

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
Using the KiCad project manager	7
Standalone mode	8
创建一个新工程	8
从其他 EDA 工具导入工程	9
Saving and loading project archives	9
KiCad 配置	11
共同偏好设置	11
鼠标和触摸板偏好设置	13
快捷键偏好设置	14
路径配置	15
库配置	17
工程模板	18
使用模板	18
模板位置	18
创建模板	19
插件和内容管理器	21
操作参考	22
KiCad 工程管理器	22

参考手册

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

Jean-Pierre Charras, Fabrizio Tappero, Jon Evans.

翻译人员

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

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

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/>

简介

KiCad 是一款用于创建电子电路原理图和印刷电路板 (PCB) 的开源软件套件。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属性的 PCB 计算器、用于检查制造文件的 Gerber 浏览器和用于检查电路行为的集成 SPICE 模拟器。

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。

该文件的最新版本可在 <https://docs.kicad.org>。

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.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。

工程文件

*.kicad_pro	工程文件, 包含原理图和 PCB 之间共享的设置。
*.pro	遗留 (KiCad 5.x 及更早版本) 工程文件。可被读取, 并将由工程管理器转换为 .kicad_pro 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	遗留 (KiCad 5.x 及更早版本) 原理图文件。可以读取, 写入时将转换为 .kicad_sch 文件。
*.lib	遗留 (KiCad 5.x 及更早版本) 原理图库文件。可以读取但不能写入。
*.dcm	遗留 (KiCad 5.x 及更早版本) 原理图库文件。可以读取但不能写入。
*-cache.lib	遗留 (KiCad 5.x 及更早版本) 原理图元件库缓存文件。为正确加载遗留原理图 (.sch) 文件所必需。
sym-lib-table	原理图编辑器中可用的符号库列表。

PCB 编辑器的文件和文件夹

*.kicad_pcb	包含除页面布局外的所有信息的板块文件。
*.pretty	封装库文件夹。该文件夹本身就是库。
*.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 .kicad_pcb 文件的自定义设计规则。
*.brd	遗留 (KiCad 4.x 及更早版本) PCB 文件。可以被当前的 PCB 编辑器读取，但不能写入。
*.mod	遗留 (KiCad 4.x 及更早版本) 封装库文件。可以被封装或 PCB 编辑器读取，但不能写入。
fp-lib-table	封装库表：在 PCB 编辑器中可用的封装库列表。
fp-info-cache	缓存，以加快加载封装库的速度。不需要与工程一起发布或置于版本控制之下。

通用文件

*.kicad_prl	Local settings for the current project; helps KiCad remember the last used settings such as layer visibility or selection filter. May not need to be distributed with the project or put under version control.
*.kicad_wks	Page layout (drawing border and title block) description file
*.net	Netlist file created from the schematic, and read by the board editor. Note that the recommended workflow for transferring information from the schematic to the board does not require the use of netlist files.
*.cmp	Association between components used in the schematic and their footprints. It can be created by the Board Editor and imported by the Schematic Editor. Its purpose is to import changes from the board to the schematic, for users who change footprints in the Board Editor (for instance using Exchange Footprints command) and want to import these changes back to the schematic. Note that the recommended workflow for transferring information from the board to the schematic does not require the use of .cmp files.

制造和文档文件

*.gbr	Gerber 文件，用于制造。
*.drl	钻孔文件（Excellon 格式），用于制造。
*.pos	位置文件（ASCII 格式），用于自动贴片机。
*.rpt	报告文件（ASCII 格式），用于记录。
*.ps	图表文件（Postscript），用于文档。
*.pdf	图表文件(PDF 格式)，用于文档。
*.svg	绘图文件（SVG 格式），用于文档。
*.dxf	图表文件(DXF 格式)，用于文档。
*.plt	绘图文件(HPGL 格式)，用于文档。

储存和发送 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，因此您可以自行备份或发送这些文件，不会有任何问题。一些重要的设计信息存储在工程文件（`.kicad_pro`）中，所以如果你要发送一个完整的设计，请确保包含这些信息。

有些文件，如工程本地设置文件（`.kicad_prl`）和 `fp-info-cache` 文件，没有必要与工程一起发送。如果你使用版本控制系统（如 Git）来跟踪你的 KiCad 工程，你可能想把这些文件添加到被忽略的文件列表中，这样它们就不会被跟踪了。

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。

请注意，KiCad 前一版本的原理图符号和封装库表将 **不会** 被导入。

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 stores the settings files in a folder inside your user directory. Each KiCad version will use a different versioned subfolder. For KiCad 8, those folders are:

Windows	%APPDATA%\kicad\8.0
Linux	~/.config/kicad/8.0
macOS	/Users/<username>/Library/Preferences/kicad/8.0

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。

The schematic editor documentation describes several particular considerations for opening [legacy schematics](#).

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。







Using the KiCad project manager

The KiCad project manager is a tool which creates and opens KiCad projects and launches the other KiCad tools (schematic and board editors, Gerber viewer, and utility tools).

