

KiCad

The KiCad Team

Table of Contents

简介.....	2
系统要求.....	2
KiCad 文件和文件夹.....	2
安装和升级 KiCad.....	5
导入设置.....	5
从早期版本迁移文件.....	5
使用 KiCad 工程管理器.....	7
工程管理器窗口.....	7
工具启动器.....	7
工程树视图.....	8
顶部工具栏.....	8
创建一个新工程.....	9
从其他 EDA 工具导入工程.....	9
KiCad 配置.....	11
共同偏好设置.....	11
鼠标和触摸板偏好设置.....	12
快捷键偏好设置.....	13
路径配置.....	14
库配置.....	15
工程模板.....	16
使用模板.....	16
模板位置:.....	17
创建模板.....	17

参考手册

版权

本文档由其贡献者拥有版权 © 2010-2018，如下所列。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，或知识共享署名许可证 (<http://creativecommons.org/licenses/by/3.0/>)，版本 3.0 或更高版本的条款分发和/或修改它。

本指南中的所有商标均属于其合法所有者。

贡献者

Jean-Pierre Charras, Fabrizio Tappero.

翻译人员

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

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>

软件版本

KiCad 6.0

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
- **图框编辑器**：创建用于原理图和 PCB 的绘图模板 (工作表) 的工具。

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 不存在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 13 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的工程师的理想工具。

KiCad 适用于 Linux, Windows 和 Apple macOS。

系统要求

KiCad 能够在多种硬件和操作系统上运行，但在低端硬件上执行某些任务可能会很慢或很困难。为获得最佳体验，建议使用 1920x1080 或更高分辨率的专用显卡和显示器。

有关最新的系统要求，请查看 KiCad 网站：<https://kicad-pcb.org/help/system-requirements/>

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：

*kicad_pro	工程文件，包含原理图和 PCB 之间共享的设置
*.pro	旧版 (KiCad 5.x 和更早版本) 工程文件。可以读取，并将由工程管理器转换为 <i>.kicad_pro</i> 文件。

原理图编辑器文件:

*.kicad_sch	包含所有信息和元件本身的原理图文件。
*.kicad_sym	原理图符号库文件, 包含元件描述: 图形形状、引脚、字段。
*.sch	旧版 (KiCad 5.x 和更早版本) 原理图文件。可以读取, 并将在写入时转换为 <i>.kicad_sch</i> 文件。
*.lib	旧版 (KiCad 5.x 和更早版本) 原理图库文件。可以读取, 但不能写入。
*.dcm	旧版 (KiCad 5.x 和更早版本) 原理图库文档。可以读取, 但不能写入。
*_cache.lib	旧版 (KiCad 5.x 和更早版本) 原理图元件库缓存文件。正确加载旧版原理图 (<i>.sch</i>) 文件所需。
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	包含除板框以外的所有信息的电路板文件。
*.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描述。
*.kicad_dru	设计规则文件, 包含某个 <i>.kicad_pcb</i> 文件的自定义设计规则。
*.brd	旧版 (KiCad 4.x 和更早版本) 电路板文件。当前电路板编辑器可以读取, 但不能写入。
*.mod	旧版 (KiCad 4.x 和更早版本) 封装库文件。可由封装或线路板编辑器读取, 但不能写入。
fp-lib-table	封装库列表 (封装库表): 线路板编辑器中可用的封装库的列表。
fp-info-cache	缓存以加速封装库的加载。

* 常用文件: *

*.kicad_wks	图框页面布局 (图形边框和标题栏) 描述文件
*.net	原理图创建的网表文件, 由电路板编辑器读取。此文件与 <i>.cmp</i> 文件相关联, 适用于偏爱元件/封装关联的单独文件的用户。
*.kicad_prl	当前工程的本地设置, 帮助 KiCad 记住上次使用的设置, 如图层可见性或选择过滤器。可能不需要与工程一起分发或置于版本控制之下。

其他文件:

*.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 <i>改变封装</i> 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。

存储和发送 KiCad 文件

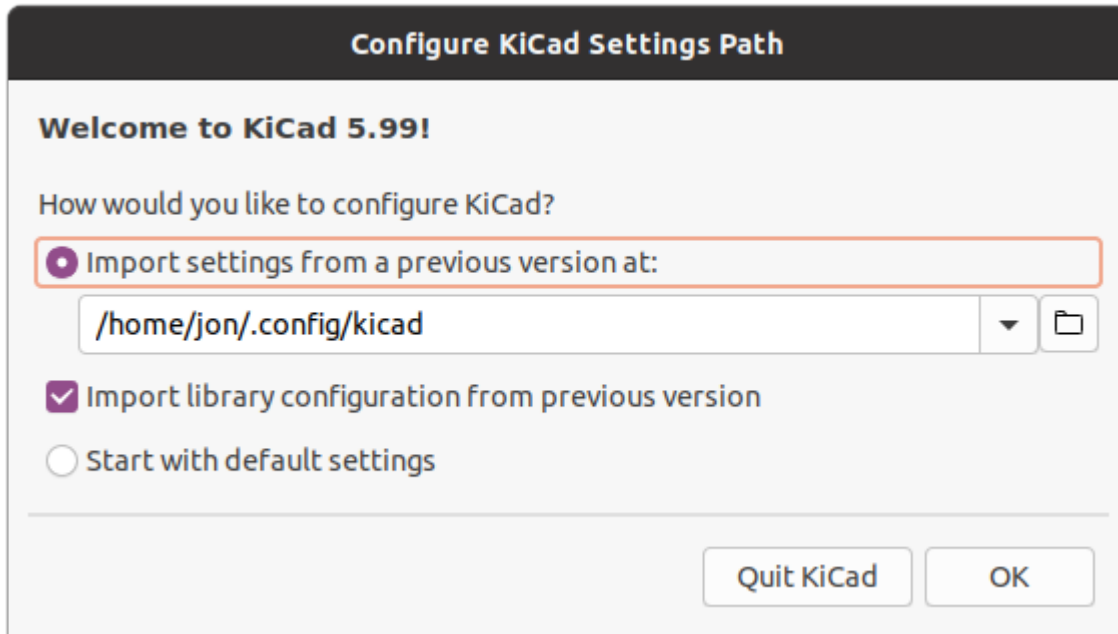
KiCad 原理图和电路板文件包含设计中使用的所有原理图符号和封装, 因此您可以自己备份或发送这些文件, 而不会出现问题。一些重要的设计信息存储在工程文件 (*.kicad_pro*) 中, 因此, 如果要发送完整的设计, 请确保将其包括在内。

