1, b2, b3, b4)。 在我 想要以引脚到引脚的方式将_Bus a_与 Bus b 接起来。我 希望不使用引脚 也是可能的)，而是使用 上的 每个 一个
6. 使用之前 明的 方法 接并 CONN_4。将引脚命名 b1, b2, b3和b4。使用  将引脚 接到一系列 到 入口，并使用  接到 4
7. 在 CONN_4的 上放一个 按[**I**]) 并命名 b[1..4]。
8. 在前一个 上放一个 按[**I**]) 并将其命名 a[1..4]。
9. 我 在可以做的是使用 有按 像的 接 a[1..4] 和 b[1..4]： 。
10. 通 将两条 接在一起，引脚a1 将自 接到引脚b1, a2将 接到b2, 依此 推。 4 示了最 果的 子。

NOTE

通 [Insert] 可 的 重 上一 可以成功用于重 期 目插入。例如， 接到 2 3和 4中 所有引脚的短 都已放置此

11. 通 [Insert] 的 重 上一 也被广泛用于使用  放置 多系列的 到 入口。





布局印刷 电路板

是在 候使用您生成的网表文件来布局 PCB 了。 是通 *Pcbnew* 工具完成的。

NOTE

如果您使用来自 *Eeschema* 的 *原理 更新到PCB* 不需要网表和步 5.您 在可以像步 6和7一 将封装放入板中，然后按 步 2~4 入表 信息和 计

使用 Pcbnew

1. 从 KiCad 工程管理器， *Pcb布局 器* 。您 可以使用 *Eeschema* 中的相 工具 按 *Pcbnew* 窗口将打 开。如果您收到一条消息，指出 `*.kicad_pcb` 文件不存在并 您是否要 建它，只需 是。
2. 首先 入一些原理 信息。 部工具 上的 面置 。将相 的 尺寸 (*A4*, *8.5x11* 等) 和 置 教程1。
3. 最好将 距和**最小布 度** 置 PCB制造商要求的 度。通常，您可以将 隙 置 *0.25mm*，将最小 道 度 置 *0.25mm*。 置 → 计 菜 如果它尚未 示， 网 器 卡。将窗口 部的 距字段更改 *0.25mm*，将 布 度字段更改 *0.25mm*，如下所示。 里的 量 位是 *mm*。

网络类	间隙	布线宽度	过孔外径	辅助孔	微孔外径	U 形辅助孔	差分对线宽	差分对间距
Default	0.254 mm	0.4 mm	1.4 mm	0.6 mm	0.5 mm	0.127 mm	0.2 mm	0.25 mm
Power	0.254 mm	0.5 mm	1.6 mm	0.6 mm	0.5 mm	0.127 mm	0.2 mm	0.25 mm

+

✖

筛选网络

网络类筛选:

网络名称筛选:

显示所有网络

应用筛选



关联网络类

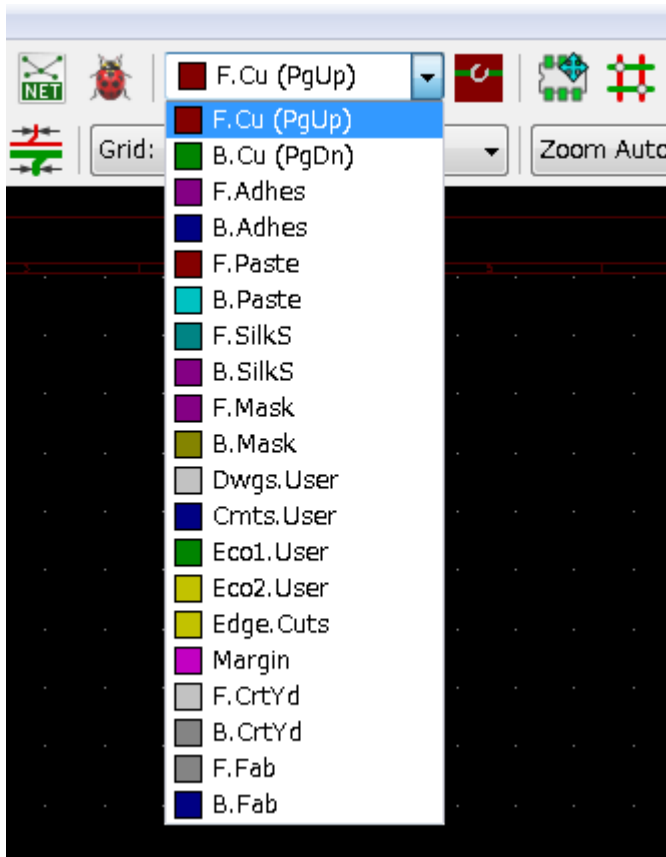
新网络类:


