# KiCad 入门

The KiCad Team

# Table of Contents


Introduction to KiCad Version 8 ..... 2
- 下载和安装 KiCad ..... 2
- 支持 ..... 2

基本概念和工作流程 ..... 3
- PCB 设计工作流程 ..... 5

教程第一部分：工程 ..... 6

教程第 2 部分：原理图 ..... 8
- 符号库表设置 ..... 8
- 原理图编辑器基础知识 ..... 8
- 原理图图框设置 ..... 9
- 将符号添加到原理图中 ..... 10
- 选择和移动对象 ..... 11
- 为原理图布线 ..... 12
- 批注、符号属性和封装 ..... 14
- 电气规则检查 ..... 16
- 物料清单 ..... 17

教程第 3 部分：电路板 ..... 19
- PCB 编辑器基础知识 ..... 19
- 板子的设置和压层 ..... 19
- 从原理图中导入更改 ..... 23
- 绘制电路板边框 ..... 24
- 放置封装 ..... 25
- 布线 ..... 27
- 放置敷铜 ..... 29
- 设计规则检查 ..... 31
- 3D 查看器 ..... 33
- 制造输出 ..... 34

教程第 4 部分：自定义符号和封装 ..... 37
- 库和库表的基础知识 ..... 37
- 创建新的全局库或工程库 ..... 38
- 新建符号 ..... 38
- 新建封装 ..... 40
- 链接符号、封装和 3D 模型 ..... 47

今后的发展方向 ..... 50
- 更多学习资源 ..... 50
- 帮助改进 KiCad ..... 50

# 版权

This document is Copyright © 2010-2024 by its contributors as listed below. You may distribute it and/or modify it under the terms of either the GNU General Public License (http://www.gnu.org/licenses/gpl.html), version 3 or later, or the Creative Commons Attribution License (http://creativecommons.org/licenses/by/3.0/), version 3.0 or later.

本指南中的所有商标均属于其合法所有者。

### 贡献人员

Graham Keeth, Jon Evans, Glenn Peterson.

### 翻译人员

taotieren <admin@taotieren.com>, 2019-2023.

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

### 以前的贡献者

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

### 反馈

The KiCad project welcomes feedback, bug reports, and suggestions related to the software or its documentation. For more information on how to submit feedback or report an issue, please see the instructions at https://www.kicad.org/help/report-an-issue/

# Introduction to KiCad Version 8

KiCad 是一个开源的软件套件，用于创建电子电路原理图、印刷电路板（PCB）和相关的零件描述。KiCad 支持综合设计工作流程，其中原理图和相应的 PCB 一起设计，也支持特殊用途的独立工作流程。KiCad 还包括一些帮助电路和 PCB 设计的实用程序，包括用于确定电路结构电气特性的 PCB 计算器、用于检查制造文件的 Gerber 查看器、用于可视化成品 PCB 的 3D 查看器以及用于检查电路行为的集成 SPICE 仿真器。

KiCad 可在所有主要的操作系统和广泛的计算机硬件上运行。它支持多达 32 个铜层的 PCB，适合创建各种复杂的设计。KiCad 是由世界各地的软件和电气工程师组成的志愿者团队开发的，其使命是创建适合专业设计师的免费和开源的电子设计软件。

The latest documentation for KiCad is available at https://docs.kicad.org. This document applies to KiCad version 8.0.

## 下载和安装 KiCad

KiCad 可在许多操作系统上运行，包括微软 Windows、苹果 macOS 和许多主要的 Linux 发行版。

你可以在 https://www.kicad.org/download/ 找到最新的说明和下载链接。这些说明没有包括在本手册中，因为它们可能随着操作系统更新的发布而改变。

| | |
|---|---|
| **IMPORTANT** | 根据 KiCad 稳定发布政策，KiCad 的稳定发布会定期进行。新的功能正在不断地被添加到开发分支中。如果你想利用这些新功能，并通过测试它们来提供帮助，请下载你的平台的最新夜间构建包。夜间构建可能会引入一些错误，如文件损坏、生成不良 Gerbers 等，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。 |

## 支持

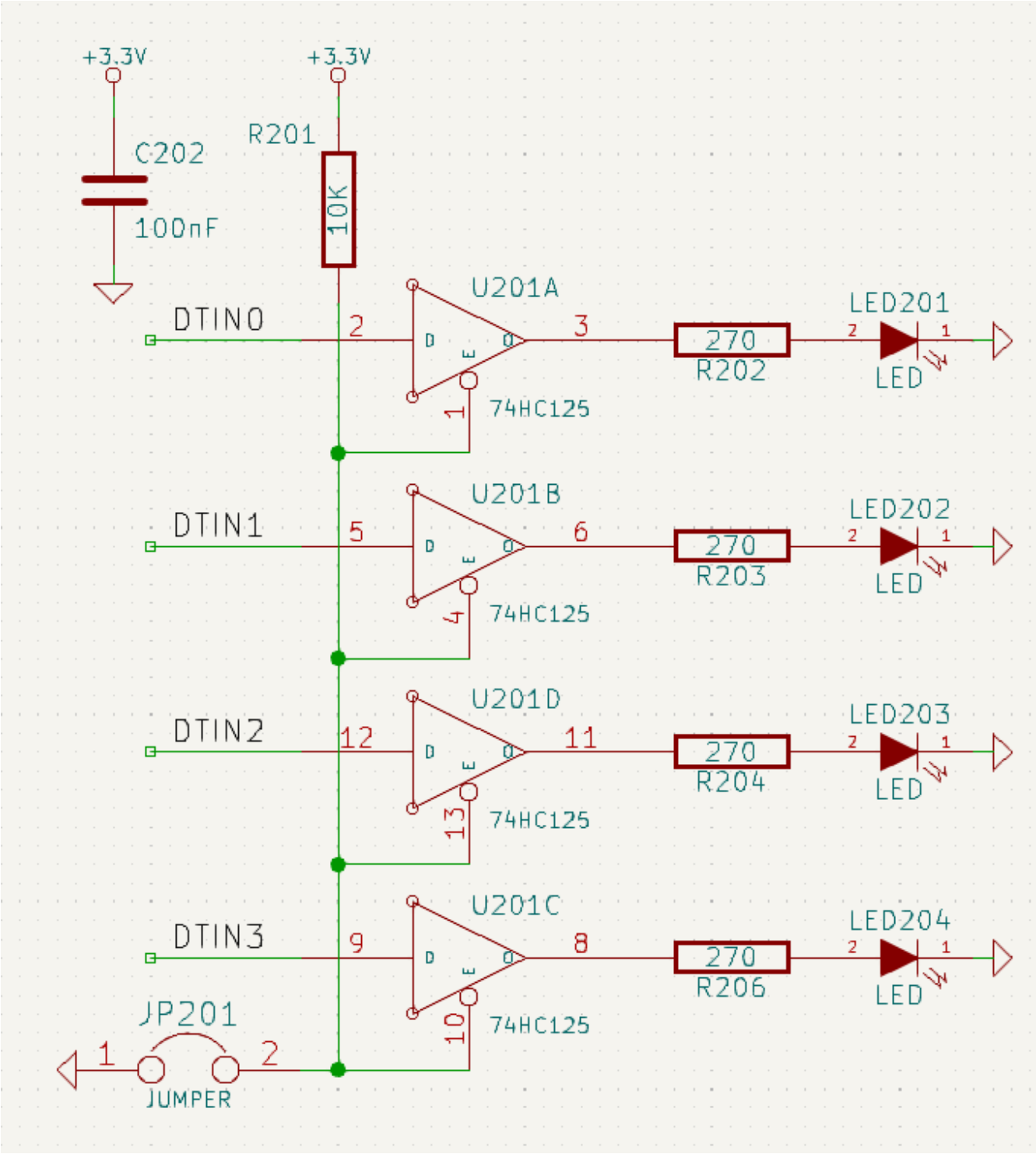
如果您有想法，评论或问题，或者您只是需要帮助：

- 这是 KiCad 官方用户论坛 是一个与其他 KiCad 用户联系并获得帮助的好地方。
- 加入我们在 Discord 或 IRC 的社区，与用户和开发者进行实时讨论。
- 查看 KiCad 网站，了解 KiCad 社区制作的 学习资源。

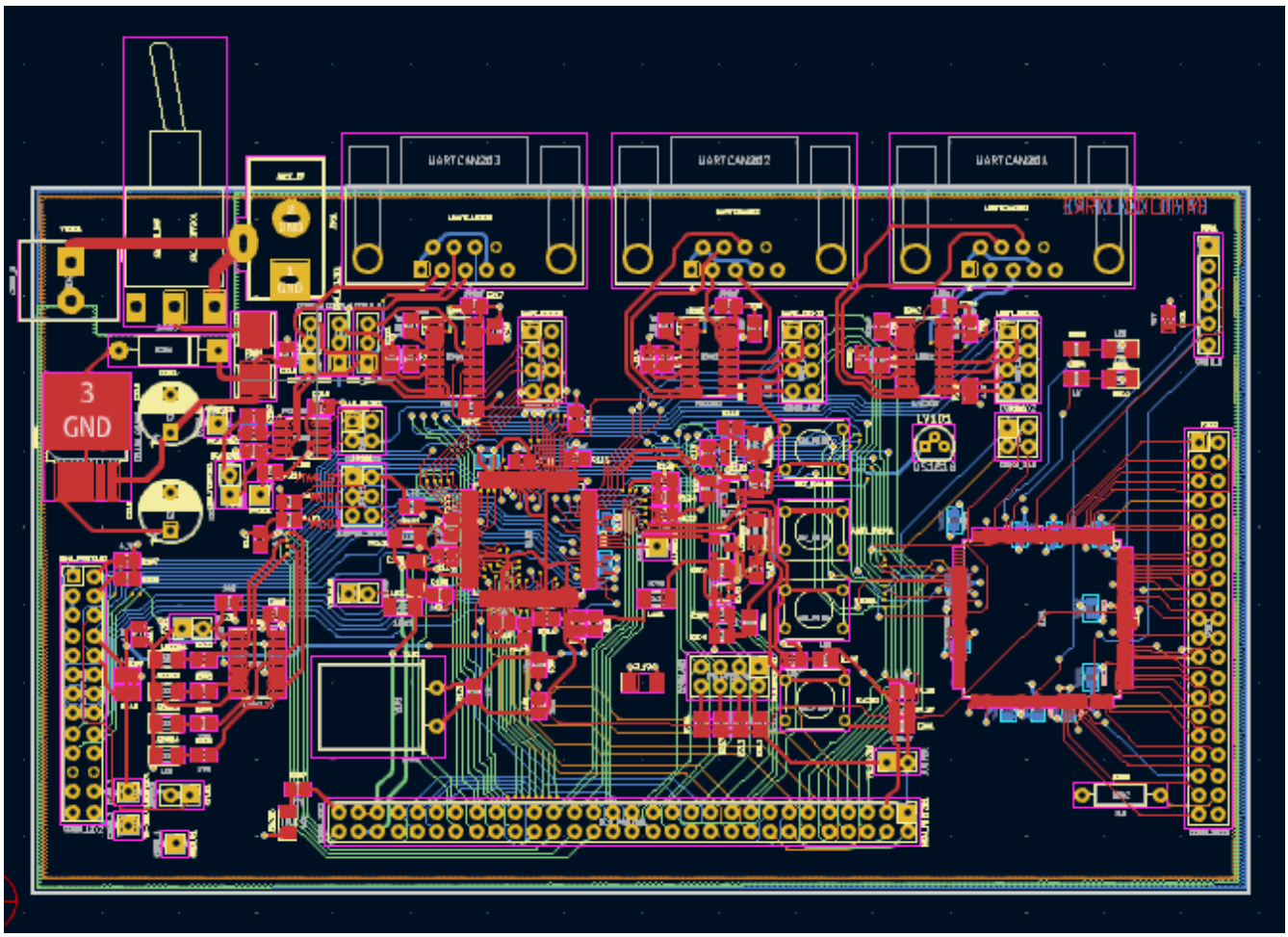
# 基本概念和工作流程

KiCad 的典型工作流程包括两个主要任务：绘制原理图和布局电路板。

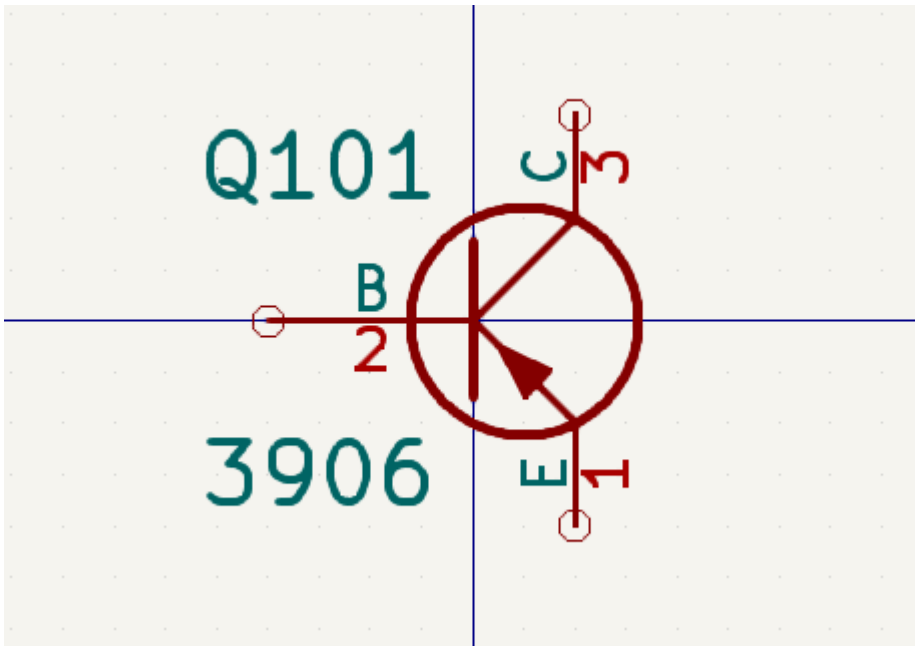
原理图是电路的符号表示：使用了哪些元件，它们之间有哪些连接。原理图符号是电子元件在原理图中的象形表示，如一个电阻的人字形或矩形，或一个运算放大器的三角形。原理图包含了设计中每个元件的符号，符号中的线连接着引脚。原理图通常是先画出来，然后再布线电路板。

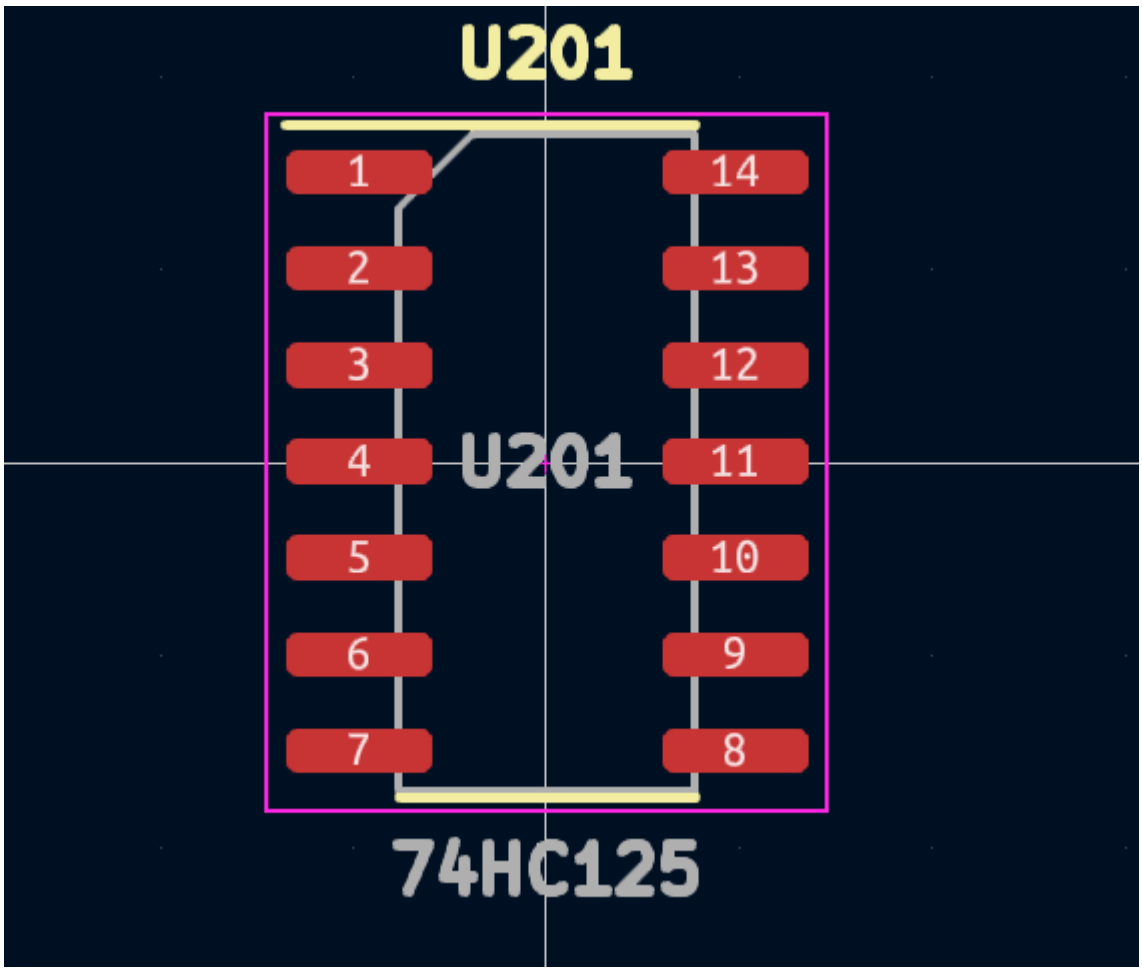



板子是原理图的物理实现，元件的封装被放置在板子上，铜线实现原理图中描述的连接。封装是一组铜焊盘，与物理元件上的引脚相匹配。当电路板被制造和组装时，该元件将被焊接到电路板上相应的封装上。




KiCad 有单独的窗口用于绘制原理图（"原理图编辑器"）、布局电路板（"PCB 编辑器"）以及编辑符号和封装（"符号编辑器 "和 "封装编辑器"）。KiCad 自带了大量高质量的、由用户提供的符号和封装库，但创建新的符号和封装或修改现有符号和封装也很简单。






最后，重要的是要了解 KiCad 有一个基于工程的工作流程。 KiCad 工程是一个包含工程文件、原理图、电路板布局的文件夹，也可以选择其他相关文件，如符号和封装库、仿真数据、采购信息等。许多与项目相关的设置，包括网络类和设计规则，都存储在工程层面。在其相关工程之外打开一个电路板可能会导致设计信息的丢失，所以要确保将所有与工程相关的文件放在一起。

## PCB 设计工作流程

通常情况下，首先绘制原理图。这意味着在原理图上添加符号，并绘制它们之间的连接。如果没有合适的符号，可能需要创建自定义符号。在这个阶段，还要为每个部件选择封装，必要时还要创建自定义封装。当原理图完成后，设计通过了电气规则检查（ERC），原理图中的设计信息被转移到电路板编辑器中，并开始布局。

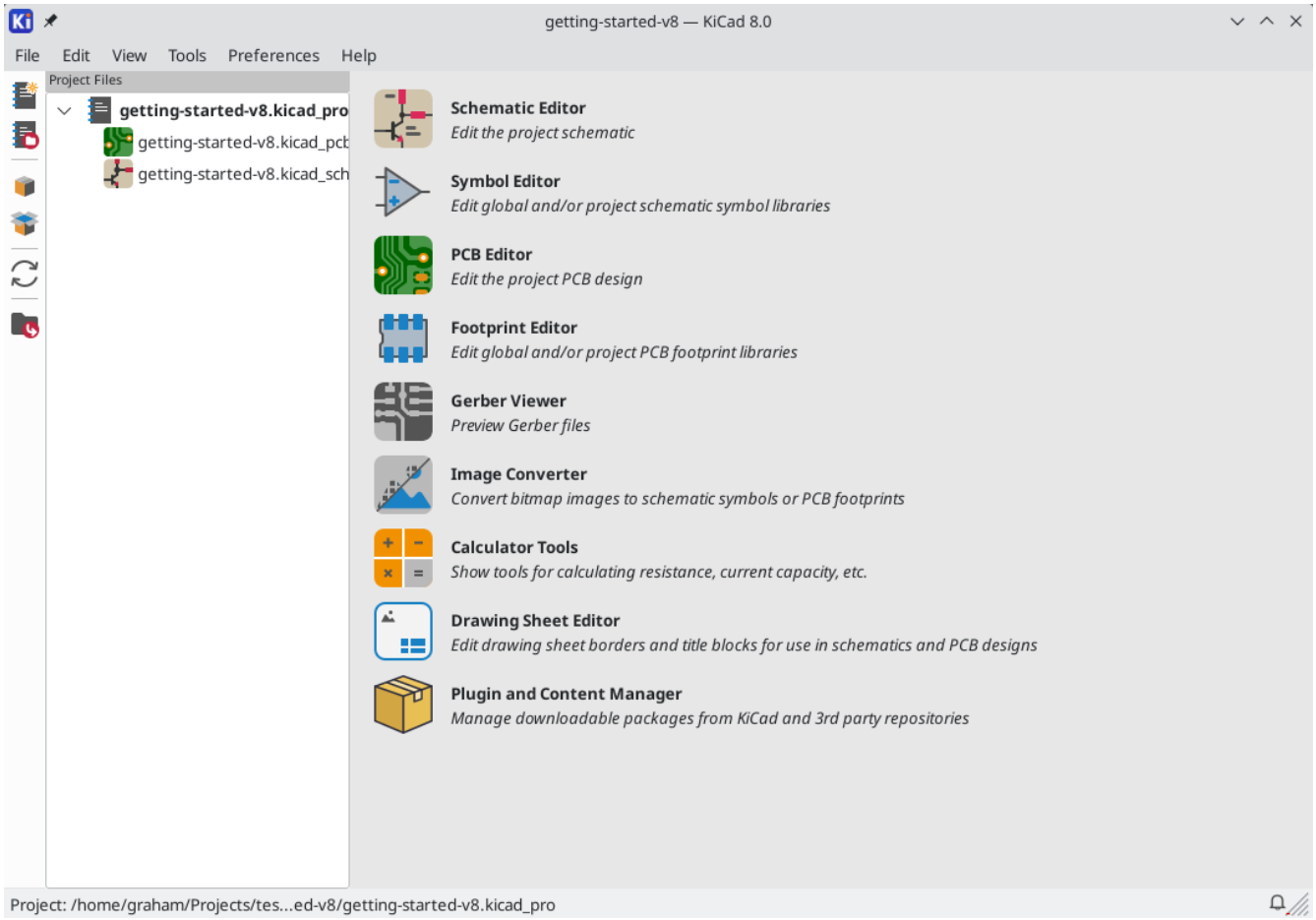
原理图描述了设计中的哪些元件以及它们的连接方式；电路板编辑器利用这些信息使布局更容易，并防止原理图和 PCB 之间的不匹配。布局过程需要在电路板上仔细放置每个封装。在元件放置完毕后，根据原理图中的连接以及其他电气方面的考虑，如跟踪电阻、控制阻抗要求、串扰等，在元件之间绘制铜线。

通常情况下，原理图在布局开始后需要更新；原理图的变化可以很容易地拉到电路板设计中。相反的情况也经常发生：在电路板布局中的任何设计变化都可以被推回到原理图中，以保持两者的一致性。

当电路板布局完成，并且电路板通过了设计规则检查（DRC），就会产生制造输出，这样电路板就可以由 PCB 制造者制造。

# 教程第一部分：工程

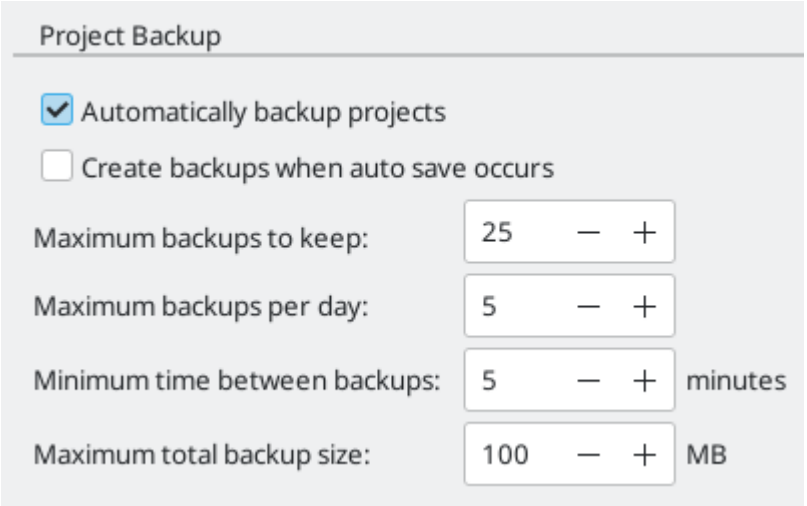
开始新设计时，首先要做的是创建一个新的工程。 打开 KiCad 会出现工程管理器。点击 **文件** → **新建工程**，浏览到你想要的位置，并给你的工程起个名字，如 "入门"。确保 **为工程创建一个新的文件夹** 复选框被选中，然后点 击**保存**。这将在一个新的子文件夹中创建你的工程文件，其名称与你的工程相同。




在左边，工程文件窗格列出了新工程中的文件。有一个扩展名为 `.kicad_pro` 的工程文件，一个扩展名为 `.kicad_sch` 的原理图文件，以及一个扩展名为 `.kicad_pcb` 的电路板文件。这些文件都与你的工程共享一个名称。

也可能有一个`备份`目录。KiCad 会在你保存时自动创建工程的备份，也可以选择在固定的时间间隔内进行备份。备份设置可以通过进入 **偏好设置** → **偏好设置** → **常用** → **工程备份** 来配置。

| | |
|---|---|
| **NOTE** | Opening the Preferences dialog may trigger a library table configuration dialog. This dialog is explained below. |