某些文件 (如工程本地设置文件 (*.kicad_prj*) 和 *fp-info-cache* 文件) 不需要随工程一起发送。如果您使用诸如 Git 之类的版本控制系统来跟踪 KiCad 工程, 则可能需要将这些文件添加到忽略的文件列表中, 这样就不会跟踪它们。

安装和升级 KiCad

导入设置

KiCad 的每个主要版本都有自己的配置，因此您可以在同一台计算机上运行多个 KiCad 版本，而不会干扰配置。首次运行新版本的 KiCad 时，系统将询问如何初始化设置：



如果检测到以前版本的 KiCad，您可以选择从该版本导入设置。系统会自动检测以前配置文件的位置，但如果需要，您可以覆盖该位置以选择其他位置。

默认情况下，还将导入早期版本 KiCad 中的原理图符号和封装库表。如果您希望从无库配置开始，请取消选中 **从以前版本导入库配置** 复选框。

如果您不想从以前的版本导入设置，也可以选择从默认设置开始。

KiCad 将设置文件存储在用户目录内的文件夹中。每个 KiCad 版本都会将其设置存储在该文件夹的子文件夹中 (KiCad 5.1 及更早版本除外，它不使用子文件夹)。这些文件夹是：

Windows	%APPDATA%\kicad
Linux	~/.config/kicad
Mac OS	/Users/<username>/Library/Preferences/kicad

从早期版本迁移文件

现代版本的 KiCad 可以打开在早期版本中创建的文件，但只能写入最新格式的文件。这意味着，通常情况下，除了打开文件外，从以前版本迁移文件不需要特殊步骤。在某些情况下，文件的扩展名从一个 KiCad 版本更改到下一个版本。打开这些文件后，它们将以新的文件扩展名以新格式保存。旧文件不会自动删除。

一般来说，一个新版本的 KiCad 创建或修改的文件 **不能** 被旧版本的 KiCad 打开。因此，在测试新的 KiCad 版本时保留项目的备份副本非常重要，直到您确信不再需要使用旧的 KiCad 版本。

NOTE

目前不会从以前的版本导入快捷键配置。您可以通过将各种 `*.hotkeys` 文件从旧版本配置目录复制到新版本配置目录中，手动导入快捷键配置。如果这样做，请注意 KiCad 不会自动检测冲突，例如一个键被分配给多个操作。

使用 KiCad 工程管理器

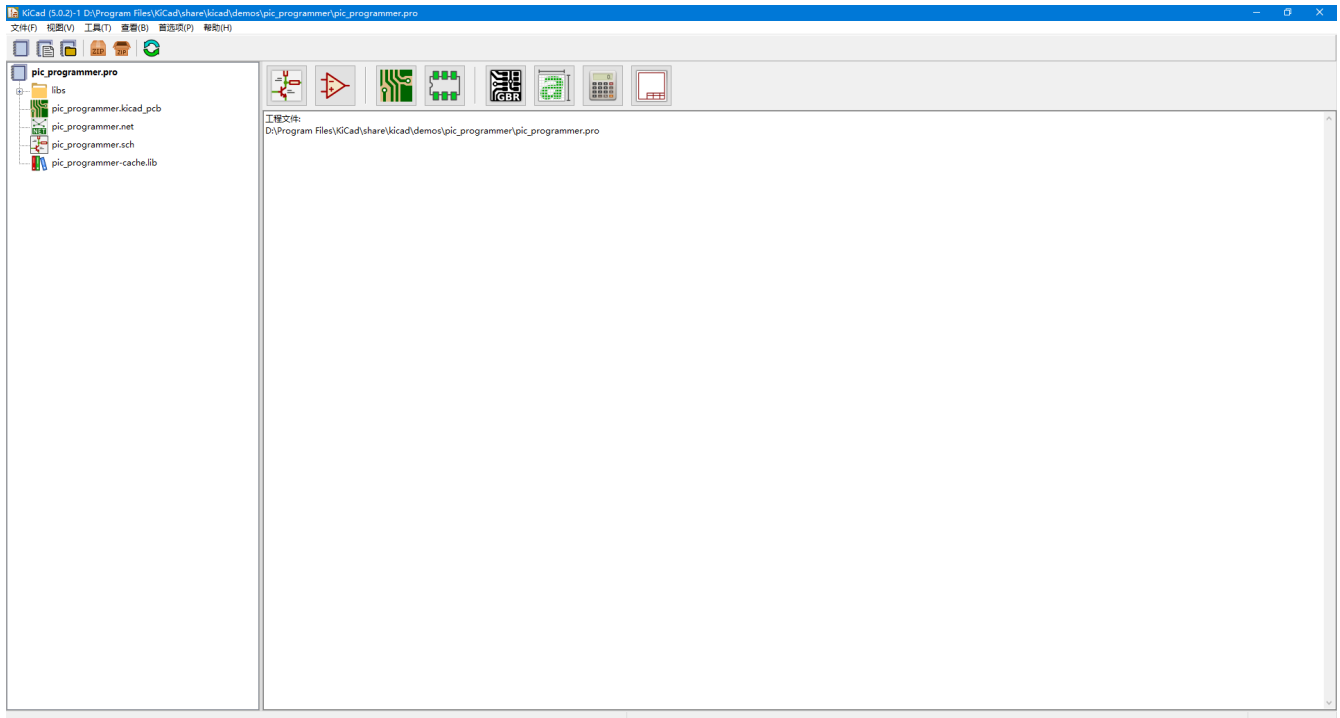
KiCad 工程管理器 (kicad 或 kicad.exe) 是一个在创建设计时可以轻松运行其他工具 (原理图和电路板编辑器、Gerber 查看器和实用工具) 的工具。

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。
- 原理图编辑器和电路板编辑器之间的设计同步 (无需创建网表文件)