分配到所有列出的网络

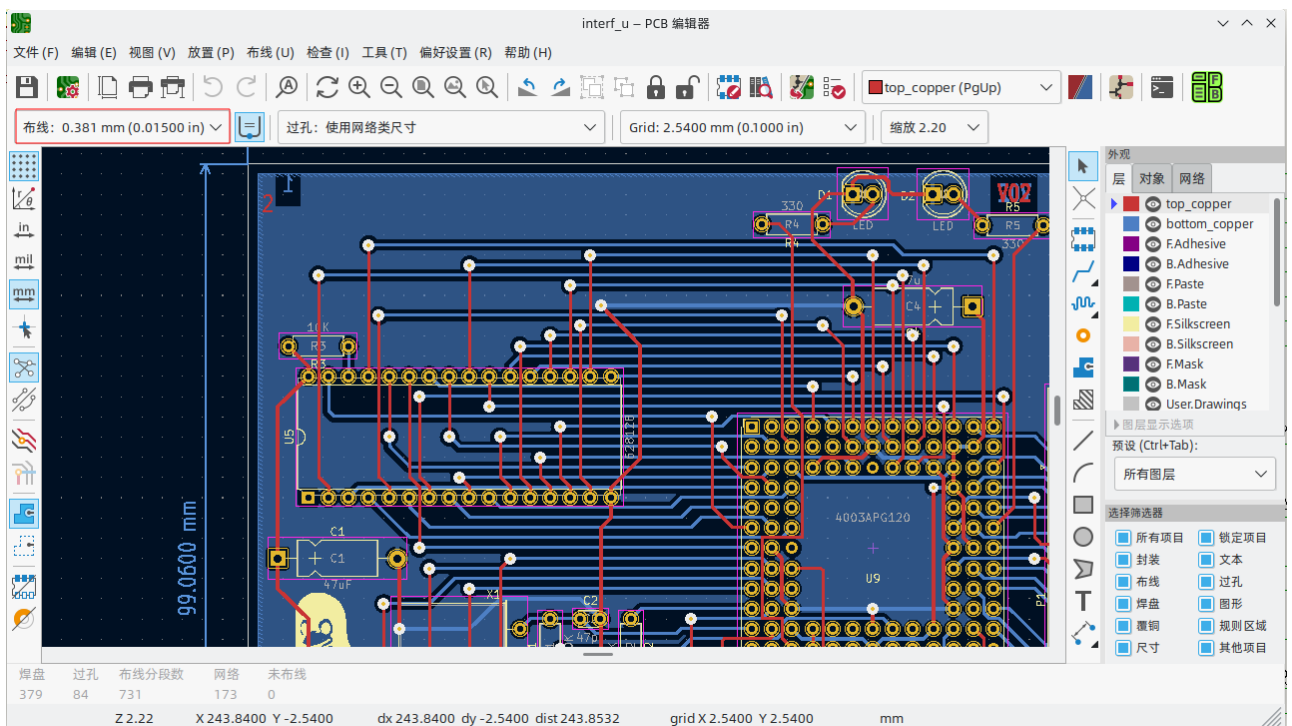
关联到选择的网络

网络	网络类
/BMH-OUT	Default
/ACK	Default
/AUTOFD-	Default
/BIT0	Default
/BIT1	Default
/BIT2	Default
/BIT3	Default
/BIT4	Default
/BIT5	Default

4. 全局 计 卡，将 **最小布 度** 置 *0.25mm*。 确定按 以提交更改并 计 器窗口。
5. 在我 将 入网表文件，如果你 建了一个。 部工具 上的 取网表 。如果网表文件 教程1.net 是从 *Eeschema* 建的， 在 网表文件 字段中 点 取当前网表。然后 按
6. 在 可以看到所有元件。 它 并按照鼠 光
7. 将元件移 到板的中 如有必要，您可以在移 元件 放大和 小。 鼠 左
8. 所有元件都通 称 的一 接。确保按下 示/ 藏板鼠 按 。通 种方式，您可以看到 接所有元件的最快速度。
9. 您可以通 将每个元件 停在其上并按 [m] 来移 它 要放置它 的位置。或者，您可以通 元件然后拖 它。按 [r] 旋 元件。移 所有元件，直到最小化 交叉的数量。



14. 例如，如果您决定改 使用 4 PCB 到 置 → 置 并将 更改 4.在 表中，您可以命名 并确定它 的含义用于。 注意，可以通 分 菜 非常有用的
15. 右 工具 上的 布 。 J1的第1 并运行 道以填充 R2。双 以 置 道 束的点。 道的 度将默认 0.250mm。您可以从 部工具 的下拉菜 中更改布 度。 注意，默认情况下，您只有一个可用的布 度。



16. 如果您想添加更多的 道 度， 到： 置 → 计 → 全局 计 卡，在此窗口的右下角添加您希望可用的任何其他 度。然后，您可以在布置 电路板 从下拉菜 中 布 的 度。 参 下面的示例（英寸）。