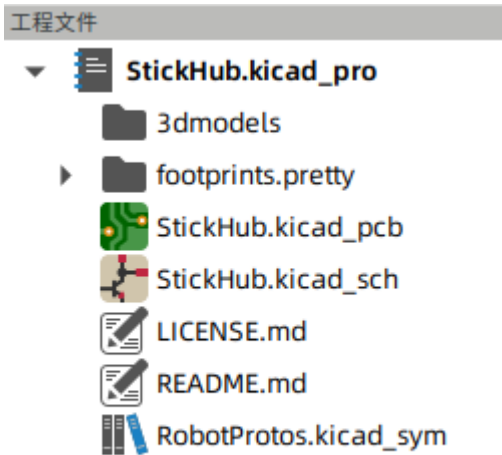


KiCad 工程管理器窗口由左侧的工作区视图和右侧的启动器组成，前者显示与打开的工程相关的文件，后者则包含各种编辑器和工具的快捷方式。

窗口左侧的工具栏为常见的工程操作提供快捷键：

	Create a new project.
	Open an existing project.
	Create a zip archive of the whole project. This includes schematic files, libraries, PCB, etc.
	Extract a project zip archive into a directory. Files in the destination directory will be overwritten.
	Refresh the tree view, to detect changes made on the filesystem.
	Open the project working directory in a file explorer.

工作区视图显示工程文件夹内的文件列表。在工作区视图中双击一个文件将在相关的编辑器中打开它。右键点击一个文件将打开一个包含一些文件操作命令的上下文菜单。



NOTE 只有 KiCad 知道如何打开的文件才会显示在工程工作区视图中。

KiCad projects contain at least a project file, a schematic, and a board design. Schematics may contain multiple sheets, each in its own file, but a project can only contain a single board. KiCad expects the project file, schematic root sheet file, and board file to all have the same name.

Standalone mode

You can also run the KiCad editor tools in *standalone* mode, by launching them directly from your operating system's application launcher rather than from the project manager. It is usually **not recommended** to run the tools in standalone mode, except for some specific situations where it is necessary, such as when importing projects from other EDA tools. When running in standalone mode, some project features are not available, including:

- cross probing between the schematic editor and the board editor
- design synchronization between the schematic and the board

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	PCB 文件。

从其他 EDA 工具导入工程

KiCad is able to import files created by some other software packages. Some software formats can be imported as complete projects. Others can only be imported as standalone schematics or boards at the moment, and must be manually linked together into a KiCad project. Currently the following types of project are supported:

.sch, .brd	Eagle 6.x 或更新版本(XML 格式)
.csa, .cpa	CADSTAR 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。

Currently the following types of documents can be imported standalone. To import these documents, start the KiCad Schematic Editor or PCB Editor standalone (do not open the KiCad project manager first) and choose **File > Import > Non-KiCad Schematic** or **File > Import > Non-KiCad Board File**. When importing Altium projects, we recommend importing the PCB first, saving the resulting project, and then copying each schematic sheet into the project after importing it in a standalone Schematic Editor window.

*.SchDoc	Altium Designer, Circuit Studio, Circuit Maker schematic documents
*.PcbDoc	Altium Designer PCB
*.CMPcbDoc	Altium Circuit Maker PCB
*.CSPcbDoc	Altium Circuit Studio PCB
*.pcb	P-Cad 200x ASCII PCB
*.txt, *.fab	Fabmaster PCB

NOTE

KiCad does not support schematics with multiple top-level sheets. When importing designs from other tools that do support this feature, each schematic sheet must be imported, and then the imported sheets must be placed as hierarchical sheets in a new KiCad project.

Saving and loading project archives

You can archive your project's files into a zip archive with the Archive tool (**File** → **Archive Project...**).

You can also unarchive a project using the Unarchive tool (**File** → **Unarchive Project...**). When you unarchive a project into the currently loaded project directory, the project will be reloaded automatically to reflect any changes that were in the archived version of the project.