KiCad 目前一次只支持打开一个工程。从 KiCad 工程管理器运行原理图和电路板编辑器时, 您只能编辑与打开的工程相关的原理图和电路板。当这些工具在 独立 模式下运行时, 您可以打开任何工程中的任何文件, 但工具之间的交叉探测可能会产生奇怪的结果。

工程管理器窗口



KiCad 工程管理器窗口由工程树视图、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。

工具启动器

工具启动器从工程管理器打开不同的 KiCad 工具。打开原理图或电路板编辑器时, 工程文件将自动加载。

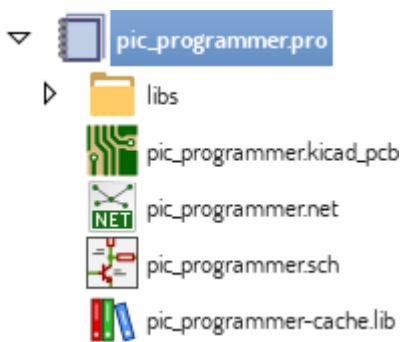
启动器中的 8 个按钮对应于以下命令(从左到右, 从 1 到 8):



1	Eeschema	原理图编辑器。
2	LibEdit	原理图符号编辑器和符号库管理器。
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	从黑白位图图像创建封装或元件以创建 logo 的工具。
7	Pcb 计算器	用于计算布线宽度和许多其他内容的工具。
8	图框编辑器	图框 (图形边框和标题栏) 编辑器

工程树视图

树状视图显示工程文件夹内的文件列表。



双击原理图文件运行原理图编辑器，在本例中打开文件 **picProgrammer.kicad_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。

NOTE | 只有 KiCad 知道如何打开的文件才会显示在工程树状视图中。

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.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。
	创建整个工程的 zip 压缩。这包括原理图文件，库，PCB 等。
	将工程 zip 压缩包解压到目录中。目标目录中的文件将被覆盖。
	刷新树视图，有时在树更改后需要。
	在文件资源管理器中打开工程工作目录。

创建一个新工程

大多数 KiCad 设计都是从创建工程开始的。从 KiCad 工程管理器创建工程有两种方式：可以创建空工程，也可以基于现有模板创建工程。本节将介绍如何创建一个新的空工程。从模板创建工程在《工程 - 模板，工程模板》一节中介绍。

若要创建新工程，请使用 **文件** 菜单中的 **新建工程...** 命令，或单击顶部工具栏中的 **新建工程** 按钮，或使用键盘快捷键 (默认为 Ctrl+N)。

系统将提示您输入工程名称。默认情况下，将使用相同的名称为您的工程创建一个目录。例如输入名称 MyProject，KiCad 会在其中创建 MyProject 目录和工程文件 MyProject/MyProject.kicad_pro。

如果您已经有一个存储工程文件的目录，可以在 **新建工程** 对话框中取消选中 *为工程创建新目录* 复选框。

NOTE | 强烈建议您将每个 KiCad 工程保存在其自己的目录中。

选择工程名称后，KiCad 将在工程目录中创建以下文件：

example.kicad_pro	KiCad 工程文件。
example.kicad_sch	主原理图文件。
example.kicad_pcb	印刷电路板文件。

从其他 EDA 工具导入工程

KiCad 可以导入由其他一些软件包创建的文件。目前支持以下工程类型：

*.sch, *.brd	Eagle 6.x 或更高版本 (XML 格式)
*.csa, *.cpa	CADSTAR 档案格式

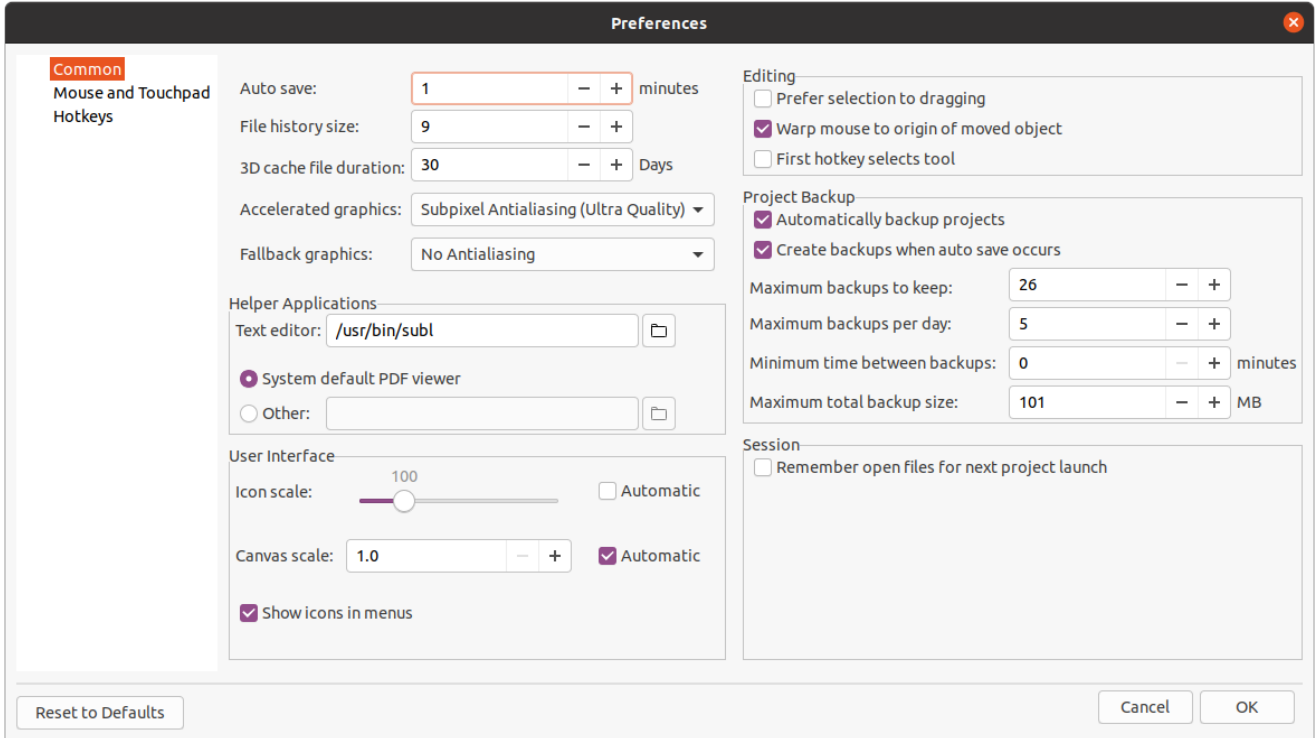
要从这些工具中导入工程，请在 **文件** 菜单的 **导入非 KiCad 工程** 子菜单中选择合适的选项。