在工程窗口的右侧，有一些按钮可以启动 KiCad 提供的各种工具。启动这些工具将自动打开当前项目中的相关设计文件（原理图或 PCB）。首先打开原理图编辑器。

# 教程第 2 部分：原理图

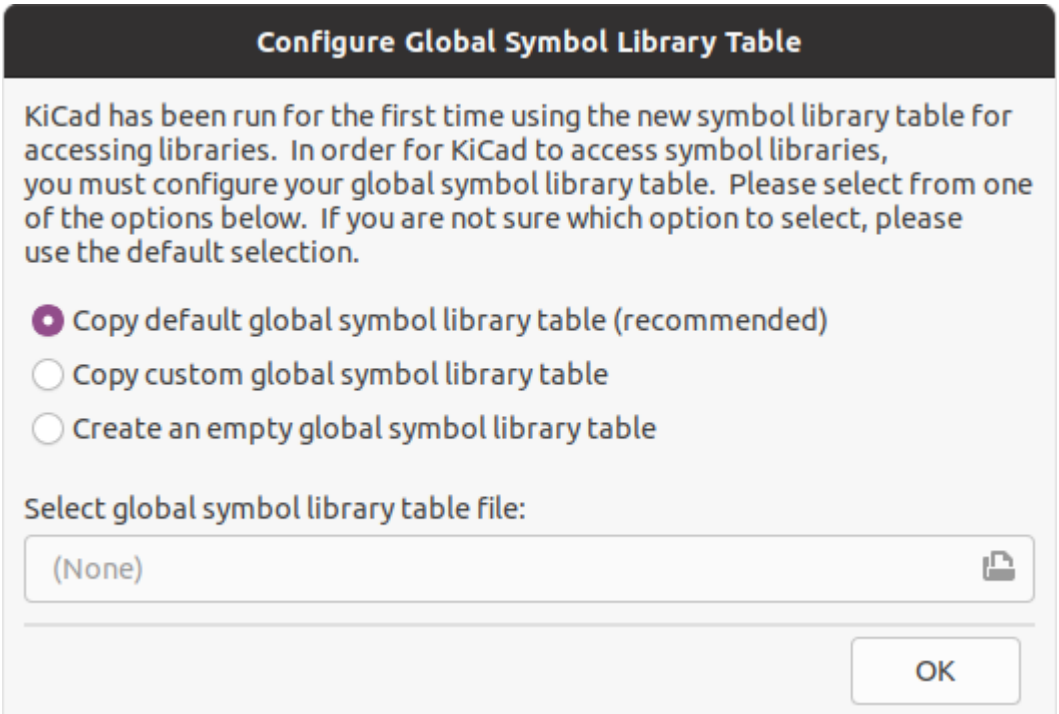
## 符号库表设置

第一次打开原理图编辑器时，会出现一个对话框询问如何配置全局符号库表。符号库表告诉 KiCad 要使用哪些符号库以及它们的位置。如果你已经安装了 KiCad 的默认库，建议你选择默认选项。**复制默认的全局符号库表（推荐）**。

如果 KiCad 无法在其预期的安装位置找到库，该选项将被禁用。在这种情况下，用户应该选择第二个选项，**复制自定义全局符号库表**。点击底部的文件夹按钮，并浏览到下面给出的位置。选择 `sym-lib-table` 文件。

默认库表文件的位置取决于操作系统，并可能根据安装位置而有所不同。下面是每个操作系统的默认值：

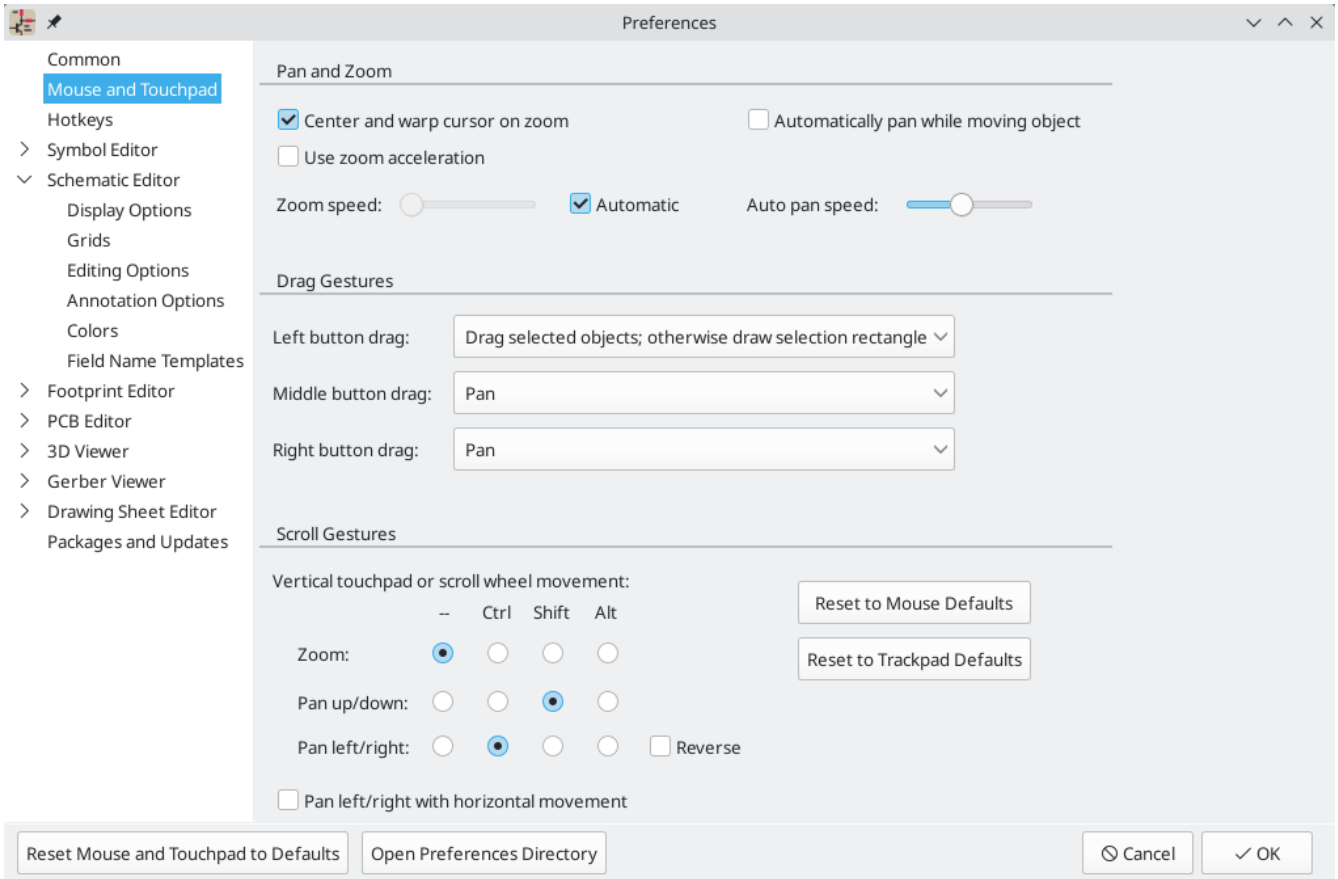
- Windows: `C:\Program Files\KiCad\8.0\share\kicad\template\`
- Linux: `/usr/share/kicad/template/`
- macOS: `/Applications/KiCad/KiCad.app/Contents/SharedSupport/template/`




## 原理图编辑器基础知识

要在原理图上移动，用鼠标中键或鼠标右键点击并拖动。用鼠标滚轮或 F1 和 F2 进行放大和缩小。笔记本电脑用户可能会发现，改变鼠标控制，使其更适合于触摸板；鼠标控制可在 **偏好设置** → **偏好设置...** → **鼠标和触摸板** 中配置。

默认情况下，KiCad 启用了一个名为 **缩放时的中心和经纬光标的鼠标设置**。启用该功能后，当用户放大或缩小时，鼠标光标会自动移动到屏幕中央。这使缩放区域始终保持在中心位置。这个功能是不寻常的，但许多用户一旦习惯了它就会发现它很有用。试着用鼠标光标在画布的不同区域进行放大和缩小。如果默认的缩放行为让人不舒服，请在鼠标和触摸板偏好设置中禁用该功能。

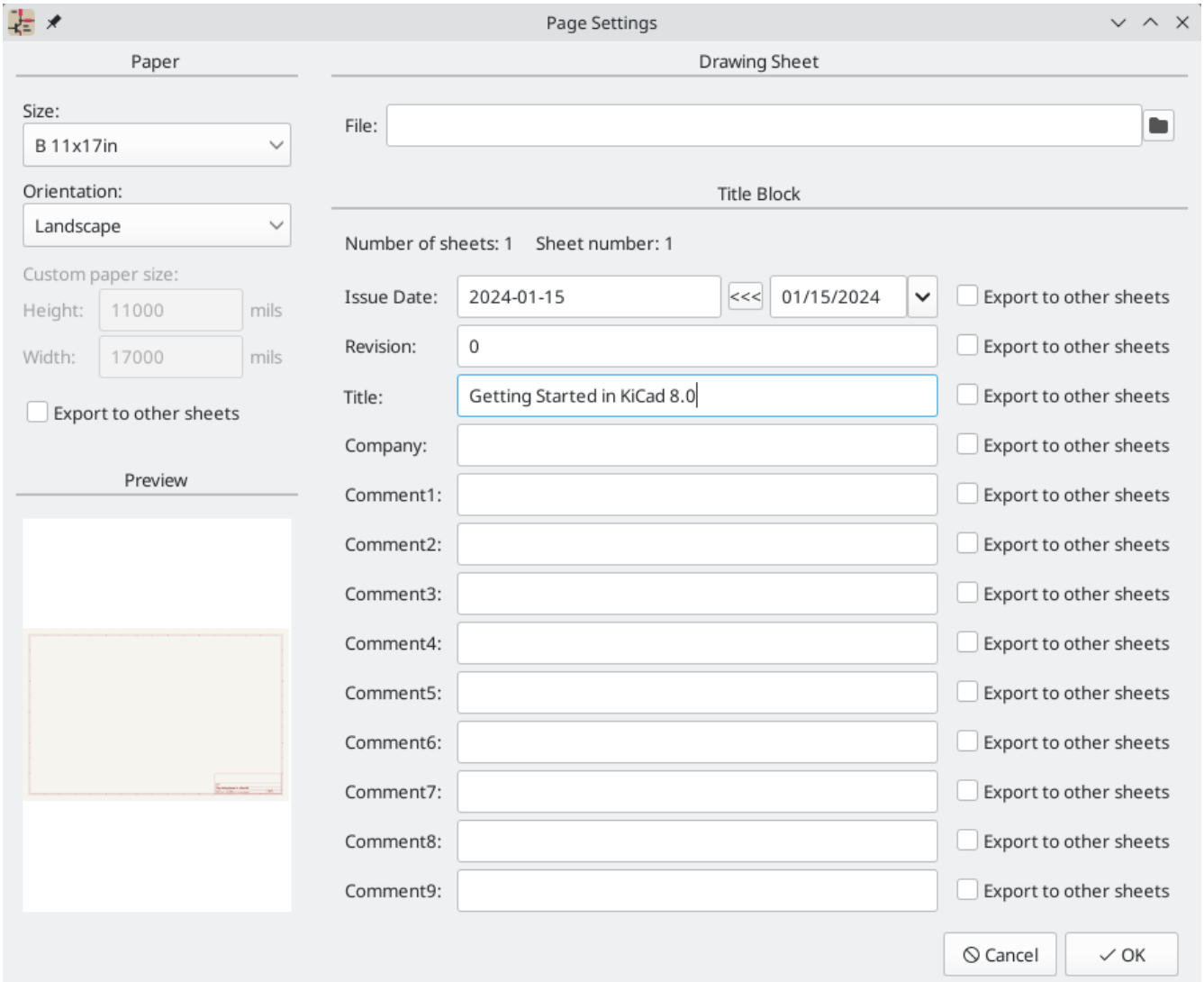



The toolbar at the left side of the schematic editor screen contains basic display settings. The toolbar at the right side of the screen contains tools for editing the schematic.

KiCad 中的大多数工具要么有默认的快捷键，要么可以分配自定义的快捷键。要查看所有快捷键，请进入 **帮助** → **列出快捷键...**。 快捷键可以在 **偏好设置** → **偏好设置...** → **快捷键** 进行更改。

## 原理图图框设置

在绘制原理图中的任何内容之前，先对原理图页面本身进行设置。 点击 **文件** → **页面设置**。给原理图一个标题和日期，如果需要的话，改变原理图大小。




## 将符号添加到原理图中

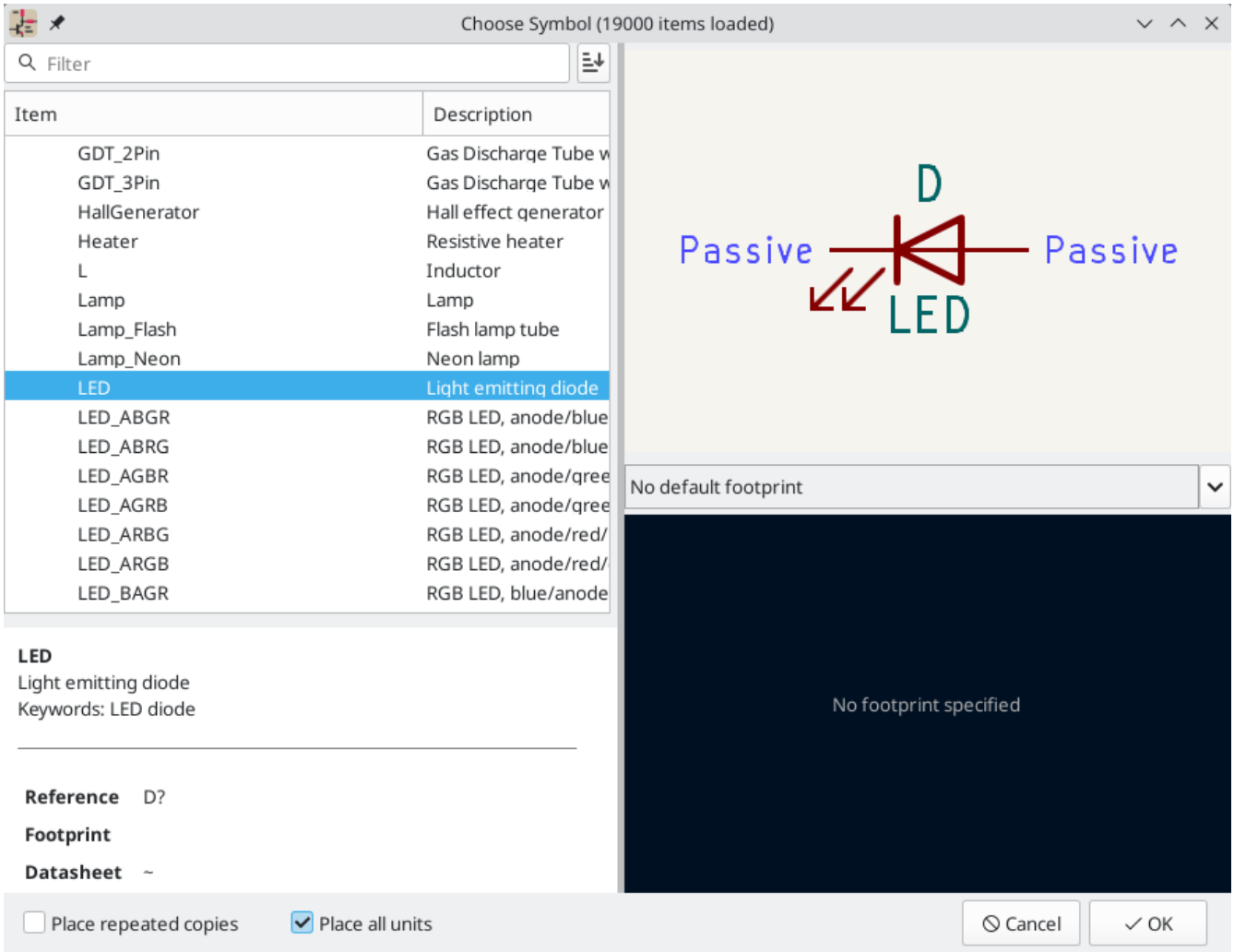
Start making the circuit by adding some symbols to the schematic. Open the Choose Symbol dialog by clicking the **Add Symbols** button on the right side of the window or pressing A.

This action will trigger the Footprint Library Table Setup dialog if it has not already been triggered. This dialog is equivalent to the Symbol Library Table Setup dialog explained earlier, but for footprints instead of symbols.

同样，选择默认选项。**复制默认全局封装库表（推荐）**。如果这个选项被禁用，请选择第二个选项，**复制自定义全局封装库表**。点击底部的文件夹按钮，浏览到《符号库表默认位置，符号库表设置说明》中给出的位置。 选择 `fp-lib-table` 文件并点击 **OK**。

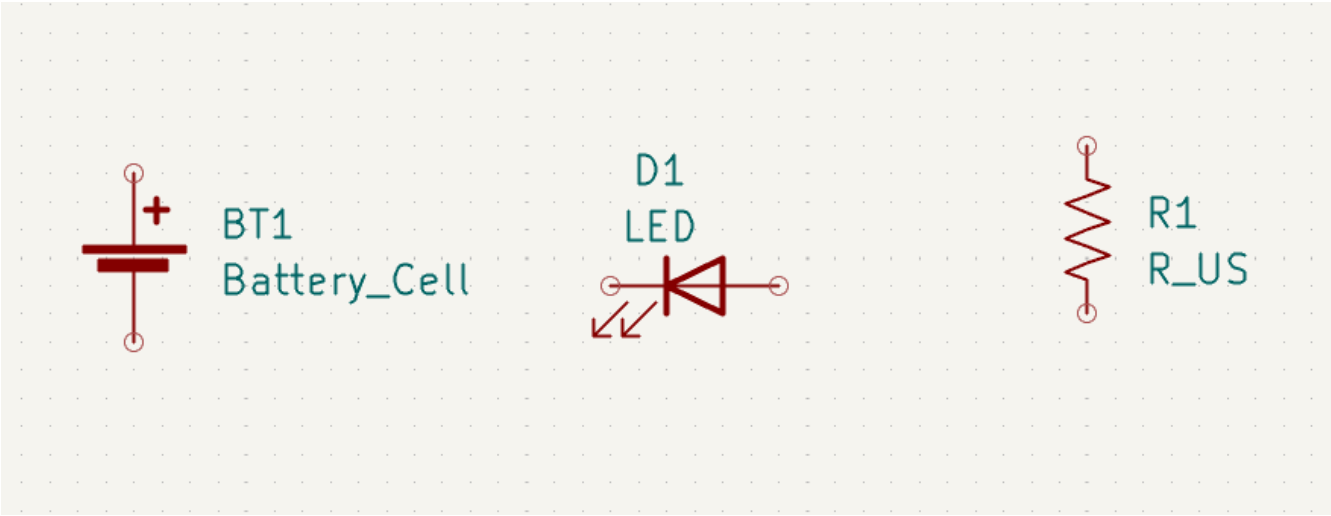
选择符号对话框列出了可用的符号库和其中包含的元件符号。基本设备如无源元件、二极管和其他通用符号在 `Device` 库中找到。特定的设备，如一个特定的 LED，可能在其他库中找到。

向下滚动到 `Device` 库，展开它，并选择 `LED` 符号。 点击 **确定**，然后再次点击，将该符号放入原理图中。



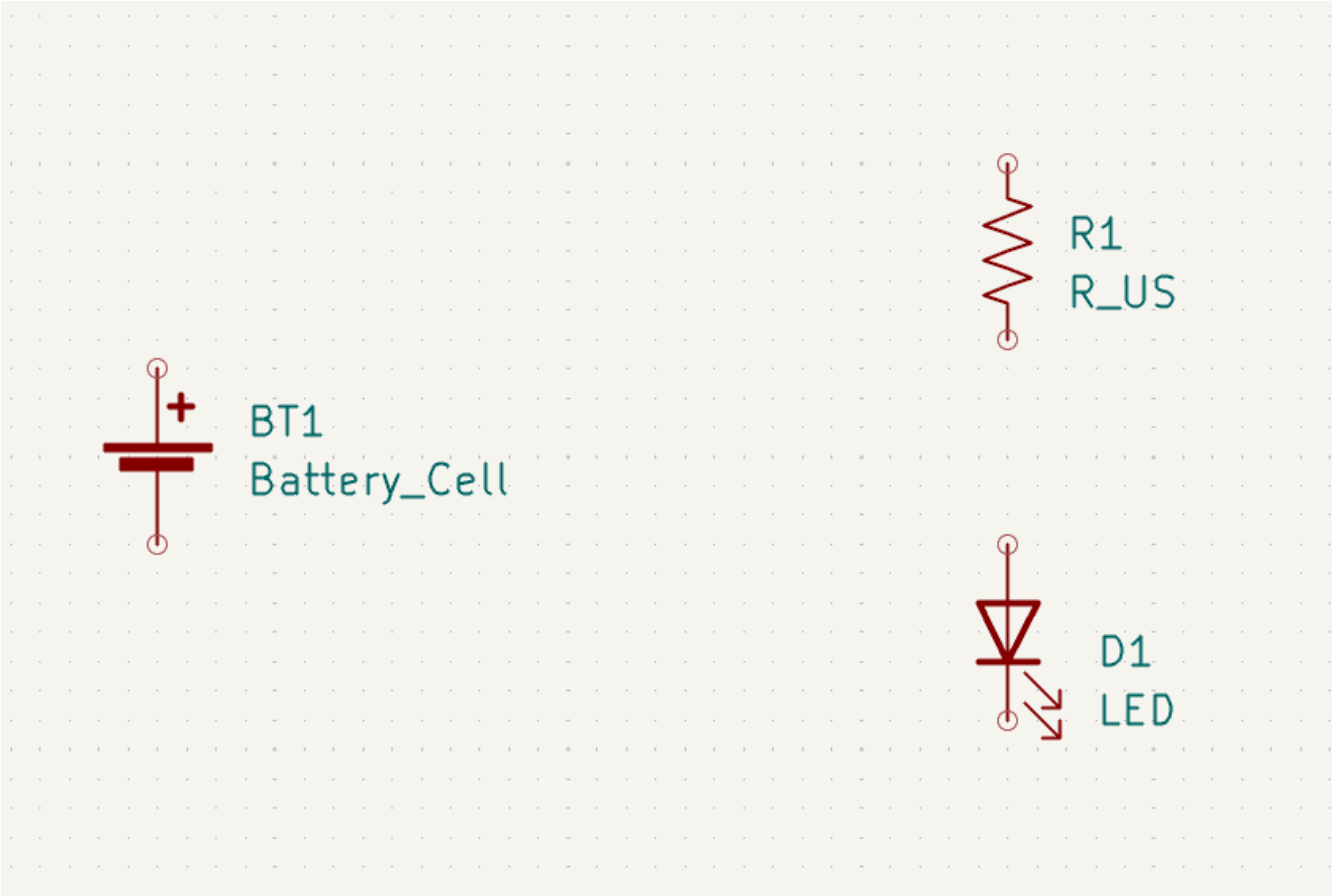

接下来，添加一个限流电阻。回到符号选择器，但这次尝试通过在顶部的过滤框中输入 `R` 来搜索一个电阻。 同样，它可以在 `Device` 库中找到。 `R` 设备是一个 IEC 风格的矩形电阻符号。对于喜欢 ANSI 风格人字形符号的用户，也有一个 `R_US` 符号。选择一个电阻符号并将其添加到原理图中。

最后，添加一个电池为LED供电。 `Device` 库有一个合适的 `Battery_Cell` 符号。




## 选择和移动对象

接下来，如截图所示，正确定位符号之间的关系。通过选择、移动和旋转符号来做到这一点。




In KiCad 8.0, objects are selected by clicking on them. Additional objects can be added to the selection with Shift+click, or removed with Ctrl+Shift+click (macOS: Cmd+Shift+click). You can toggle an item's selection state with Ctrl+click (macOS: Cmd+click).

拖动选择也是可以的；从左到右拖动可以选择完全被选择框包围的对象，而从右到左拖动也可以选择部分被选择框包围的对象。 Shift, Ctrl+Shift (Cmd+Shift), 和 Ctrl (Cmd) 也可以和拖动选择一起使用，分别从选择中增加、减少或切换。

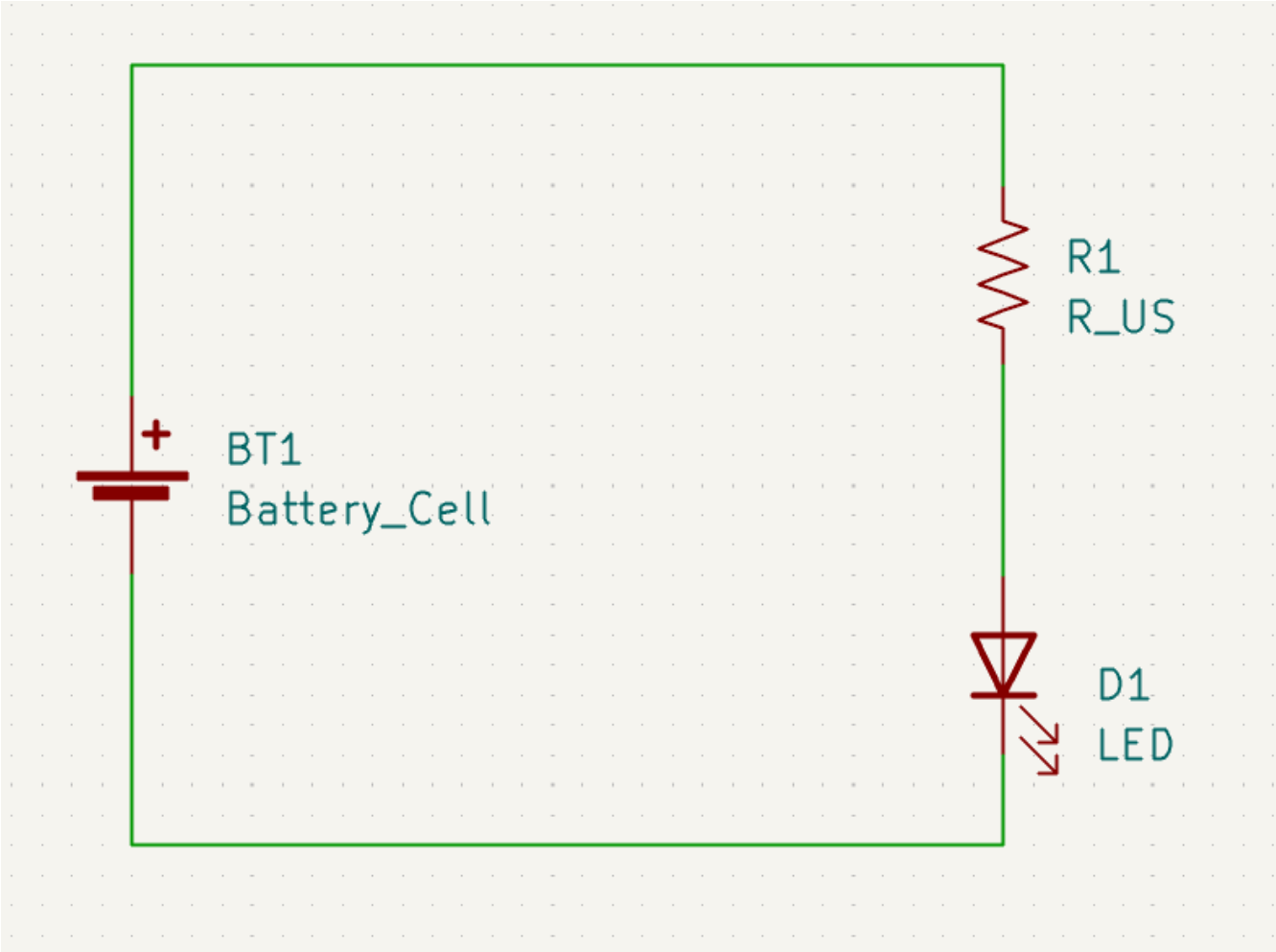
注意，可以选择整个符号（通过点击符号形状本身）或选择符号中的一个文本字段而不选择符号的其他部分（通过点击文本）。当只选择一个文本字段时，所执行的任何操作将只作用于所选择的文本，而不作用于符号的其他部分。