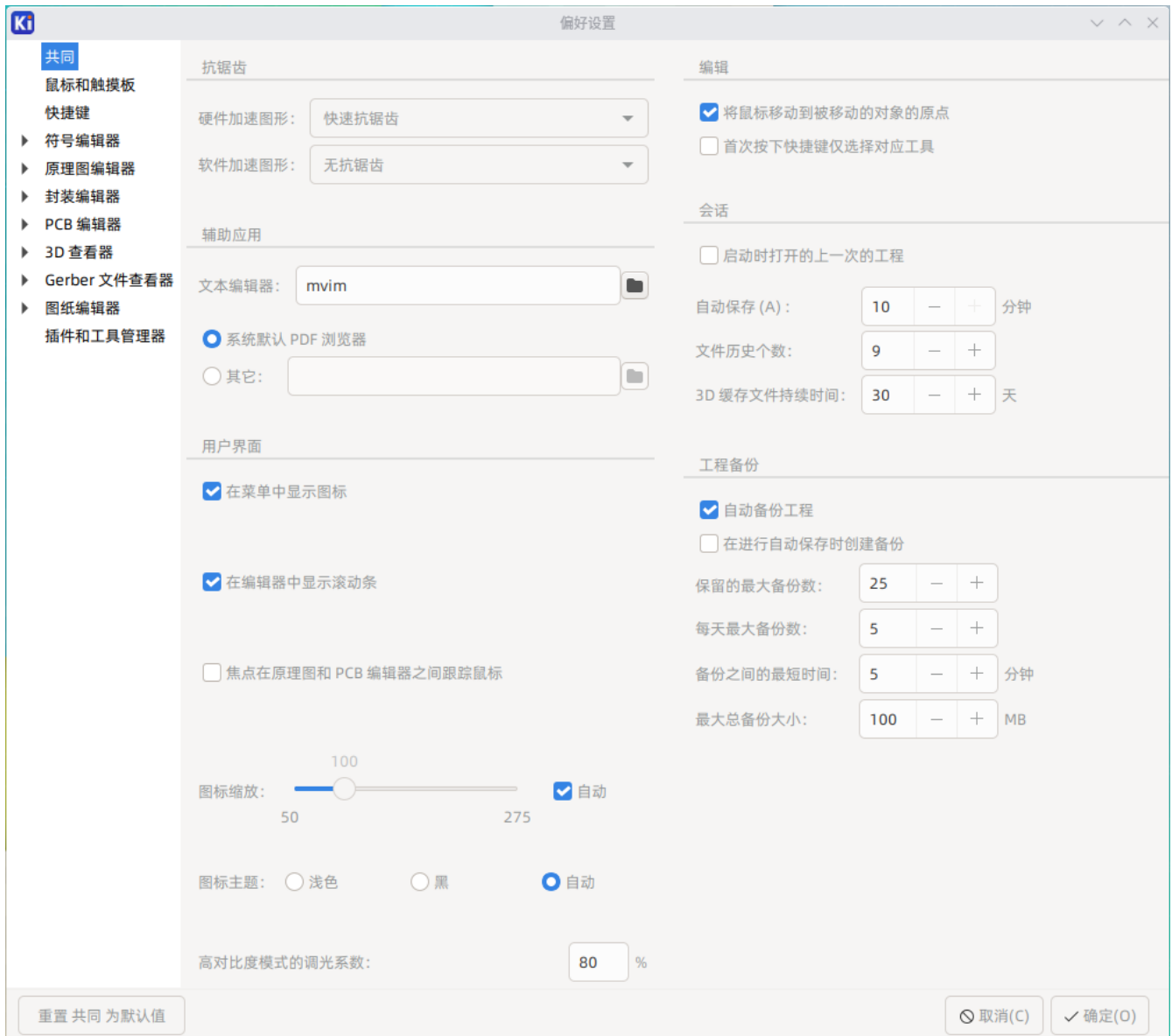
The archive tool saves the following files from your project folder into the archive:

<code>*.kicad_prl, *.kicad_pro, *.kicad_sch, *.kicad_sym, *.kicad_pcb, *.kicad_mod, *.kicad_dru, *.kicad_wks, *.wbk, *.fp-lib-table, *.sym-lib-table</code>	KiCad design files
<code>*.pro, *.sch, *.lib, *.dcm, *.cmp, *.brd, *.mod</code>	Legacy KiCad design files
<code>*.stp, *.step</code>	3D models
<code>*.g?, *.g??, *.gm??, *.gbrjob</code>	Gerber files
<code>*.pos, *.drl, *.nc, *.xnc, *.d356, *.rpt</code>	Manufacturing files
<code>*.net</code>	Netlists
<code>*.py</code>	Python scripts
<code>*.pdf, *.txt</code>	Documentation files
<code>*.cir, *.sub, *.model</code>	SPICE models
<code>*.ibs, *.pkg</code>	IBIS models

KiCad 配置

The KiCad preferences can always be accessed from the **Preferences** menu, or by using the hotkey (default **Ctrl + P**). The Preferences dialog is shared between the running KiCad tools. Some preferences apply to all tools, and some are specific to a certain tool (such as the schematic or board editor).

共同偏好设置



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： 选择从工程工作区视图打开文本文件时要使用的文本编辑器。