系统将提示您在导入文件浏览器对话框中选择原理图或电路板文件。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）。一旦选择了请求的文件，系统将要求您选择一个目录来存储生成的 KiCad 工程。

KiCad 配置

KiCad 偏好设置始终可以从 **偏好设置** 菜单访问，也可以使用快捷键（默认为 `Ctrl+,`）访问。偏好设置对话框在运行的 KiCad 工具之间共享。有些偏好设置适用于所有工具，有些则特定于某个工具（如原理图或电路板编辑器）。

共同偏好设置



自动保存： 在编辑原理图和电路板文件时，KiCad 会定期自动保存您的工作。设置为 0 将禁用此功能。

文件历史大小： 配置最近打开的文件列表中的条目数

3D 缓存文件时长： KiCad 创建 3D 模型缓存，以提高 3D 观看速度。您可以配置在删除旧文件之前保留此缓存的时间。

硬件加速抗锯齿： KiCad 在使用显卡渲染时可以使用不同的方法来防止走样（锯齿线）。不同的方法在不同的硬件上可能看起来更好，所以您可能想要尝试一下，找出对您来说最好的一种。

软件加速抗锯齿： KiCad 在使用软件加速模式时也可以应用抗锯齿。启用此功能可能会导致某些硬件的性能下降。

文本编辑器： 选择从工程树状视图打开文本文件时要使用的文本编辑器。

PDF 查看器： 选择打开 PDF 文件时使用的程序

图标比例： 设置 KiCad 中菜单和按钮使用的图标大小。选择 *自动* 根据您的操作系统设置自动选择合适的图标比例。

画布比例： 设置 KiCad 编辑器中使用的画布比例。选择 *自动* 根据您的操作系统设置自动选择适当的画布比例。

在菜单中显示图标： 在整个 KiCad 用户界面中启用下拉菜单中的图标。

NOTE | 菜单中的图标在某些操作系统上不显示。

偏好选择而不是拖动： 启用后，即使开始在对象上拖动，单击并拖动鼠标也会始终执行选择操作。禁用时，在可以移动的对象顶部单击并拖动将执行移动操作。

将鼠标扭曲到移动对象的原点： 启用后，当您在对象上启动移动命令时，鼠标光标将重新定位（扭曲）到该对象的原点。

第一个快捷键选择工具： 禁用时，按下 *添加线* 等命令的快捷键将立即在当前光标位置启动该命令。启用后，第一次按快捷键将仅选择 *添加线* 工具，但不会立即开始一条线。

自动备份工程： 开启后，KiCad 工程将根据以下设置自动存档为 ZIP 文件。存档将保存在工程文件夹的子文件夹中。在工程中保存文件时会创建备份。

自动保存时创建备份： 启用后，每次自动保存文件时都会创建备份（如果下面的设置允许备份）。如果自动保存间隔设置为 0（禁用），则此设置无效。

要保留的最大备份数： 创建新备份时，将删除最旧的备份文件，以将备份文件总数控制在此限制以下。

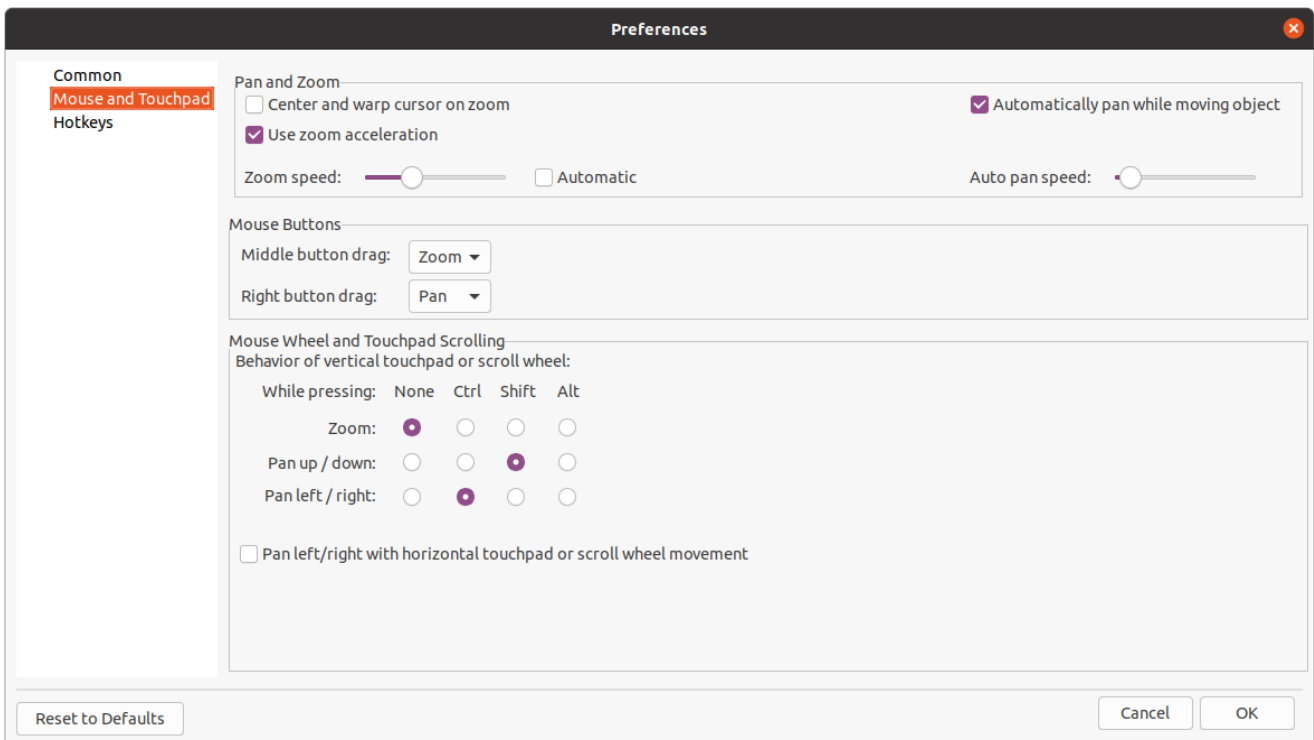
每日最大备份次数： 新建备份时，将删除当天创建的最旧备份文件，使其保持在该限制以下。

