



KiCad

December 30, 2020

Contents

1 简介	1
1.1 KiCad	1
1.2 KiCad 文件和文件夹	2
2 安装和配置	4
2.1 显示选项	4
2.2 初始化默认配置	4
2.3 修改默认配置	4
2.4 路径配置	4
2.5 外部工具的初始化	5
2.5.1 选择文本编辑器	6
2.5.2 选择 PDF 查看器	6
2.6 创建一个新工程	6
2.7 导入外部工程	7
3 使用 KiCad 工程管理器	8
3.1 工程管理器窗口	8
3.2 实用程序启动窗格	9
3.3 工程树视图	9
3.4 顶部工具栏	10
4 工程模板	11
4.1 使用模板	11
4.2 模板位置:	12
4.3 创建模板	12
4.3.1 所需文件:	13
4.3.2 可选文件:	14

I	从版本 4 升级到版本 5	15
5	原理图符号库	17
5.1	全局符号库表。	17
5.2	符号库表映射	17
5.3	重新映射搜索顺序	18
5.4	符号名称和符号库昵称限制	18
6	符号缓存库可用性	19
7	板文件格式更改	20
7.1	全局封装库表。	20

参考手册

Copyright

本文件是以下列出的贡献者的版权 (c) 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 或更高版本或知识共享归因许可证 (<http://creativecommons.org/licenses/by/3.0/>) 版本的条款分发和/修改它 3.0 或更高版本。

本指南中的所有商标均属于其合法所有者。

Contributors

Jean-Pierre Charras, Fabrizio Tappero.

翻译

taotieren <admin@taotieren.com>, 2019

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

反馈

请将任何错误报告、建议或新版本引导到此处:

- 关于 KiCad 文档: <https://gitlab.com/kicad/services/kicad-doc/issues>
- 关于 KiCad 软件: <https://gitlab.com/kicad/code/kicad/issues>
- 关于 KiCad 翻译: <https://gitlab.com/kicad/code/kicad-i18n/issues>

出版日期和软件版本

2015 年 5 月 21 日。

Chapter 1

简介

1.1 KiCad

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 融合了以下软件工具的优雅集合：

- **KiCad**：工程管理器
- **Eeschema**：原理图编辑器和元件编辑器
- **Pcbnew**：电路板布局编辑器和封装编辑器
- **GerbView**：Gerber 查看器

还包括 3 个实用工具：

- **Bitmap2Component**：LOGO 的元件制造器。它创建了一个原理图元件或来自位图图片的封装。
- **PcbCalculator**：一个计算器，有助于计算调节器元件，轨道宽度与电流，传输线等
- **Pl Editor**：图框编辑器。

这些工具通常从工程管理器运行，但也可以作为独立工具运行。

KiCad 没有任何电路板尺寸限制，它可以处理多达 32 个铜层，14 个技术层和 4 个辅助层。

KiCad 可以创建构建印刷电路板所需的所有文件，包括：

- Gerber 文件为光绘机
- 钻孔文件
- 元件位置文件

作为开源 (GPL 许可) 软件, KiCad 是面向有意创建开源电子硬件的工程工程师的理想工具。

KiCad 适用于 Linux, Windows 和 Apple macOS。

1.2 KiCad 文件和文件夹

KiCad 创建并使用具有以下特定文件扩展名（和文件夹）的文件进行原理图和电路板编辑。

工程管理器文件：

*.pro	包含当前工程的一些参数的小文件，包括元件库列表。
-------	--------------------------

原理图编辑器文件：

*.sch	原理图文件，不包含元件本身。
*.lib	原理图元件库文件，包含元件描述：图形形状，引脚，字段。
*.dcm	原理图元件库文档，包含一些元件描述：评论，关键字，数据表参考。
*_cache.lib	原理图元件库缓存文件，包含原理图工程中使用的元件的副本。
sym-lib-table	符号库列表（符号库表）：原理图编辑器中可用的符号库列表。

板编辑器文件和文件夹：

*.kicad_pcb	板文件，包含除图框之外的所有信息。
*.pretty	封装库文件夹。文件夹本身就是库。
*.kicad_mod	封装文件，每个文件包含一个封装说明。
*.brd	旧版格式的电路板文件。当前的电路板编辑器可以读取但不能写入。
*.mod	旧格式的封装库。可以通过封装或电路板编辑器读取，但不能写入。
fp-lib-table	封装库列表（封装库表）：加载的封装库列表（各种格式）由板或封装编辑或 CvPcb。

* 公共文件：*

*.kicad_wks	图框描述文件，适用于需要工作表的用户具有自定义外观。
.net	由原理图创建的网表文件，由电路板编辑器读取。对于喜欢单独文件的用户，此文件与.cmp 文件相关联用于元件/封装关联。