预定义布线和过孔标注：

布线

宽度
0.381
0.762

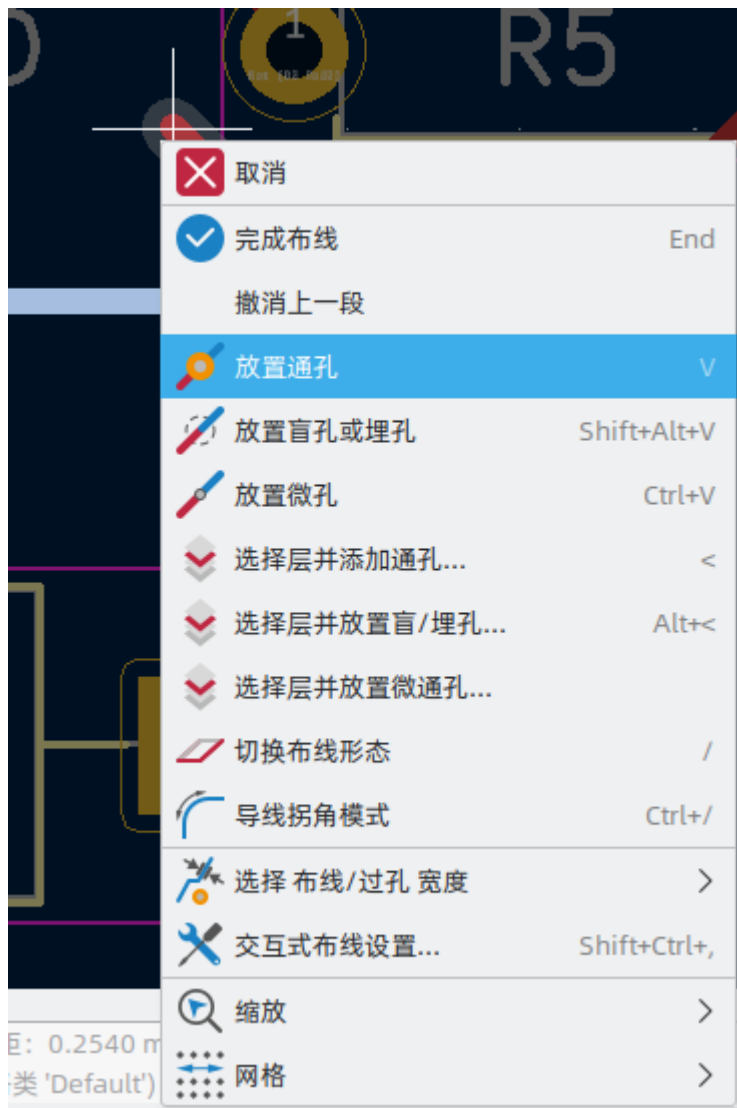
过孔



尺寸	孔
1.524	0.762

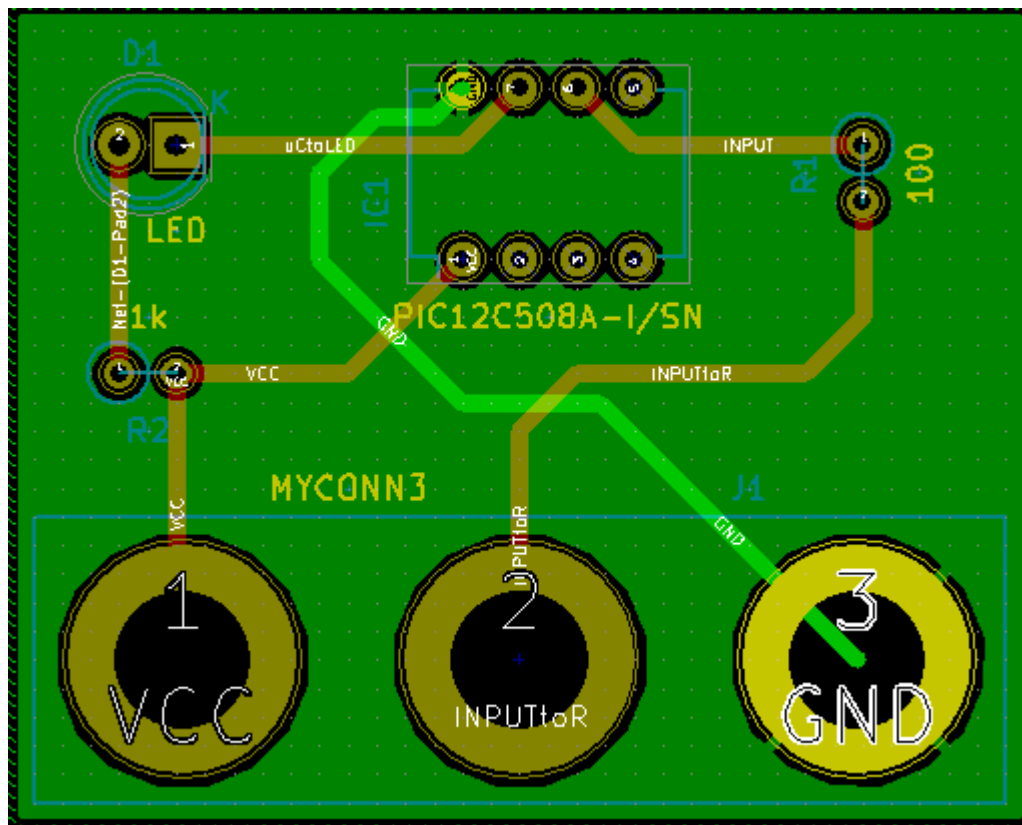
+

+

17. 或者，您可以添加一个网，在其中指定一 到 置 → 计 → 网 器并添加一个名 源的新 将布 度从 8mil（表示 0.0080）更改 24mil（表示 0.0240）。接下来，将除地面之外的所有内容添加到 源（左 默认，右 _ 源_ 并使用箭
18. 如果要更改网格大小，右 → 网格。在放下元件并将它 与 道 接在一起之前或之后， 必 合适的网格尺寸。
19. 重 此 程，直到 接除 J1 的引脚3之外的所有 您的 路板 如下所示。