最小备份间隔时间： 如果触发备份（例如保存电路板文件），如果已有备份文件超过该限制，则不会创建备份。

最大总备份大小： 创建新的备份文件时，将删除最旧的备份文件，以使备份文件目录的总大小低于该限制。

记住下次启动工程时打开的文件： 勾选后，如果上次关闭工程管理器时原理图和电路板编辑器处于打开状态，KiCad 将重新打开它们。

鼠标和触摸板偏好设置



缩放时居中和扭曲光标： 启用后，使用快捷键或鼠标滚轮进行缩放将导致视图在光标位置居中。

使用缩放加速： 启用后，滚动鼠标滚轮或触摸板的速度越快，缩放的变化也会越快。

缩放速度： 控制鼠标滚轮或触摸板滚动一定量时的缩放变化量。使用 *自动* 根据您的操作系统设置默认值。

移动对象时自动平移： 开启后，可以通过靠近画布边缘的方式在移动对象时平移视图。

自动平移速度： 控制画布移动对象时的平移速度。

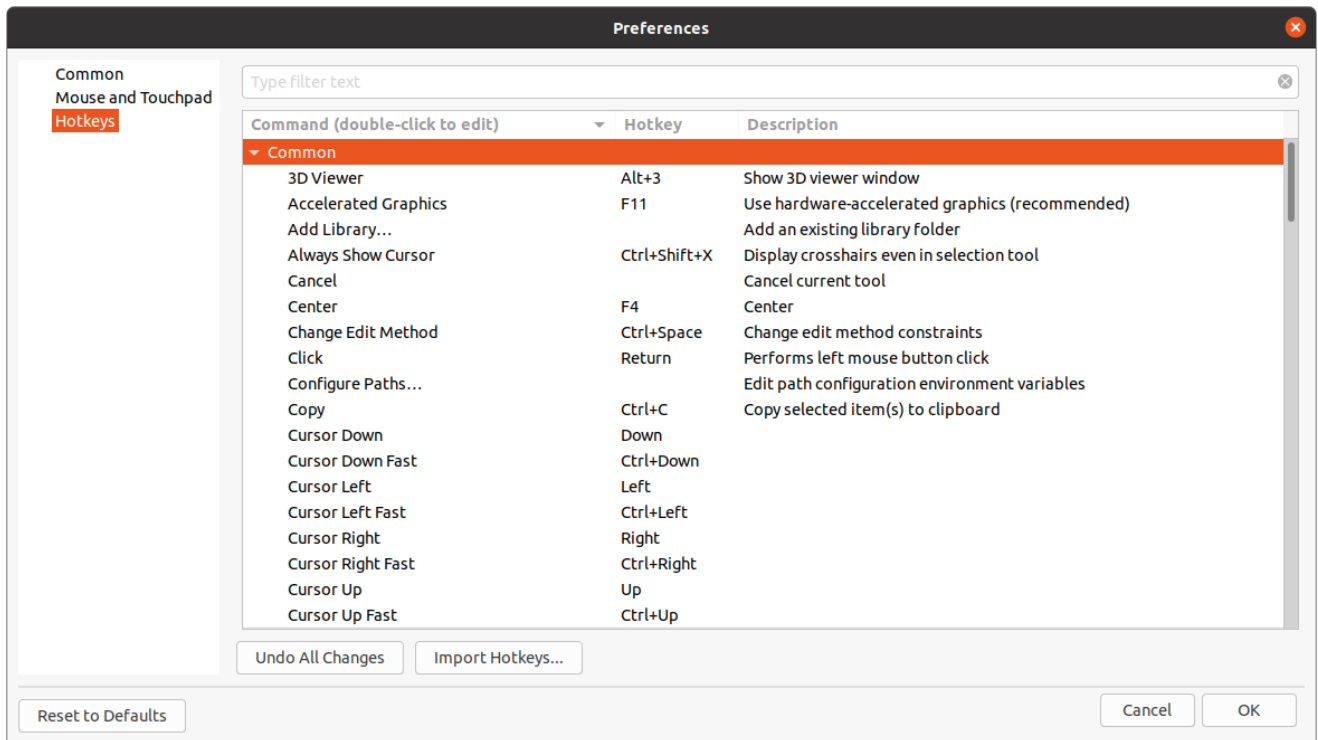
鼠标键： 您可以设置拖动鼠标中键和鼠标右键来缩放视图、平移视图或无效果的行为。

NOTE | 鼠标左键始终用于选择和操作对象。

鼠标滚轮和触摸板滚动： 您可以设置在按特定修改键的同时滚动鼠标滚轮或触摸板垂直运动的行为。

使用水平触摸板或滚轮移动平移左/右： 启用后，您可以使用触摸板或水平滚轮 (如果鼠标上存在) 平移视图。

快捷键偏好设置



可以使用此对话框自定义用于控制 KiCad 的快捷键。公用部分中的快捷键在每个 KiCad 程序之间共享。当程序运行时，会显示每个特定 KiCad 程序的快捷键。您可以将相同的快捷键分配给不同 KiCad 程序 (例如原理图编辑器和电路板编辑器) 中的不同操作，但不能将一个快捷键分配给同一程序中的多个操作。

有许多可用命令，因此并非所有命令都默认分配了快捷键。您可以通过双击列表中的命令将快捷键添加到任何命令。如果选择已分配给其他命令的快捷键，则可以选择在所选命令上使用该快捷键，这将从冲突的命令中删除指定的快捷键。

您对快捷键分配所做的更改会在命令名的末尾显示一个 * 字符。您可以通过右键单击特定命令并选择 **撤消更改** 来撤消对该命令的更改，也可以使用命令列表下面的按钮撤消所有更改。