特殊文件：

*.cmp	原理图中使用的元件与其封装之间的关联。它可以由 Pcbnew 创建并由 Eeschema 导入。其目的是为用户导入从 Pcbnew 到 Eeschema 的更改谁更改 Pcbnew 内的封装（例如使用 改变封装命令）并希望在原理图中导入这些更改。
-------	---

其他文件：

它们由 KiCad 生成用于制作或文档。

*.gbr	Gerber 文件，用于制作。
*.drl	钻孔文件（Excellon 格式），用于制作。
*.pos	位置文件（ASCII 格式），用于自动插入机器。
*.rpt	报告文件（ASCII 格式），用于文档。

*.ps	绘图文件 (Postscript), 用于文档。
*.pdf	绘图文件 (PDF 格式), 用于文档。
*.svg	绘图文件 (SVG 格式), 用于文档。
*.dxf	绘图文件 (DXF 格式), 用于文档。
*.plt	绘图文件 (HPGL 格式), 用于文档。

Chapter 2

安装和配置

2.1 显示选项

Pcbnew 和 Gerbview 中的硬件加速渲染器需要支持 OpenGL v2.1 或更高版本的显卡。

2.2 初始化默认配置

名为 **kicad.pro** 的默认配置文件在 `kicad/template` 中提供。它作为任何新工程的模板，用于设置 Eeschema 加载的库文件列表。Pcbnew 的一些其他参数（默认文本大小，默认行粗细等）也存储在此处。

可能存在另一个名为 **fp-lib-table** 的默认配置文件。它只会被用一次来创建封装库列表；否则列表将从头开始创建。

2.3 修改默认配置

如果需要，可以自由修改默认的 **kicad.pro** 文件。

验证您是否具有 `kicad/template/kicad.pro` 的写入权限

运行 KiCad 并加载 **kicad.pro** 工程。

通过 KiCad 管理器运行 Eeschema。修改和更新 Eeschema 配置，以设置每次创建新工程时要使用的库列表。

通过 KiCad 管理器运行 Pcbnew。修改和更新 Pcbnew 配置，尤其是封装库列表。Pcbnew 将创建或更新名为 **封装库列表** 的库列表文件。有 2 个库列表文件（名为 `fp-lib-table`）：第一个（位于用户主目录中）对于所有工程都是全局的，第二个（位于工程目录中）是可选的，专用于工程。

2.4 路径配置

在 KiCad 中，可以使用 *environment* 变量定义路径。一些环境变量由 KiCad 在内部定义，可用于定义库，3D 形状等的路径。

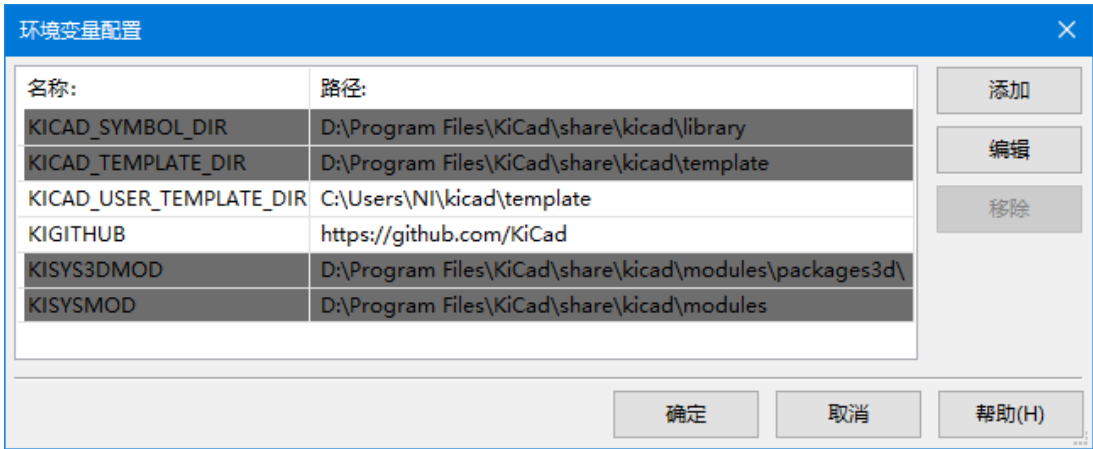
当绝对路径未知或可能发生变化时（例如，当您将工程传输到另一台计算机时），以及许多类似工程共享一个基本路径时，这非常有用。请考虑以下可能安装在不同位置的内容：

- Eeschema 元件库
- Pcbnew 封装库
- 封装定义中使用的 3D 形状文件

例如, 使用 **KISYSMOD** 环境变量时, *connect.pretty* 封装库的路径将被定义为 ***\${KISYSMOD}/connect.pretty*** 此选项允许您使用环境变量定义路径，并根据需要添加您自己的环境变量以定义个人路径。