按 M 可以移动选中的对象，按 R 可以旋转。 G 热快捷键（拖动）也可用于移动对象。对于移动未连接的符号，G 和 M 的行为是相同的，但对于连接有导线的符号， G 移动符号并保持导线连接，而 M 移动符号并留下导线。选定的对象可以用 Del 键删除。

## 为原理图布线

符号引脚上都有小圆圈，表明它们没有连接。通过在符号引脚之间画导线来解决这个问题，如截图中所示。点击右侧工具栏上的 **添加导线** 按钮 或使用 W 快捷键。点击开始画导线，通过点击符号引脚或双击任何地方完成画导线。按 Escape 键将取消画导线。

另一种方便的画导线方法是将鼠标悬停在一个未连接的引脚上。 鼠标指针会改变，表示可以从该位置开始画导线。点击该引脚就会自动开始画导线。

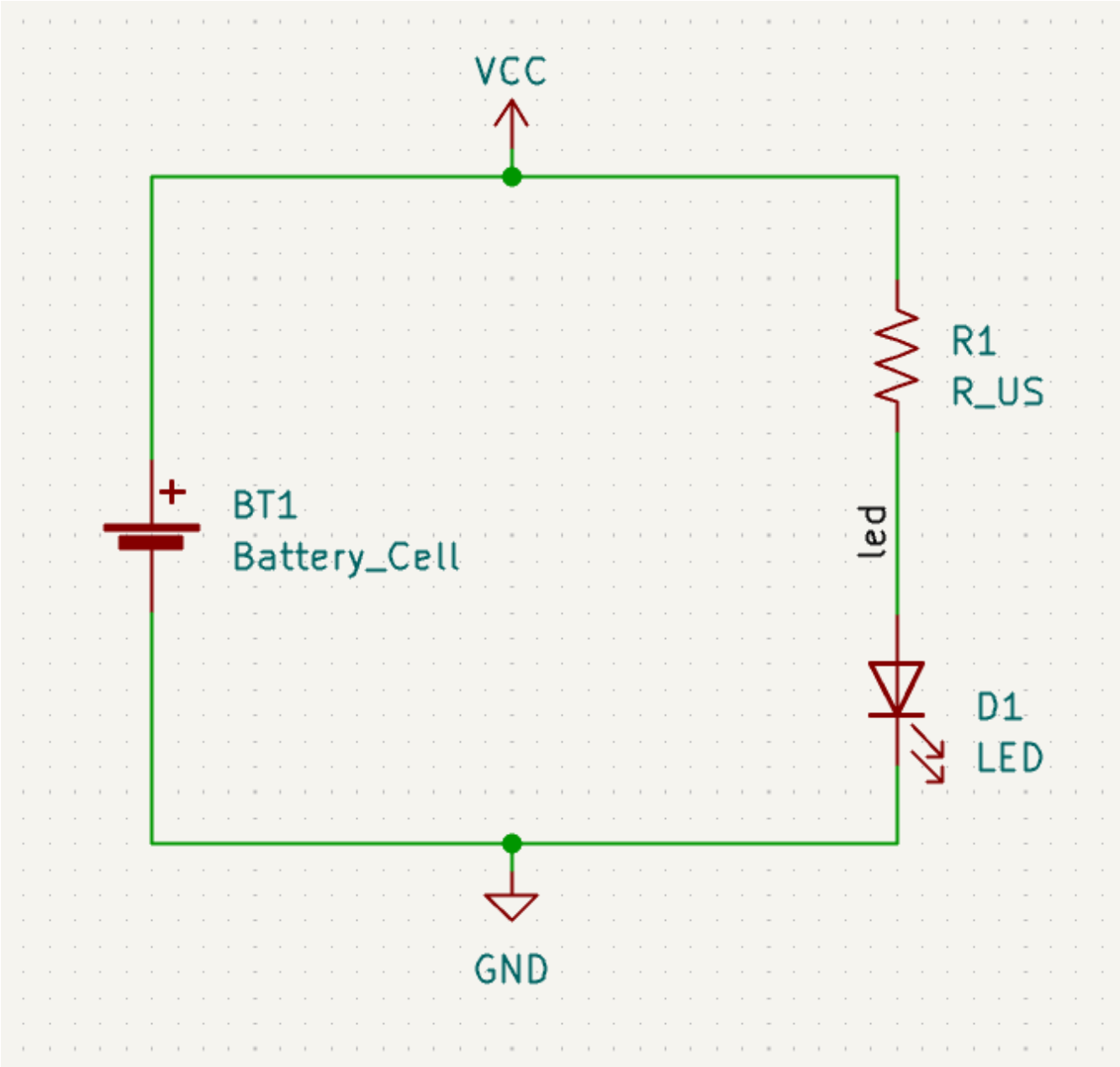



接下来，在原理图上添加电源和接地符号。虽然在这样一个简单的原理图中没有严格意义上的必要，但这些符号使大型原理图更容易理解。

在 `Power` 符号库中，有许多电源和地线符号。 然而，有一个添加这些符号的快捷方式：点击 **添加电源符号*** 按钮 或使用 `P` 快捷键。这将弹出 **选择符号*** 对话框，但只显示包含电源符号的符号库。

添加一个 VCC 符号和 GND 符号，然后用导线将它们连接到电路上。

Finally, add a label to the wire between the LED and resistor. Again, this may not be necessary in a simple circuit, but it is good practice to label important nets. Click the **Draw Net Labels** button in the right toolbar (`L`), type a label name (`led`), and place the label into the schematic so that the square attachment point overlaps with the wire. Rotate and align the label as necessary.




请注意，同名的标签和电源符号是连接在一起的。 在这个原理图页面上，另一个 GND 符号或标记为` led` 的导线将与现有的导线短路，即使没有导线将它们直观地连接起来。

## 批注、符号属性和封装

### 批注

每个符号都需要为其分配一个独特的位号。这个过程也被称为注解。

By default in KiCad 8.0, symbols are automatically annotated when they are added to the schematic. Automatic annotation can be enabled or disabled using the button in the left toolbar.

虽然在本指南中没有必要，但可以使用顶部工具栏中的 **填写原理图符号位号** 按钮（ ）对符号进行手动批注或重新批注。

### 符号属性

接下来，填入每个元件的数值。选择 LED，右键单击，并选择属性...（ E ）。这个工程将使用一个红色的 LED，所以将 `值` 字段改为 `红色`。在一个真正的工程中，在这里写上 LED 制造商的零件编号可能会更好。注意，可以在每个

符号的属性中单独编辑位号。

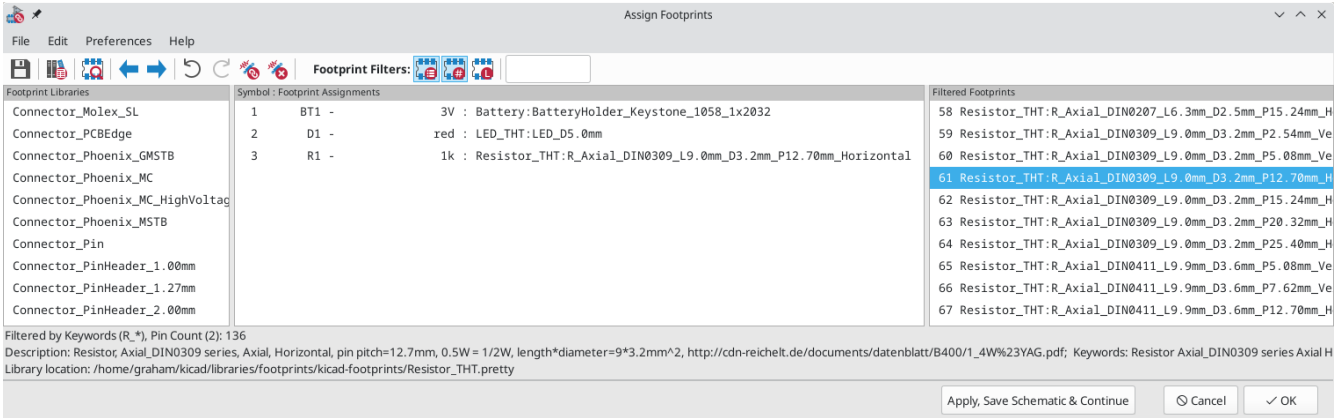
这个工程将使用一个 3V 的纽扣电池，所以把 `BT1` 的 `值` 字段改为 `3V'`。将电阻的值改为 `1k`。

## 封装分配

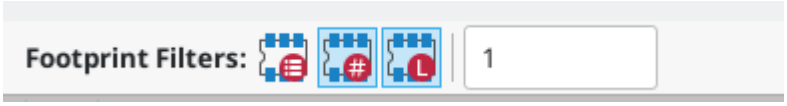
最后，给每个元件指定一个封装。这定义了每个元件将如何连接到 PCB 上。有些符号预留了封装，但对许多元件来说，有多种可能的封装，所以用户需要选择适当的一个。

有几种分配封装的方法，但有一种方便的方法是使用封装分配工具，点击顶部工具栏的 按钮。

The left pane of this window lists the available footprint libraries. The middle pane shows the symbols in the schematic. All of these symbols will get footprints assigned to them. The right pane shows the footprints that can be chosen for the symbol selected in the middle pane. To assign a footprint to a symbol, select the symbol in the middle pane and double click on the desired footprint in the right pane. You can preview a footprint by right-clicking the footprint's name and selecting **View selected footprint**.




KiCad 中包含了许多封装，因此封装分配工具提供了几种方法来筛选掉与有关符号不相关的封装。




- 最左边的按钮 可以激活每个符号中可以定义的筛选器。例如，一个运算放大器符号可以定义筛选器，只显示 SOIC 和 DIP 封装。有时，这些预定义的筛选器缺失或限制性太强，所以在某些情况下，关闭这个筛选器可能会很有用。
- 中间的按钮 按引脚数筛选，所以只有 8 个引脚的符号才会显示 8 个引脚的封装。这个筛选器几乎总是有用的。
- 右边的按钮 按所选库筛选。库在左侧窗格中被选中；不在所选库中的封装将被筛选掉。只要知道哪个库包含正确的封装，这个筛选器就很有用。通常情况下，最好使用这个筛选器或符号筛选器，但不能同时使用。
- 文本框筛选掉与框中文本不匹配的封装。当方框为空时，这个筛选器被禁用。

Using the filters, find each of the footprints shown in the central column in the screenshot above. Assign each footprint to a symbol by selecting the symbol in the central column and then double clicking the appropriate footprint in the right column. Once all of the footprints have been assigned, click **OK**.

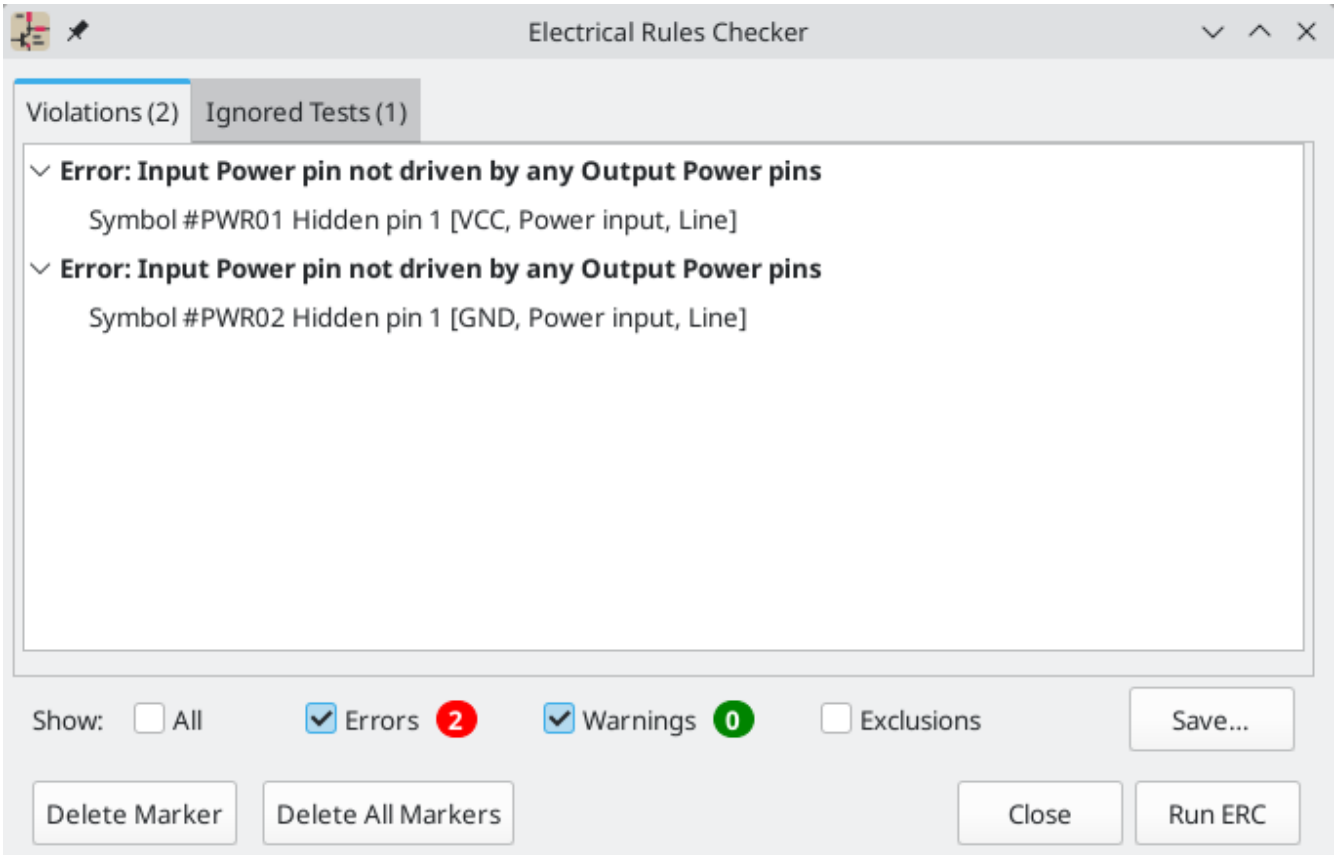
There are other ways to assign footprints; one way is through the symbol properties window. For more information on assigning footprints, see the manual.

## 电气规则检查

The last remaining thing to do in the schematic is to check for electrical errors. KiCad's Electrical Rules Checker (ERC) cannot make sure that the design in the schematic will work, but it can check for some common connection issues such as unconnected pins, two power outputs shorted together, or a power input that isn't powered by anything. It also checks for some other mistakes like symbols that aren't annotated and typos in net labels. To see the full list of electrical rules and to adjust their severity, go to **File** → **Schematic Setup...** → **Electrical Rules** → **Violation Severity**. It is a good idea to run ERC before starting layout.

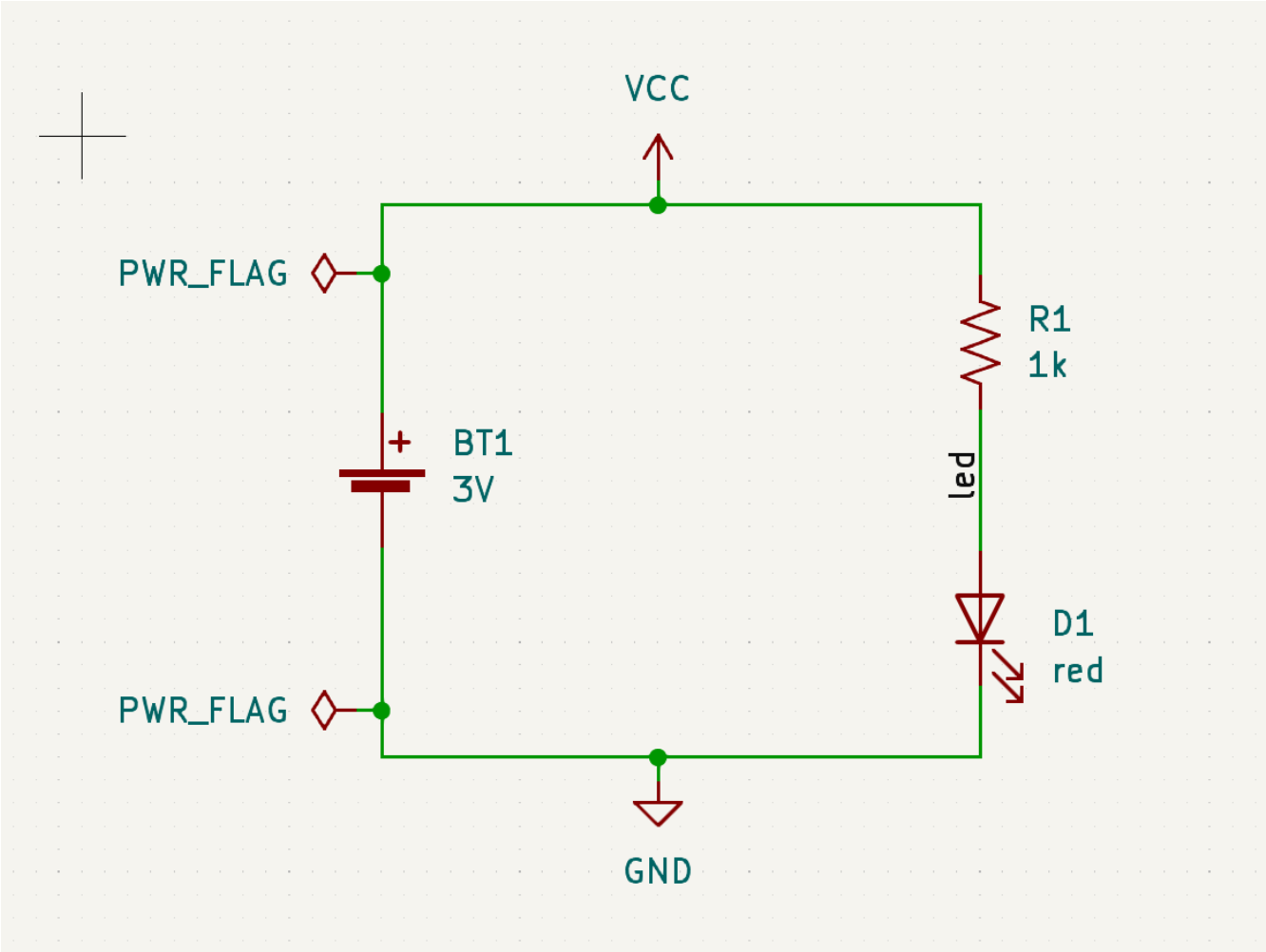
通过点击顶部工具栏的 **ERC** 按钮（ ），然后点击 **运行 ERC**，运行电气规则检查。

即使在这个简单的原理图中，KiCad 也发现了两个潜在的错误。这些错误列在 ERC 窗口中，箭头指向原理图中的违规位置。在 ERC 窗口中选择一个错误，就会显示出相应的箭头。




你可以通过右键点击每条错误信息来排除个别违规行为或忽略整个违规类别。然而，为了得到一份干净的 ERC 报告，避免遗漏真正的问题，通常值得处理这些违规行为，即使它们不是实际的设计错误。

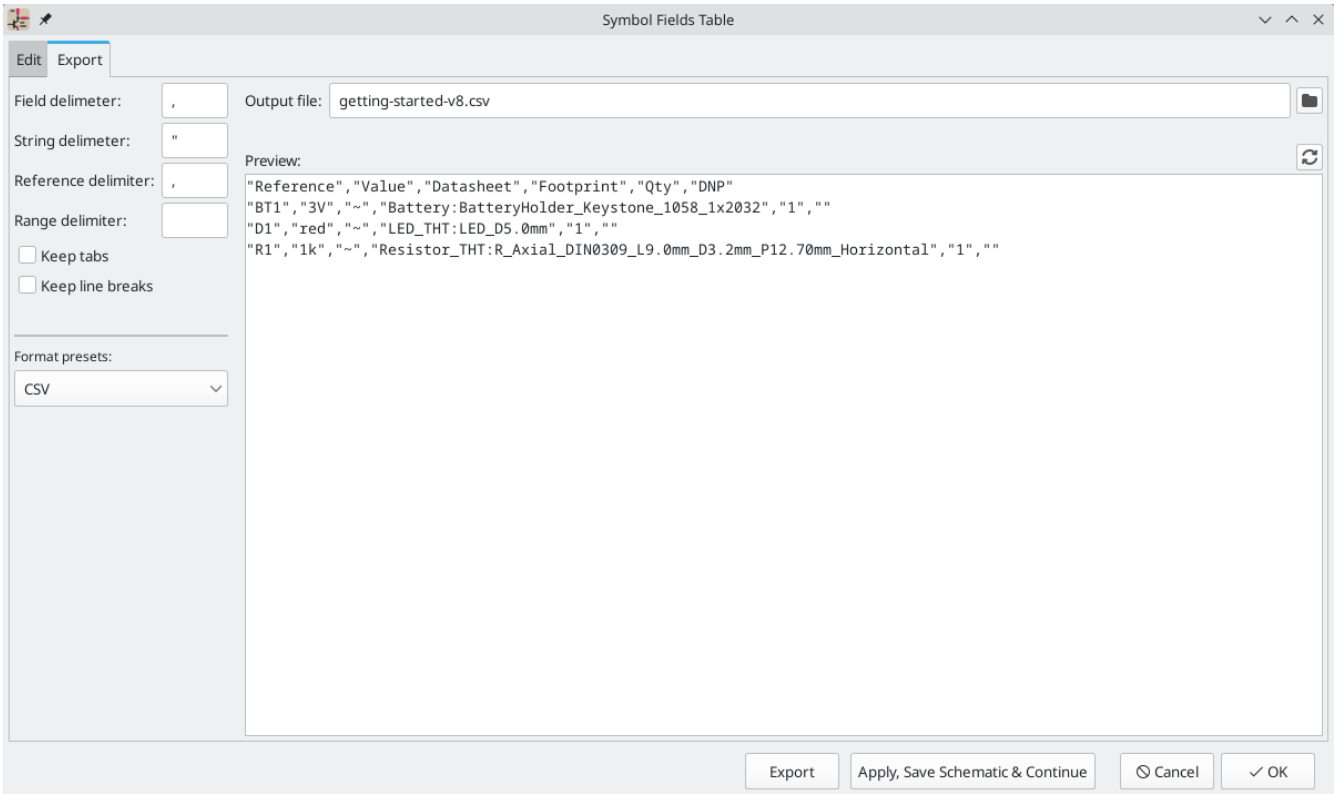
在这种情况下，KiCad 对 `VCC` 和 `GND` 网络报告 "输入电源引脚没有被任何输出电源引脚驱动"。这是一个常见的 KiCad ERC 错误。 电源符号被设置为需要一个电源输出引脚，如电压调节器的输出，在同一网络上；否则 KiCad 认为该网络未被驱动。 对人来说，`VCC` 和 `GND` 显然是由电池驱动的，但有必要在原理图中明确显示。




在 `Power` 符号库中有一个特殊的 `PWR_FLAG` 符号，用于解决这个问题，告诉 KiCad 这些网络实际上是被驱动的。将这个符号添加到 `VCC` 和 `GND` 网络上，然后重新运行 ERC。当 ERC 通过而没有任何违反时，原理图就完成了。

## 物料清单

A final optional step is to generate a Bill of Materials listing all components used in the project. Click **Tools → Generate Bill of Materials...**.




KiCad 8.0 has a GUI for BOM export. You can configure which symbol metadata will be exported, and how the symbols are grouped, in the **Edit** tab, and configure the output format in the **Export** tab.

For now, the default settings should be fine, but you can see a raw view of what will be exported in the preview pane. Specify an output file in the **Export** tab, then press the **Export** button.

