KiCad

The KiCad Team

Table of Contents

参考手册

Contributors

Jean-Pierre Charras, Fabrizio Tappero, Jon Evans, Graham Keeth.

翻译人员

taotieren <admin@taotieren.com>, 2019-2024.

Feedback

KiCad 项目欢迎与软件或其文档相关的反馈、错误报告和建议。有关如何提交反馈或报告问题的详细信息，请参阅 https://www.kicad.org/help/report-an-issue/ 上的说明

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。

Many of these files include important design information, especially the project file (`.kicad_pro`), the schematic file(s) (`.kicad_sch`), and the board file (`.kicad_pcb`). Other files may also be necessary. Such files should always be included when distributing the project. Some files are not necessary to distribute with the project, such as the project-local settings file (`.kicad_prl`) or the `fp-info-cache` file. Files that are unnecessary to distribute are noted in the table below.

工程文件

`*.kicad_pro`	Project file, containing settings that are shared between the schematic and PCB
`*.pro`	Legacy (KiCad 5.x and earlier) project file. Can be read and will be converted to a `.kicad_pro` file by the project manager.

原理图编辑器文件

`*.kicad_sch`	Schematic files, containing all symbol and connection information.
`*.kicad_sym`	Schematic symbol library file, containing the symbol descriptions: graphic shape, pins, fields.
`*.kicad_blocks`	Schematic design block library folders. The folder itself is the library.
`*.kicad_block`	Schematic design block folder for defining a reusable schematic design. The folder is the design block, and contains a `.kicad_sch` file defining the design block's schematic and a `.json` file defining the design block's metadata.
`*.wbk`	Simulator workbook file containing SPICE simulation setup information.
`*.sch`	Legacy (KiCad 5.x and earlier) schematic file. Can be read and will be converted to a `.kicad_sch` file on write.
`*.lib`	Legacy (KiCad 5.x and earlier) schematic library file. Can be read but not written.
`*.dcm`	Legacy (KiCad 5.x and earlier) schematic library documentation. Can be read but not written.
`*-cache.lib`	Legacy (KiCad 5.x and earlier) schematic component library cache file. Required for proper loading of a legacy schematic (`.sch`) file.
`sym-lib-table`	Symbol library table: list of symbol libraries available in the schematic editor.
`design-block-lib-table`	Design block library table: list of design block libraries available in the schematic editor.

PCB 编辑器的文件和文件夹

`*.kicad_pcb`	Board file containing all info but the page layout.
`*.pretty`	Footprint library folders. The folder itself is the library.
`*.kicad_mod`	Footprint files, containing one footprint description each.
`*.kicad_dru`	Design rules file, containing custom design rules for a certain `.kicad_pcb` file.
`*.brd`	Legacy (KiCad 4.x and earlier) board file. Can be read, but not written, by the current board editor.
`*.mod`	Legacy (KiCad 4.x and earlier) footprint library file. Can be read by the footprint or the board editor, but not written.
`fp-lib-table`	Footprint library table: list of footprint libraries available in the board editor.
`fp-info-cache`	Cache to speed up loading of footprint libraries. Does not need to be distributed with the project or put under version control.

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. Does not need to be distributed with the project or put under version control.
`*.kicad_wks`	Page layout (drawing border and title block) description file.
`*.kicad_jobset`	Jobset definition file containing output jobsets.
`*.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。

Some files, such as the project-local settings file (`.kicad_prl`) and the `fp-info-cache` file, are not necessary to send with your project. If you use a version control system such as Git to keep track of your KiCad projects, you can add these files to the list of ignored files so that they are not tracked.