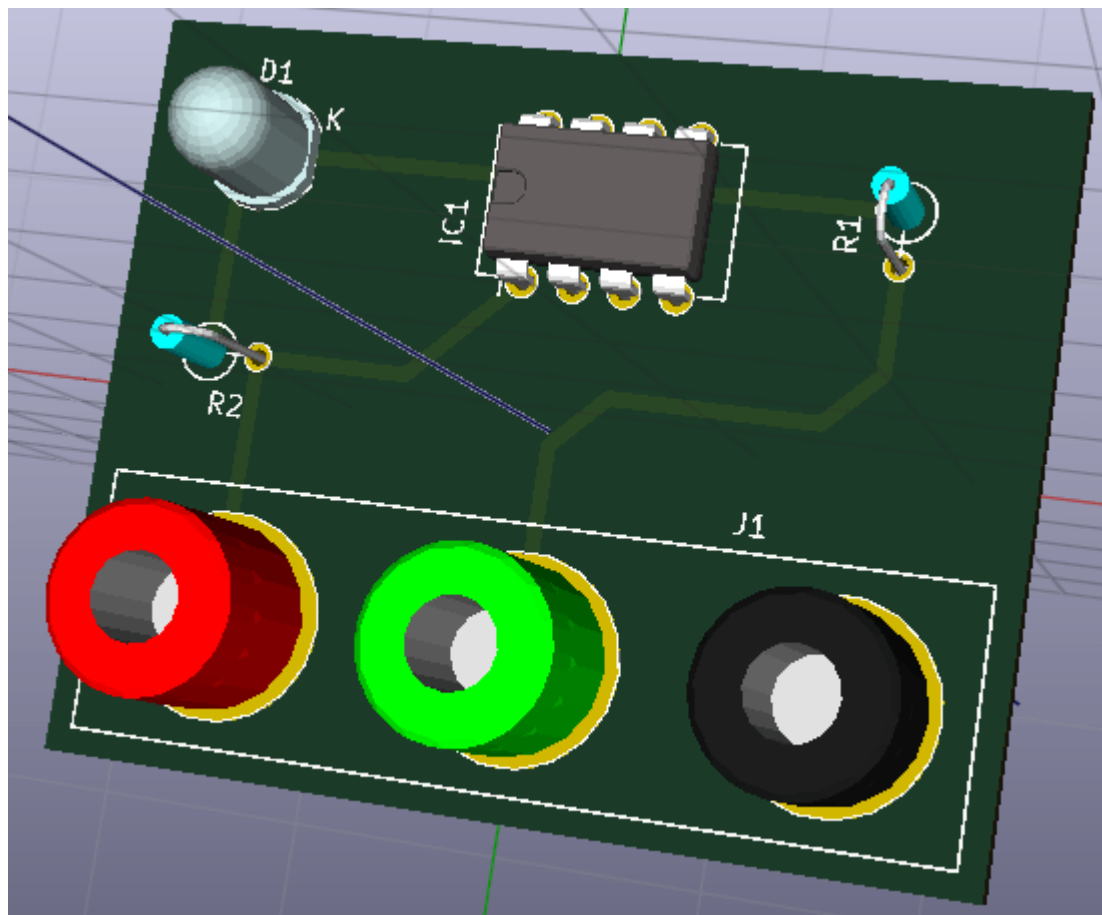


22. 当您想要 特定 接 可以 右 工具 上的 高亮网 。点 J1 的引脚3。布 本身和 接到它的所有 都 高亮 示。
23. 在我 将制作一个 接到所有 GND 引脚的接地 右 工具 上的 添加填充区域 。我 将在板周 布 一 个矩形，因此 您想要其中一个角的位置。在出 的 框中，将默认 接 置 防散 花 ，将 廓角度 置 限H, V和45度，然后 确定。
24. 通 旋 中的每个角来跟踪板的 廓。通 第二次 第一个角完成矩形。右 跟踪的区域。 区域 → 填充 或重新填充所有区域。板 填充 色，看起来像



25. 在工具栏上的运行计数器。点开始DRC。没有点列表未接。没有未接的目的。确定DRC控制框。

26. 文件 → 保存保存文件。要以3D方式观察您的电路板，→ 3D查看器。



27. 您可以拖鼠标来旋转PCB。

28. 你的板是完整的。要将其 送 制造商，您需要生成所有 Gerber 文件。


生成 Gerber 文件

完成 PCB 后，您可以 每一 生成 Gerber 文件，并将它 送 您最喜 的 PCB 制造商，他 将 您制作 电路板。

- 1. 从 KiCad, 打开 *Pcbnew* 板 器。
- 2. 点 **文件** → **制 Gerber 作 制格式并 放置所有 Gerber 文件的文件 制按**
- 3. 要生成 取文件， 从 *Pcbnew* 再次 到 **文件** → **制 默认 置 没**
- 4. 些是您制作典型的2 PCB 需要 的

	KiCad 名	默认 Gerber 展	“使用 Protel 文件 展名”已启用
Bottom Layer	B.Cu	.GBR	.GBL
Top Layer	F.Cu	.GBR	.GTL
Top Overlay	F.SilkS	.GBR	.GTO
Bottom Solder Resist	B.Mask	.GBR	.GBS
Top Solder Resist	F.Mask	.GBR	.GTS
Edges	Edge.Cuts	.GBR	.GM1

使用 Gerbview

- 1. 要 看所有 Gerber 文件， 到 KiCad 工程管理器并 *GerbView* 在下拉菜 或 管理器中， 形 1。
文件 → **打开 Gerber 文件** 或 。 并打开所有生成的 Gerber 文件。注意它 如何一个 示在另一个之上。
- 2. 使用 **文件** → **打开 Excellon 取文件** 打开 取文件。
- 3. 使用右 的 管理器 /取消 要 示的 在 送生 之前仔 每一
- 4. 与 *Pcbnew* 似。在 内右 并 网格以更改网格。

使用 FreeRouter 自 布

手 布 板既快速又有趣，但 于具有大量元件的 电路板，您可能希望使用自 布 器。 住，您 首先手 布 迹 然后 置自 布 器以 行无聊位。它的工作只会解 未布 的痕迹。我 将在 里使用的自 布 器是 FreeRouting。

NOTE

FreeRouting 是一个开源的 Java 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 网搜索_freerouting_找到它 它可以在源代 形式或 的 Java 包中找到。


- 1. 从 *Pcbnew* **文件** → **出** → *** Specctra DSN *** 并在本地保存文件。后 *FreeRouter* 并 *打开您自己的 计按 dsn* 文件并加 它。
- 2. *FreeRouter* 具有 KiCad 目前不具 的一些功能，包括手 布 和自 布 *FreeRouter* 主要有两个步 首先， 路 板 行布 然后 其 行 化。完全 化可能需要很 但您可以随 停止它。

您可以通过 部 上的 自 布 器 按 来后 自 布 底 您提供有 正在 行的布 程的信息。如果 通 计数超 30 您的 电路板可能无法使用此布 器自 行。更多地展开您的元件或更好地旋 它 并再 一次。零件的旋 和 位置的目 是降低速率的交叉走 的数量。

4. 左 鼠 可以停止自 布 并自 后 化 程。另一次左 将停止 化 程。除非你真的需要停下来，否 最好让 FreeRouter 完成它的工作。

5. 文件 → 出 Spectra 会 文件 菜 并使用 .ses 展名保存板文件。您 上不需要保存 FreeRouter 文件。





6. 回到 Pcbnew。您可以通过 文件 → 入 → * Spectra 会 * 并 .ses 文件来 入 布 的 电路板。

如果有任何您不喜 的布 可以使用 [Delete] 和布 工具 除它并重新布 它， 是布  在右 工具 上添加。

在 KiCad 中向前批注

完成 子原理 封装分配, 电路板布局和生成 Gerber 文件后, 您就可以将所有内容 送 PCB 制造商, 以便您的 电路板 成

通常, 种 性工作流程并非如此 向。例如, 当您必 修改/ 展您或其他人已 完成此工作流程的板 您可能需要移 元件, 替 其他元件, 更改占用空 等等。在此修改 程中, 您不想做的是从 开始重新布 整个 电路板。相反, 是你 如何做到 一点:

1. 假 您想要用 CON2 替 假 的 接器 CON1。
2. 您已 有完整的原理 和完全布 的 PCB。
3. 从 KiCad 开始 *Eeschema*, 通 除 CON1 并添加 CON2 行修改。使用 像保存原理 工程  并点 部工具 上的 网表生成 。
4. 点 网表然后点 保存。保存 默认文件名。你必 重写旧的。
5. 在 CON2 分配封装。 部工具 上的 运行Cvpcb 。将封装分配 新 CON2。其余元件仍然具有分配 它的先前封装。 Cvpcb。
6. 返回原理 器, 文件 → 保存整个原理 工程保存工程。 原理 器。
7. 从 KiCad 工程管理器, Pcbnew Pcbnew 窗口将打开。
8. 旧的, 已 布 的 电路板 自 打开。让我 入新的网表文件。 部工具 上的 取网表 。
9. 网表文件按 在文件 框中 网表文件, 然后 取当前网表。然后 按
10. 此 您 能 看到已布置所有先前元件的布局。在左上角, 您 看到所有未布 的元件, 在我 的例子中是 CON2。用鼠 CON2。将元件移 到板的中
11. 放置 CON2 并布 它。完成后, 保存并 照常生成 Gerber 文件。




里描述的 程可以根据需要 松地重 多次。除了上面描述的向前批注方法之外, 有另一种方法称 向后批注。此方法允 您从 Pcbnew 修改已布 的 PCB, 并在原理 和网表文件中更新 些修改。然而, 向后批注方法没有那么有用, 因此 里不描述它。

在 KiCad 中制作原理 符号

有 您想要放置在原理 中的符号不在 KiCad 中。 很正常，没有理由担心。在本 中，我 将了解如何使用 KiCad 快速 建新的原理 符号。不 住，您始 可以在互 网上找到 KiCad 元件。


在 KiCad 中，符号是一段以 *DEF* 开 并以 *ENDDEF* 尾的文本。一个或多个符号通常放在 文件中， 展名 *.lib*。如 果要将符号添加到 文件，只需使用文本 器的剪切和粘 命令即可。

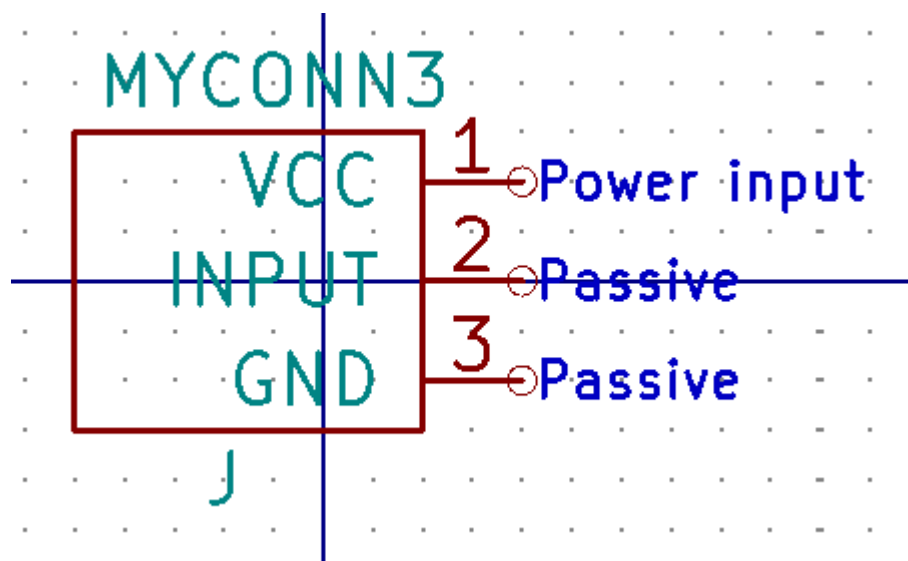
使用元件 器





1. 我 可以使用_元件 器_ (*Eeschema* 的一部分) 来 建新元件。在我 的工程文件 教程1 中，让我 建一个名 的文件 在我 建新元件后，我 将在内部放置新的 文件 *myLib.lib*。
2. 在我 可以开始 建新元件了。从 KiCad 开始 *Eeschema* 器 ，然后 新元件 。将出 元件属性 窗口。将新元件命名 *MYCONN3*，将 默认参考指示符 置 *J*，将 每个包的 位数 置 *1*。 确定。如果出 警告， 是。此 元件 由其 成。让我 添加一些引脚。 右 工具 上的 添加引脚 。要放 置引脚， 在 *MYCONN3* 正下方的零件 器工作表中 鼠 左
3. 在出 的引脚属性窗口中，将引脚名称 置 *VCC*，将引脚 号 置 *1*，将 气 型 置 源 入，然后 确定。



4. 在 *MYCONN3* 下方 您想要的位置放置引脚。
 5. 重 行引脚步 次 引脚名 入，引脚号 *2*，气 型 被 。
 6. 重 放置引脚步 次 引脚名称 *GND*，引脚 号 *3*，气 型 被 。
- 将 一个放在另一个的 部。元件 *MYCONN3* 位于 面的中心 (交叉的位置)。