KiCad 环境变量：

KICAD_PTEMPLATES	工程创建期间使用的模板（5.0.0-rc2 版本开始已经弃用，改为使用 KICAD_TEMPLATE_DIR）。如果您使用此变量，则必须定义它。
KICAD_SYMBOL_DIR	符号库文件的基本路径。
KIGITHUB	经常用于示例封装库表。如果您使用此变量，则必须定义它。
KISYS3DMOD	3D 形状文件的基本路径，并且必须定义，因为通常不使用绝对路径。
KISYSMOD	封装库文件夹的基本路径，如果在封装库名称中未使用绝对路径，则必须定义。
KICAD_TEMPLATE_DIR	使用 KiCad 安装的模板的位置。
KICAD_USER_TEMPLATE_DIR	个人模板的位置。



另请注意，环境变量 **KIPRJMOD** 始终由 KiCad 内部定义，并且是 **当前工程绝对路径**。

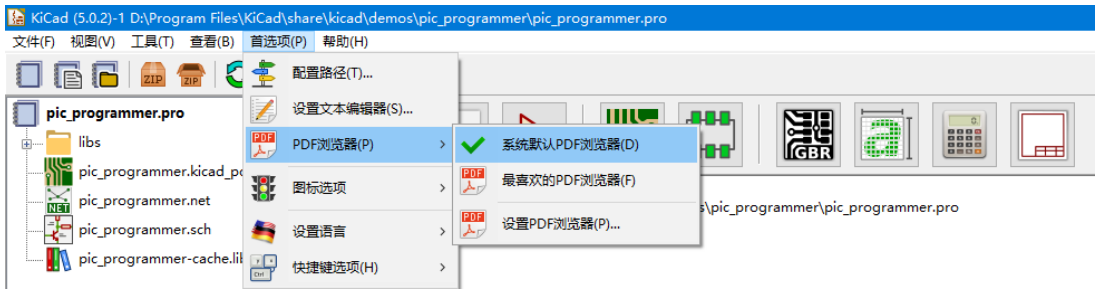
例如，***\${KIPRJMOD}/connect.pretty*** 总是在 *connect.pretty* 文件夹（漂亮的封装库）中找到的 **当前工程文件夹**。

如果您修改路径配置，请退出并重新启动 KiCad 避免路径处理中的任何问题。

2.5 外部工具的初始化

您可以定义自己喜欢的文本编辑器和 PDF 查看器。只要您想打开文本或 PDF 文件，就会使用这些设置。

可以从“首选项”菜单访问这些设置：



2.5.1 选择文本编辑器

在使用文本编辑器浏览/编辑当前工程中的文件之前，必须选择要使用的文本编辑器。

选择 **首选项** → **设置文本编辑器** 以设置要使用的文本编辑器。

2.5.2 选择 PDF 查看器

您可以使用默认的 PDF 查看器或选择自己的 PDF 查看器。

要更改默认 PDF 查看器，请使用 **首选项** → **PDF 查看器** → **设置 PDF 查看器** 以选择 PDF 查看器程序，然后选择 **首选项** → **PDF 查看器** → **偏好的 PDF 查看器**。

在 Linux 上，已知默认的 PDF 查看器是脆弱的，因此建议选择自己的 PDF 查看器。

2.6 创建一个新工程

为了管理由原理图文件，印刷电路板文件，补充库，用于照片跟踪，钻孔和自动元件放置文件的制造文件组成的 KiCad 工程，建议按如下方式创建工程：

- 为工程创建一个工作目录（使用 KiCad 或其他方式）。
- 在此目录中，使用 KiCad 创建工程文件（带有的文件扩展名.pro）通过“创建一个新工程”或“从模板创建新工程”图标。



Warning

为每个 KiCad 工程使用唯一的目录。不要将多个工程组合到一个目录中。

KiCad 创建一个扩展名为.pro 的文件，该文件维护了许多工程管理参数（例如原理图中使用的库列表）。主要原理图文件和印刷电路板文件的默认名称都是从工程名称派生的。因此，如果在名为 **example** 的目录中创建了名为 **example.pro** 的工程，则将创建默认文件：

example.pro	工程管理文件。
example.sch	主要原理图文件。
example.kicad_pcb	印刷电路板文件。
example.net	网表文件。
example.*	由其他实用程序创建的各种文件。
example-cache.lib	由自动创建和使用的库文件原理图编辑器，包含原理图中使用的组件的备份。

2.7 导入外部工程

KiCad 能够导入使用其他软件包创建的文件。目前仅支持 Eagle 6.x 或更新版本（XML 格式）。

要导入外部工程，需要在导入文件浏览器对话框中选择原理图文件或电路板文件。导入的原理图和电路板文件应具有相同的基本文件名（例如 project.sch 和 project.brd）。选择所请求的文件后，系统将要求您选择存储导入文件的目录，这些目录将保存为 KiCad 工程。