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 9, those folders are:

Windows	`%APPDATA%\kicad\9.0`
Linux	`~/.config/kicad/9.0`
macOS	`/Users/<username>/Library/Preferences/kicad/9.0`

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。

原理图编辑器文档介绍了打开 旧版原理图 时的几个特别注意事项。

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 工程管理器是一种创建和打开 KiCad 工程并启动其他 KiCad 工具（原理图和电路板编辑器、Gerber 查看器和实用工具）的工具。

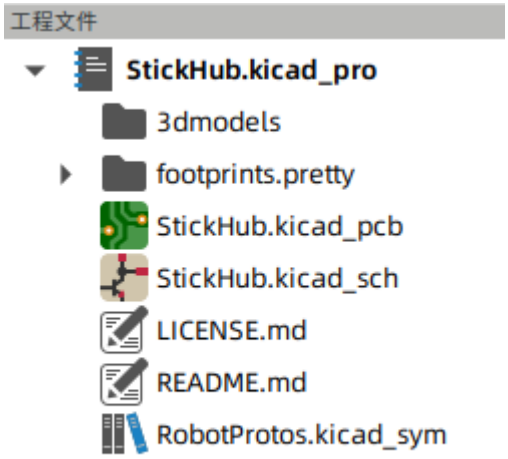
KiCad 工程管理器窗口由左侧的工作区视图和右侧的启动器组成，前者显示与打开的工程相关的文件，后者则包含各种编辑器和工具的快捷方式。

窗口左侧的工具栏为常见的工程操作提供快捷键：

	创建一个新的工程。
	打开一个已存在的工程。
	将整个工程打包成 ZIP 存档，包括原理图文件、库、PCB 等所有相关文件。
	解压工程 ZIP 存档至指定目录。目标目录中的文件将被覆盖。
	刷新树状视图，以检测文件系统上所做的更改。
	在文件浏览器中打开工程的工作目录。

The tree view shows a list of files inside the project folder. Double-clicking on a file in the tree view will open it in the associated editor. Right-clicking on a file will open a context menu with some file manipulation

commands. If the project is part of a Git repository, the tree shows icons indicating the version control status of each file and lists the active branch next to the project name.



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 工程至少包含一个工程文件、一个原理图和一个电路板设计。原理图可以包含多个图纸，每个图纸都有自己的文件，但一个项目只能包含一块电路板。KiCad 希望工程文件、原理图根页文件和电路板文件都具有相同的名称。

单机模式

您也可以在 *单机模式* 模式下运行 KiCad 编辑器工具，方法是直接从操作系统的应用程序启动器而不是从工程管理器启动这些工具。通常 **不建议** 在单机模式下运行工具，除非在某些特殊情况下必须这样做，例如从其他 EDA 工具导入工程时。 在单机模式下运行时，某些工程功能不可用，包括：

- 原理图编辑器和电路板编辑器之间的交叉探测
- 原理图和电路板之间的设计同步

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 能够导入某些其他软件包创建的文件。 有些软件格式可以作为完整工程导入。 其他格式目前只能作为独立的原理图或电路板导入，必须手动连接到 KiCad 工程中。目前支持以下类型的工程：

`*.sch`, `*.brd`	Eagle 6.x 或更新版本(XML 格式)
`*.csa`, `*.cpa`	CADSTAR 压缩格式
`\*.zip`	EasyEDA (JLCEDA) 标准版备份
`*.epro`, `*.zip`	EasyEDA (JLCEDA) 专业版工程

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。

目前可在单机模式导入以下类型的文档。 要导入这些文件，请单独启动 KiCad 原理图编辑器或 PCB 编辑器（不要先打开 KiCad 工程管理器），然后选择 **文件** > **导入** > **非 KiCad 原理图** 或 **文件** > **导入** > **非 KiCad 电路板文件**。 导入 Altium 工程时，我们建议先导入 PCB，保存生成的工程，然后在单机模式的原理图编辑器窗口中导入工程后，将每个原理图页复制到工程中。

`*.SchDoc`	Altium Designer, Circuit Studio, Circuit Maker 原理图文档
`*.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 不支持具有多个顶层工作表的原理图。 从支持该功能的其他工具导入设计时，必须导入每个原理图工作表，然后将导入的工作表作为分层工作表放置在新的 KiCad 工程中。