接下来，制元件的 廓。 添加矩形 。我 想在引脚旁 制一个矩形，如下所示。 此， 矩形左上角的位置（不要按住鼠 按 再次 矩形右下角的位置。





8. 如果要用黄色填充矩形， 在 偏好 置 → 色方案 中将填充 色 置 黄色4， 然后使用 [e] 在 屏幕中 矩形， 填充背景。
9. 将元件保存在 myLib.lib 中。 新建  航到 教程1/ / 文件 并保存名 myLib.lib 的新 文件。
10. 到 偏好 置 → 元件 并在 用 定义的搜索路径中添加 教程1/ /， 在 元件 文件中添加 mylib.lib。
11. 工作 。在 窗口中， myLib， 然后 确定。注意窗口的 如何表示当前正在使用的 在 是 myLib。
12. 部工具 中的 更新当前 中的当前元件 。 部工具 中的 在磁 上保存当前加 的  保存所有 更改。在出 的任何确认消息中 是。 在， 新的 示意 元件已在窗口 中指示的 中完成并可用。
13. 您 在可以 元件 器窗口。您将返回到原理 器窗口。您的新元件 在可以从 myLib 中使用。
14. 您可以通 将 .lib 文件添加到 路径来使其可用。从 Eeschema 开始， 到* 偏好 置 * → * 并在 用 定义的搜索路径中添加路径， 在 元件 文件中添加_file.lib_。

出， 入和修改 元件

而不是从 开始 建 元件， 有 从已 制作并修改它更容易。在本 中， 我 将了解如何将元件从 KiCad 准 出到 您自己的 myOwnLib.lib， 然后 行修改。



1. 从 KiCad 开始 Eeschema， 点 器 ， 点 工作  并 的 。 从当前 中加 元件 行  并 入 RELAY_2RT。
2. “ 出元件”  航到 / 文件 并保存名 myOwnLib.lib 的新 文件
3. 您可以将此元件和整个 myOwnLib.lib 添加到 路径中， 从而使您可以使用它。从 Eeschema 到 *偏好 置 * → * 元件 * 并在 用 定义的搜索路径中添加 library/， 在 元件 文件中添加 myOwnLib.lib。 窗口。
4. 工作 。在 窗口中， myOwnLib 并 OK。注意窗口的 如何表示当前正在使用的 它 是 myOwnLib。
5. 从当前 中加 元件 行  并 入 RELAY_2RT。

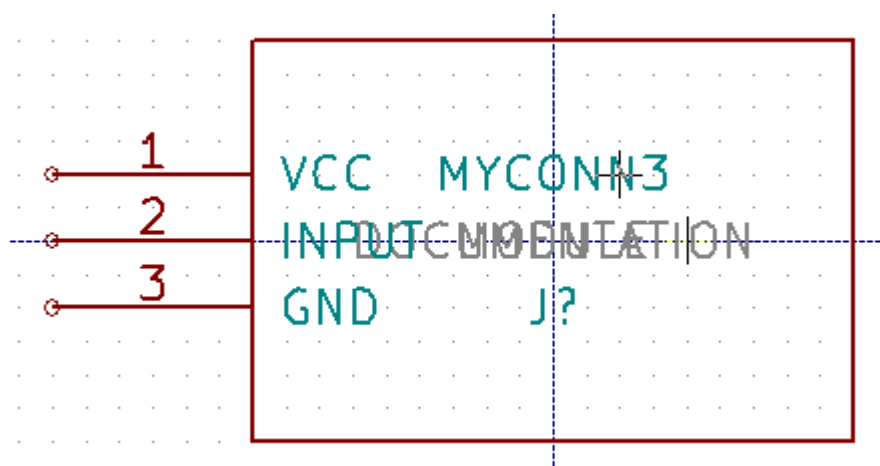
您可以在可以根据需要修改元件。将鼠标停在 *RELAY_2RT* 上，按 [e] 并将其重命名 *MY_RELAY_2RT*。

7. 部件工具 中的 更新当前 中的当前元件 。部件工具 中的 在磁 上保存当前加的  保存所有更改。

使用 quicklib 制作原理 元件

本节介绍使用互联网工具 *quicklib* 构建原理元件的另一种方法（参见上面的《*myconn3*, *MYCONN3*》）。

1. 前往 *quicklib* 网 <http://kicad.rohrbacher.net/quicklib.php>
2. 使用以下信息填写 面：元件名称：MYCONN3 参考前 J 引脚布局 式：SIL 引脚数，N:3
3. 分配引脚 使用以下信息填写 面：引脚1：VCC;引脚2：输入引脚;3：GND。 型：所有3个引脚都被
4. 如果您 意， 构建 元件。下 文件并将其重命名 教程1/ /myQuickLib.lib。你完成了！
5. 使用 KiCad 看它。从 KiCad 工程管理器，后 *Eeschema* “ 器”  “ 入元件”  航到 教程1/ /并 *myQuickLib.lib*



6. 您可以通过将此元件和整个 *myQuickLib.lib* 添加到 KiCad 路径来使其可用。从 *Eeschema* 到 偏好 置 → 元件 并在 用 定义的搜索路径中添加 *library*，在 元件 文件中添加 *myQuickLib.lib*。

正如您可能猜到的，当您想要 构建具有大引脚数的元件 种 构建 元件的方法非常有效。

制作高引脚数的原理 元件

在 *quicklib* 中 制作原理 元件的部分中，我 了解了如何使用 *quicklib* 基于 Web 的工具制作原理 元件。但是，您偶 会 需要 构建一个具有大量引脚（几百个引脚）的原理 元件。在 KiCad 中， 不是一 非常 的任

1. 假 您要 具有 50 个引脚的器件 构建原理 元件。通常的做法是使用多个低引脚数的 来 制它，例如两个 每个 有 25 个引脚。 元件表示允 的引脚 接。
2. 构建元件的最佳方法是使用 *quicklib* 分 生成两个 25 引脚元件，使用 Python 脚本重新 号它 的引脚，最后通 使用 制和粘 将它 合并 一个 独的 DEF 和 ENDDF 元件。
3. 您将在下面找到一个 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 号 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 文件 *in.txt* 中的所有行都已完成。