PDF 阅读器： 选择一个在打开 PDF 文件时使用的程序。

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。

在编辑器中显示滚动条：当启用时，滚动条会显示在每个工具的编辑画布旁边。当禁用时，不显示滚动条。

在原理图和 PCB 编辑器之间焦点跟随鼠标：启用后，鼠标指针下的窗口将自动成为焦点。

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。

图标主题：设置是为浅色窗口背景设计的图标主题，还是为深色窗口背景设计的图标主题。自动的默认设置将根据操作系统窗口主题的亮度选择主题。

高对比度模式的调光系数：设定在高对比度显示模式下非焦点项目的调光程度。

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。

记住下次启动项目时打开的文件：启用后，当重新打开工程时，KiCad 会自动重新打开之前打开的所有文件。

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。

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。

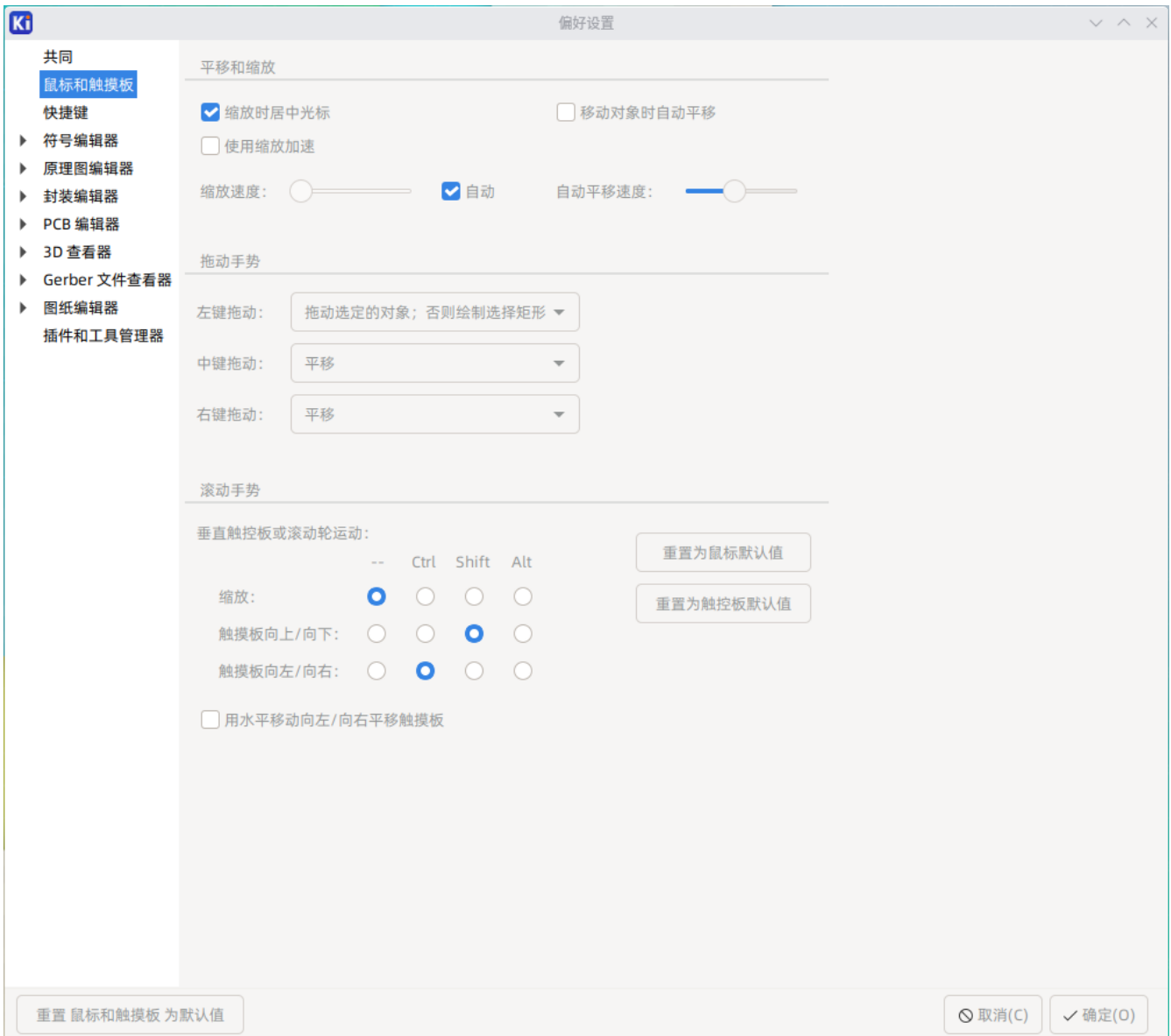
要保留的最大备份数：创建新备份时，将删除最旧的备份文件，以将备份文件总数控制在此限制以下。

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。

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： 启用后，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。