保存和加载工程档案

您可以使用存档工具将工程的文件存档到 zip 存档中（**文件** → **存档工程...**）。

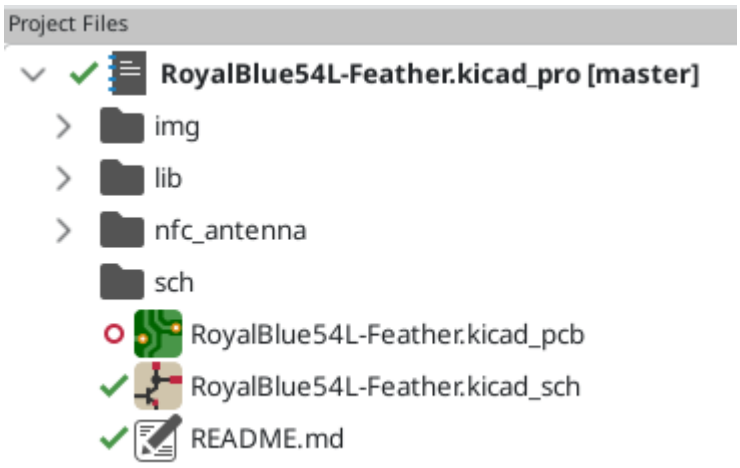
您也可以使用解压缩工具（**文件** → **解压缩项目...**）来解压缩工程。将项目解压缩到当前加载的工程目录中时，工程将自动重新加载，以反映归档版本中的任何更改。

存档工具会将工程文件夹中的以下文件保存到存档中：

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

Git integration

The KiCad Project Manager integrates with the Git version control tool for tracking changes in your projects. It can work with an existing local Git repository, clone a project from a remote repository, or create a new repository in an existing project. You can use the tool to commit changes from your project, push and pull from a remote repository, and switch branches.

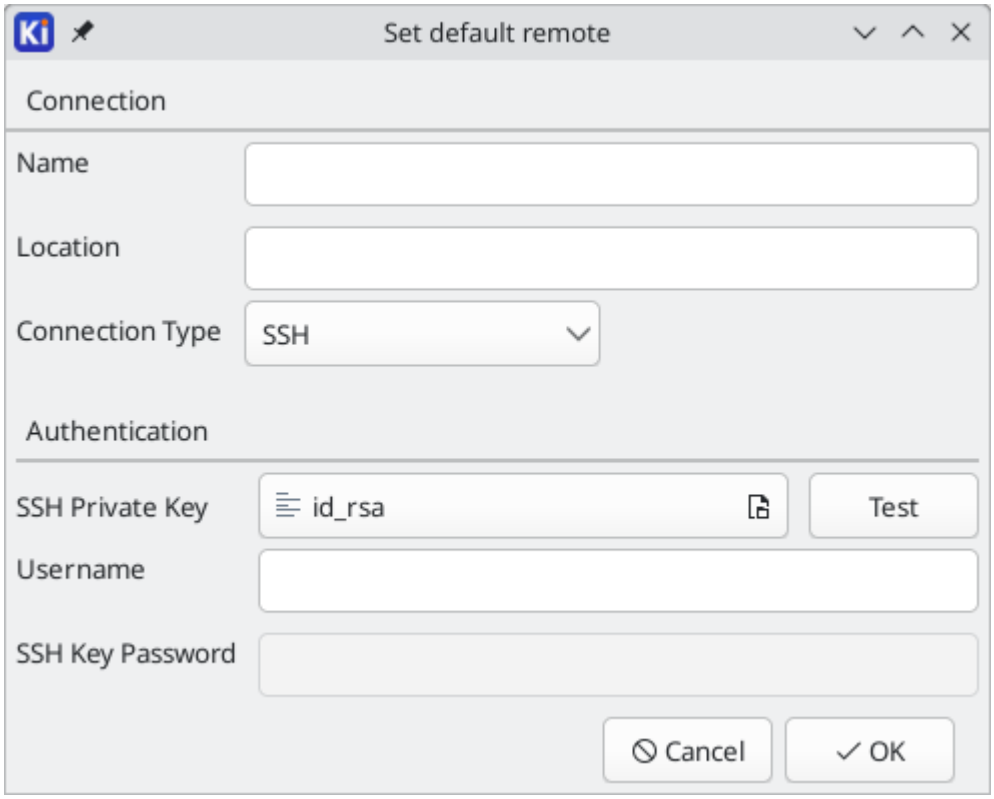


If you open a project that is already under version control with Git, i.e. it is part of an existing Git repository, you can use KiCad's version control features to track changes in the project without any additional

configuration. The active branch is displayed next to the project name, and the version control status of each file in your project is shown graphically in the project files tree. For example, the ✓ icon indicates a file is unchanged, ○ indicates a file has uncommitted changes, and + indicates a file is not tracked. No icons are shown if the project is not part of a Git repository.