Chapter 3

使用 KiCad 工程管理器

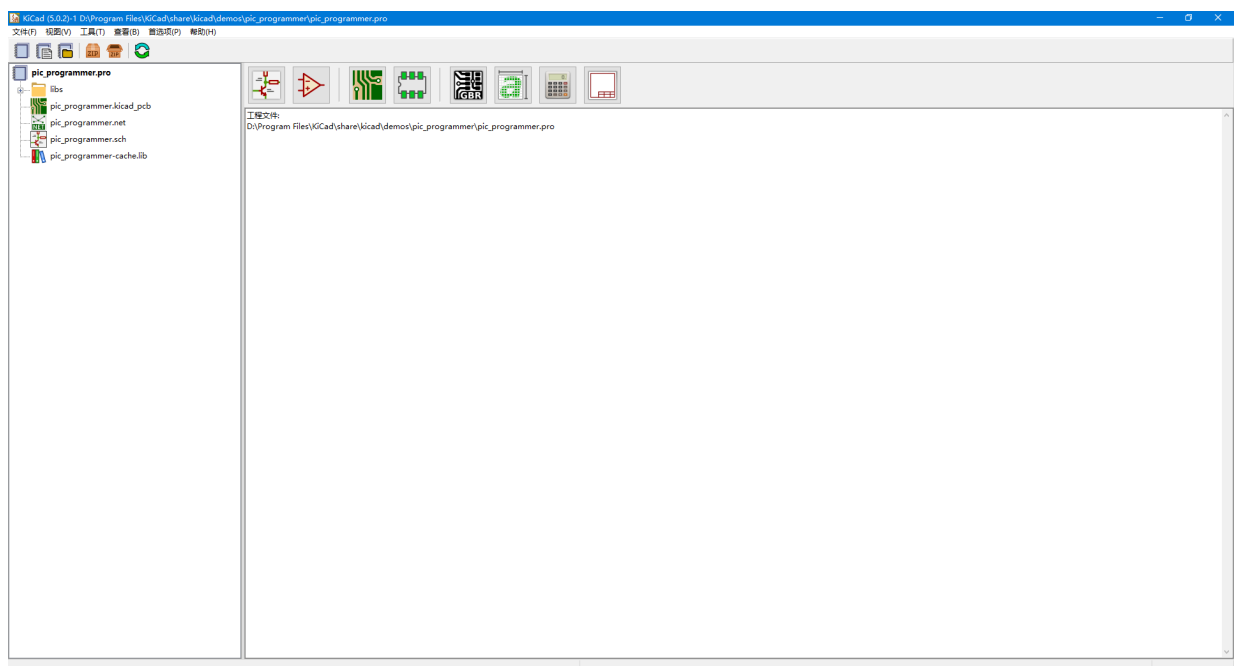
KiCad 工程管理器 (kicad 或 kicad.exe) 是一个工具，可以在创建设计时轻松运行其他工具（原理图和 PCB 编辑器，Gerber 查看器和实用工具）。

从 KiCad 管理器运行其他工具有一些优点：

- 在原理图编辑器和电路板编辑器之间交叉探测。
- 在原理图编辑器和封装选择器 (CvPcb) 之间进行交叉探测。

但是，您只能编辑当前工程文件。当这些工具在 独立模式下运行时，您可以在任何工程中打开任何文件，但在工具之间交叉探测会产生奇怪的结果。

3.1 工程管理器窗口



主 KiCad 窗口由工程树视图，包含用于运行各种软件工具的按钮的启动窗格和消息窗口组成。菜单和工具栏可用于创建，读取和保存工程文件。

3.2 实用程序启动窗格

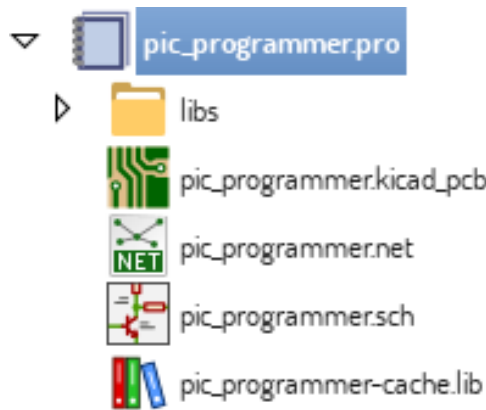
KiCad 允许您运行随附的所有独立软件工具。

启动窗格由下面的 8 个按钮组成，对应于以下命令（从左到右为 1 到 8）：



1	Eeschema	原理图编辑器。
2	LibEdit	元件编辑器和元件库管理器。
3	Pcbnew	电路板布局编辑器。
4	FootprintEditor	封装编辑器和封装库管理器。
5	Gerbview	Gerber 文件查看器。它还可以显示钻孔文件。
6	Bitmap2component	用于构建封装或元件的工具用于创建 LOGO 的 B & W 位图图像。
7	Pcb 计算器	计算轨道宽度的工具，以及许多其他工具的东西。
8	Pl 编辑器	图框编辑器，用于创建/自定义框架引用。