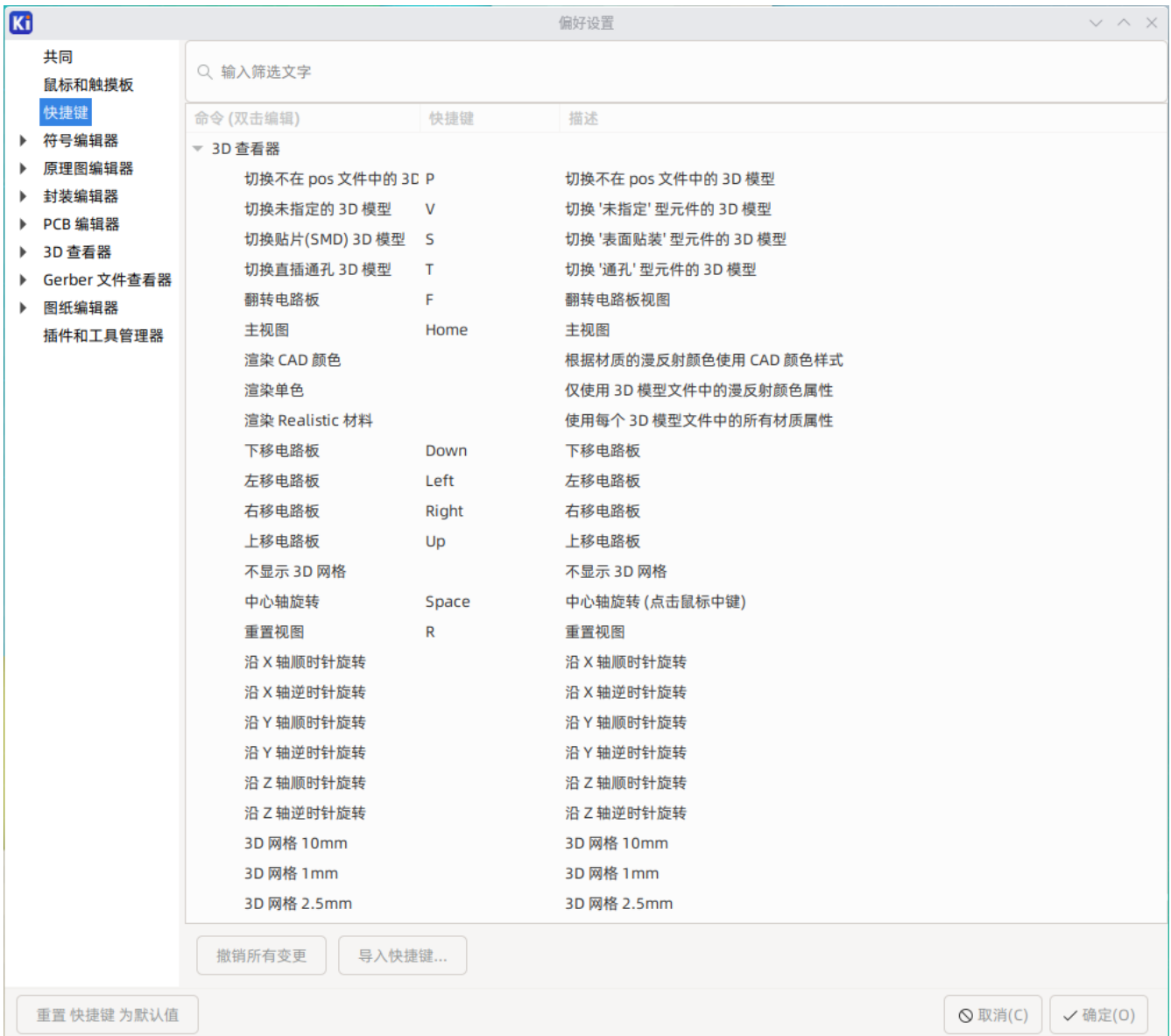
鼠标键： 您可以设置拖动鼠标中键和鼠标右键来缩放视图、平移视图或无效果的行为。您还可以根据编辑画布中是否已选择任何对象来设置拖动鼠标左键的行为。

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。

触摸板水平移动左右平移： 启用后，您可以使用触摸板或水平滚轮 (如果鼠标上有) 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。

你对快捷键分配所做的改变会在命令名称的末尾显示一个 * 字符。你可以通过右键单击某个特定的命令并选择 **撤销更改** 来撤销对该命令的更改，或者你可以通过命令列表下面的按钮撤销所有的更改。

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

In KiCad, one can define paths using a **path variable**. A few path variables are internally defined by KiCad, and can be used to define paths for libraries, 3D shapes, etc.

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 模型文件

For instance, the path to the `connect.pretty` footprint library, when using the `KICAD8_FOOTPRINT_DIR` path variable, would be defined as `${KICAD8_FOOTPRINT_DIR}/connect.pretty`.