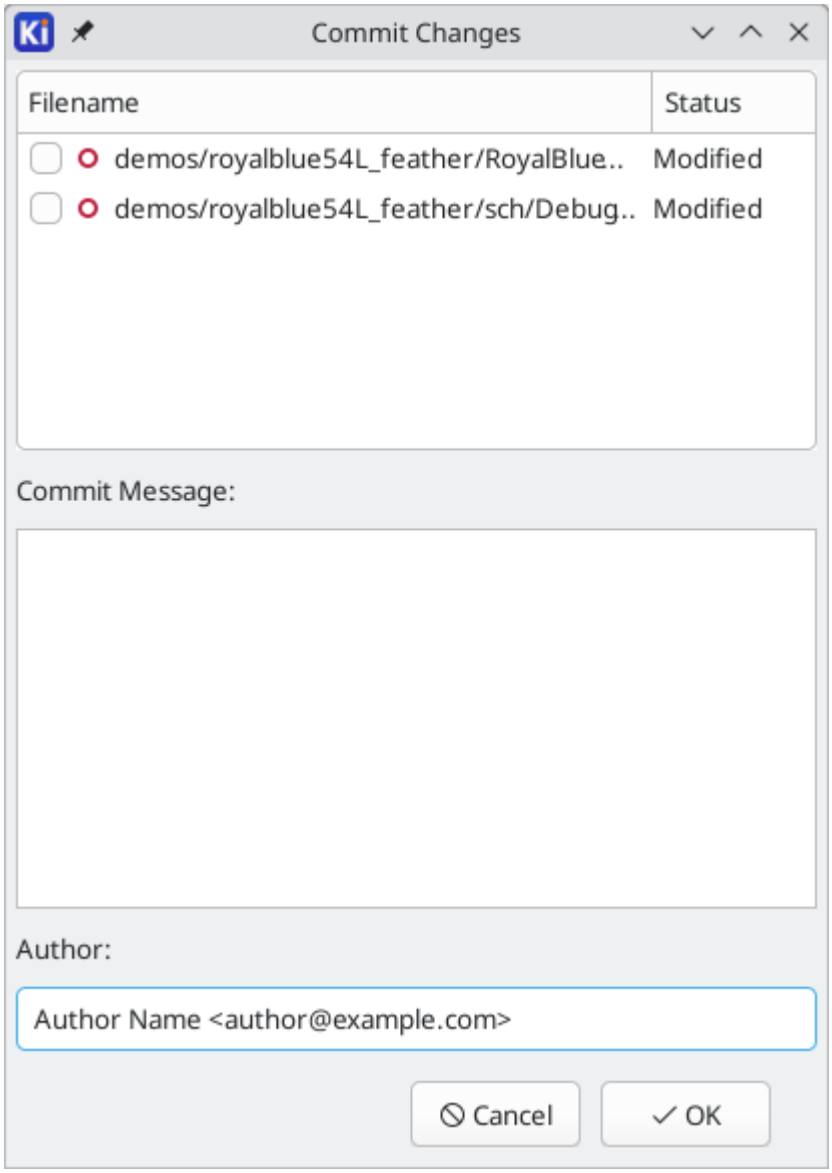
If an existing project is not already under version control, you can initialize a new Git repository in the project by right clicking on one of the files in the project files tree and clicking **Version Control → Add Project to Version Control...**. You must configure a remote when initializing a repository in this way. Configuring the repository requires the following information:

- **Name**: A name for the repository. This field can be anything and is not used.
- **Location**: The URL or file path to the remote.
- **Connection Type**: The protocol for connecting to the remote. This can be HTTPS, SSH, or local (file). HTTPS connections use username and password authentication. SSH connections use a username, private key, and an optional password for the keyfile. Local connections do not use authentication. You can check the connection and authentication by clicking the **Test** button.



To clone an existing repository and open the cloned project, use **File → Clone Project from Repository...**. You can clone a remote repository using SSH or HTTPS, or clone a local repository. The configuration settings for cloning are the same as the settings for configuring a new repository and remote for an existing project.