The BOM should look like the below image when it is opened in a spreadsheet program.

| | A | B | C | D | E | F |
|---|---|---|---|---|---|---|
| 1 | Reference | Value | Datasheet | Footprint | Qty | DNP |
| 2 | BT1 | 3V | ~ | Battery:BatteryHolder_Keystone_1058_1x2032 | 1 | |
| 3 | D1 | red | ~ | LED_THT:LED_D5.0mm | 1 | |
| 4 | R1 | 1k | ~ | Resistor_THT:R_Axial_DIN0309_L9.0mm_D3.2mm_P12.70mm_Horizontal | 1 | |
| 5 | | | | | | |

# 教程第 3 部分：电路板

原理图完成后，回到工程窗口并打开 PCB 编辑器，可以通过点击 PCB 编辑器按钮或打开电路板文件。

## PCB 编辑器基础知识

PCB 编辑器中的导航与原理图编辑器相同：用鼠标中键或右键拖动平移，用滚轮或 F1 / F2 缩放。

PCB 编辑器的主要部分是一块画布，将在这里设计电路板。 左边的工具栏有各种板子的显示选项，包括单位和布线、过孔、焊盘和区域的边框/填充显示模式的切换。画布右边的工具条包含了设计 PCB 的工具。

**NOTE**

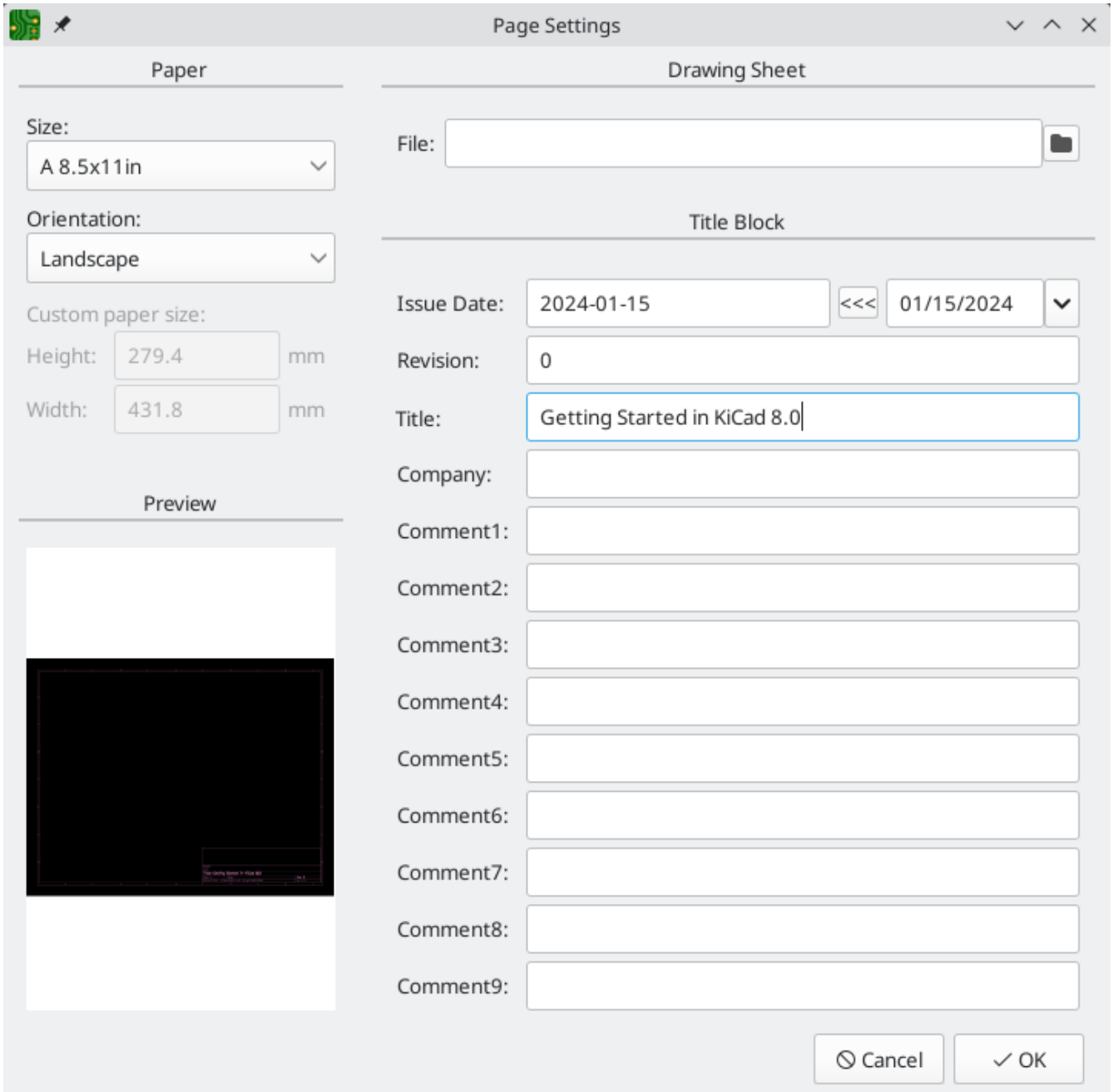
右边工具栏上的一些按钮在右下角有一个小三角形 。这个三角形表示该按钮有一个扩展的调色板，包含几个相关的工具，比如说不同种类的尺寸。要选择一个备用工具，请点击并按住按钮直到调色板出现，然后点击备用工具。另一种使用调色板的方法是点击按钮并向左拖动，直到调色板出现，然后在所需的工具被突出显示时释放鼠标按钮。

最右边的是外观面板和选择筛选器。外观面板用于改变 PCB 层、对象和网络的可见性、颜色和不透明度。通过点击层的名称可以改变活动层。

在外观面板下面是选择筛选器，它可以启用和禁用各种类型的 PCB 对象的选择。这对于在拥挤的布局中选择特定的项目很有用。

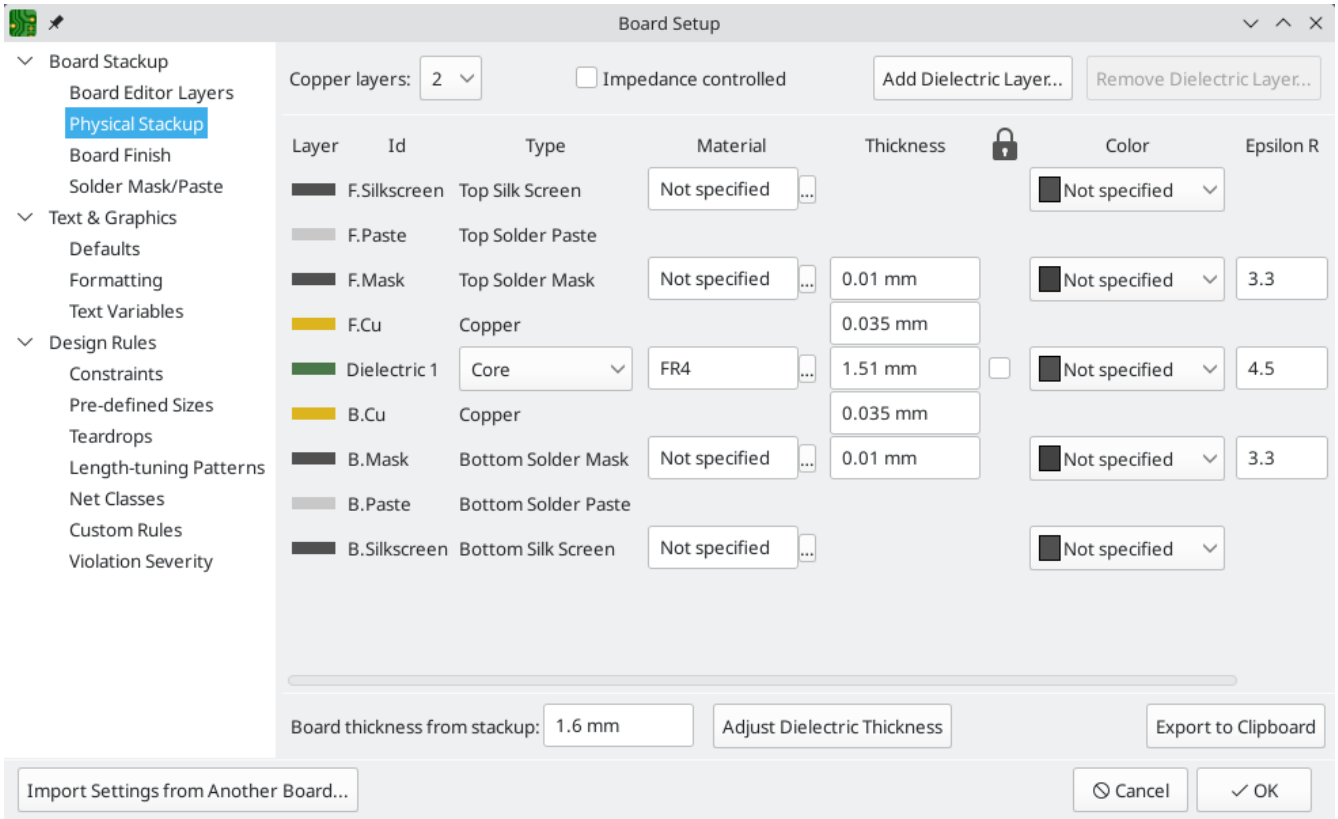
## 板子的设置和压层

在设计电路板报之前，先设置页面大小，并在标题栏中添加信息。点击 **文件** → **页面设置...**，然后选择一个合适的纸张尺寸，并输入日期、修订和标题。

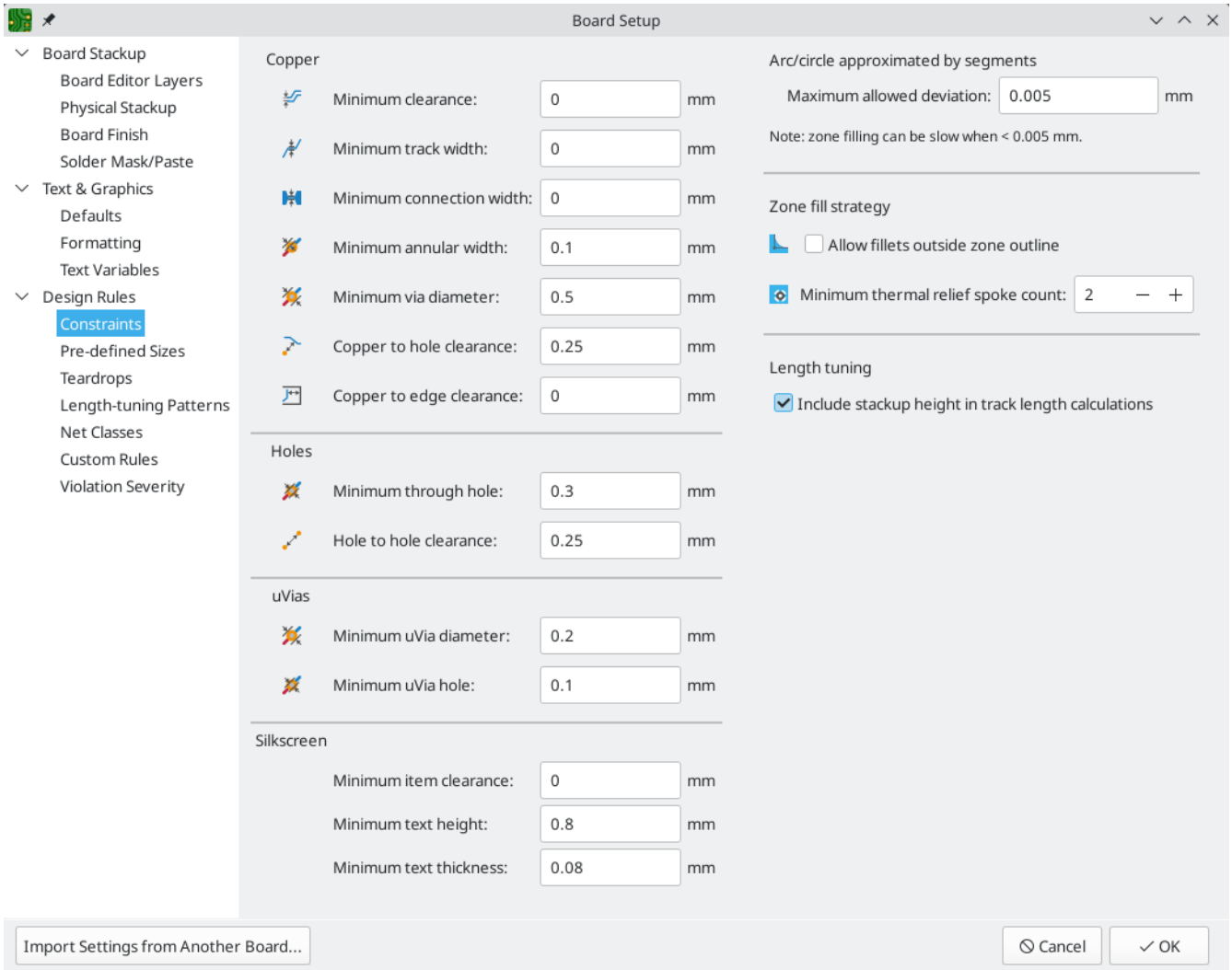



接下来，进入 **文件** → **电路板设置...**，定义如何制造 PCB。最重要的设置是压层，即 PCB 将有哪些铜层和介质层（以及它们的厚度），以及设计规则，例如布线和过孔的尺寸和间距。

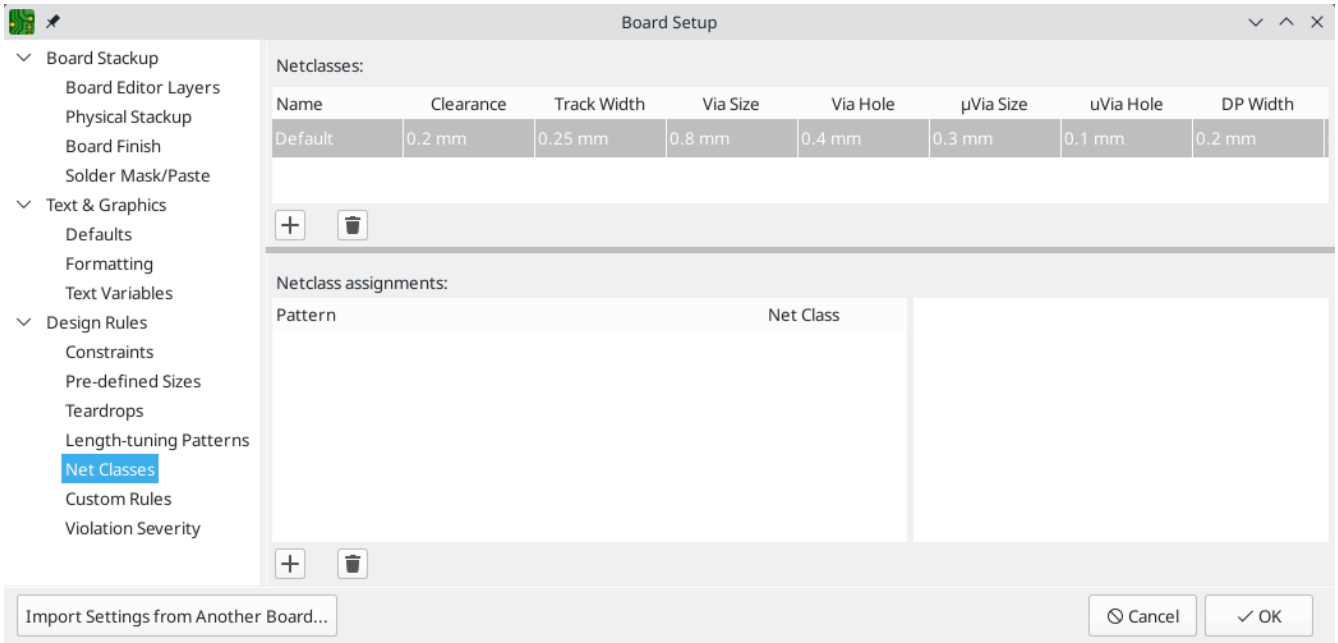
要设置压层，打开电路板设置窗口的 **电路板压层** → **物理压层** 页面。在本指南中，将铜层的数量定为 2，但更复杂的工程可能需要更多的铜层。




接下来，进入 **设计规则** → **约束** 页面。这个页面上的设置为电路板设计中的所有内容指定了首要的设计规则。就本指南的目的而言，默认值是好的。然而，对于一个真正的工程来说，应该根据 PCB 工厂的能力来设置这些规则，这样才能使 PCB 设计具有可制造性。




最后，打开 **设计规则** → **网络类** 页面。网络类是一组与特定网络组相关的设计规则。这个页面列出了设计中每个网络类的设计规则，并允许将网络类分配给每个网络类。



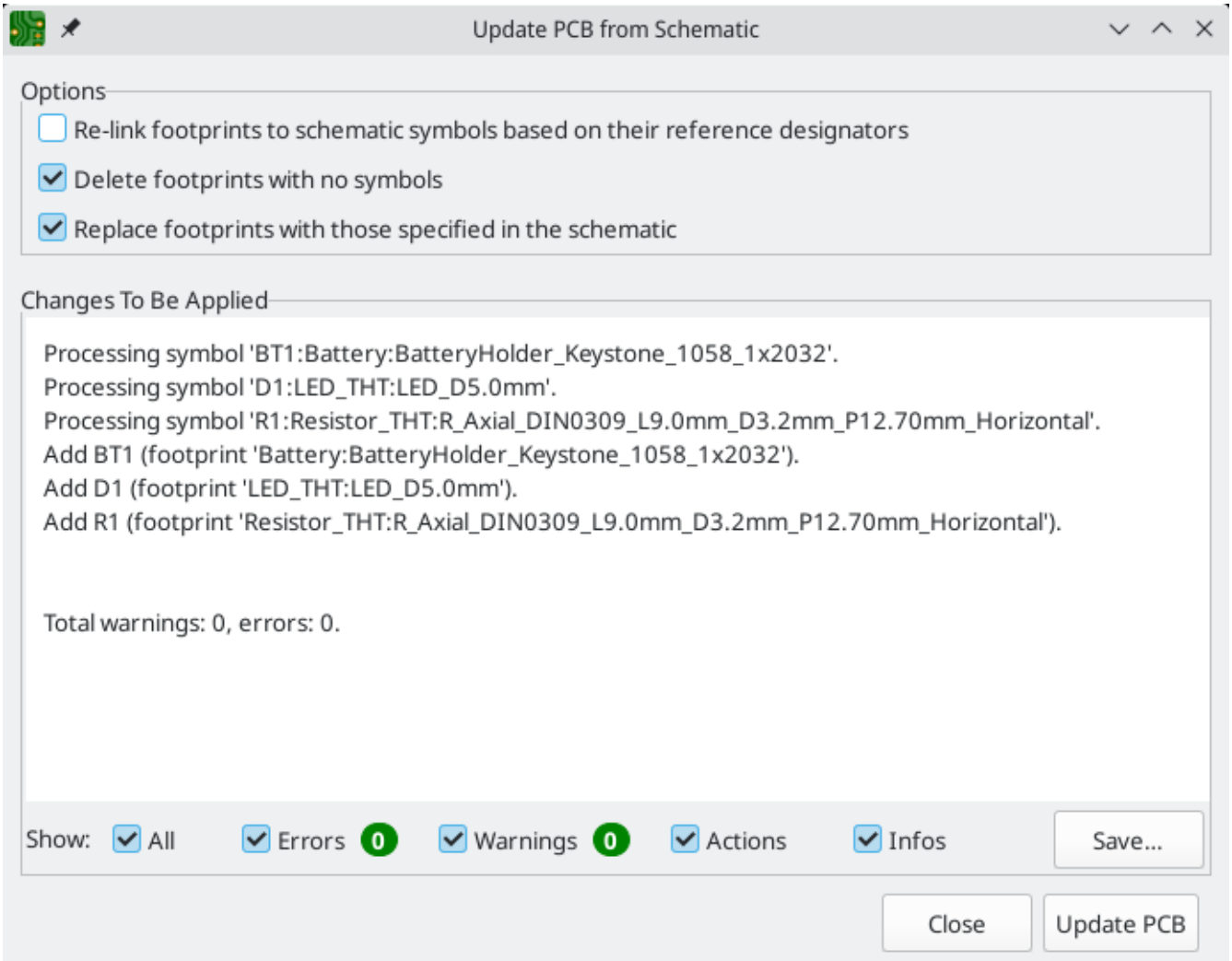

布线的宽度和间距可以由设计者在布局过程中手动管理，但推荐使用网络类，因为它们提供了一种自动管理和检查设计规则的方法。

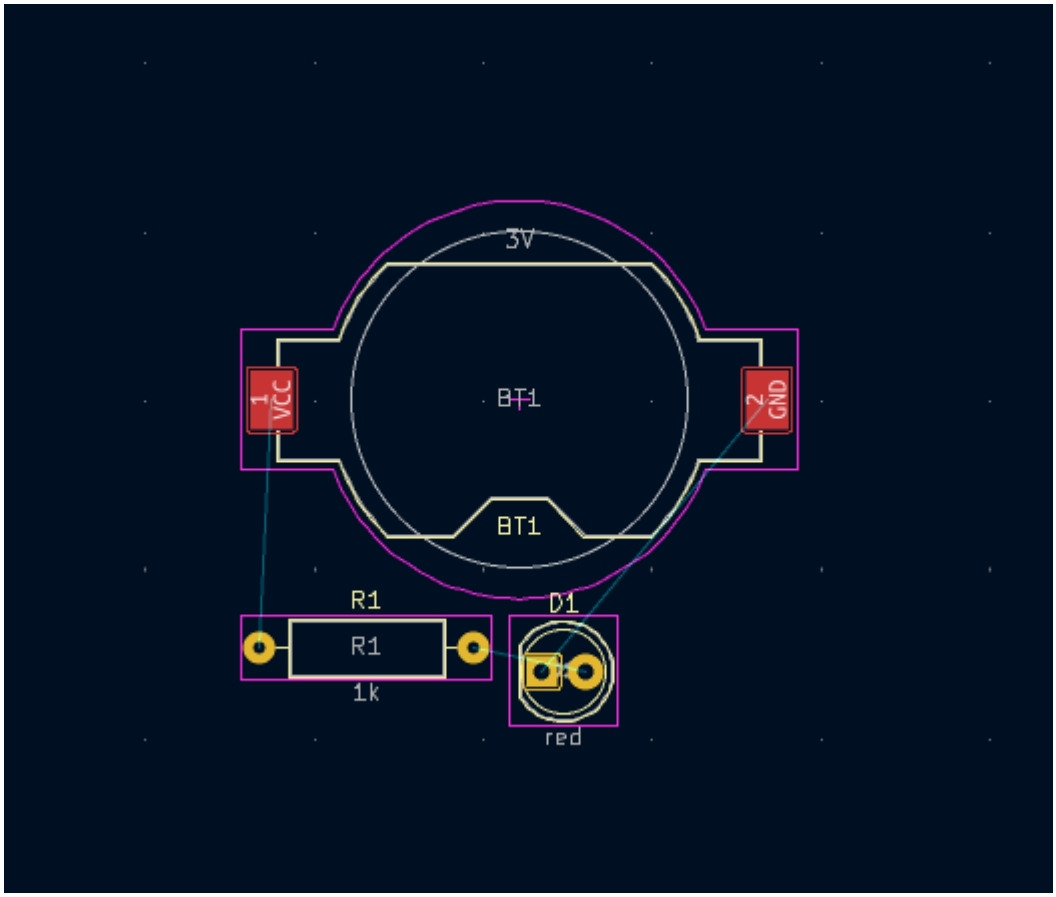
在这个设计中，没有指定网络类，所以所有的网络都属于 `默认` 网络类。这个网络类的默认设计规则对这个工程来说是可以接受的，但是其他设计可能有多个网络类，每个网络类有不同的设计规则。例如，一块板子可能有一个宽线 `高电流` 网络类，有，或者一个 `50 ohm` 网络类，有特定的宽度和 50 欧姆控制阻抗线的间隙规则。

## 从原理图中导入更改

原理图已经完成，但布局中还没有任何元件。 要从原理图中导入设计数据到布局中，请点击 **工具** → **从原理图更新 PCB...**，或按 F8。在顶部的工具栏上还有一个按钮 。

仔细阅读 **要应用的改变** 窗口中的信息，其中会说原理图中的三个元件将被添加到电路板上。点击 **更新 PCB**，**关闭**，然后点击画布来放置这三个封装。每个封装相对于其他封装的位置将在以后被改变。



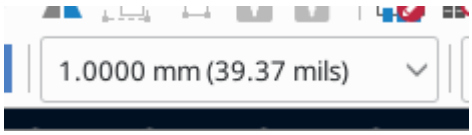



在 KiCad 中，根据原理图的变化更新 PCB 是一个手动过程：设计者决定何时适合根据原理图的修改来更新 PCB。每次编辑原理图时，设计者必须使用 **从原理图更新 PCB** 工具来保持原理图和布局的同步。

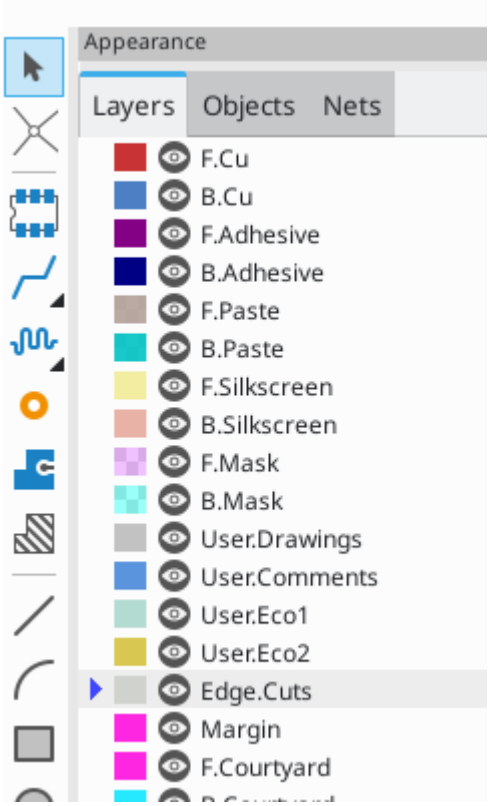
## 绘制电路板边框

现在这三个元件已经被放置好了，但是电路板本身还没有被定义。通过在 `Edge.Cuts` 层上画一个板的边框来定义电路板。

用粗大的网格来绘制电路的板框通常是很有用的，这样可以很容易得到画布尺寸的整数。在画布上方的网格下拉菜单中选择 1 毫米，就可以切换到粗略的网格。




To draw on the `Edge.Cuts` layer, click **Edge.Cuts** in the Layers tab of the Appearance panel at right. Choose the rectangle tool in the right toolbar, click on the canvas to place the first corner, then click again to place the opposite corner so that the rectangle roughly surrounds the three footprints. The other graphic tools (line , arc , circle , polygon , or a combination of them) could also be used to define the board outline; the only requirement is that the outline is a single closed shape that doesn't intersect itself.