3.3 工程树视图

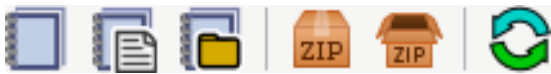


双击原理图文件运行原理图编辑器，在这种情况下打开文件 `pic_programmer.sch`。







双击电路板文件运行布局编辑器，在这种情况下打开文件 `pic_programmer.kicad_pcb`。

右键单击工程树中的任何文件都可以进行通用文件操作。

3.4 顶部工具栏



KiCad 顶部工具栏允许一些基本的文件操作：

	创建一个新工程。如果找到默认模板文件（kicad.pro） kicad/template ， 将其复制到工作目录中。
	从现有模板创建新工程。
	打开现有工程。
	更新并保存当前工程树。
	创建整个工程的 zip 压缩。这包括原理图文件，库，PCB 等
	刷新树视图，有时在树更改后需要。

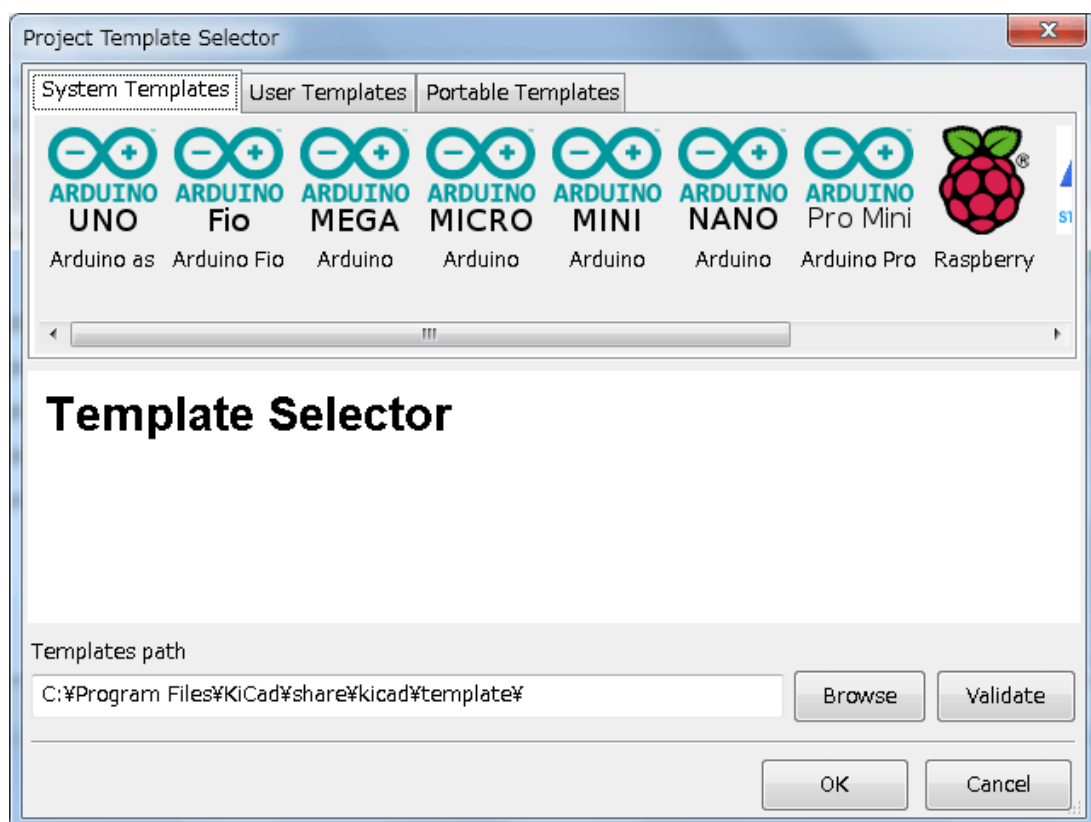
Chapter 4

工程模板

使用工程模板有助于使用预定义设置设置新工程。模板可能包含预定义的电路板边框，连接器位置，原理图元素，设计规则等。甚至可以包括用作新工程的种子文件的完整原理图和/或 PCB。