When you have made changes that you want to commit, you can commit either the entire project (right click → **Version Control → Commit Project...**) or a specific file (right click the file → **Version Control → Commit File...**). Both actions open the Commit Changes dialog, but the Commit Project action shows all changed files in the repository, while the Commit File action shows only the file that was right clicked. The Commit Changes dialog lets you select the changed files you want to include in the commit, provide a commit message and author, and commit the changes.



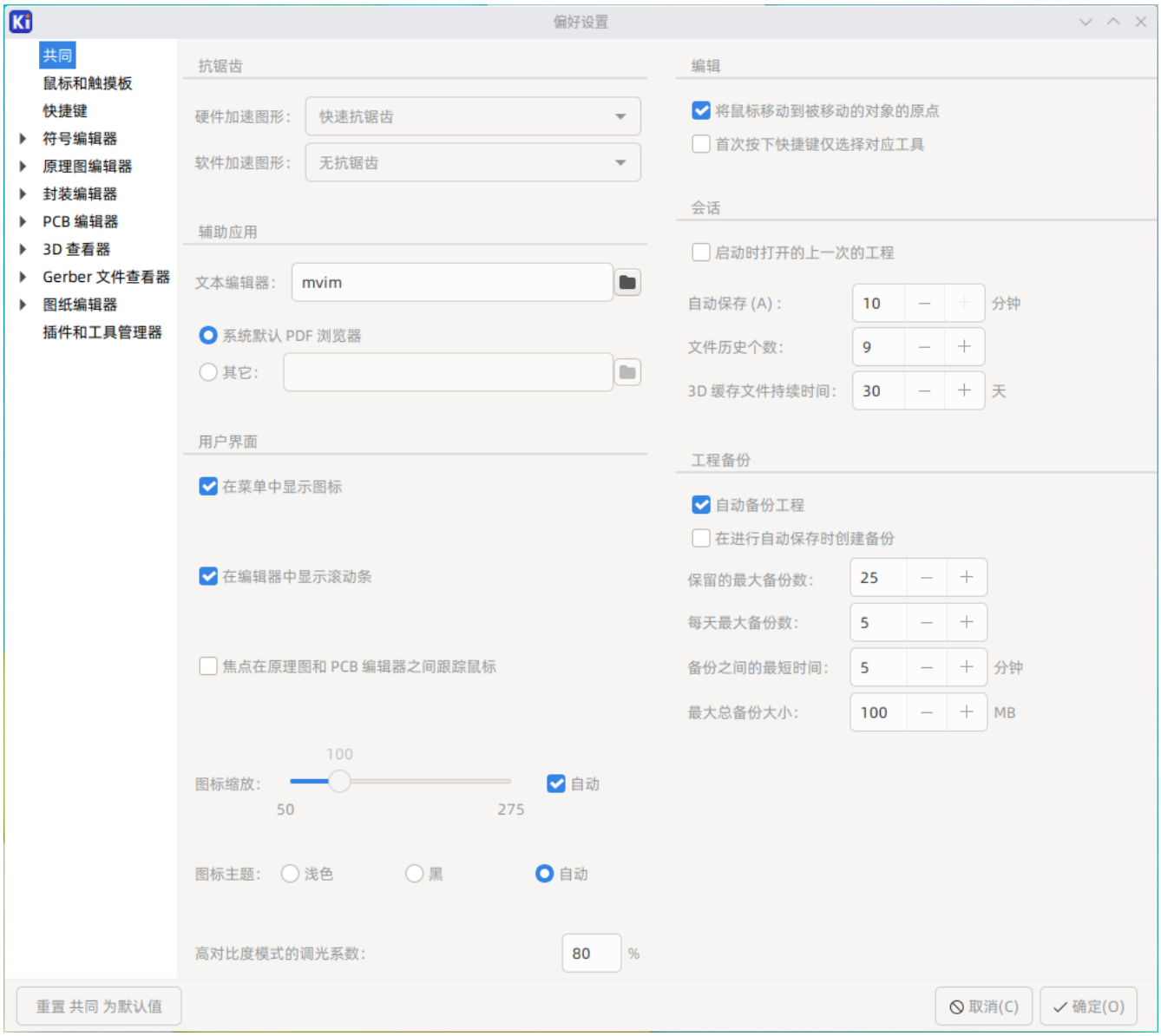
To push changes to the remote, right click in the project files tree and select **Version Control → Push**. To pull from the remote, right click and select **Version Control → Pull**. You can switch branches by selecting the desired branch from the **Version Control → Switch to Branch** menu.