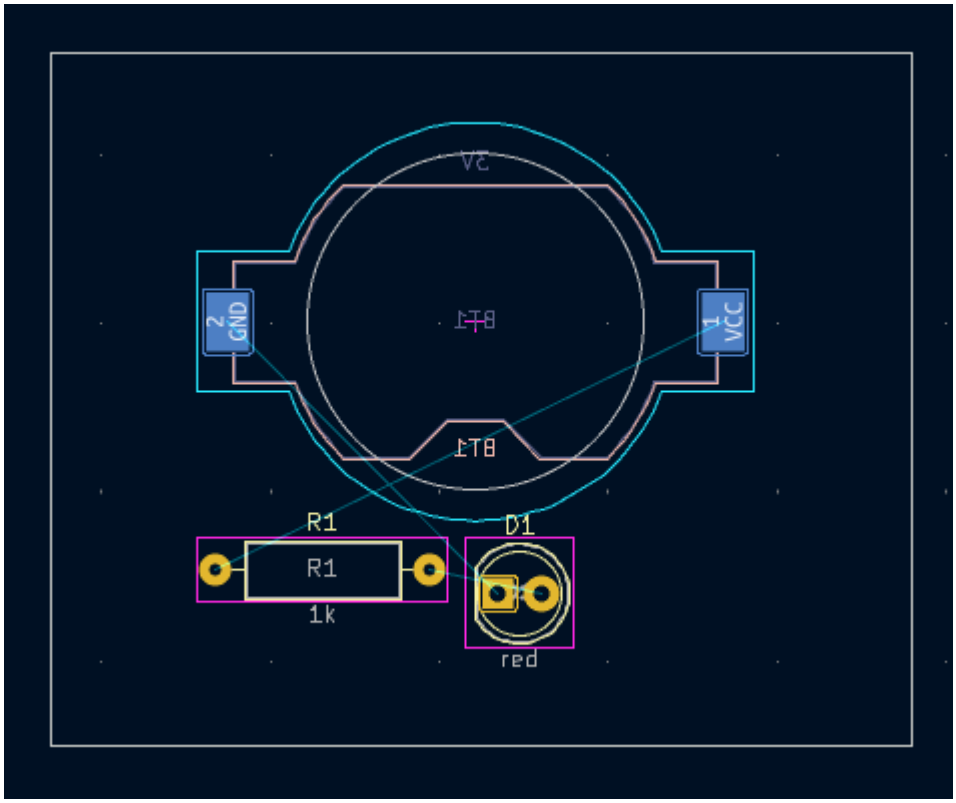

## 放置封装

布局过程中的下一步是在电路板上安排封装。 一般来说，在定位封装时有几个考虑因素：

- 有些封装可能对其位置有确切的要求，如连接器、指示灯或按钮和开关。
- 一些元件可能需要根据电气方面的考虑来放置。 旁路电容应靠近相关 IC 的电源引脚，敏感的模拟元件应远离数字干扰。
- 几乎所有的元件都有一个 "边框"（或两个，如果前面和后面都有定义）。一般来说，边框不应该相交。
- 否则元件的位置应便于布线。连接的元件一般应紧靠在一起，并将布线的复杂性降到最低。飞线（表示焊盘之间连接的细线）对于确定如何将封装相对于其他封装的最佳位置很有用。

在本指南中，唯一的放置目标是使布线过程尽可能简单。

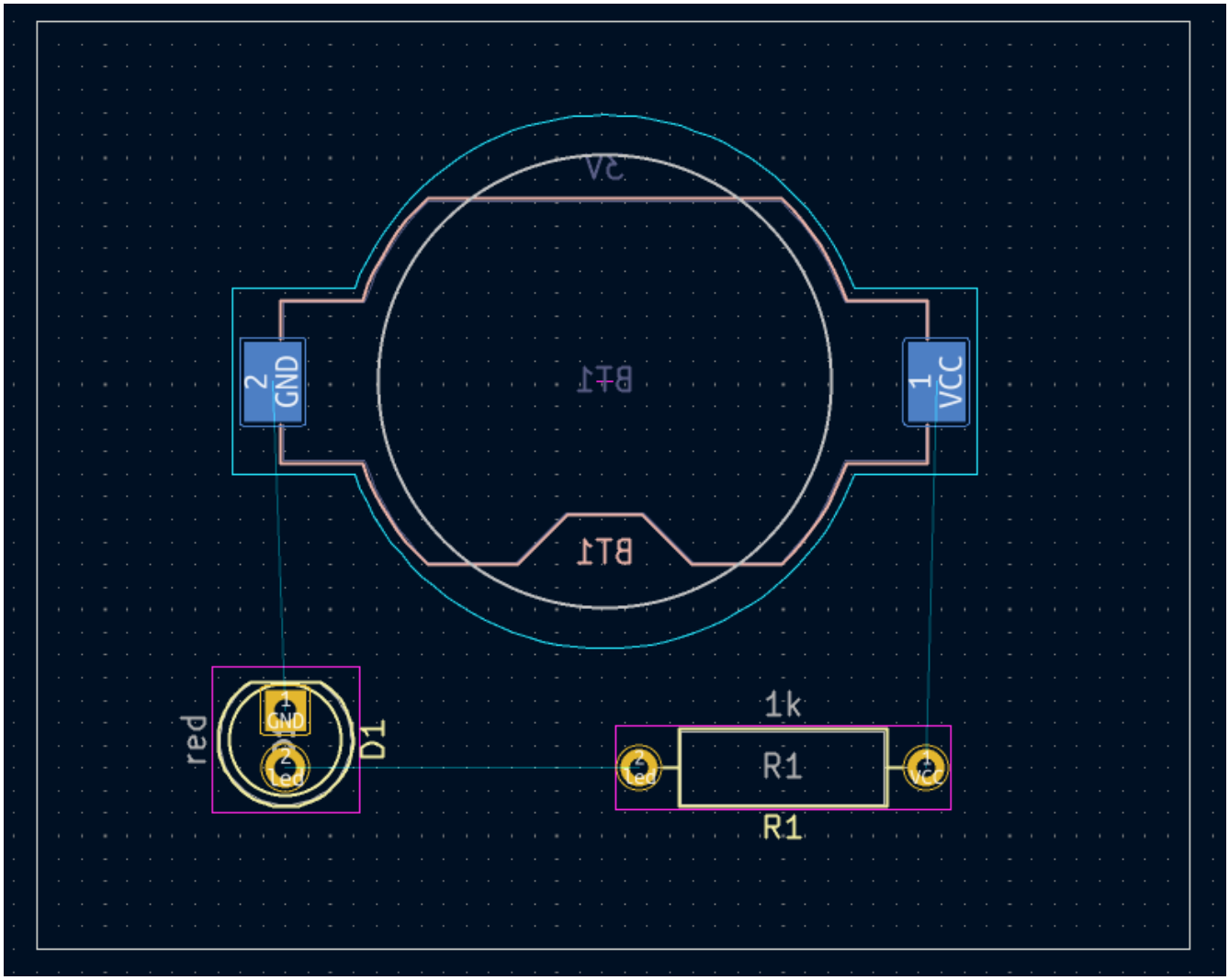
首先将电池座 `BT1` 移到电路板的背面。点击它来选择它，然后按 M 来移动它。按 F 将其翻转到另一侧；现在它看起来是镜像的，它的焊盘从红色变成了蓝色。




所有的 PCB 层都是从板子的正面看的。因此，电路板底部的封装是颠倒的，看起来是镜像的。

每个 PCB 层都有一个独特的颜色，由外观面板的层标签中的色板显示。在默认的颜色方案中，`F.Cu` （正面铜）层上的项目是红色的，而 `B.Cu` （背面铜）上的项目是蓝色。

现在放置另外两个元件。一次一个，选择每个元件，然后用 M 和 R 移动和旋转它。观察每个焊盘之间的飞线，以选择最简单的元件排列；一个好的排列将使这些线不被缠住。一个可能的安排显示在下面的截图中。

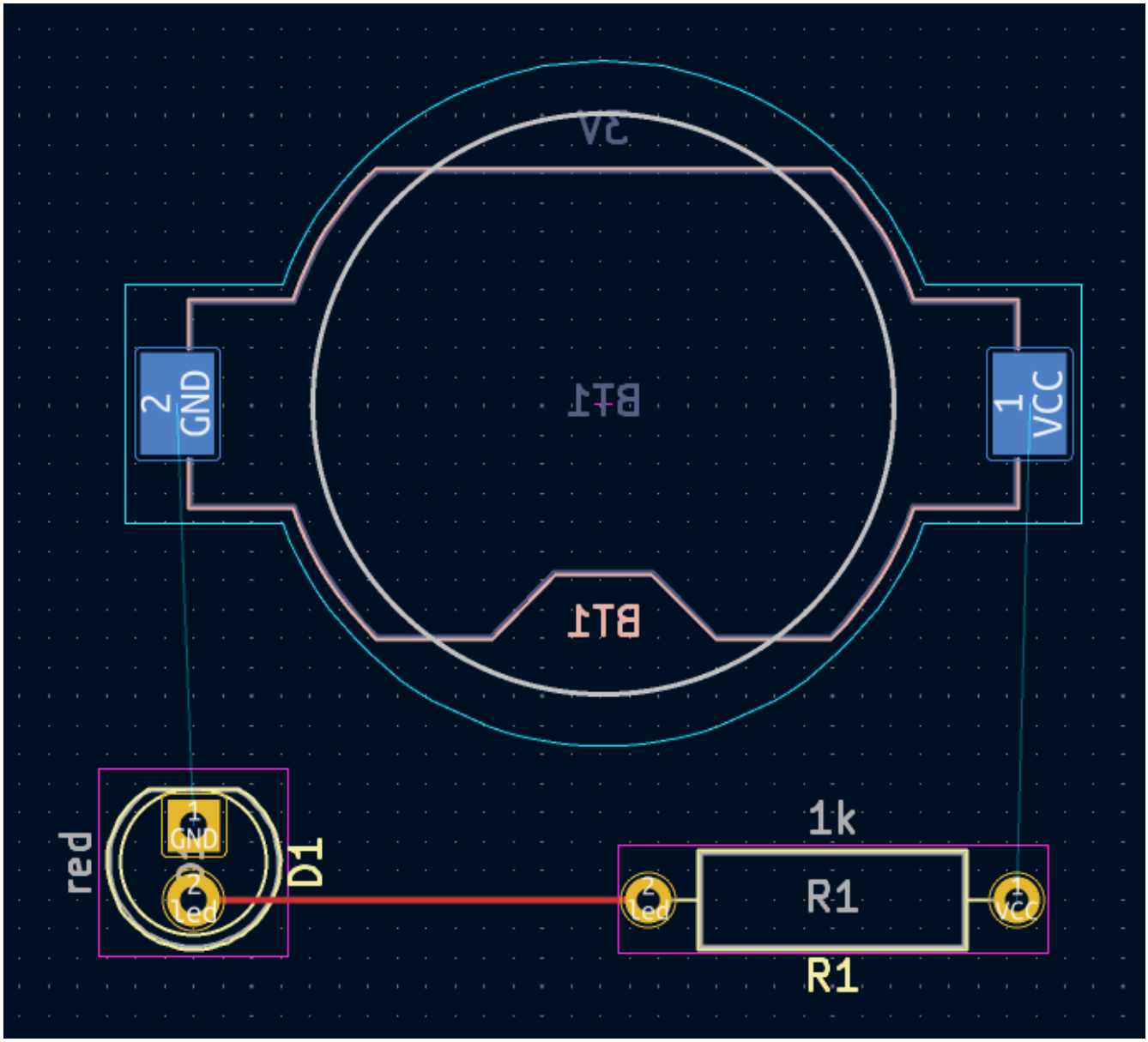



## 布线

元件就位后，现在是用铜线连接焊盘的时候了。

第一条布线将被画在电路板的正面，所以在外观面板的层标签中把活动层改为 `F.Cu`。

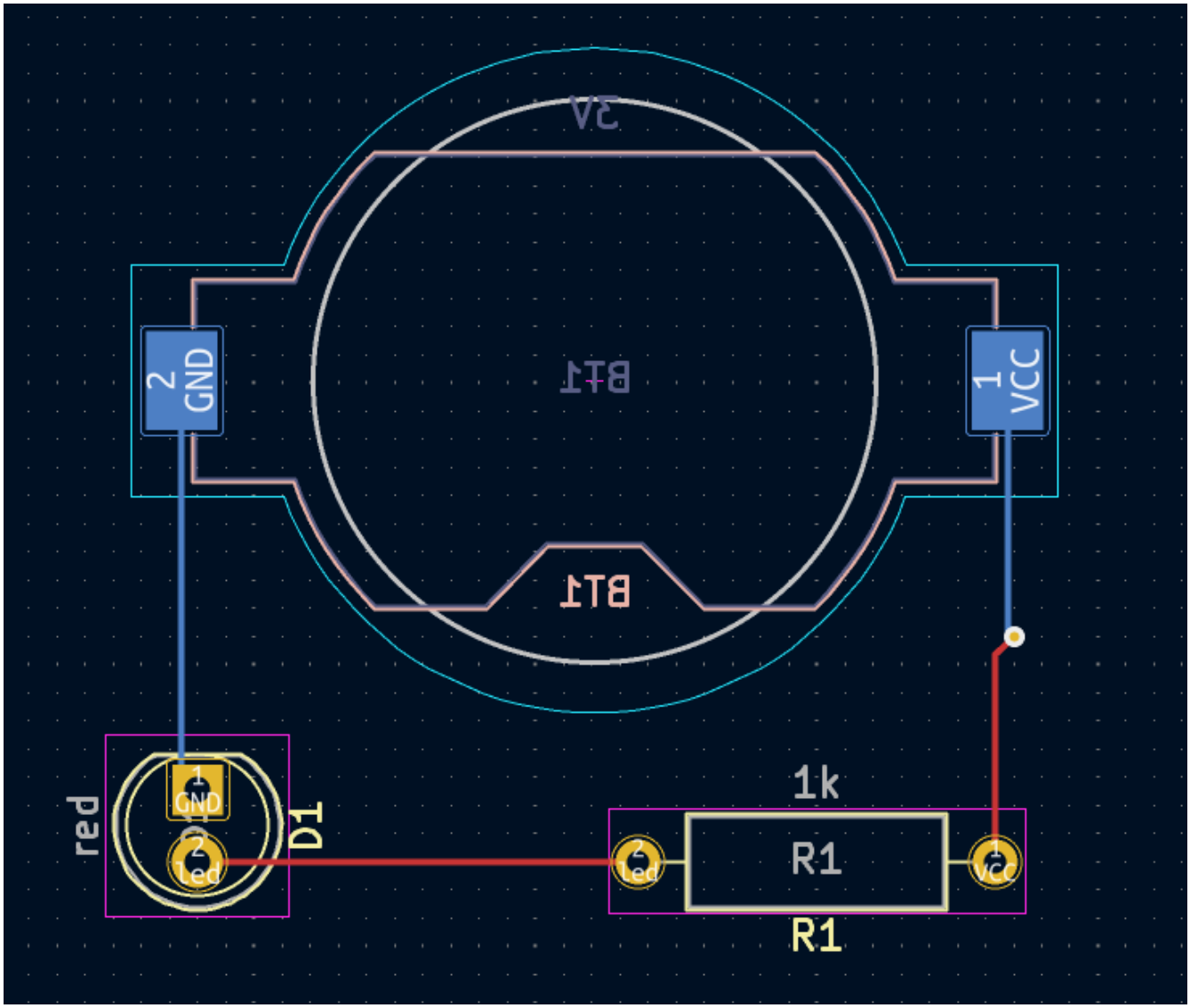
点击右侧工具栏中的 **布线** 或按 `X`。点击 `D1` 的 `led` 焊盘。鼠窝线表明有一个与 `R1` 的 `led` 焊盘的飞线布线连接，所以点击该焊盘来画一个连接这两个焊盘的布线。点击第二个焊盘就完成了布线。 `led` 引脚之间的飞线不再被画出，因为连接已经在铜中完成。




现在在 `BT1` 和 `D1` 的 `GND` 焊盘之间布线，从电路板背面的 `BT1` 焊盘开始。注意在点击 `BT1` 焊盘后，活动层自动变成了 `B.Cu`。点击 `D1` 焊盘来完成这个布线。

While `BT1` has surface mount pads that are only on the bottom of the board, `D1` has through hole pads that can connect to tracks on both the front and back. Through hole pads are one way to make a connection between multiple layers. In this case, `D1` is a component on the front side of the board, but its through hole pads are used to connect to a trace on the back of the board.

另一种跨层连接的方法是用过孔。在电路板背面的 `BT1` 的 `VCC` 焊盘上开始布线。按 `V` 并在 `BT1` 和 `R1` 之间点击，插入一个过孔，同时将活动层切换到 `F.Cu`。通过点击 `R1` 的 `VCC` 焊盘完成电路板顶部的布线。



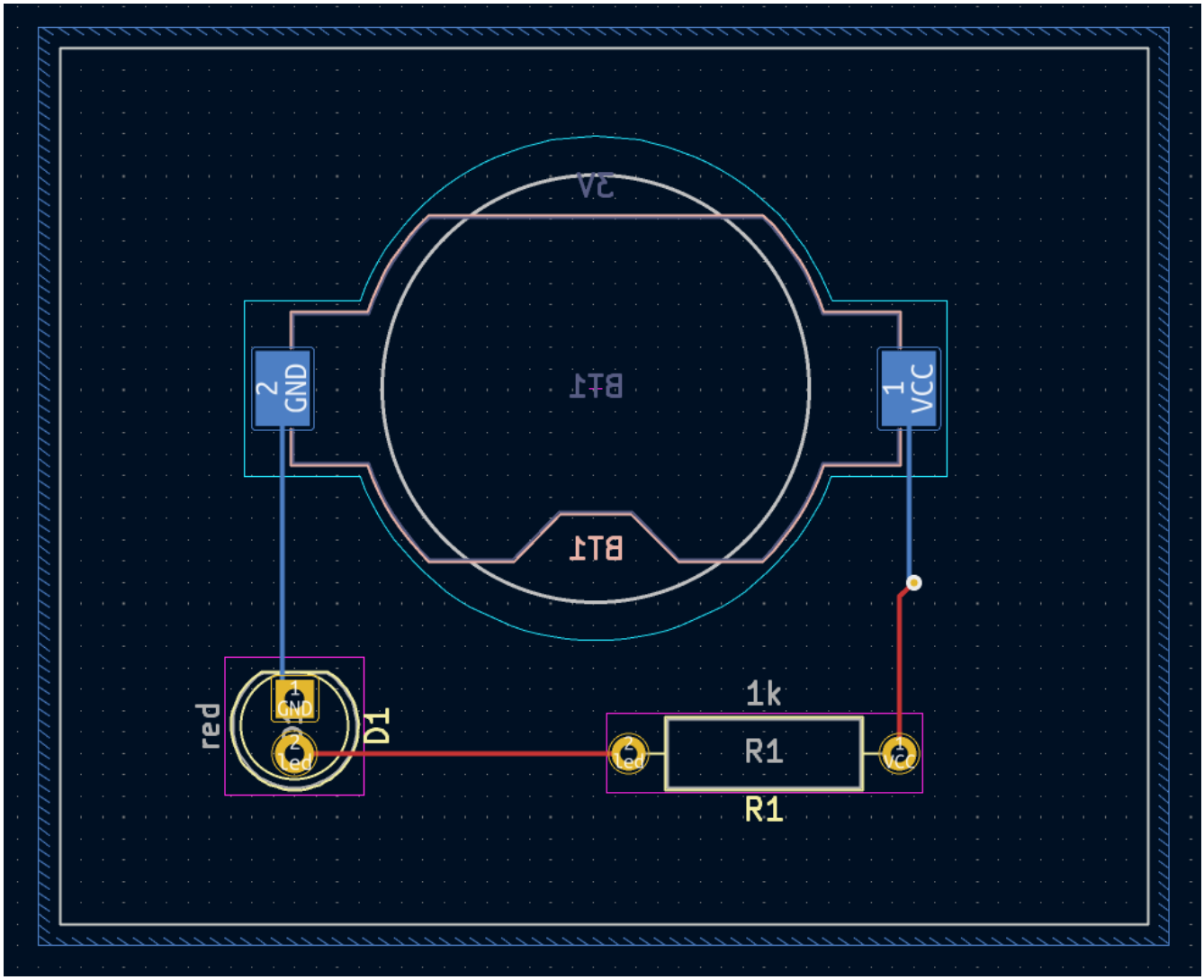

在这一点上，所有的连接都被布线了。这可以通过查看窗口左下方的状态屏幕来确认，其中未被布线的网络数量为 0。

## 放置敷铜

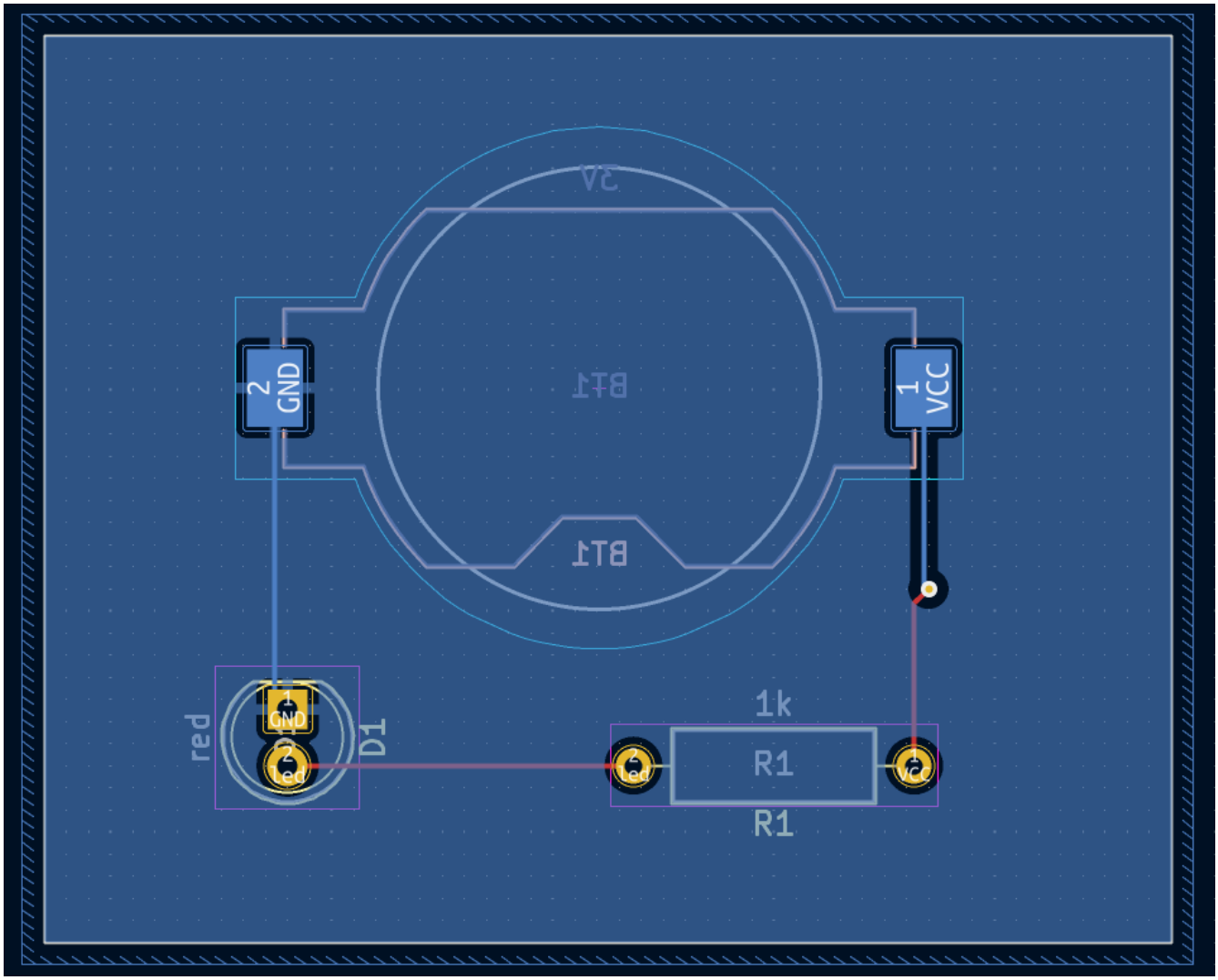
Copper zones are regions of copper with an associated net. Zones automatically connect to copper objects belonging to the same net and avoid objects belonging to other nets. Zones are often used for ground and power connections because they provide a lower impedance connection than traces.

通过切换到底部铜层并点击右侧工具栏中的 **添加敷铜** 按钮 ，在电路板的底部添加一个 `GND` 敷铜。在 PCB 上点击，放置该敷铜的第一个角。

在出现的 "敷铜属性" 对话框中，选择 `GND` 网络并确保 `B.Cu` 层被选中。点击 **确定**，然后点击放置该敷铜的其他三个角。在放置最后一个角时，双击来完成该敷铜。




敷铜边框显示在画布上，但区还没有被填满 — 在敷铜的范围内没有铜，因此敷铜没有进行任何电气连接。用 **编辑 → 填充所有敷铜** (B)填充该敷铜。铜已经被添加到该敷铜，但是它没有连接到 `VCC` 或 `led` 焊盘和导线，并且被板子的边缘剪掉。它与前面画的 `GND` 线重叠，并通过细线与 `GND` 焊盘连接。这些是散热条，使焊盘更容易焊接。 散热条和其他敷铜设置可以在敷铜属性对话框中修改。




在 KiCad 中，当第一次绘制或修改敷铜时，或移动敷铜内的封装时，敷铜不会被自动填充。敷铜是通过手动填充和运行《DRC，DRC》时重新填充。在生成《制造，制造输出》之前，请确保敷铜填充是最新的。

有时，在一个拥挤的画布设计中，填满的敷铜会让人很难看到其他的对象。可以使用左侧工具栏上的 **只显示敷铜边界** 按钮隐藏敷铜，。当只显示敷铜的边框时，敷铜会保留其填充状态—隐藏敷铜的填充与取消填充是不同的。

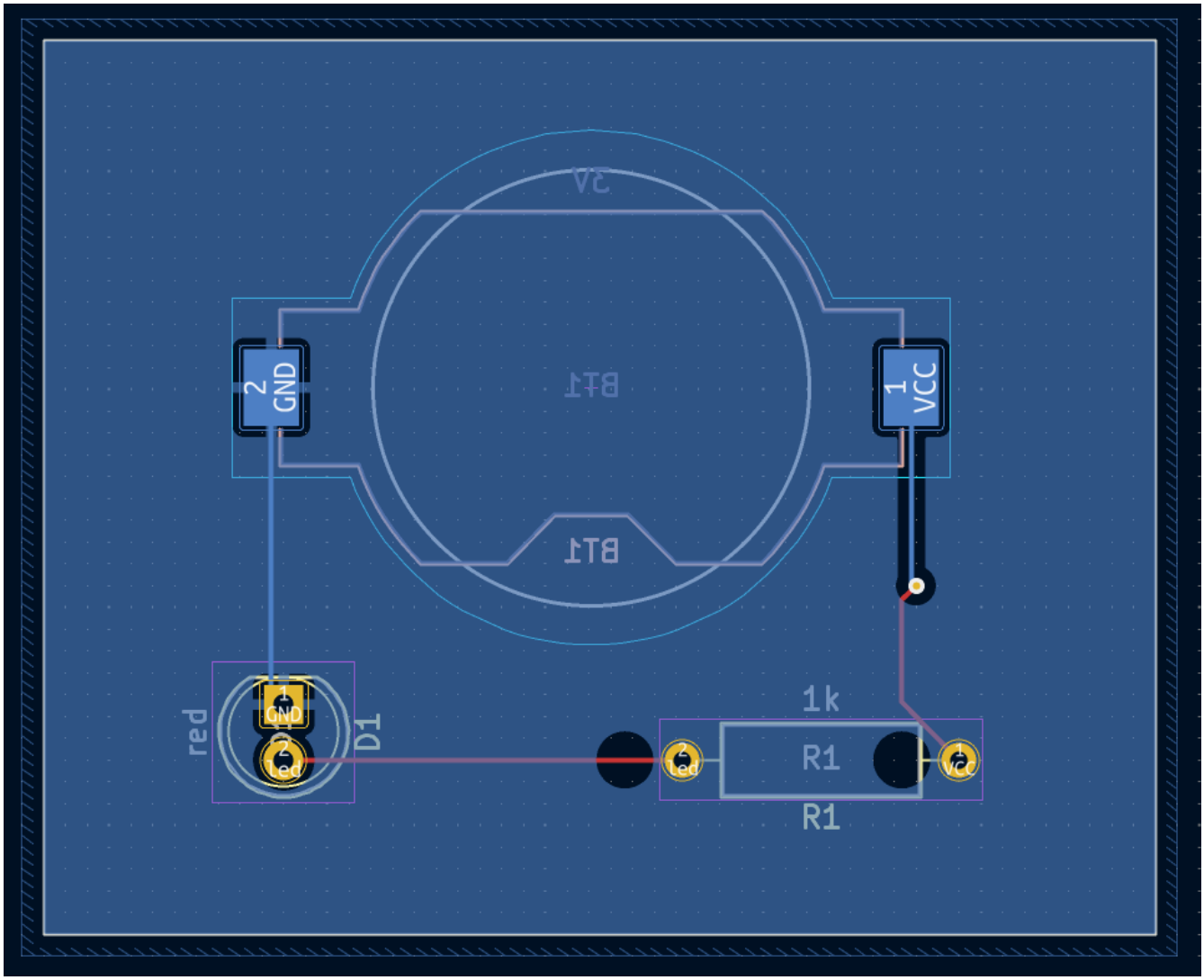
敷铜也可以使用外观面板使之透明，非活动层也可以使用外观面板中的 **层显示选项** 隐藏或调暗。

## 设计规则检查

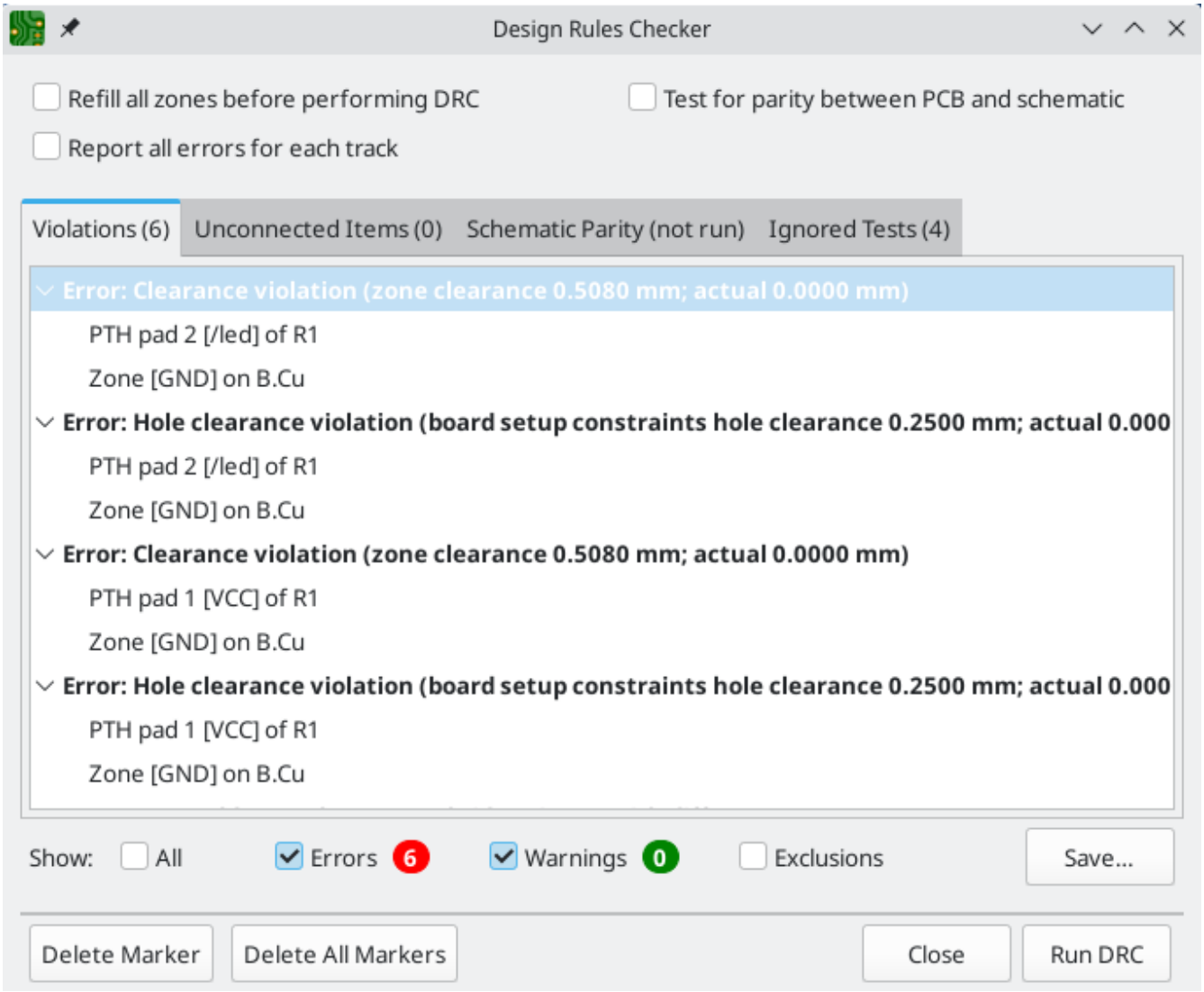
Design Rule Checking is the layout equivalent of Electrical Rule Checking for the schematic. DRC looks for design mistakes like mismatches between the schematic and layout, copper regions that have insufficient clearance or are shorted together, and traces that do not connect to anything. Custom rules can also be written in KiCad 8.0. To view the full list of design rules that are checked and to adjust their severity, go to **File** → **Board Setup...** → **Design Rules** → **Violation Severity**. Running DRC and fixing all errors is strongly advised before generating fabrication outputs.