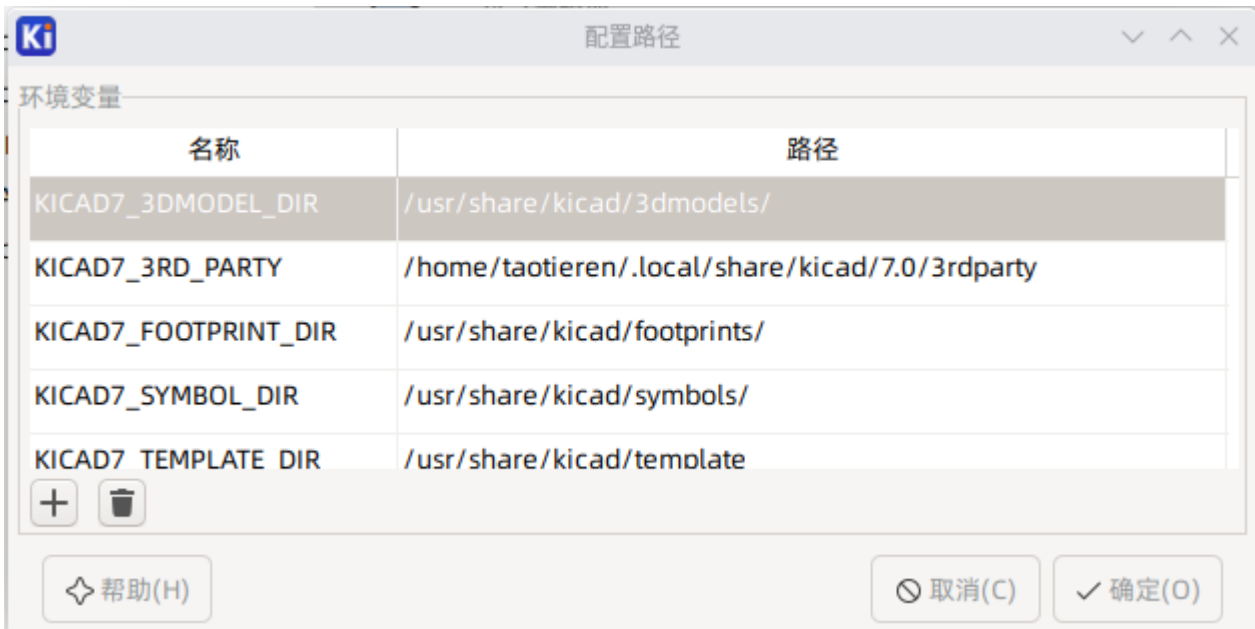
The **Preferences** → **Configure Paths...** menu allows you to define paths for some built-in KiCad path variables, and add your own path variables to define personal paths, if needed.

NOTE

KiCad will automatically resolve versioned path variables from older versions of KiCad to the value of the corresponding variable from the current KiCad version, as long as the old variable is not explicitly defined itself. For example, `${KICAD7_FOOTPRINT_DIR}` will automatically resolve to the value of `${KICAD8_FOOTPRINT_DIR}` if there is no `KICAD7_FOOTPRINT_DIR` variable defined.

KiCad path variables

<code>KICAD8_3DMODEL_DIR</code>	Base path of KiCad's standard 3D footprint model library files.
<code>KICAD8_3RD_PARTY</code>	Location for plugins, libraries, and color themes installed by the Plugin and Content Manager .
<code>KICAD8_FOOTPRINT_DIR</code>	Base path of KiCad's standard footprint library files.
<code>KICAD8_SYMBOL_DIR</code>	Base path of KiCad's standard symbol library files.
<code>KICAD8_TEMPLATE_DIR</code>	Location of KiCad's standard project template library files.
<code>KICAD_USER_TEMPLATE_DIR</code>	Location of personal project templates.
<code>SPICE_LIB_DIR</code>	Location of personal simulation model libraries . This variable is not defined by default.
<code>KIPRJMOD</code>	Absolute path to the current project directory. This variable is set automatically and cannot be redefined.



在配置路径对话框中设置的路径是 KiCad 内部的，在 KiCad 之外不可见环境变量。它们被保存在《配置文件位置，KiCad 的用户配置文件》。

Paths can also be set as system environment variables outside of KiCad, which will override any settings in the user's configuration.

NOTE You cannot override a system environment variable that has been set outside of KiCad by using the Configure Paths dialog. Any variable that has been set externally will be shown as read-only in the dialog.

Note also that the path variable `KIPRJMOD` is **always** internally defined by KiCad, and expands to the **current project absolute path**. For instance, `${KIPRJMOD}/connect.pretty` is always the `connect.pretty` folder (the footprint library) inside **the current project folder**. The `KIPRJMOD` variable cannot be changed in the Configure Paths dialog or overridden by an external environment variable.

高级环境变量

Some advanced environment variables can be set to customize where KiCad expects certain files to be located. By default, these variables are set based on your platform, but they can be overridden by system environment variables. These variables are not shown in the Configure Paths dialog and cannot be used in path substitutions.

改变这些变量不会导致 KiCad 将任何文件从默认位置移动到新的位置，所以如果你改变这些变量，你将需要手动复制任何需要的设置或文件。

KICAD_CONFIG_HOME	Base path of KiCad configuration files. Subdirectories will be created within this directory for each KiCad minor version.
KICAD_DOCUMENTS_HOME	Base path of KiCad user-modifiable documents, such as projects, templates, Python scripts, libraries, etc. Subdirectories will be created within this directory for each KiCad minor version. This directory is provided as a suggested user data location, but does not need to be used.

WARNING

如果您修改了路径的配置，请退出并重新启动 KiCad，以避免在路径处理方面出现任何问题。

库配置

The **Preferences** → **Manage Symbol Libraries...** menu lets you manage the list of symbol libraries ([symbol library table](#)).

Likewise, use the **Preferences** → **Manage Footprint Libraries...** menu to manage the list of footprint libraries ([footprint library table](#)).