导入快捷键

快捷键偏好设置存储在 KiCad 设置目录的 `.hotkeys` 文件中 (有关设置目录在操作系统上的位置，请参阅《设置，设置》一节)。如果您在一台计算机上以您喜欢的方式配置了 KiCad 快捷键，则可以通过导入适当的 `.hotkeys` 文件将该配置传输到另一台计算机。

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：

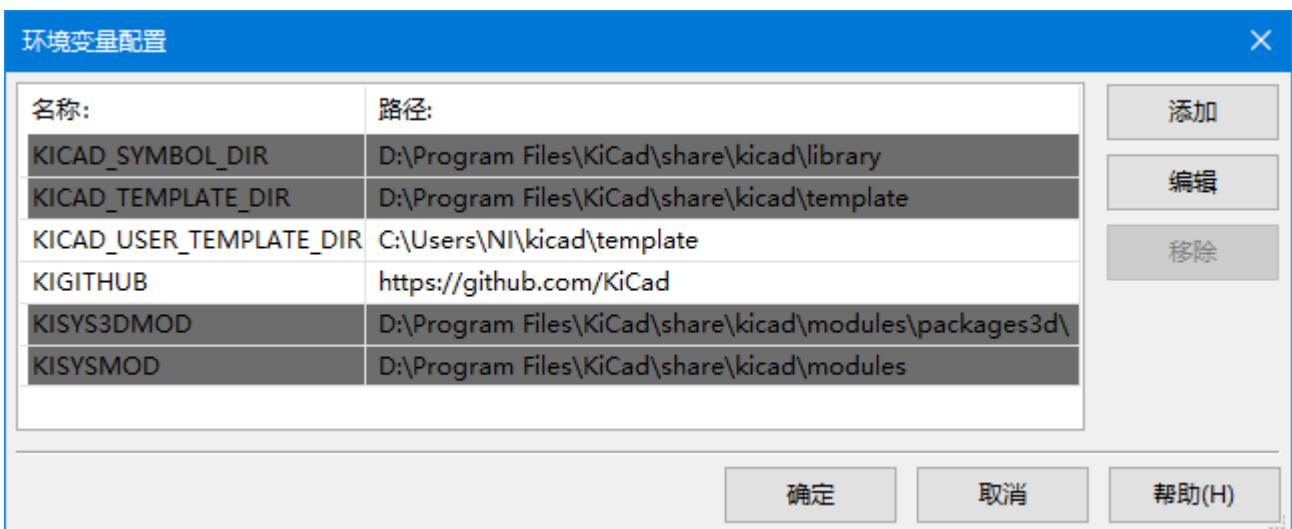
- 原理图符号库
- 封装库
- 封装定义中使用的 3D 形状文件

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`**

在 **偏好设置** → **配置路径...** 菜单中，您可以为一些内置的 KiCad 环境变量定义路径，如果需要，还可以添加自己的环境变量来定义个人路径。

KiCad 环境变量：

KICAD6_SYMBOL_DIR	符号库文件的基本路径。
KICAD6_FOOTPRINT_DIR	封装库文件的基本路径。
KICAD6_3DMODEL_DIR	封装中使用的 3D 模型的基本路径。
KICAD6_TEMPLATE_DIR	随 KiCad 一起安装的工程模板的位置。
KICAD_USER_TEMPLATE_DIR	个人工程模板的位置。
KICAD6_SCRIPTING_DIR	随 KiCad 一起安装的 Python 脚本的位置。
KICAD6_USER_SCRIPTING_DIR	个人 Python 脚本的位置。



可以设置一些高级环境变量来定制 KiCad 的行为。默认情况下，这些变量不会显示在环境变量配置对话框中。更改这些变量不会导致 KiCad 将任何文件从默认位置移动到新位置，因此，如果更改这些变量，则需要手动复制任何所

需的设置或文件。

其他环境变量：

KICAD_CONFIG_HOME	KiCad 配置文件的基本路径。将在此目录中为每个 KiCad 次要版本创建子目录。
KICAD_DOCUMENTS_HOME	KiCad 用户可修改文档的基本路径，如模板、Python 脚本、库等。将在此目录中为每个 KiCad 次要版本创建子目录。