的脚本

```
#!/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
```

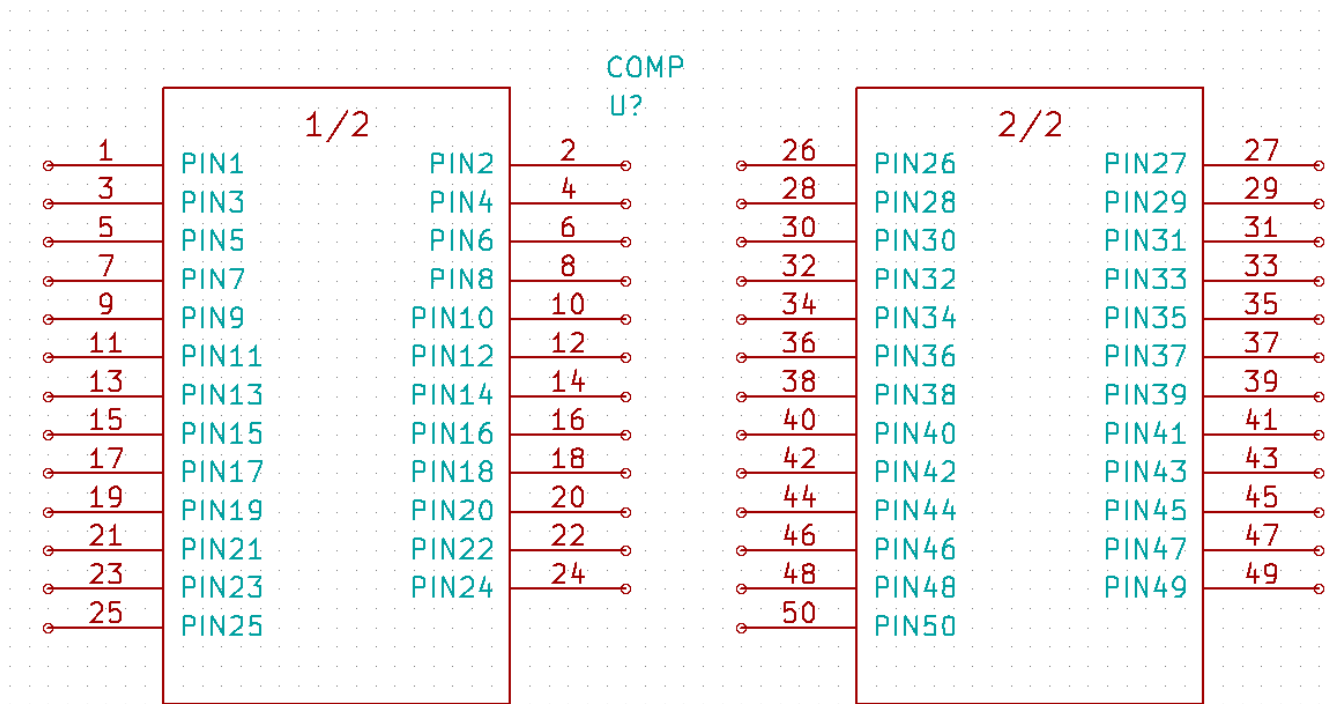
1. 在将两个元件合并 一个元件 有必要使用 Eeschema 的 器移 第一个元件，以便第二个元件不会在它上面移
您将在下面找到最 的 .lib 文件及其在 *Eeschema* 中的表示形式。