对于每一种类型的库（符号和封装），都有2个库表：全局和工程专用。全局库表位于《配置文件位置，用户配置目录》中，包含所有工程可用的库列表。工程专用库表是可选的，包含工程专用的库列表。它位于工程目录中。

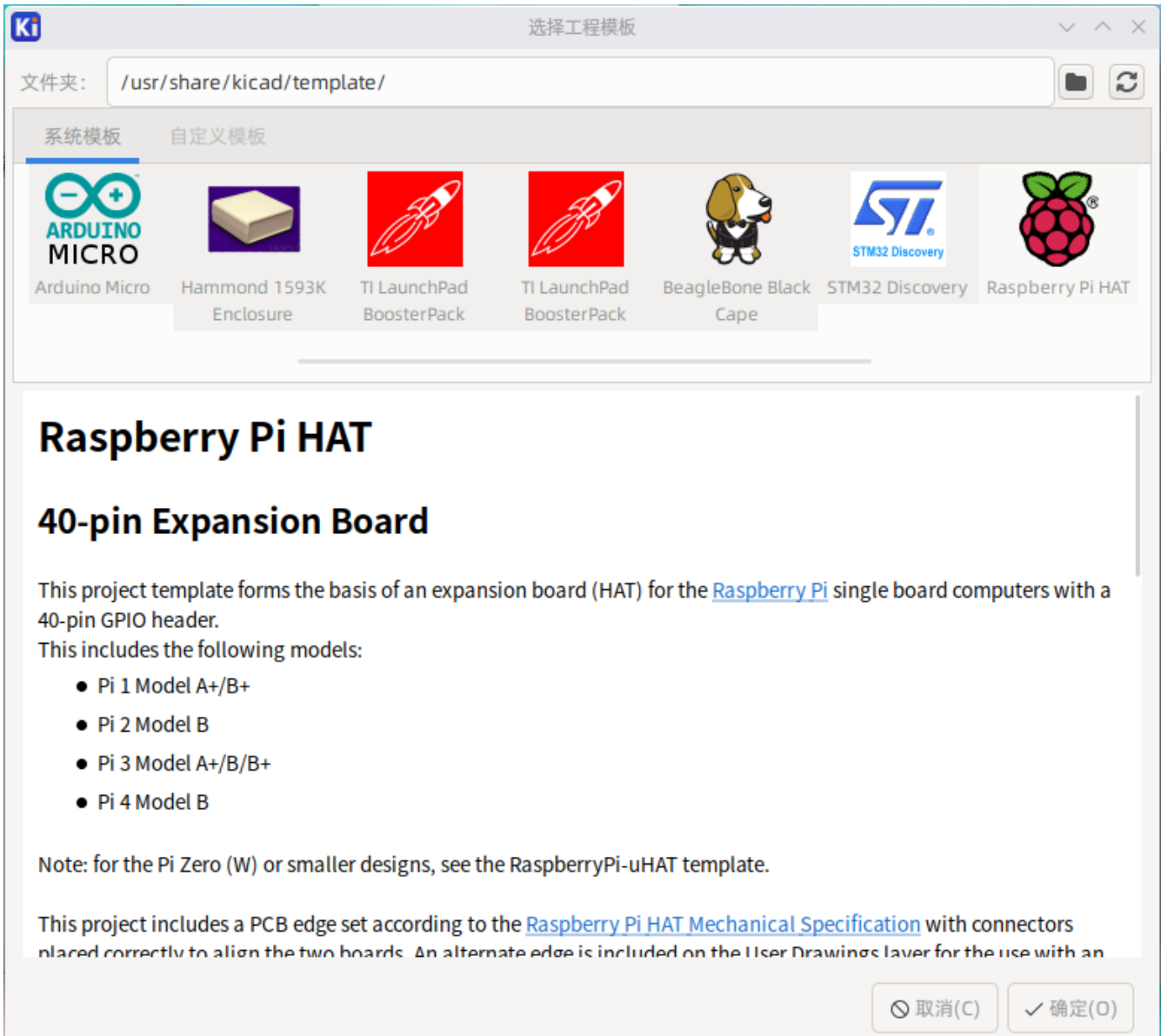
工程模板

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 → 从模板新建工程 菜单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。



模板位置

KiCad looks for system templates in the path defined in the `KICAD8_TEMPLATE_DIR` path variable, and user templates in the path defined in `KICAD_USER_TEMPLATE_DIR`. However, you can browse for templates in an arbitrary directory using the **Folder** control at the top of the dialog.

创建模板

A KiCad template is simply a directory containing the template project files, as well as some required metadata for the template in a subdirectory named `meta`. The name of the directory containing the template files determines the name of the template. When you create a project from a template, KiCad copies the template files to the new project directory, renaming them to match the new project name as described below.

All files in the template are copied, with two exceptions:

- Files with names beginning with the `.` character (dotfiles) are not copied. There is a special case for files named `.gitignore` or `.gitattributes`, which are copied if they exist.
- The `meta` directory is not copied

The `meta` directory must contain an HTML file named `info.html`, which is displayed in the KiCad template browser and should contain basic information describing the template. Basic HTML features are supported, including images. Any images referenced by `info.html` should also be stored in the `meta` directory.