用 **检查** → **设计规则检查** 来运行 DRC 检查，或者使用顶部工具栏的 按钮。单击 **运行 DRC**。当检查完成后，不应该有错误或警告报告。关闭 DRC 窗口。

现在故意造成 DRC 错误，将电阻的封装移动到与该敷铜的填充区域重叠。使用 D （拖动）稍微移动电阻的封装，同时保持导线连接到其焊盘上。这就产生了间隙违规，因为电阻的 `VCC` 和 `led` 焊盘与 `GND` 区的填充物短路了。通常情况下，这可以通过重新填充敷铜来解决，但现在不要重新填充敷铜。




Run DRC again, but make sure to uncheck the **Refill all zones before performing DRC** checkbox. DRC reports 6 violations: for each pad of `R1`, there is a clearance violation between the pad and the zone, another clearance violation between the pad's through hole and the zone, and a third violation where the pad's solder mask opening exposes the copper of two different nets (the `GND` fill and the trace connected to the pad). Arrows point to each violation in the canvas. Clicking on each violation message zooms in on the respective violation.

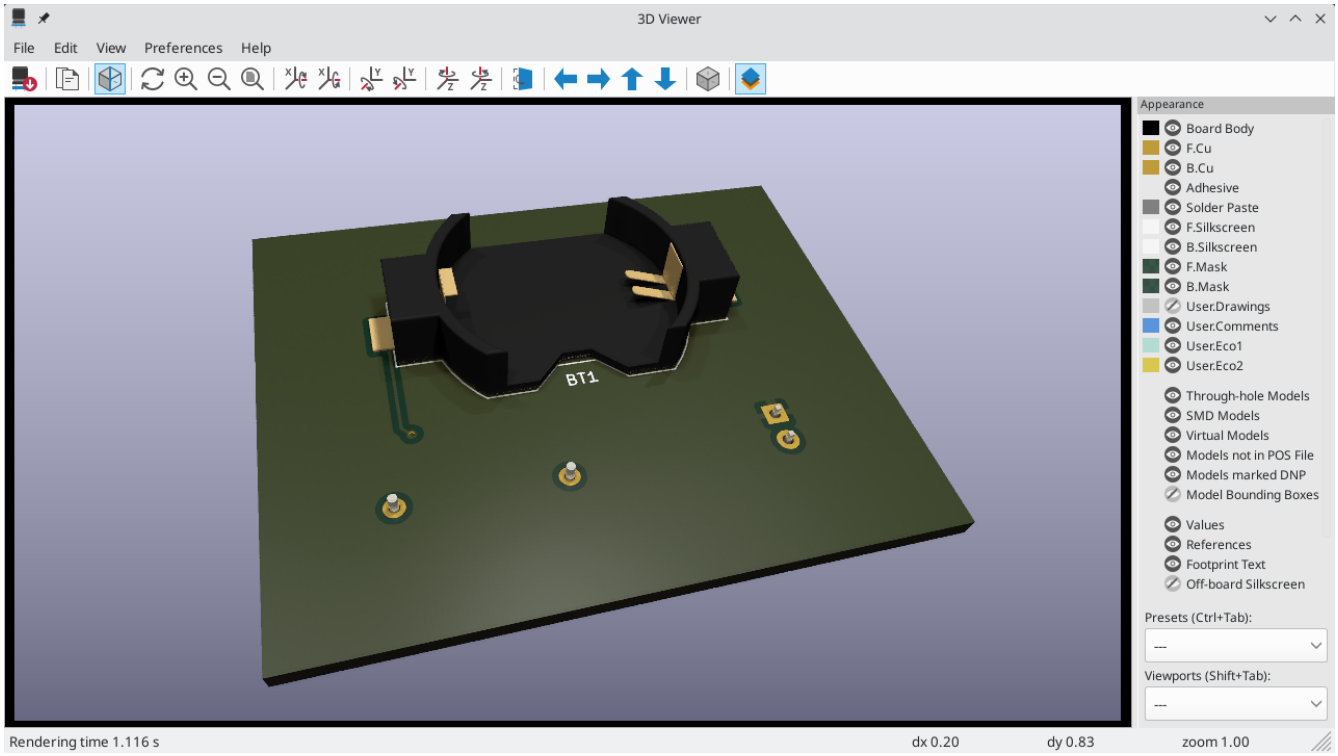



关闭 DRC 对话框，按 B 重新填充敷铜，并重新运行 DRC。 或者，勾选 **在执行 DRC 之前重新填充所有敷铜** 复选框并重新运行 DRC。所有的违规行为都被修复。

## 3D 查看器

KiCad 提供了一个 3D 查看器，对检查 PCB 很有用。用 **查看** → **3D 查看器** 打开 3D 查看器。用鼠标中键拖动进行平移，用鼠标左键拖动进行绕行。绕着 PCB 板运行，可以看到顶部的 LED 和电阻，以及底部的电池座。

有一种光线追踪模式，它比较慢，但能提供更精确的渲染。用 **偏好设置** → **光线追踪** 切换到光线追踪模式。



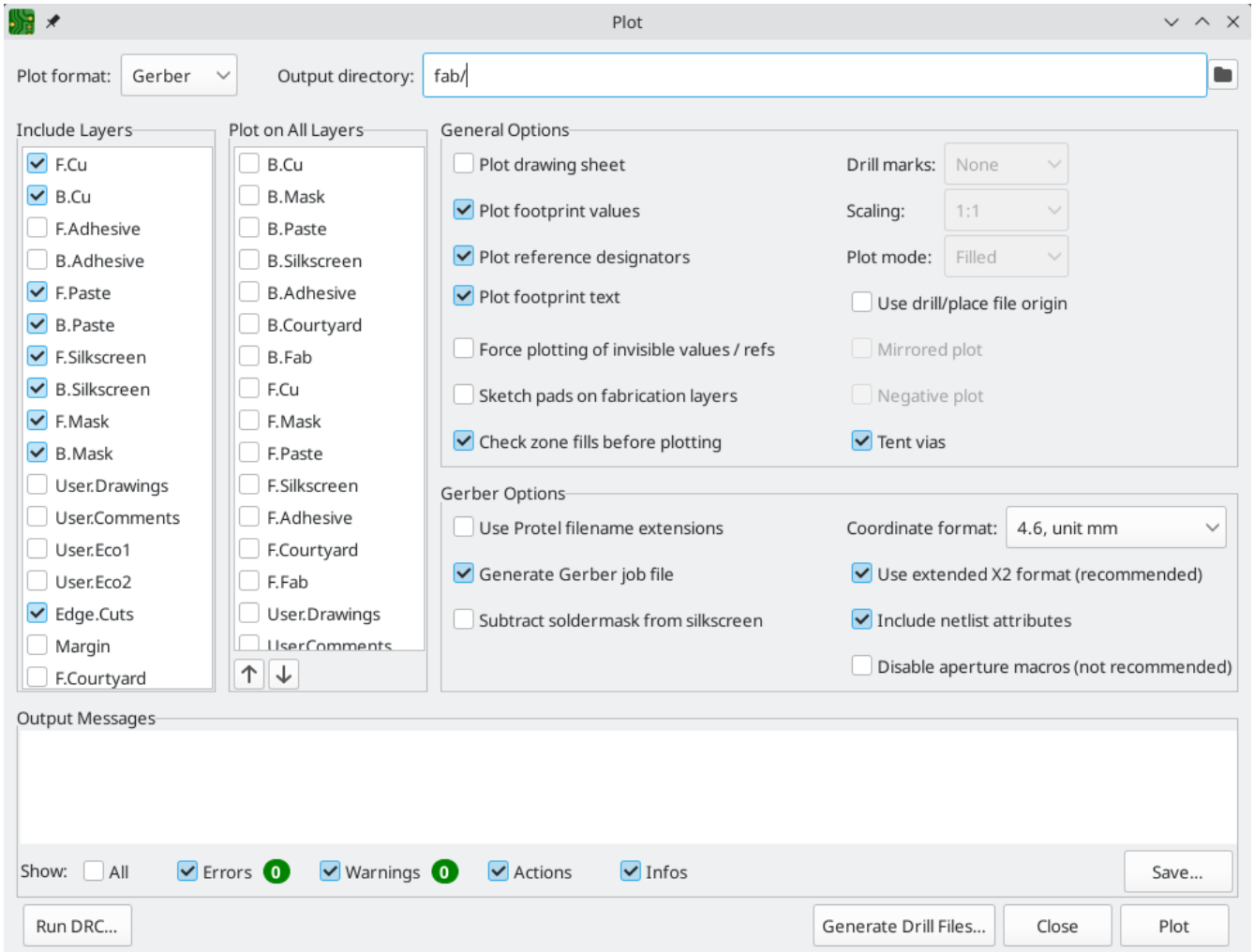

KiCad 库中的许多封装都带有 3D 模型，包括本指南中使用的所有封装。有些封装没有附带 3D 模型，《封装和 3D 模型，但用户可以添加自己的模型》。

## 制造输出

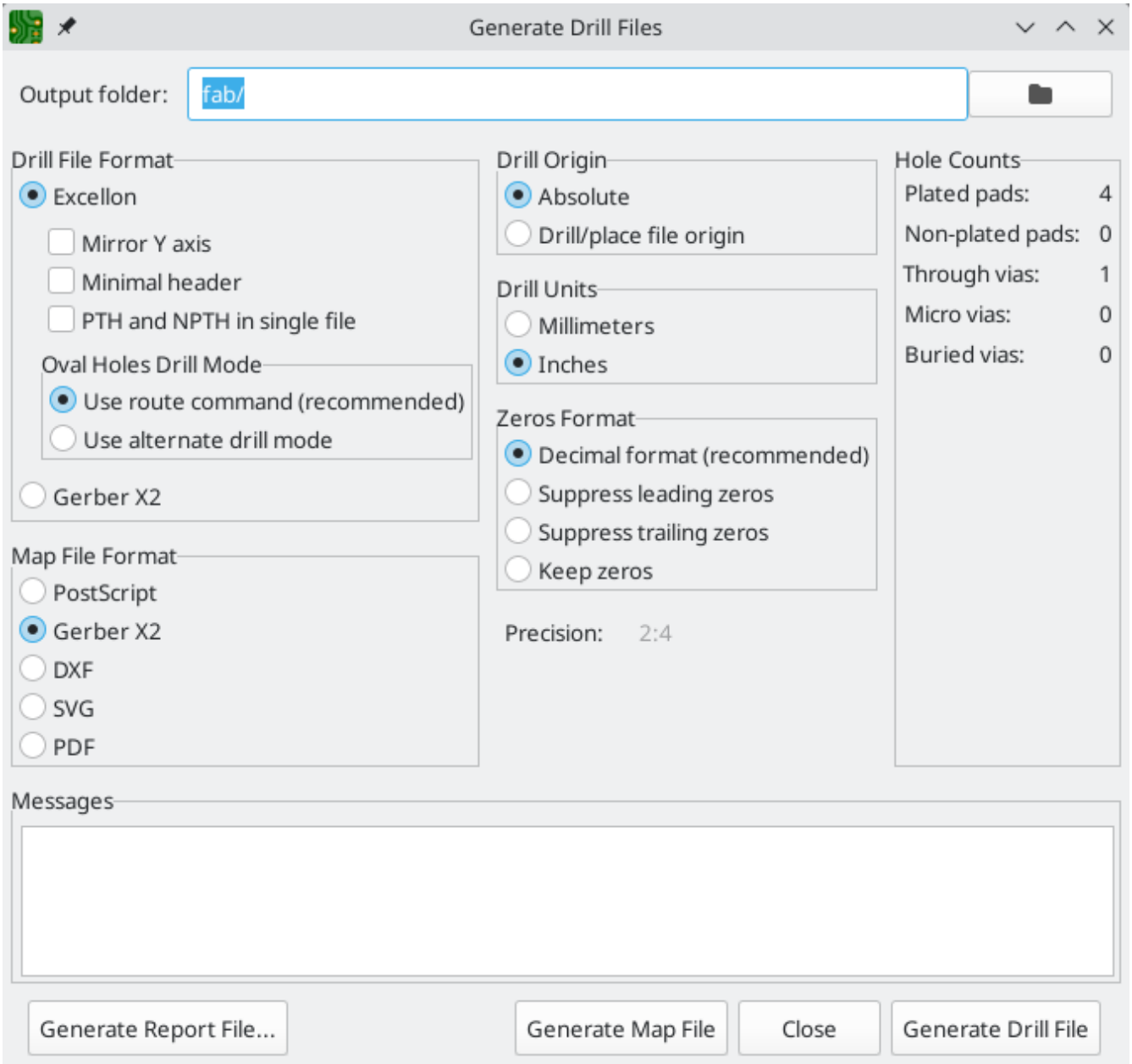
随着电路板设计的完成，最后一步是生成制造输出，这样电路板就可以被制造出来。

用 **文件** → **绘制...** 打开绘制对话框。这个对话框可以将设计绘制成几种格式，但 Gerber 通常是向 PCB 制造厂家制造的正确格式。

指定一个输出目录，这样绘制的文件就会被收集到一个文件夹中。否则，默认设置就可以了，但要确保所有必要的层都被选中：包括铜层（`*.Cu`）、电路板边框（`Edge.Cuts`）、阻焊层（`*.Mask`）和丝印（`*.Silkscreen`）。锡膏层（`*.Paste`）对于制造锡膏模版（钢网）很有用。粘合层（`*.Adhesive`）只有在组装过程中任何元件将被粘在电路板上时才需要。其他层可能对绘图有帮助，但通常对 PCB 的制作没有必要。




点击 **绘制** 来生成 Gerber 文件。同时点击 **生成钻孔文件...**，然后点击 **生成钻孔文件** 来创建文件，指定所有将在电路板上钻孔的位置。最后，关闭 "绘制" 对话框。设计就完成了。


Generate Drill Files
Output folder:
fab/
Drill File Format
Excellon
Mirror Y axis
Minimal header
PTH and NPTH in single file
Oval Holes Drill Mode
Use route command (recommended)
Use alternate drill mode
Gerber X2
Map File Format
PostScript
Gerber X2
DXF
SVG
PDF
Drill Origin
Absolute
Drill/place file origin
Drill Units
Millimeters
Inches
Zeros Format
Decimal format (recommended)
Suppress leading zeros
Suppress trailing zeros
Keep zeros
Precision:
2:4
Hole Counts
Plated pads: 4
Non-plated pads: 0
Through vias: 1
Micro vias: 0
Buried vias: 0
Messages
Generate Report File...
Generate Map File
Close
Generate Drill File

# 教程第 4 部分：自定义符号和封装

通过添加一个开关来打开和关闭 LED，该电路将得到改善。添加这个开关的过程需要创建一个新的符号和封装库，绘制一个开关符号，并为开关创建一个封装。

本指南将使用的具体开关是 NKK M2011S3A1W03，一种 SPST（单刀单置） 拨动开关。许多其他开关也可以使用，但引脚编号和封装尺寸可能需要调整。

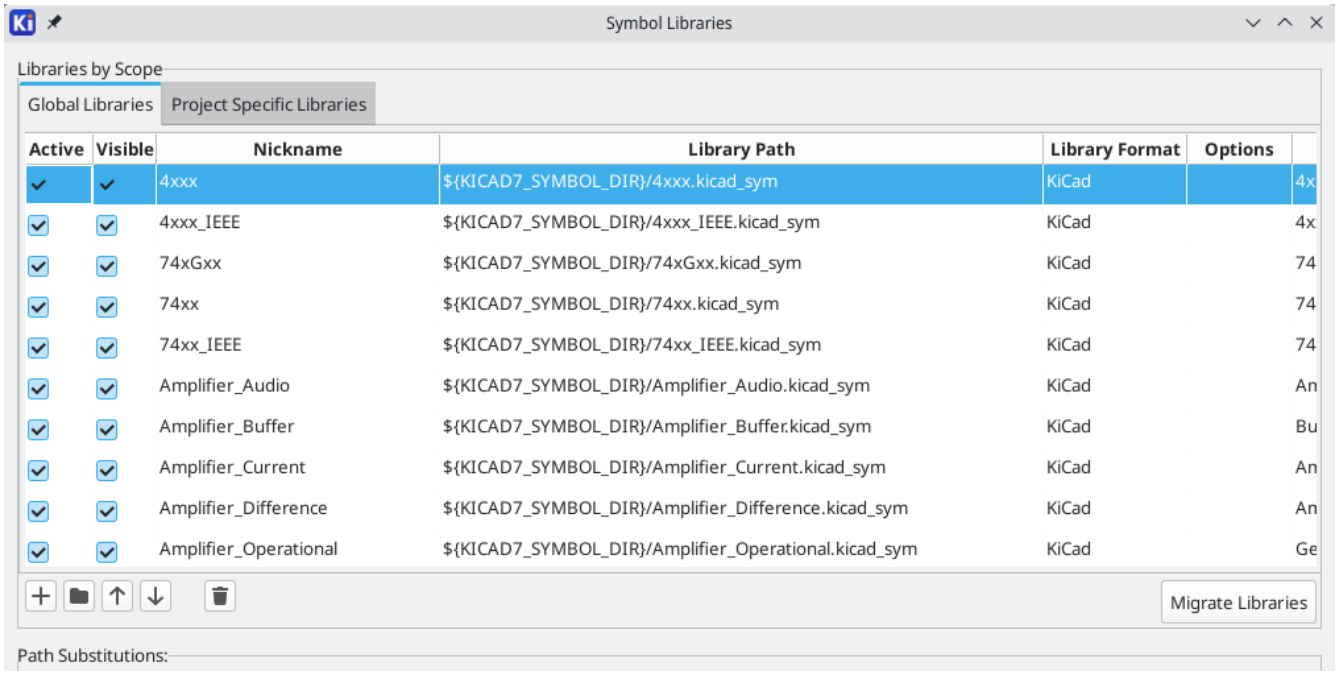
## 库和库表的基础知识

符号和封装被组织到库中。一个库可以容纳符号或封装，但不能同时容纳。

KiCad 在符号库表和封装库表中分别记录了用户的符号库和封装库。每个库表都是一个库名的列表，以及每个库在硬盘上存在的位置。

除了全局的符号和封装库表，还有用于符号和封装的工程库表。添加到全局表的符号和封装在所有工程中都可用，而专用工程表中的符号和封装只对该专用工程有效。用户可以将自己的库添加到全局库表或工程专用表中。

符号库表可以在原理图编辑器或符号编辑器窗口中通过 **偏好设置** → **管理符号库...** 来查看或编辑。 封装库表可以在电路板编辑器或封装编辑器中通过 **偏好设置** → **管理封装库...** 查看或编辑。这两个库表也可以从工程管理器中访问。




Often, paths to libraries are defined with path substitution variables. This enables a user to move all of their libraries to a new location without modifying the library tables. The only thing that needs to change is to redefine the variable to point to the new location. KiCad's path substitution variables are edited with **Preferences** → **Configure Paths...** in the Project Manager or any of the Editor windows.

一个有用的路径替换变量是 `${KIPRJMOD}`。这个变量总是指向当前的工程目录，所以它可以用来包括存储在工程目录内的工程专用库。

在第一次运行时，KiCad《符号库表设置，提示用户设置符号库表》和《封装库表设置，提示用户设置封装库表》。要再次进行这种设置，请删除或重命名符号库表或封装库表文件。在删除这些表之前，请对它们进行备份。

符号和封装库表文件的位置取决于操作系统。

- Windows: `%APPDATA%\kicad\8.0\sym-lib-table` and `%APPDATA%\kicad\8.0\fp-lib-table`
- Linux: `~/.config/kicad/8.0/sym-lib-table` and `~/.config/kicad/8.0/fp-lib-table`
- macOS: `~/Library/Preferences/kicad/8.0/sym-lib-table` and `~/Library/Preferences/kicad/8.0/fp-lib-table`

## 创建新的全局库或工程库

绘制一个新的符号或封装的第一步是选择一个库来存储它。在本指南中，开关符号和封装将进入新的工程专用库。

从工程管理器中打开符号编辑器。点击 **文件** → **新建库**，并选择 **工程**。为新库选择一个名字（例如：`getting-started.kicad_sym`）并保存在工程目录中。空的新库现在在左边的库窗格中被选中，并被自动添加到工程库表中（在 **偏好设置** → **管理符号库...** 中检查工程专用库标签）。

## 新建符号

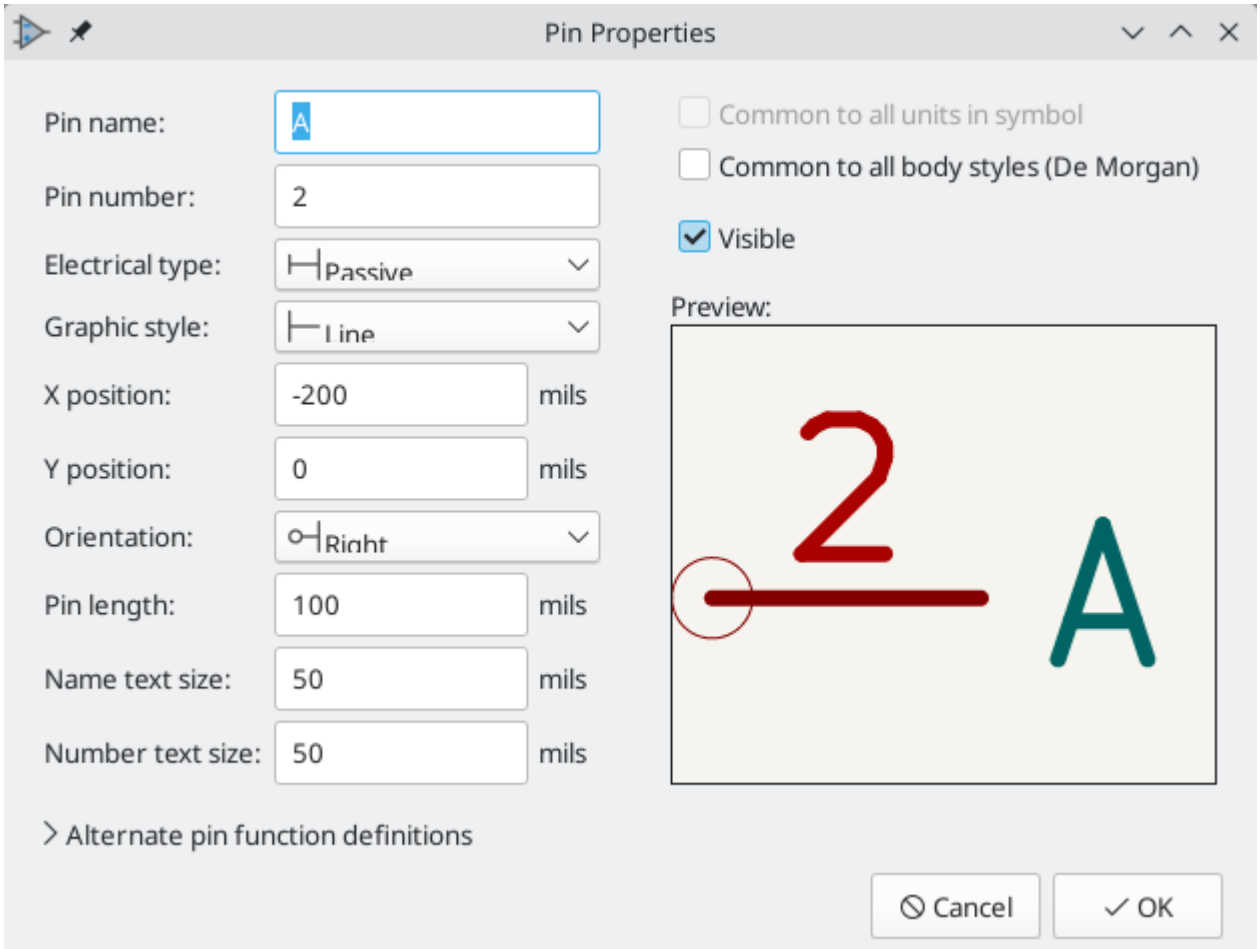
现在在新库中创建开关符号。在 "库" 窗格中选择 `getting-started` 库，点击 **文件** → **新建符号...**。在 **符号名称** 栏中，输入元件编号：`M2011S3A1W03`。开关符号的位号应以 `SW` 开头，所以将 **默认位号** 字段改为 `SW`。所有其他字段可以保持默认值。

In the Libraries pane, the `M2011S3A1W03` symbol now appears under the `getting-started` library. In the canvas, a cross indicates the center of the footprint, and text has been added for the reference designator. For now, move the text away from the center of the symbol to get it out of the way.