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。

库配置

在 **偏好设置** → **管理符号库...** 菜单中，可以管理名为 **符号库表**(sym-lib-table)的库列表文件。

同样，使用 **偏好设置** → **管理封装库...** 菜单管理称为 **封装库表**(fp-lib-table)的库列表文件。

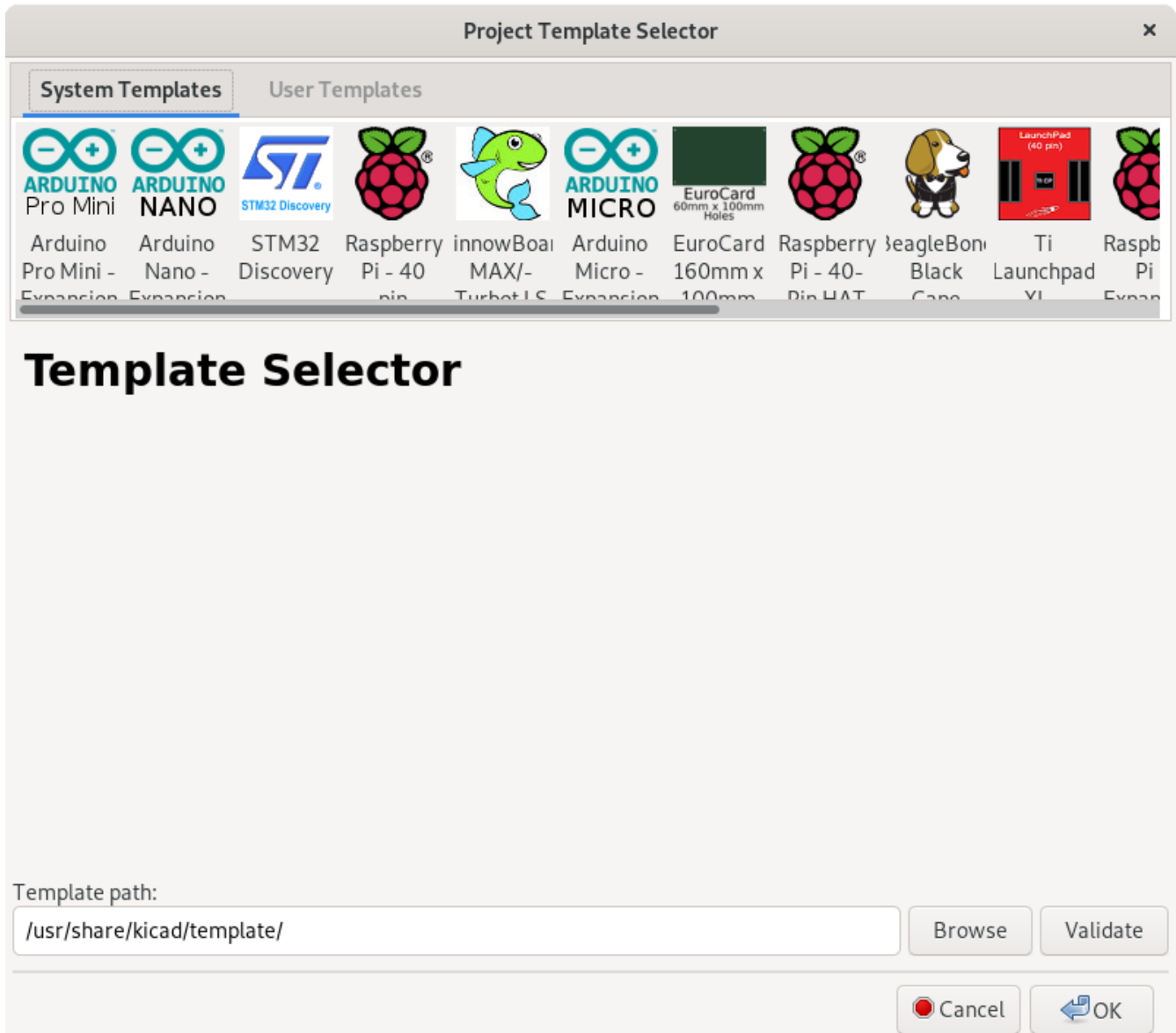
有两个库列表文件：第一个文件(位于用户主目录中)是所有工程的全局文件，第二个文件(位于工程目录中)是可选的，且特定于工程。

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。

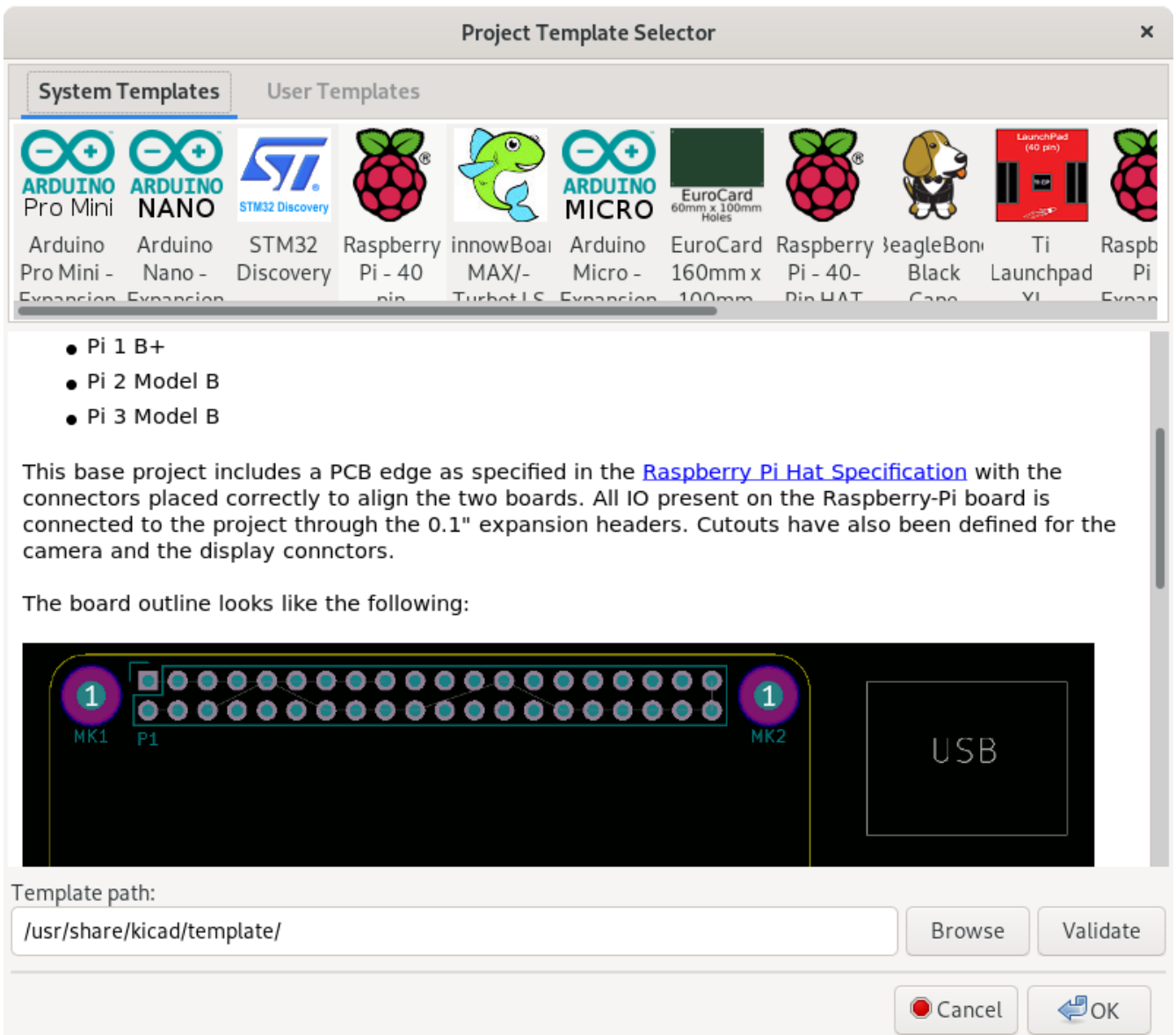
使用模板

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：



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/template/
 - Windows: C:\Documents and Settings\username\My Documents\kicad\template or C:\Users\username\Documents\kicad\template
 - Mac: ~/Documents/kicad/template/