The `<title>` tag determines the name of the template that is displayed during template selection. Note that the project template name will be cut off if it's too long. This display name does not need to be the same as the template directory name.

这里有一个 `info.html` 文件的例子：

```
<!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 - Expansion Board</TITLE>
</HEAD>
<BODY LANG="fr-FR" DIR="LTR">
<P>This project template is the basis of an expansion board for the
<A HREF="http://www.raspberrypi.org/" TARGET="blank">Raspberry Pi $25
ARM board.</A> <BR><BR>This base project includes a PCB edge defined
as the same size as the Raspberry-Pi PCB with the connectors placed
correctly to align the two boards. All IO present on the Raspberry-Pi
board is connected to the project through the 0.1" expansion
headers. <BR><BR>The board outline looks like the following:
</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>
```

Finally, `meta` can optionally contain an image named `icon.png`, which will be used as the template's icon in the template selection dialog. The icon should be a 64 x 64 pixel PNG image.

Template file renaming

All files and directories in a template are copied to the new project path when a project is created using a template, except `meta` and any dotfiles. Files and directories containing the template directory name will be renamed with the new project file name.

For example, using a template named `example` (left) to create a project named `newproject` (right), with renamed files shown in **bold**:

模板 <code>example</code> 目录下的文件	在工程 <code>newproject</code> 目录下创建的文件
<code>example.kicad_pro</code>	<code>newproject.kicad_pro</code>
<code>example.kicad_sch</code>	<code>newproject.kicad_sch</code>
<code>example.kicad_pcb</code>	<code>newproject.kicad_pcb</code>
<code>example-first.kicad_sch</code>	<code>newproject-first.kicad_sch</code>
<code>second-example.kicad_sch</code>	<code>second-newproject.kicad_sch</code>
<code>third.kicad_sch</code>	<code>third.kicad_sch</code>
<code>third.kicad_pcb</code>	<code>third.kicad_pcb</code>

A template does not need to contain a complete project. If a required project file is missing, KiCad will create the file using the default create project behavior:

模板 <code>example</code> 目录下的文件	在 <code>newproject</code> 目录下创建的文件
<code>example.kicad_sch</code>	<code>newproject.kicad_sch</code>
<code>first-example.kicad_sch</code>	<code>first-newproject.kicad_sch</code>
<code>first-example.kicad_pcb</code>	<code>first-newproject.kicad_pcb</code>
<code>second-example.kicad_sch</code>	<code>second-newproject.kicad_sch</code>
<code>second-example.kicad_pcb</code>	<code>second-newproject.kicad_pcb</code>
	<code>newproject.kicad_pro</code> (default)
	<code>newproject.kicad_pcb</code> (default)

As an exception to the template name renaming rule, if the template contains one project file (`.kicad_pro`) and its name doesn't match the template name, KiCad will do the renaming based on that project file name instead:

模板 <code>example</code> 目录下的文件	在 <code>newproject</code> 目录下创建的文件
<code>example.kicad_sch</code>	<code>example.kicad_sch</code>
<code>example.kicad_pcb</code>	<code>example.kicad_pcb</code>
<code>first-example.kicad_pro</code>	<code>newproject.kicad_pro</code>
<code>first-example.kicad_sch</code>	<code>newproject.kicad_sch</code>
<code>first-example.kicad_pcb</code>	<code>newproject.kicad_pcb</code>
<code>second-example.kicad_sch</code>	<code>second-example.kicad_sch</code>
<code>second-example.kicad_pcb</code>	<code>second-example.kicad_pcb</code>

NOTE

不建议创建包含多个工程文件的模板。

插件和内容管理器

NOTE

This section of the KiCad documentation has not yet been written. We appreciate your patience as our small team of volunteer documentation writers work to update and expand the documentation.

操作参考

以下是 KiCad 工程管理器中每个可用的 **操作** 列表：可以分配给快捷键的命令。

KiCad 工程管理器

以下操作在 KiCad 工程管理器中是可用的。热键可以分配给偏好设置中的 **快捷键** 部分的任何动作。

Action	Default Hotkey	Description
Close Project		Close the current project
Image Converter	Ctrl + B	Convert bitmap images to schematic or PCB components
Drawing Sheet Editor	Ctrl + Y	Edit drawing sheet borders and title block
Footprint Editor	Ctrl + F	Edit PCB footprints
PCB Editor	Ctrl + P	Edit PCB
Schematic Editor	Ctrl + E	Edit schematic
Symbol Editor	Ctrl + L	Edit schematic symbols
Clone Project from Repository...		Clone a project from an existing repository
New Project from Template...	Ctrl + T	Create new project from template
New Project...	Ctrl + N	Create new blank project
Open Demo Project...		Open a demo project
Open Project...	Ctrl + O	Open an existing project
Open Text Editor		Launch preferred text editor
Plugin and Content Manager	Ctrl + M	Run Plugin and Content Manager
Calculator Tools		Run component calculations, track width calculations, etc.
Gerber Viewer	Ctrl + G	Preview Gerber output files