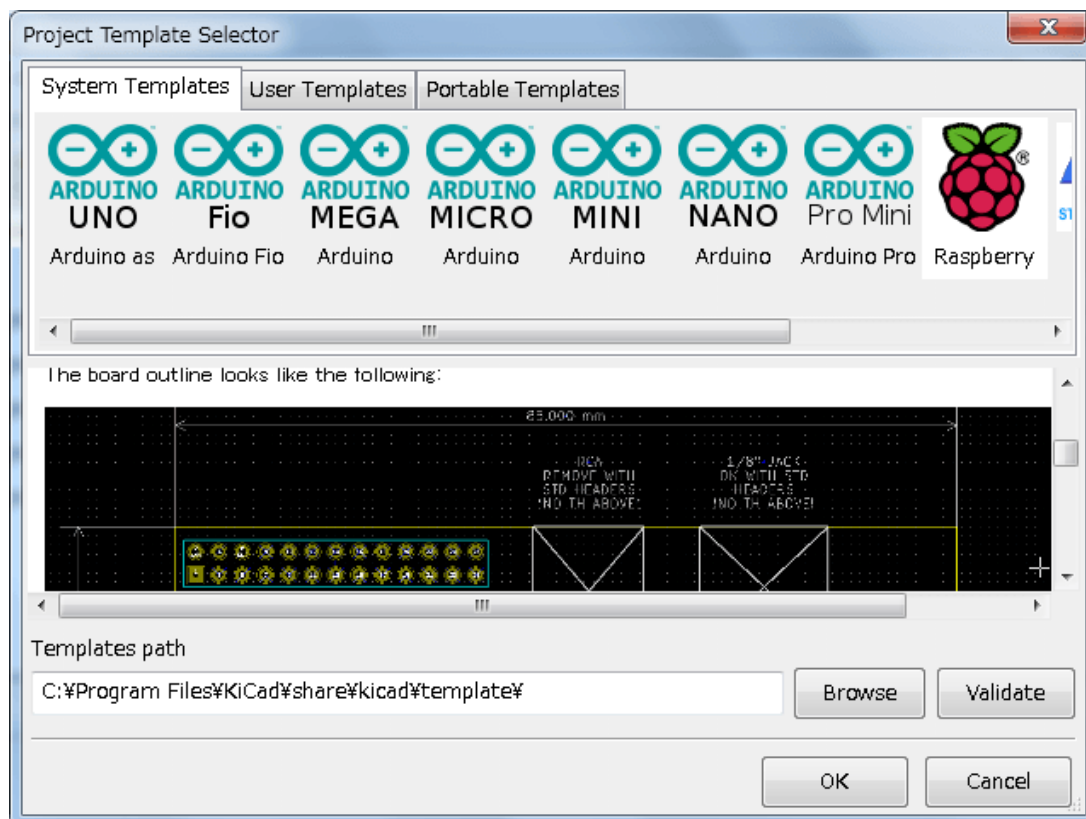
4.1 使用模板

文件 → 新工程 → 模板菜单中的新工程将打开工程模板选择器对话框：



单击模板的图标将显示模板信息，再单击“确定”按钮将创建新工程。模板文件将复制到新工程位置并重命名以反映新工程的名称。

选择模板后：



4.2 模板位置：

KiCad 在以下路径中查找模板文件：

- 环境变量 KICAD_USER_TEMPLATE_DIR 中定义的路径
- 环境变量 KICAD_TEMPLATE_DIR 中定义的路径
- 系统模板： <kicad bin dir>/../share/kicad/template/
- 用户模板：
 - Unix: ~/kicad/templates/
 - Windows: C:\Documents and Settings\username\My Documents\kicad\templates
 - Mac: ~/Documents/kicad/templates/
- 当定义环境变量 KICAD_PTEMPLATES 时,会有第三个选项卡,即可移植模板,其中列出了在 KICAD_PTEMPLATES 路径 (DEPRECATED) 中找到的模板。

4.3 创建模板

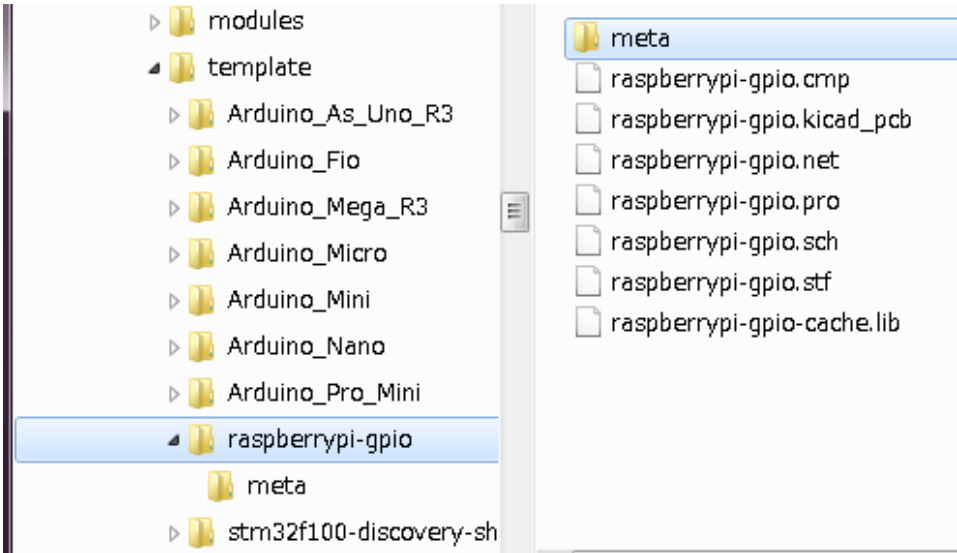
模板名称是存储模板文件的目录名称。元数据目录是名为 **meta** 的子目录，其中包含描述模板的文件。