### 符号引脚

Start drawing the symbol by adding a pin. Click the **Add a pin** button on the right toolbar. The Pin Properties dialog will appear. Set **Pin name** to `A`, **Pin number** to `2`, **Electrical type** to `Passive`, and **Orientation** to `Right`. Set **X Position** to `-200 mils` and **Y Position** to `0`. Click **OK**, then click on the canvas to place the pin. If the pin moved after clicking **OK**, it might be necessary to edit the pin's properties again (double click the pin, or right click on the pin → **Properties...**) to set the position correctly.

| | |
|---|---|
| **NOTE** | It is strongly recommended to always use mils (or inches) in the Schematic Editor and Symbol Editor, particularly for symbol pins. Mixing imperial and metric units will result in symbol pins that cannot be properly connected because they do not align with wires or other pins. The KiCad libraries use a 50 mil grid for their pins; using a different unit system will prevent your symbols and schematics from connecting with KiCad's builtin symbols. Note that unlike in the PCB editor, units in the schematic have no physical meaning. |




添加第二个引脚，但这次不要使用 **添加引脚** 工具。相反，按 `Insert`。一个编号为 3 的新引脚被添加到符号中，就在引脚 2 下面。

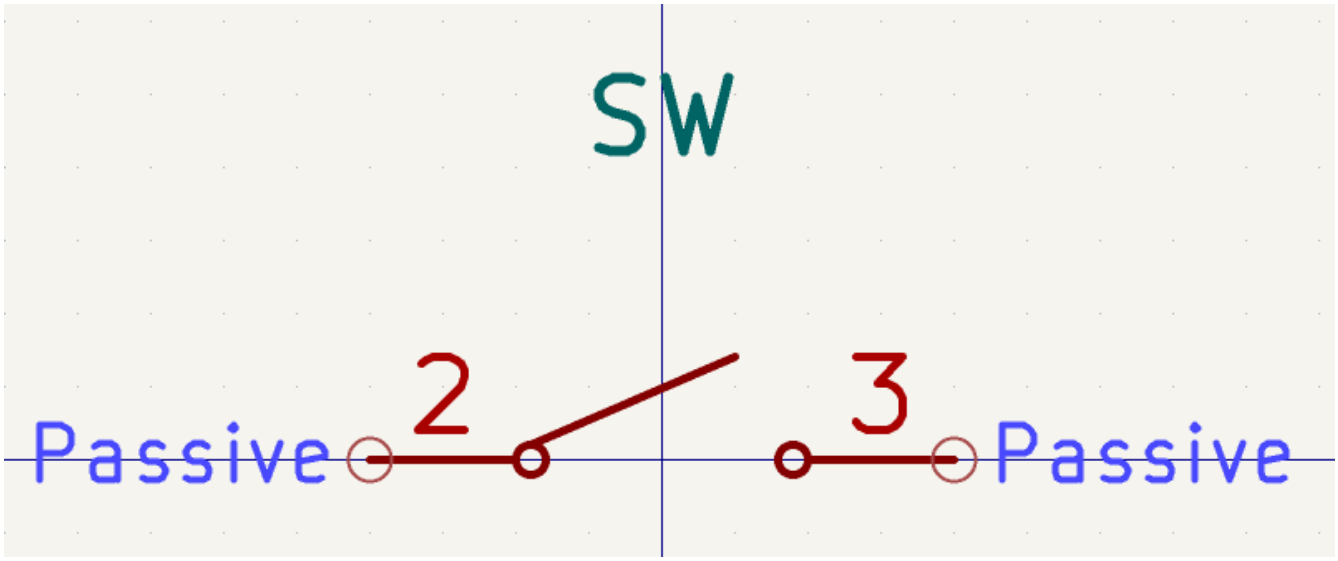
> **TIP**
> 在 KiCad 的许多地方，按 `Insert` 会重复上次的操作。新项目的位置将被移位，并根据情况自动增加编号。在符号编辑器中，这可以用来快速放置大量的引脚。在原理图编辑器中，它可以用来重复放置一个元件，或者用编号标签标记一个大元件的引脚。`Insert` 在封装和电路板编辑器中也很有用。

We want Pin 3 to be on the right side of the switch symbol, so edit the properties of pin 3 and change **X position** to `200 mils`, **Y position** to `0`, and **Orientation** to `Left`. Also, change **Pin name** to `B`.

## 图形特征

放好引脚后，使用圆圈 和直线 工具，使符号看起来像一个 SPST（单刀单置）开关。对于这一步，切换到一个更小的网格将是有用的：右击画布，在 **网格** 子菜单中选择一个更小的网格。添加图形后，再切换回 50 mil 的网格。

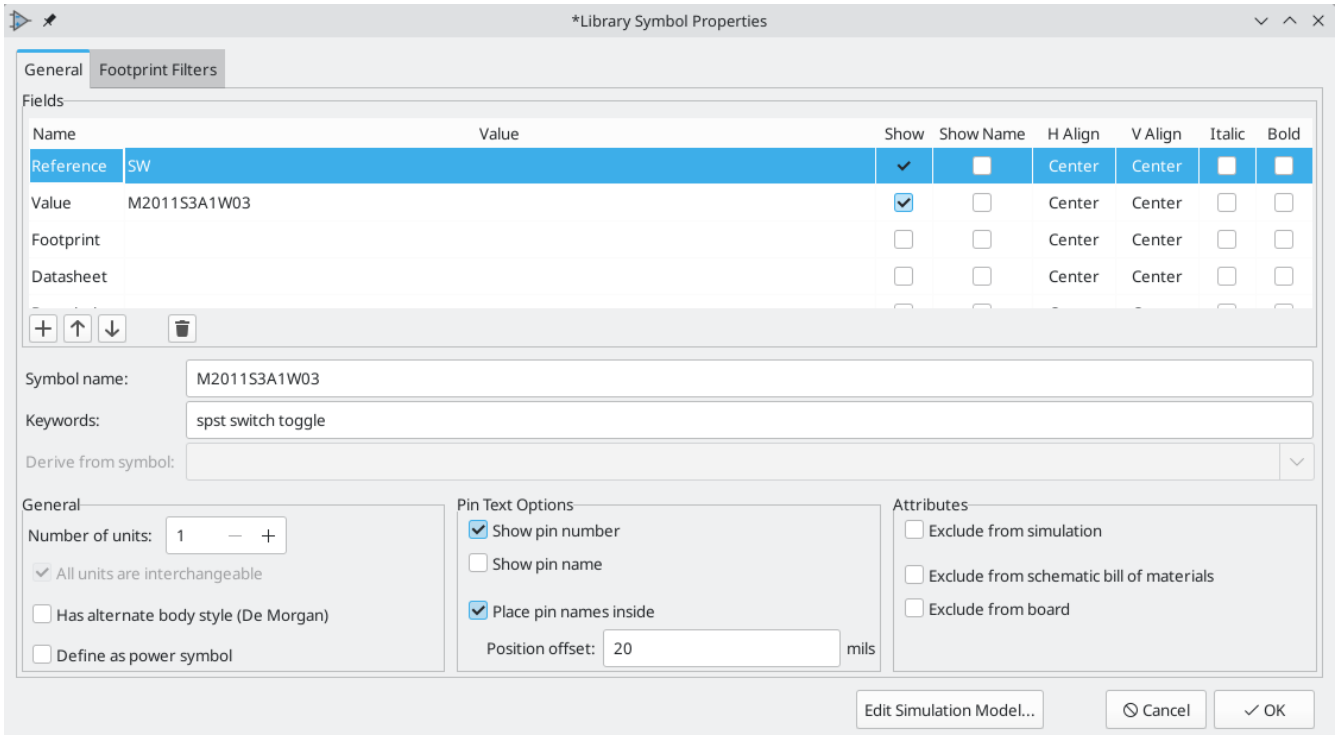
> **WARNING**
> 小网格对图形特征很有用，但 **符号引脚必须始终放在 50 mil（1.27 mm）的网格上**。没有对准 50 mil 网格的引脚将无法与原理图中的导线连接。




## 符号属性

现在用 **文件** → **符号属性** 编辑整个符号的属性，或者在画布上双击。将 **值** 字段改为 `M2011S3A1W03'，并在 **关键字** 字段中添加 `spst switch toggle'，以便通过搜索更容易找到该符号。对于这个符号，引脚名称并没有增加任何有用的信息，所以取消勾选 **显示引脚名称**，使符号在视觉上更简单。

点击 **确定** 后，将值的文字移至较低位置，使其不与符号图形重叠。




符号现在已经完成。保存它并继续创建一个封装。

## 新建封装

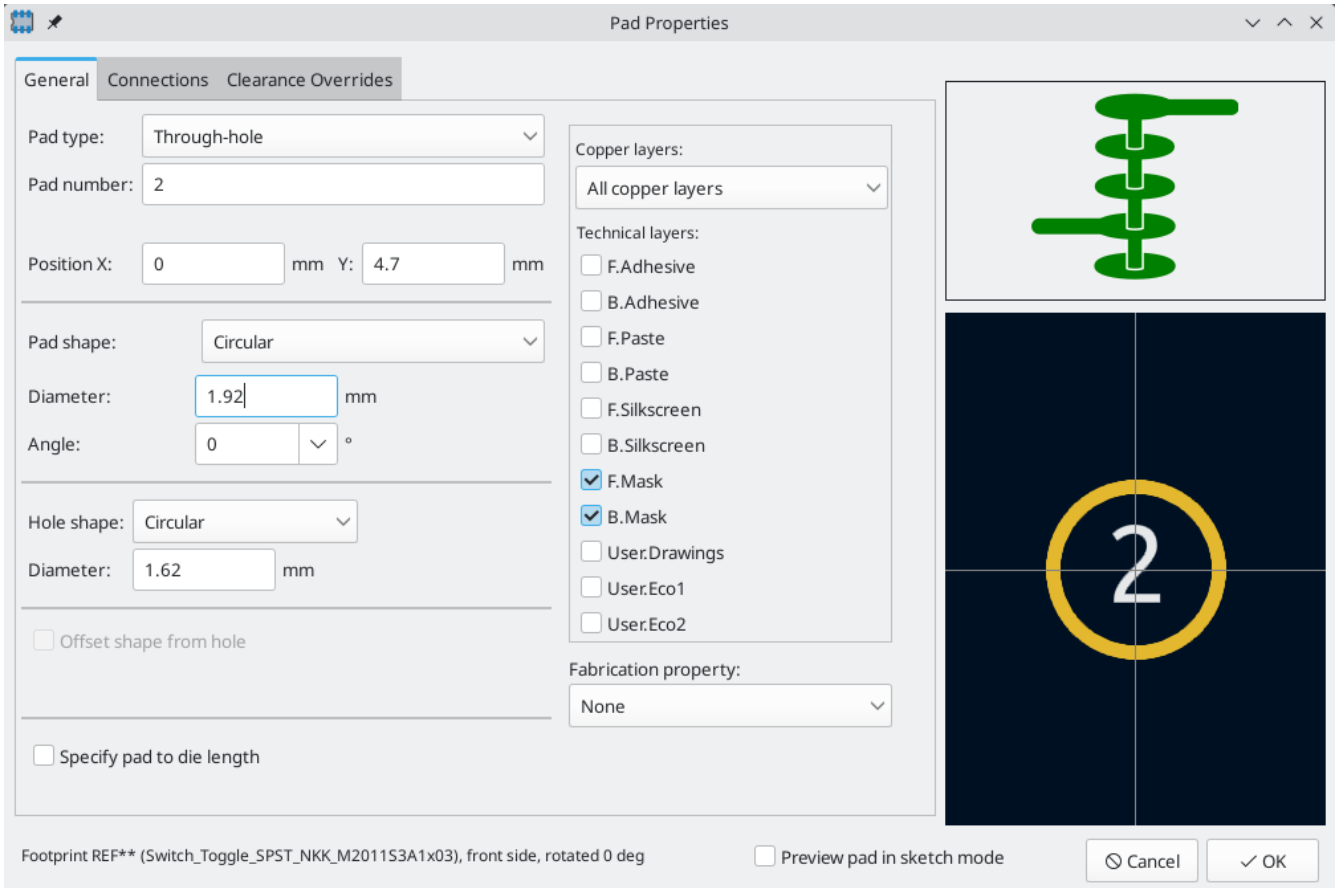
打开封装编辑器，创建一个新的工程专用封装库，命名为 `getting-started.pretty`（**文件** → **新建库...**）。和符号库一样，新的封装库被添加到工程库表中。在 "库" 窗格中选择新的库，创建一个新的封装（**文件** → **新建封装...**）。设置名称为 `Switch_Toggle_SPST_NKK_M2011S3A1x03`，类型为 "通孔"。

## 封装焊盘

The switch has two pins, numbered in the datasheet as 2 and 3, and spaced 4.7 mm apart. For ease of placement, adjust the grid to match the pad spacing. Open the grid settings by clicking **Preferences** → **Preferences...** → **Footprint Editor** → **Grids**, or by right-clicking the button in the left toolbar and selecting **Edit Grids...**. Add a new grid with the plus button at the bottom. Set the new grid's X size to `4.7 mm`. Accept the dialog, go back to the editing canvas, and then switch to the new grid in the grid dropdown in the top toolbar, if it is not already selected.

按照惯例，通孔封装的引脚 1 位于(0,0)处，并以引脚 1 在左上方的方向排列。这个开关的 SPST 版本没有引脚 1，所以封装会把 (0,0) 空出来，把焊盘 2 和 3 放在 (0, 4.7 mm) 和 (0, 9.4 mm) 处。请注意，在 KiCad 的默认坐标系中，正 Y 轴的方向是向下的。

Use the **Add a pad** tool in the right toolbar to place a pad one grid division below the origin, which is (0, 4.7 mm). Press Esc to exit the pad tool, then double click on the pad to edit its properties. Change the pad number to `2` and verify that the position is correct. The switch pins are 1.17 mm x 0.8 mm, which gives a diagonal (maximum pin dimension) of 1.42 mm. Therefore set the hole diameter to 1.42 mm + 0.2 mm = `1.62 mm`, and the pad diameter to 1.62 mm + 2*0.15 mm = `1.92 mm` to provide a sufficient annular ring.




现在再次使用 **添加焊盘** 工具，将另一个焊盘放在 (0, 9.4mm) 处。 注意，焊盘的编号会自动递增，包括焊盘尺寸和通孔的大小在内的属性会从前一个焊盘上复制下来。

在放置了两个焊盘后，圆环形环看起来略小。如果将圆环形环做大，开关将更容易焊接，机械上也更坚固。通过编辑焊盘 2 并将焊盘尺寸改为 1.62 mm + 2 * 0.3 mm = `2.22 mm`，将圆环形环的厚度从 0.15 mm 增加到 0.3 mm。不要改变通孔的尺寸。请注意，焊盘尺寸字段接受数学表达式，所以 `1.62 + 2 * 0.3` 可以直接输入，并将评估为 2.22 mm。

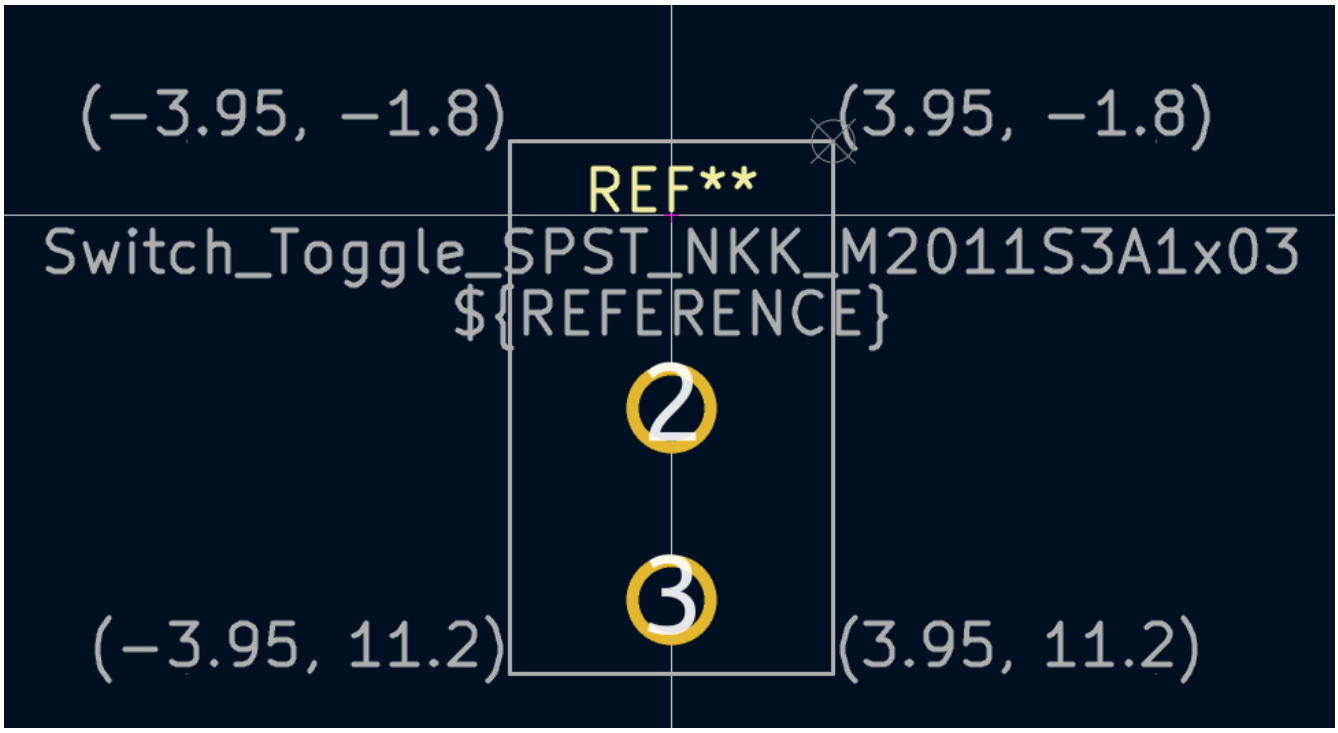
**TIP** KiCad 中的许多文本框支持数学表达，包括单位转换。

对另一个焊盘也做同样的圆环形圈修改。作为一个快捷方式，在焊盘 2 上点击右键，点击 **将焊盘属性推到其它焊盘上...**，然后点击 **改变当前封装上的焊盘**。

## 封装图形

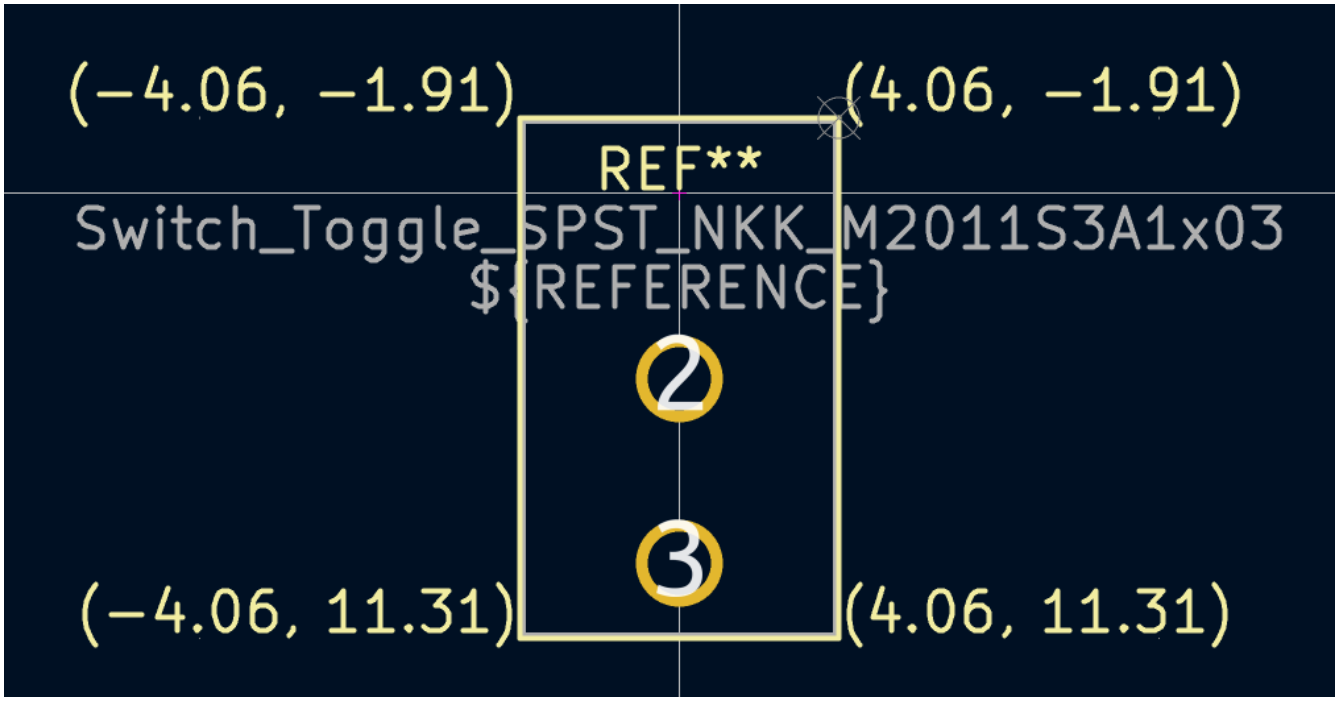
一个好的封装会在制造层（`F.Fab`）上画出准确的元件边框，在丝印层（`F.Silkscreen`）上画出稍大的边框，并在整个封装周围有一个边框（`F.Courtyard`）以防止与其他封装重叠。

Switch to the front fabrication layer by clicking **F.Fab** in the Layers panel at right. The fabrication outline should precisely match the physical dimensions of the part, which is 7.9 mm wide and 13 mm tall. Use the line , rectangle , or polygon tools to draw the outline of the part as shown in the screenshot below. One way to precisely place the outline is to create a new grid with an X spacing of `7.9 mm` and a Y spacing of `13 mm`. Then adjust the grid origin by right-clicking the button and selecting **Grid Origin...**. Set the X origin is 7.9 mm / 2 = `3.95 mm` and the Y origin to 4.7 mm - (13 mm / 2) = `-1.8 mm`.



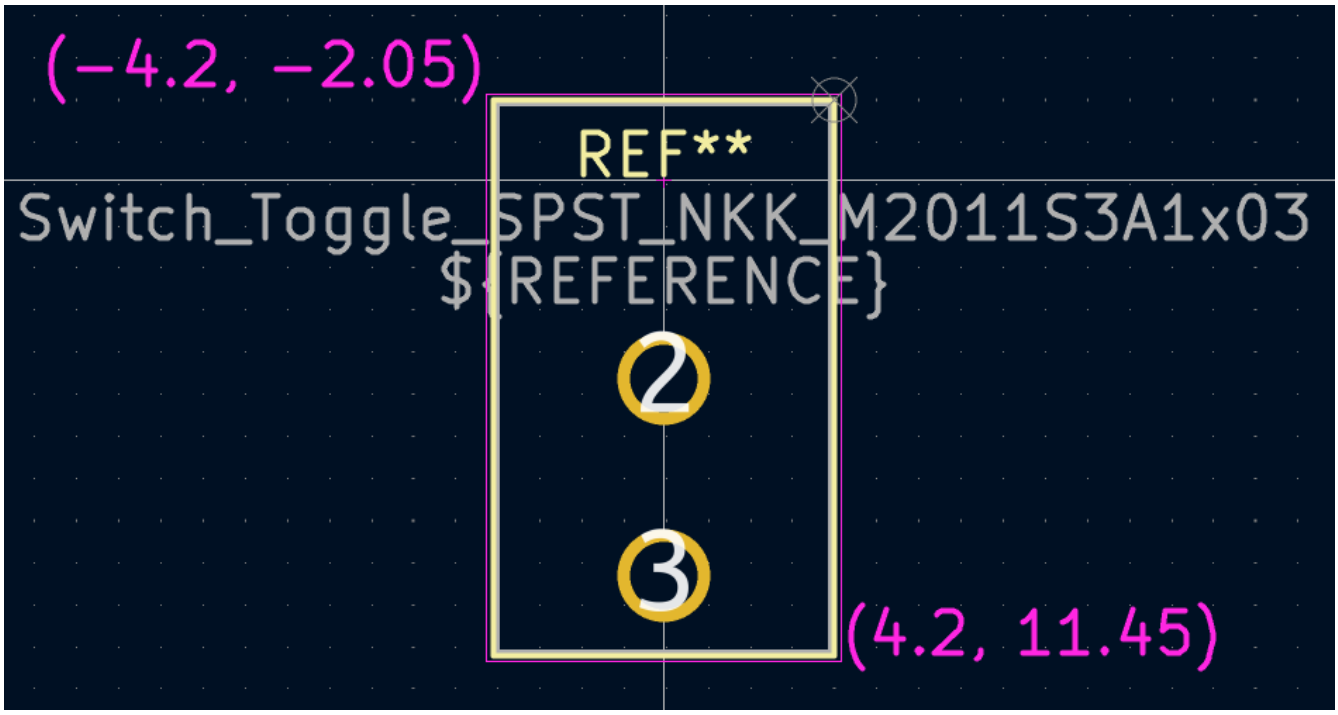

接下来，切换到 `F.Silkscreen` 层。丝网的边框应该正好在元件边框之外，所以丝网线要从制造图中向外移动 0.11 mm（0.10 mm 制造线宽度的一半，加上 0.12 mm 丝网线宽度的一半）。准确的坐标显示在下面的截图中，有用的网格设置是:

- X grid: 7.9 mm + 2 * 0.11 mm = `8.12 mm`
- Y grid: 13 mm + 2 * 0.11 mm = `13.22 mm`
- X origin: 8.12 mm / 2 = `4.06 mm`
- Y origin: -1.8 mm - 0.11 mm = `-1.91 mm`

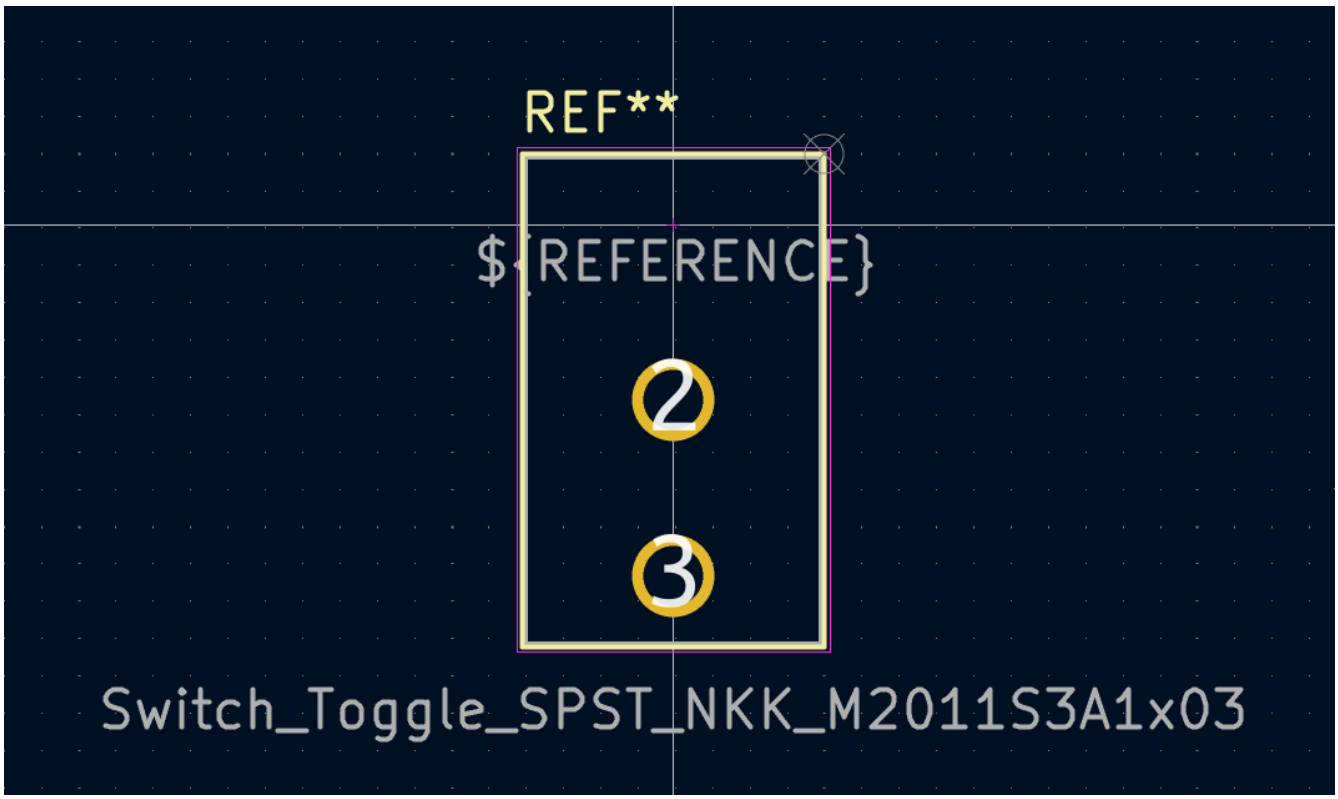



最后，选择 `F.Courtyard` 层。边框的边框应该以 0.25 mm 的间隙围绕着这个元件。

Use a different strategy to draw this layer. Switch to a 1mm grid instead of a custom grid, and draw a rectangle roughly surrounding the footprint. Double click on the rectangle to edit its properties, and enter the corner coordinates of the rectangle directly — they are shown in the screenshot below.