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。

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。

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的工程路径，但 **meta** 除外。包含模板名称的文件和目录将使用新的工程文件名重命名。

例如，从名为 **示例** 的模板创建名为 **新工程** 的工程：

模板 示例 目录中的文件。	在工程 新工程 目录中创建的文件
示例.kicad_pro	新工程.kicad_pro
示例.kicad_sch	新工程.kicad_sch
示例.kicad_pcb	新工程.kicad_pcb
示例-第一次.kicad_sch	新工程-第一次.kicad_sch
第二次-示例.kicad_sch	第二次-新工程.kicad_sch
第三次.kicad_sch	第三次.kicad_sch
第三次.kicad_pcb	第三次.kicad_pcb

模板不需要包含完整的工程，如果缺少所需的工程文件，KiCad 将使用其默认的创建工程行为创建该文件：

模板 示例 目录中的文件。	在 新工程 目录中创建的文件
示例.kicad_sch	新工程.kicad_sch
第一次-示例.kicad_sch	第一次-新工程.kicad_sch
第一次-示例.kicad_pcb	第一次-新工程.kicad_pcb
第二次-示例.kicad_sch	第二次-新工程.kicad_sch
第二次-示例.kicad_pcb	第二次-新工程.kicad_pcb
	新工程.kicad_pro (默认)
	新工程.kicad_pcb (默认)

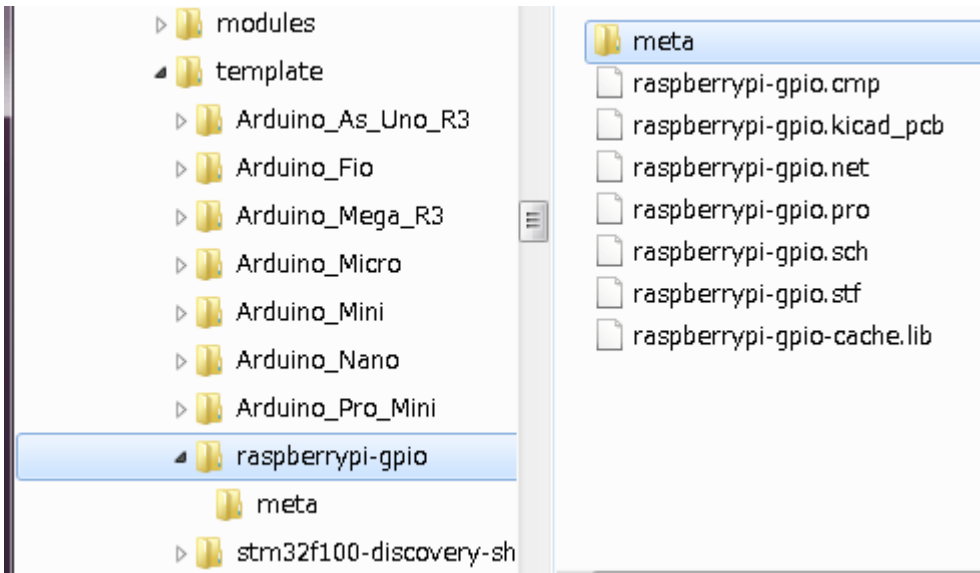
作为模板名称重命名规则的例外，如果存在一个工程文件(.kicad_pro)，并且其名称与模板名称不匹配，则 KiCad 将基于该工程文件名称进行重命名：

模板 示例 目录中的文件。	在 新工程 目录中创建的文件
示例.kicad_sch	示例.kicad_sch
示例.kicad_pcb	示例.kicad_pcb
第一次-示例.kicad_pro	新工程.kicad_pro
第一次-示例.kicad_sch	新工程.kicad_sch
第一次-示例.kicad_pcb	新工程.kicad_pcb
第二次-示例.kicad_sch	第二次-示例.kicad_sch
第二次-示例.kicad_pcb	第二次-示例.kicad_pcb

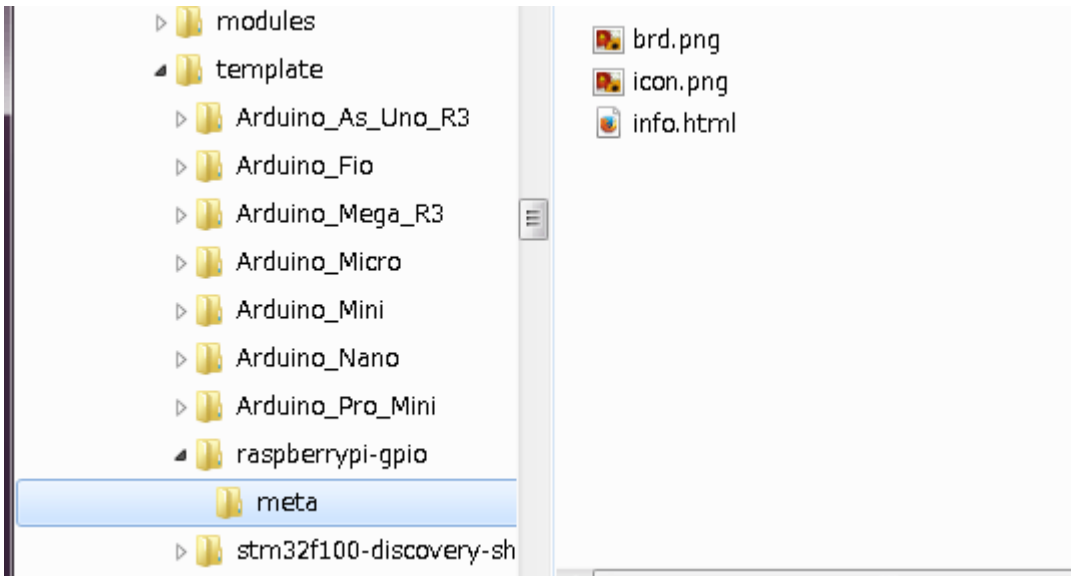
NOTE | 不建议创建包含多个工程文件的模板。

模板示例

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



和元数据文件:



所需文件:

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

<title> 标记确定向用户公开以供选择模板的模板的实际名称。请注意，如果工程模板名称太长，它将被截断。

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>

```

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。