Finally, you can remove version control entirely, deleting all tracked history from the local repository, by right clicking and selecting **Version Control → Remove Version Control**.

KiCad 配置

KiCad 偏好设置可随时从 **偏好设置** 菜单访问，或使用热键（默认 `Ctrl`+`,`）访问。运行中的 KiCad 工具共享偏好设置对话框。 有些偏好设置适用于所有工具，有些则针对特定工具（如原理图或电路板编辑器）。

共同偏好设置



Accelerated graphics antialising: KiCad can use different methods to prevent aliasing (jagged lines) when rendering using a graphics card. Different methods may look better on different hardware, so you may want to experiment to find the one that looks best to you.

Fallback graphics antialiasing: KiCad can also apply antialiasing when using the fallback graphics mode. Enabling this feature may result in poor performance on some hardware.

Text editor: Choose a text editor to use when opening text files from the project tree view.

PDF viewer: Choose a program to use when opening PDF files.

Show icons in menus: Enables icons in drop-down menus throughout the KiCad user interface.

NOTE | 菜单中的图标在某些操作系统上不显示。

Show scrollbars in editors: When enabled, scrollbars are displayed next to the editing canvases in each tool. When disabled, scrollbars are not shown.

Focus follows mouse between schematic and PCB editors: When enabled, the window under the mouse cursor will automatically become focused.

Icon scale: Sets the size of the icons used in menus and buttons throughout KiCad. Choose *Automatic* to pick an appropriate icon scale automatically based on your operating system settings.

Icon theme: Sets whether to use the icon theme designed for light window backgrounds or dark window backgrounds. The default setting of Automatic will choose the theme based on the lightness of the operating system window theme.

High-contrast mode dimming factor: Sets how much non-focused items are dimmed in high-contrast display mode.

Warp mouse to origin of moved object: When enabled, the mouse cursor will be repositioned (warped) to the origin of an object when you start a move command on that object.

First hotkey selects tool: When disabled, pressing the hotkey for a command such as *Add Wire* will immediately start the command at the current cursor location. When enabled, pressing the hotkey the first time will just select the *Add Wire* tool but will not immediately begin a wire.

Remember open files for next project launch: When enabled, KiCad will automatically re-open any files that were previously open when a project is re-opened.

Auto save: When editing schematics and board files, KiCad can automatically save your work periodically. Set to `0` to disable this feature.

File history size: Configure the number of entries in the list of recently-opened files

3D cache file duration: KiCad creates a cache of 3D models in order to speed up the 3D viewer. You can configure how long to keep this cache before deleting old files.

Automatically backup projects: When enabled, KiCad projects will be archived to ZIP files automatically according to the settings below. The archives will be stored in a subfolder of the project folder. Backups are created when saving files in the project.

Create backups when auto save occurs: When enabled, a backup will be created every time an automatic file save occurs (if the backup is permitted by the settings below). This setting has no effect if the auto save interval is set to `0` (disabled).