*.lib 文件的内容

```
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. 里介 的 Python 脚本是一个非常 大的工具，用于操作引脚号和引脚 然而， 注意，它的所有功能都来自于 神秘而且非常有用的正 表达式 法：<http://gskinner.com/RegExr/>.





制作元件封装

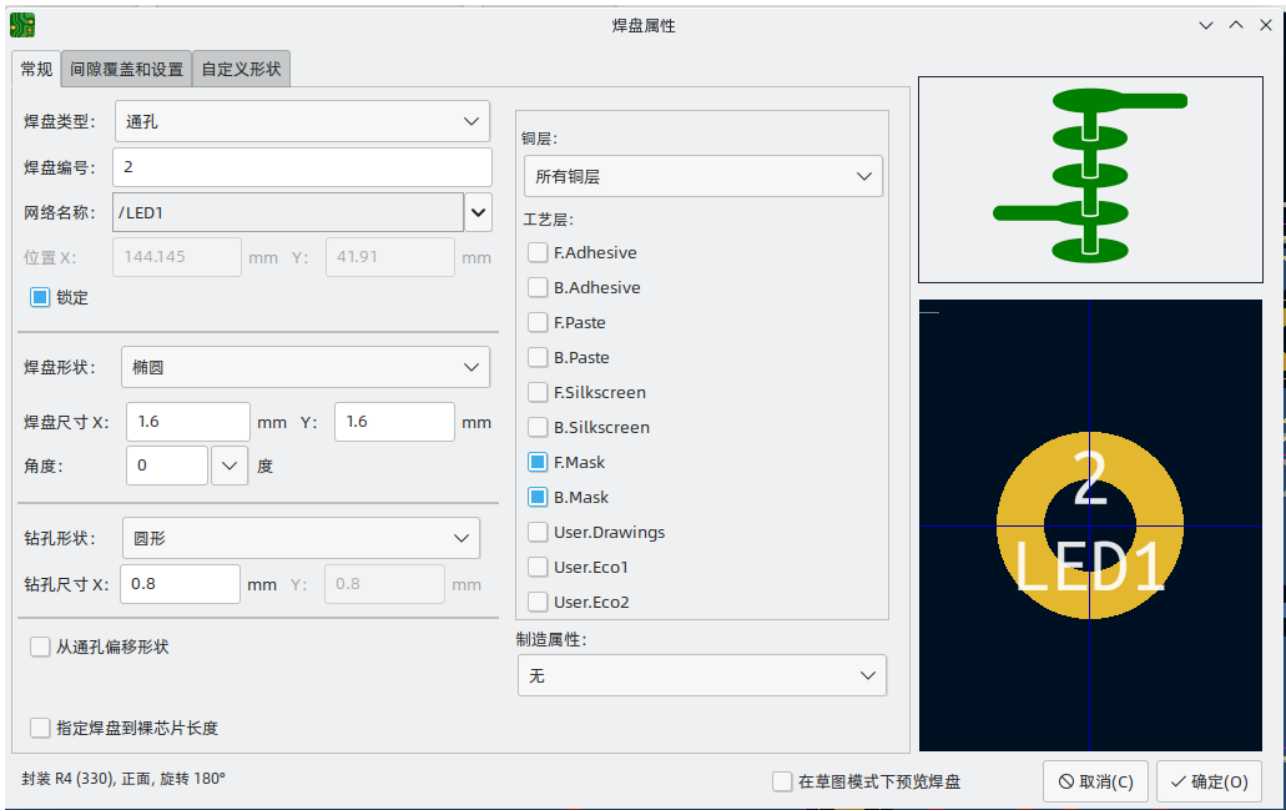
与其他 EDA 件工具不同，其中一种 型的 包含原理 符号和封装 化，KiCad.lib 文件包含原理 符号，.kicad_mod 文件包含封装。Cvpcb 用于将封装映射到符号。

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

有一个广泛的封装 与 KiCad，但有 您可能会 您需要的封装不在 KiCad 中。以下是在 KiCad 中 建新 PCB 封装 的步骤

使用封装 器

1. 从 KiCad 工程管理器开始 Pcbnew 工具。 部工具 上的 打开封装 器 。 将打开 封装 器。
2. 我 将在新的封装 myfootprint 中保存新的封装 MYCONN3。在 教程1/ 工程文件 中 建一个新文件 myfootprint.pretty。 偏好 置 → 封装 管理器 并按 附加 按 在表格中， 入 myfootprint 作 昵称， 入 \${KIPRJMOD}/myfootprint.pretty 作 路径， 并 入 KiCad 作 插件 型。按 确定 PCB 表窗口。 部工具 上的 活 。 myfootprint
3. 部工具 上的 新封装 。 入 MYCONN3 作 封装名称。在屏幕中 将出 MYCONN3 在 下， 您 可以看到 REF* 右 MYCONN3 并将其移到 REF* 上方。右 REF*_ 文本 并将其重命名 SMD。将 示 置 不可。
4. 在右 工具 中 添加 。 工作表以放置 右 新 然后 。 你也可以用 [e]。



5. 将 号 置 1， 形状 置 矩形， 型 置 SMD， 形状大小 X 置 0.4， 形状大小 Y 置 0.8。 确定。再次点 添加 并再放两个
6. 如果要更改网格大小， 右 → 网格 。在放下元件之前， 必 合适的网格尺寸。

将 MYCONN3 和 SMD 移开，使其看起来像上 所示。

8. 放置 通常需要 量相 距离。将光 放在您想要相 坐 点_ (0,0) _的位置，然后按空格 移 光 您将看到 光 在 面底部位置的相 指示。可以随 按空格 置新原点。

9. 在添加一个封装 框。 右 工具 中的 添加 形 或多 形按 。 制元件周 接器的 框。

10. 部工具 上的 在活 中保存封装 , 使用默认名称 MYCONN3。

于 KiCad 工程文件的可移植性的注意事项

您需要将哪些文件 送 某人才能完全加 和使用您的 KiCad 工程？

当你有一个 KiCad 工程与某人分享 重要的是原理 文件 `.sch`, 板文件 `.kicad_pcb`, 工程文件 `.pro` 和网表文件 `.net`, 与两个原理 一起 送 文件 `.lib` 和封装文件 `.kicad_mod`。只有 人 才能完全自由地修改原理 和 电路板。

使用 KiCad 原理 人 需要包含符号的 `.lib` 文件。需要在 *Eeschema* 偏好 置中加 些 文件。另一方面, 使用板 (`.kicad_pcb` 文件), 封装可以存 在 `.kicad_pcb` 文件中。您可以向某人 送 `.kicad_pcb` 文件, 而不是其他任何内容, 他 仍然可以 看和 板。但是, 当他 想要从网表加 元件 脚本 (`.kicad_mod` 文件) 需要存在并加 到 *Pcbnew* 偏好 置中, 就像原理 一 此外, 有必要在 *Pcbnew* 的偏好 置中加 `.kicad_mod` 文件, 以便在 *Cvpcb* 中 示 些封装。

如果有人向您 送了一个 有封装的 `.kicad_pcb` 文件, 您可以在另一个板上使用, 您可以打开封装 器, 从当前板上 加 封装, 并将其保存或 出到另一个封装 中。您也可以通 * *Pcbnew* * -> 文件 -> -> 封装 -> 建封装 将 建一个新的一次 出 `.kicad_pcb` 文件中的所有封装。 `.kicad_mod` 文件包含所有板的封装。

最重要的是, 如果 PCB 是你想要分 的唯一 西, 那么 电路板文件 `.kicad_pcb` 就足 了。但是, 如果您想让人 完全能 使用和修改您的原理 其元件和 PCB 烈建 您 并 送以下工程目

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

有 KiCad 文档的更多信息

是 KiCad 中大多数功能的快速指南。有更 的 明，参 可通 每个 KiCad 模 的帮助文件。点 **帮助** → **手册**。

KiCad 其所有四个 件元件提供了一套非常好的多 言手册。

所有 KiCad 手册的英文版都随 KiCad 一起 布。

除了手册之外，KiCad 随本教程一起 布，教程已被翻 成其他 言。本教程的所有不同版本均免 分 所有最新版本的 KiCad。本教程以及手册 与您的 KiCad 版本一起打包在您的指定平台上。

例如，在 Linux 上，典型位置位于以下目 中，具体取决于您的确切分布：

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

在Windows上它位于：

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

在 OS X 上：

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

网上的 KiCad 文档

最新版本的 KiCad 文档可以在 <http://docs.kicad.org> 上找到多种 言的版本