完成边框后，将文字定位如下图所示。封装就完成了。




## Kicad 库公约

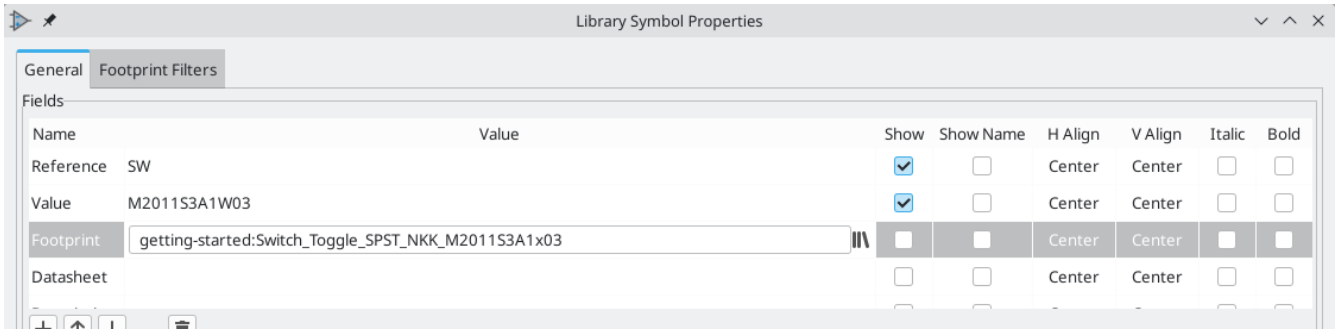
为了维护高质量的符号和封装库，KiCad 有一个 KiCad 库公约 文件，这是一套符号和封装的准则。个人库不一定要遵循这些约定，但它们是一个很好的出发点。 官方库 中的封装和符号都需要遵循 KLC。KLC 被用作本指南中的符号和封装的基础。

## 在原理图中添加开关

现在，封装已经完成，可以修改开关符号，以便默认为其使用匹配的封装。

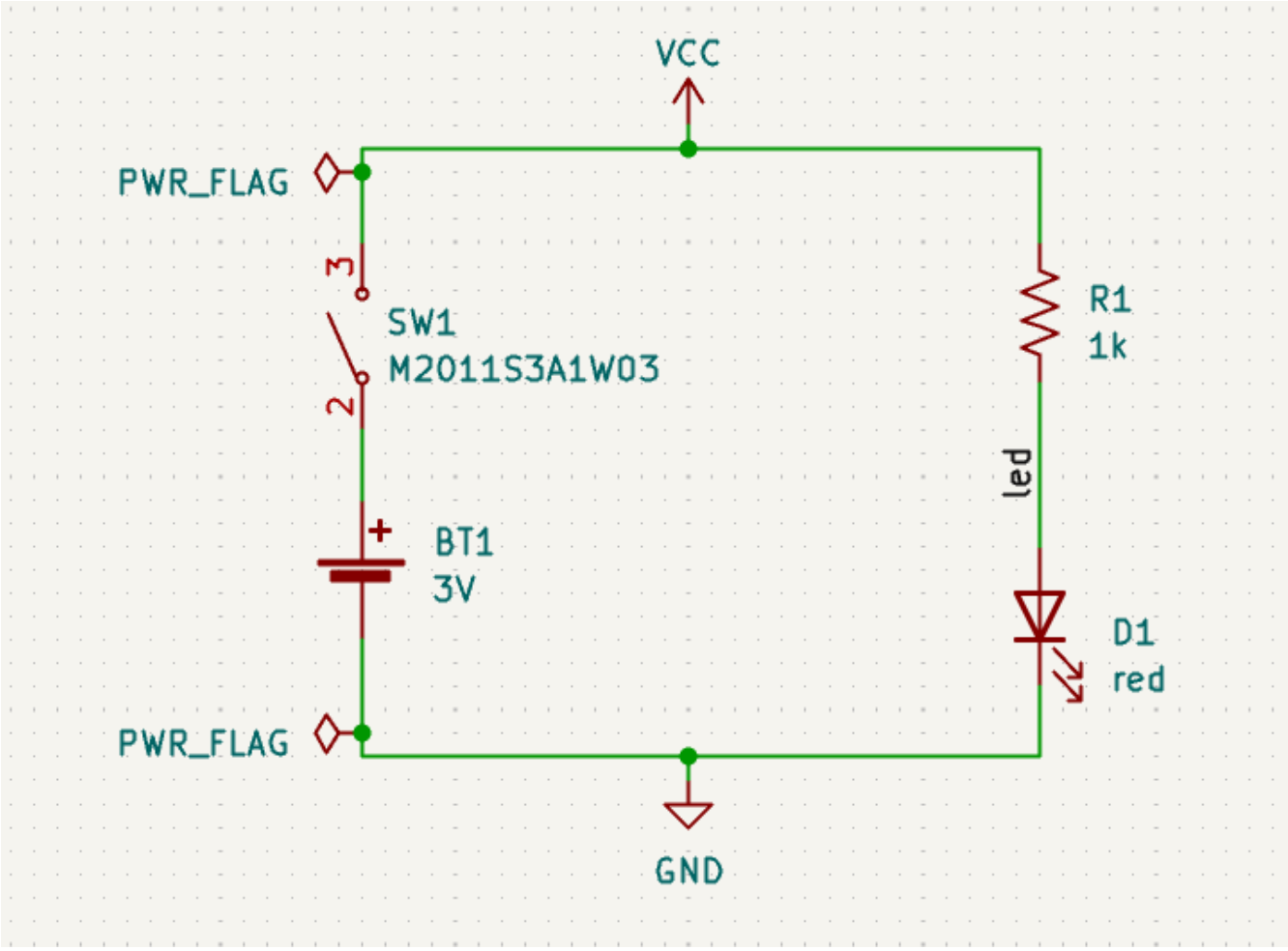
回到符号编辑器中，打开开关符号。编辑符号属性。点击 `封装` 字段，然后点击出现的库图标 。浏览到工程封装库，双击开关封装。保存该符号。

现在，开关封装被默认分配给该符号；每次将该符号添加到原理图中时，都不需要手动选择封装。



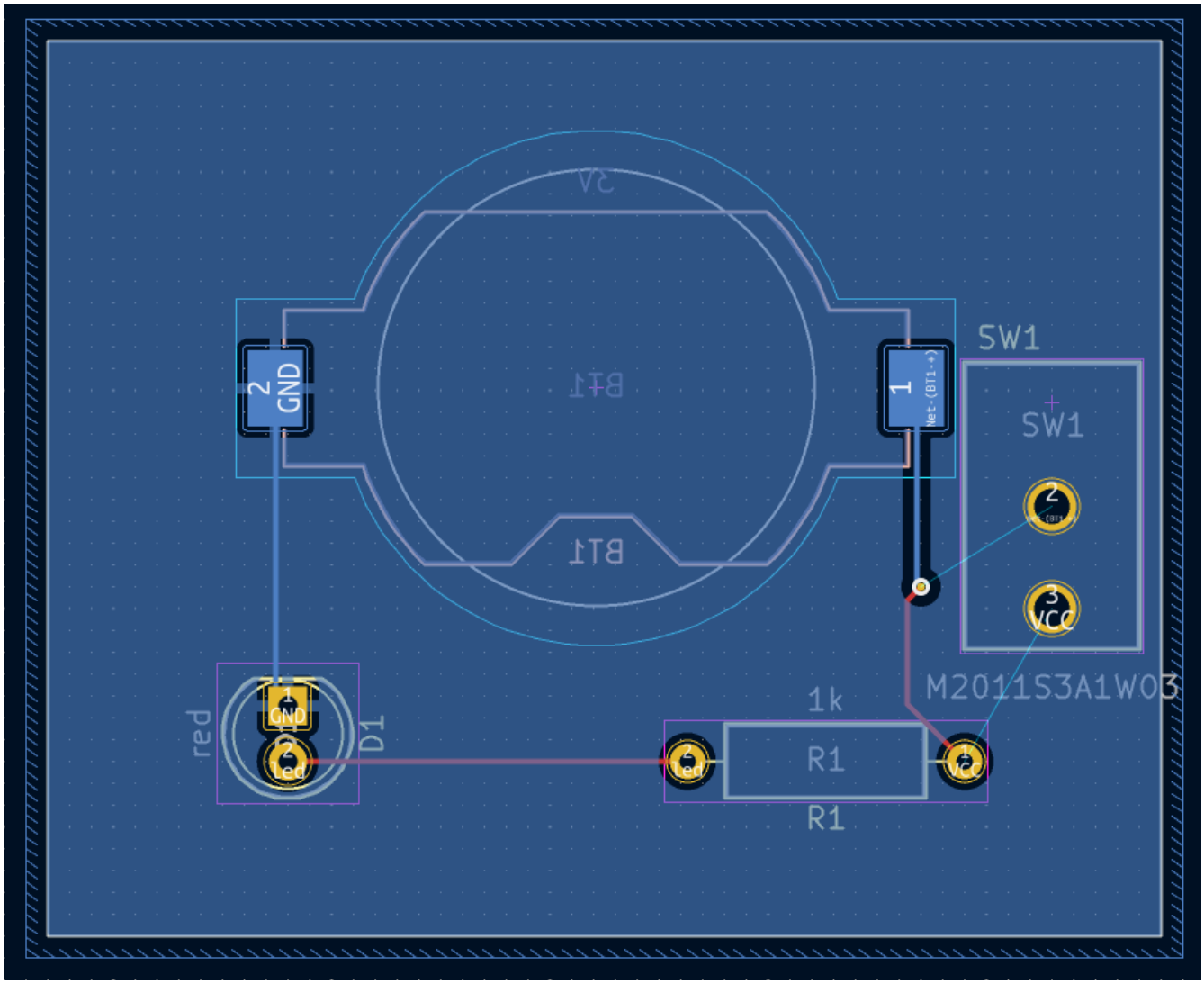

打开原理图，添加一个新符号，并选择新的开关符号。给它接线以连接或断开 LED 和电池。

It isn’t necessary to manually choose a footprint for the switch, because the switch symbol already specifies a footprint. Run ERC to make sure that the modified schematic doesn’t violate any electrical rules.




## 在布局中添加开关

确保原理图被保存，然后打开电路板编辑器，将符号添加到布局中。使用 **工具** → **从原理图更新 PCB...** 更新 PCB，并将开关的封装放到电路板上，如图所示。

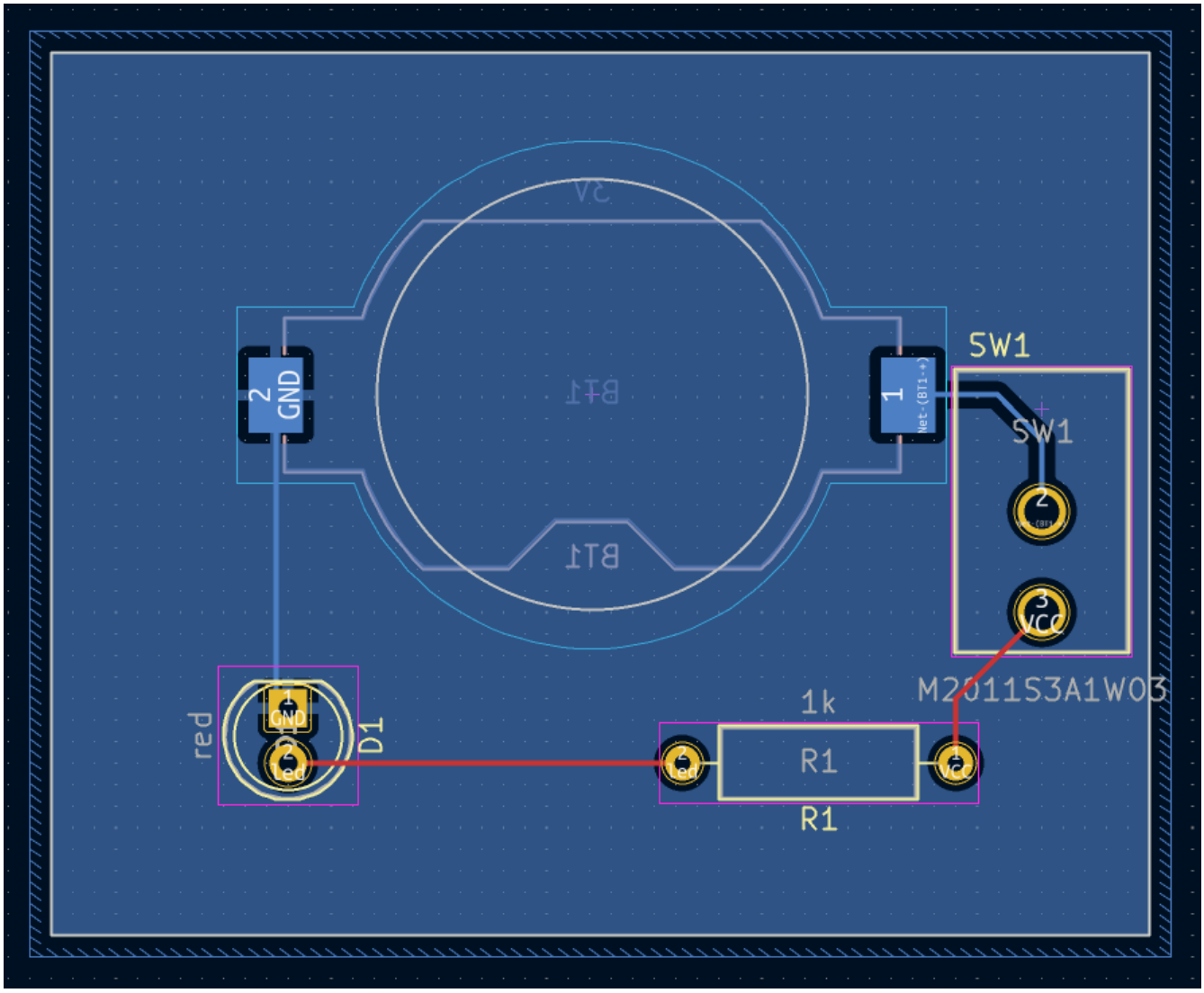



飞线显示了需要布线的新连接。此外，电池和电阻之间的连接也需要删除。

First, delete the unneeded traces. Select one of the traces between the battery and the resistor — it doesn't matter which. Press `U` several times to expand the selection to include all the segments between the battery and resistor. Press `Delete` to remove the connection.

在电池和开关之间，以及开关和电阻之间布局新的导线。按 `B` 键，重新填充各敷铜。

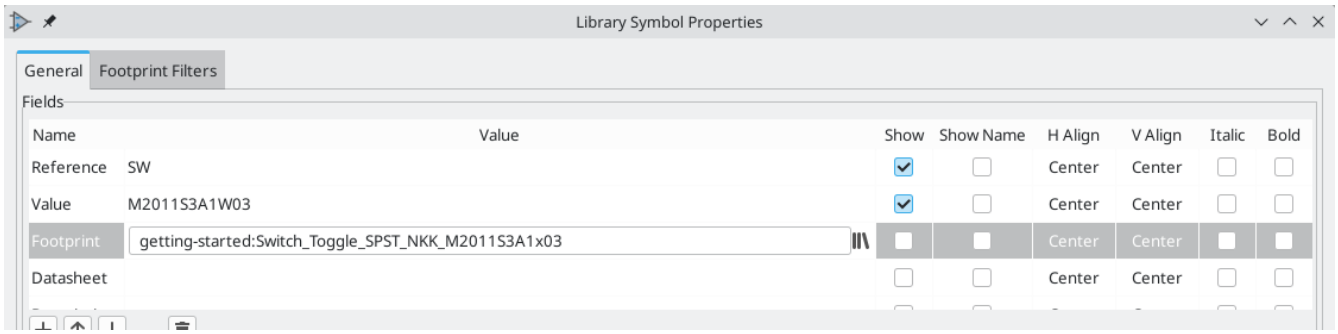
最后，重新运行 DRC 以确保修改后的电路板不违反任何设计规则。




## 链接符号、封装和 3D 模型

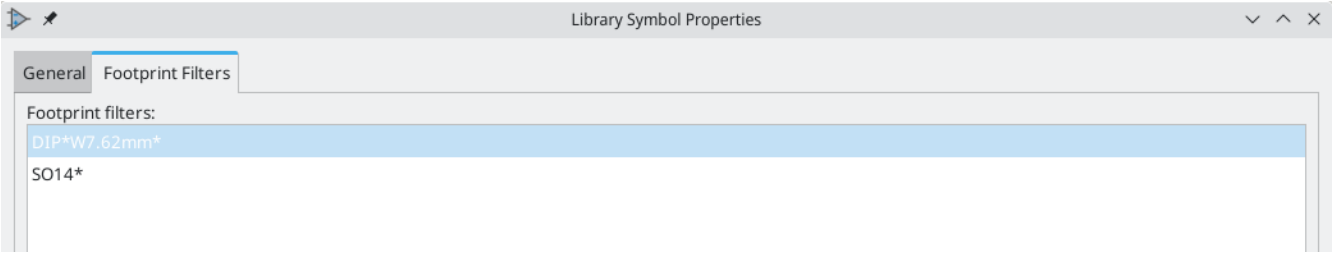
### 符号和封装

正如《封装分配，封装分配部分》中所述，在原理图输入过程中，原理图中的每个符号都需要为其分配一个封装。分配给每个符号的封装名称被存储在符号属性的 `封装` 字段中。




符号可以指定一个预选的封装。封装不需要手动分配给这些符号，因为在创建符号的时候就已经选择了一个封装。用户可以在封装分配过程中正常覆盖预选的封装。定义一个默认的封装对于那些通常或总是被分配相同封装的符号来说是个好主意，例如一个只在一个包中可用的元件。开关封装《默认封装，被设置为其匹配符号的默认封装》

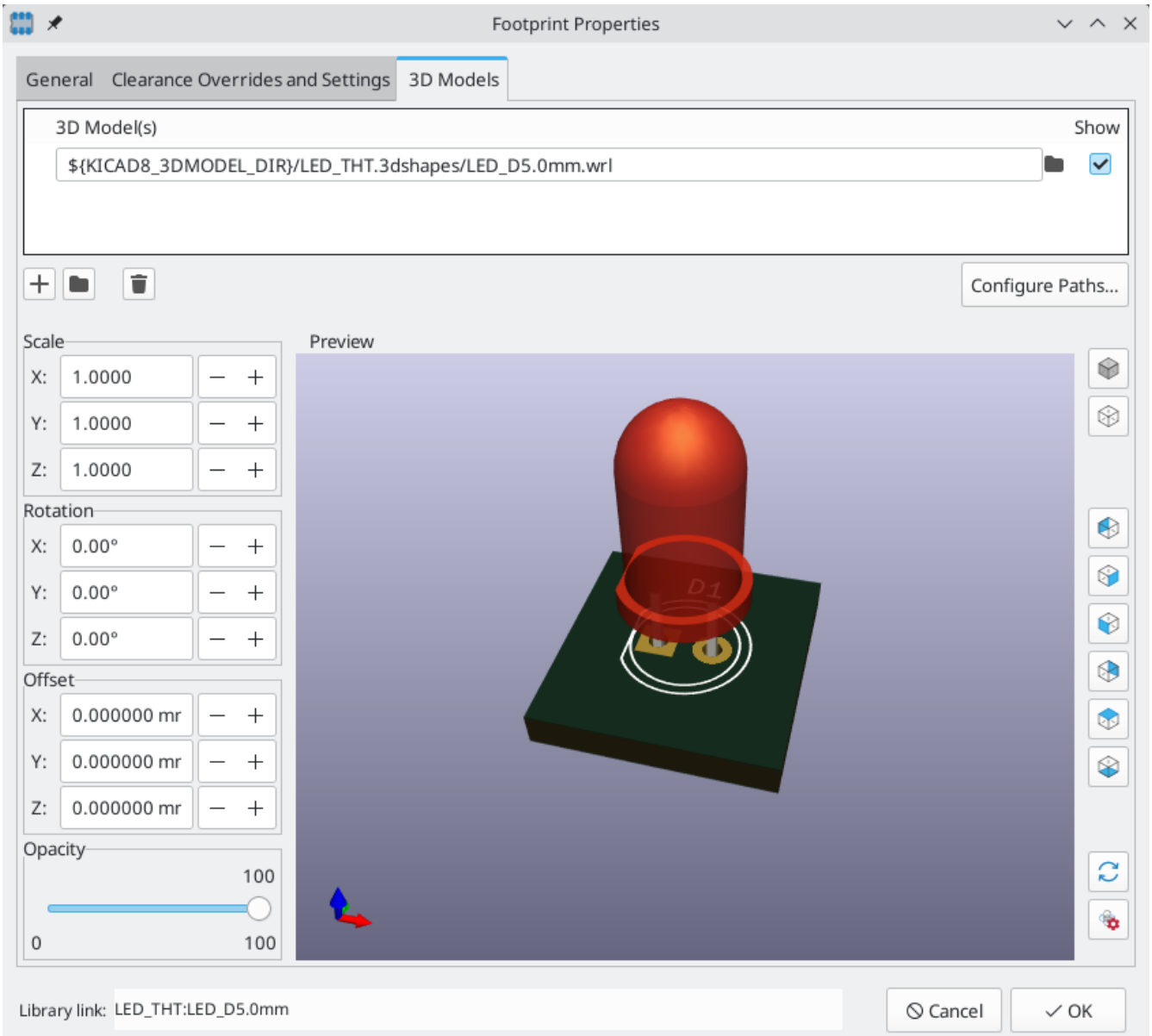
符号也可以指定封装筛选器，可以用来隐藏与符号不兼容的封装。例如，`74HC00` 符号有封装过滤器，导致只有适用的 DIP 和 SO14 封装显示在《封装筛选器，分配封装工具》中。




这个 KLC 对封装筛选器的要求 包含一些有效的封装筛选器的提示。

## 封装和 3D 模型

元件的 3D 模型被保存在单独的文件中。元件的 3D 模型的文件名被保存在封装中。每个封装可以添加任何数量的 3D 模型。3D 模型的文件名，以及模型的比例、旋转、偏移和不透明度，都在 "封装属性" 的 **3D 模型** 标签中设置。




支持 STEP（`.step`）和 VRML（`.wrl`）两种 3D 模型格式。STEP 文件在需要尺寸精度的情况下非常有用，而 VRML 文件可以用来制作更具视觉吸引力的渲染图。KiCad 库中的许多封装都有相关的 3D 模型；这些模型都以 VRML 和 STEP 格式提供。 两种模型中只有一种需要在封装中列出（通常是给出 VRML 文件名）。在为机械 CAD 目的导出电路板的 3D 模型时，KiCad 可以自动替换 STEP 版本。

| | |
|---|---|
| **NOTE** | 并非所有 KiCad 库中的封装都提供了 3D 模型，但所有封装都列出了 3D 模型的文件名，即使 3D 模型不存在。这样做是为了以后可以添加 3D 模型，而不需要编辑封装。 |

FreeCAD 与 StepUp Workbench 一起对创建元件式 3D 模型很有用；KiCad 库中的许多模型都使用了它们。StepUp 用于生成具有正确位置、缩放和旋转的 STEP 和 VRML 文件。

# 今后的发展方向

## 更多学习资源

关于如何使用 KiCad 的更多信息，请参阅 手册。

其他资源包括 KiCad 官方用户论坛，https://www.kicad.org/community/chat/[Discord 或 IRC]，以及来自 KiCad 社区的其他 学习资源。

要想了解 KiCad 的更多功能，请浏览网站的 用 KiCad 设计 部分，或打开 KiCad 附带的演示工程（**文件** → **打开演示工程...**）。

## 帮助改进 KiCad

要报告一个错误或要求一个功能，请使用 **帮助** → **报告一个错误** 或在 Gitlab 上打开一个问题。

要为 KiCad 的发展做出贡献，请参见 开发者贡献页面。用户也可以通过为 库 或 文档和翻译 提供帮助。 最后，请考虑 捐赠支持 继续开发 KiCad。