使用模板创建工程时，模板中的所有文件和目录都将复制到新工程路径，**meta** 除外。

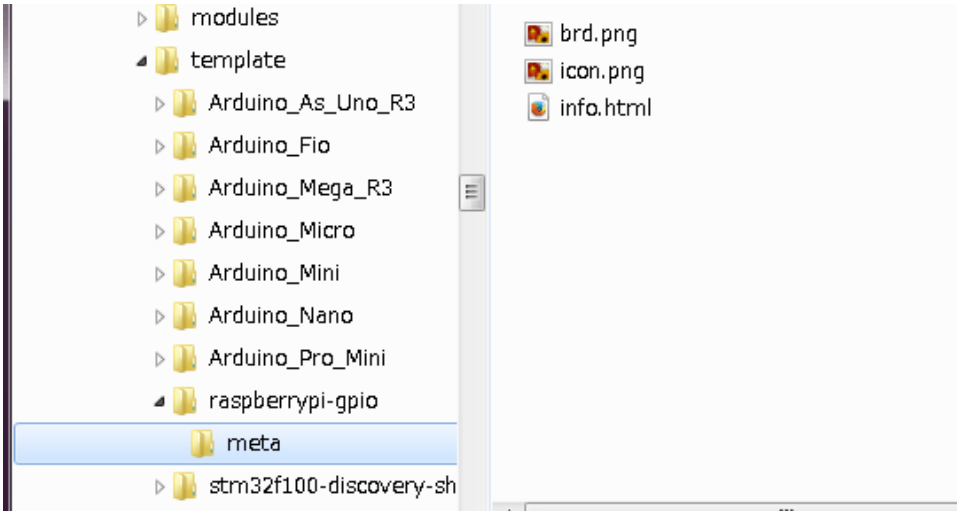
从模板创建新工程时，将使用新工程文件名重命名以模板名称开头的所有文件和目录，不包括文件扩展名。

元数据由一个必需文件组成，可能包含可选文件。所有文件必须由用户使用文本编辑器或以前的 KiCad 工程文件创建，并放入所需的目录结构中。

这是一个显示 **raspberrypi-gpio** 模板的工程文件的示例：



和元数据文件：



4.3.1 所需文件：

meta/info.html	描述模板的 HTML 格式信息。
----------------	------------------

<title> 标签确定向用户公开的模板的实际名称以进行模板选择。请注意，如果工程模板名称太长，则会将其截断。由于字体字距调整，通常可以显示 7 或 8 个字符。

使用 HTML 意味着可以轻松地对图像进行内联，而无需发明新方案。本文档中只能使用基本 HTML 标记。

这是一个示例 **info.html** 文件：

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<HTML>
<HEAD>
<META HTTP-EQUIV="CONTENT-TYPE" CONTENT="text/html;
charset=windows-1252">
<TITLE>Raspberry Pi - 扩展板</TITLE>
</HEAD>
<BODY LANG="fr-FR" DIR="LTR">
<P>这个工程模板是扩展板的基础
<A HREF="http://www.raspberrypi.org/" TARGET="blank">Raspberry Pi $25
ARM 板。</A> <BR><BR>该基础工程包括一个PCB边缘，
其尺寸与Raspberry-Pi PCB相同，
连接器正确放置以对齐两块板。Raspberry-Pi 板上的所有 IO 都通过
0.1"连接到工程。扩展
头。 <BR><BR>板轮廓如下所示：
</P>
<P><IMG SRC="brd.png" NAME="brd" ALIGN=BOTTOM WIDTH=680 HEIGHT=378
BORDER=0><BR><BR><BR><BR>
</P>
<P>(c)2012 Brian Sidebotham<BR>(c)2012 KiCad Developers</P>
</BODY>
</HTML>
```

4.3.2 可选文件：

meta/icon.png	一个 64 x 64 像素的 PNG 图标文件，用作模板选择对话框中的可单击图标。
---------------	---

meta/info.html 使用的任何其他图像文件（例如上面对话框中的电路板文件图像）也会放在此文件夹中。

Part I

从版本 4 升级到版本 5

在版本 5 开发期间对 KiCad 的行为进行了更改，这可能会影响使用旧版 KiCad 创建的工程。本节是升级到 KiCad 第 5 版时确保最顺畅路径的指南。

Chapter 5

原理图符号库

不再使用符号（在版本 4 中称为元件）查找列表来访问原理图符号库。符号库现在由符号库表管理，其行为类似于封装库表。此更改是一项重大改进，但某些原理图在转换为版本 5 时可能需要手动干预。

在以前的版本中，当在 Eeschema 文件中定位符号时，KiCad 使用库文件列表进行搜索。在定位符号时，将搜索每个路径，并使用包含符号名称的第一个库。

从 v5 开始，KiCad 符号名称以昵称为前缀，并使用 [lookup 表将昵称与库路径匹配](#) 找到包含符号的库。该表称为“符号库表”，并根据存储在用户的 KiCad 配置目录和当前加载的工程目录中的配置文件构建。

要将 KiCad 工程从 v4 升级到 v5，需要创建所有库文件的昵称，然后原理图符号名称需要以正确的昵称作为前缀。

5.1 全局符号库表。

首次启动时，Eeschema v5 将自动创建一个全局符号表。您将有机会跳过此操作并手动创建自己的全局符号表。如果根本不使用 KiCad 符号库，则只需执行此操作。否则，更容易修改自动生成的全局符号表。

Note

如果您跟踪 [symbol 库存储库](#)，则 KiCad 不会跟踪对默认全局符号库表所做的更改。您必须手动保持全局符号库表是最新的。

5.2 符号库表映射

每当打开未重新映射的原理图时，将执行符号的自动重新映射。您应该提前采取一些步骤，以使重映射最有效。

Note

如果您一直在使用 KiCad 的开发版本，请从安装有 KiCad 库的模板文件夹或中复制完整的默认全局符号库表文件 (sym-lib-table) [KiCad library repo](#) 到你的 KiCad 用户配置文件夹。这将取代 Eeschema 创建的空（最有可能）。如果你不这样做，你很可能最终得到一堆破碎的符号链接。

**Warning**

重新映射的原理图与旧版本的 KiCad 不兼容。“重映射符号”对话框将备份原理图文件，如果手动重新映射，则应该执行相同的操作。+

1. 如果可能，请在您的系统上安装 KiCad 的第 4 版，除非您从未使用过与 KiCad 一起分发的任何符号库。
2. 如果在启动 Eeschema 版本 4 时收到有关缺少库的警告，请确保在尝试重新映射原理图之前修复缺少的库（如果它们包含原理图中的符号）。否则，将找不到正确的符号，您的原理图中最终会出现符号链接断开。您可以通过左键单击原理图中的符号并验证是否未从缓存库加载符号来对此进行测试。如果从缓存库加载符号，则 Eeschema 无法在系统或工程符号库中找到您的元件。如果需要缓存部件可供系统上的其他工程使用，则需要手动将其集成到系统或工程库中。
3. 如果在重映射过程中需要符号恢复，请不要忽略它。无法恢复符号将导致符号链接断开或原理图中链接的符号错误。
4. 在重映射过程中，未在全局符号库表中找到的符号库将用于创建工程特定的符号库表。如果这是您的首选项，您可以手动将它们移动到全局符号库表。
5. 要获得最准确的重映射，请通过将工程缓存文件（project-name-cache.lib）复制到其他文件并将其添加到符号库列表的顶部来创建工程库。您必须在符号库表实现之前使用 KiCad 版本才能执行此操作。

Note

提供了一个工具来尝试修复重映射问题。如果原理图中缺少符号库链接，可以通过打开“工具 -> 编辑符号库参考...”菜单条目并单击“映射孤立”按钮来修复它们。

5.3 重新映射搜索顺序

重新映射符号时，KiCad 按以下顺序继续将库分配给符号：

1. 全局符号库表：符号优先映射到全局符号库表（如果存在）。
2. 工程专用库：接下来搜索工程库列表中列出的不在全局符号库表中的库。
3. 工程缓存文件：如果上面列出的库中不存在符号，则首先获取该符号 — 从缓存中复制并放置在 *proj-rescue.lib* — 在符号映射到此新文件之前，救援库。

5.4 符号名称和符号库昵称限制

符号名称不能包含 “<SPACE>”，“:”，“/”。

库昵称可能不包含 “<SPACE>”，“:”。

必须通过手动编辑相关的原理图和库文件来重命名具有这些字符的现有符号名称。

Chapter 6

符号缓存库可用性

缓存库不再显示在符号库查看器或符号库编辑器中。永远不应编辑缓存，因为下一个原理图保存会覆盖任何更改。

Chapter 7

板文件格式更改

Pcbnew 中添加了一些新功能，这些功能会影响电路板文件格式。在电路板设计中使用这些新功能将阻止使用以前版本的 Pcbnew 打开它们。

- 圆角矩形封装焊盘。
- 定制形状封装焊盘。
- 封装填充名称超过四个字符。
- 将禁止布线层（Keep out）保留在多个图层上。
- 3D 模型偏移保存为毫米而不是英寸。
- 封装文本锁定。

7.1 全局封装库表。

如果您跟踪 [封装库存储库](#)，则 KiCad 不会跟踪对默认全局封装库表所做的更改。您必须手动保持全局覆盖库表是最新的。