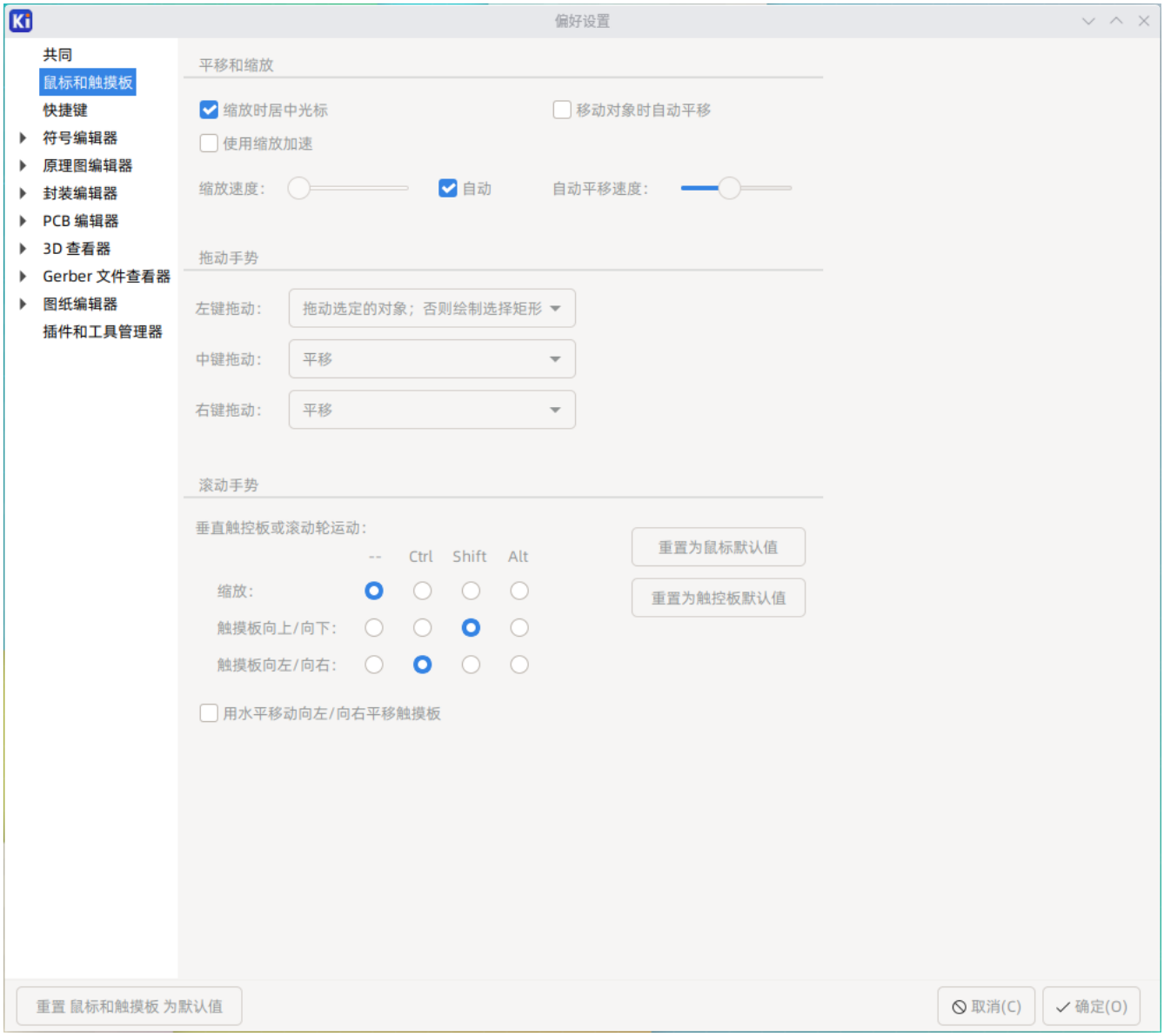
Maximum backups to keep: When creating a new backup, the oldest backup file will be deleted to keep the total number of backup files below this limit.

Maximum backups per day: When creating a new backup, the oldest backup file created on the current day will be deleted to stay below this limit.

Minimum time between backups: If backup is triggered (for example, by saving a board file), the backup will not be created if an existing backup file is newer than this limit.

Maximum total backup size: When creating a new backup file, the oldest backup files will be deleted to keep the total size of the backup files directory below this limit.

鼠标和触摸板偏好设置



Center and warp cursor on zoom: When enabled, zooming using the hotkeys or mouse wheel will cause the view to be centered on the cursor location.

Use zoom acceleration: When enabled, scrolling the mouse wheel or touchpad faster will cause the zoom to change faster.

Zoom speed: Controls how much the zoom changes for a given amount of scrolling the mouse wheel or touchpad. Use *Automatic* to set a default value depending on your operating system.

Automatically pan while moving object: When enabled, the view can be panned while moving an object by moving close to the edge of the canvas.

Auto pan speed: Controls how fast the canvas pans while moving an object.

Mouse buttons: You can set the behavior of dragging the middle and right mouse buttons to zoom the view, pan the view, or have no effect. You can also set the behavior of dragging the left mouse button depending

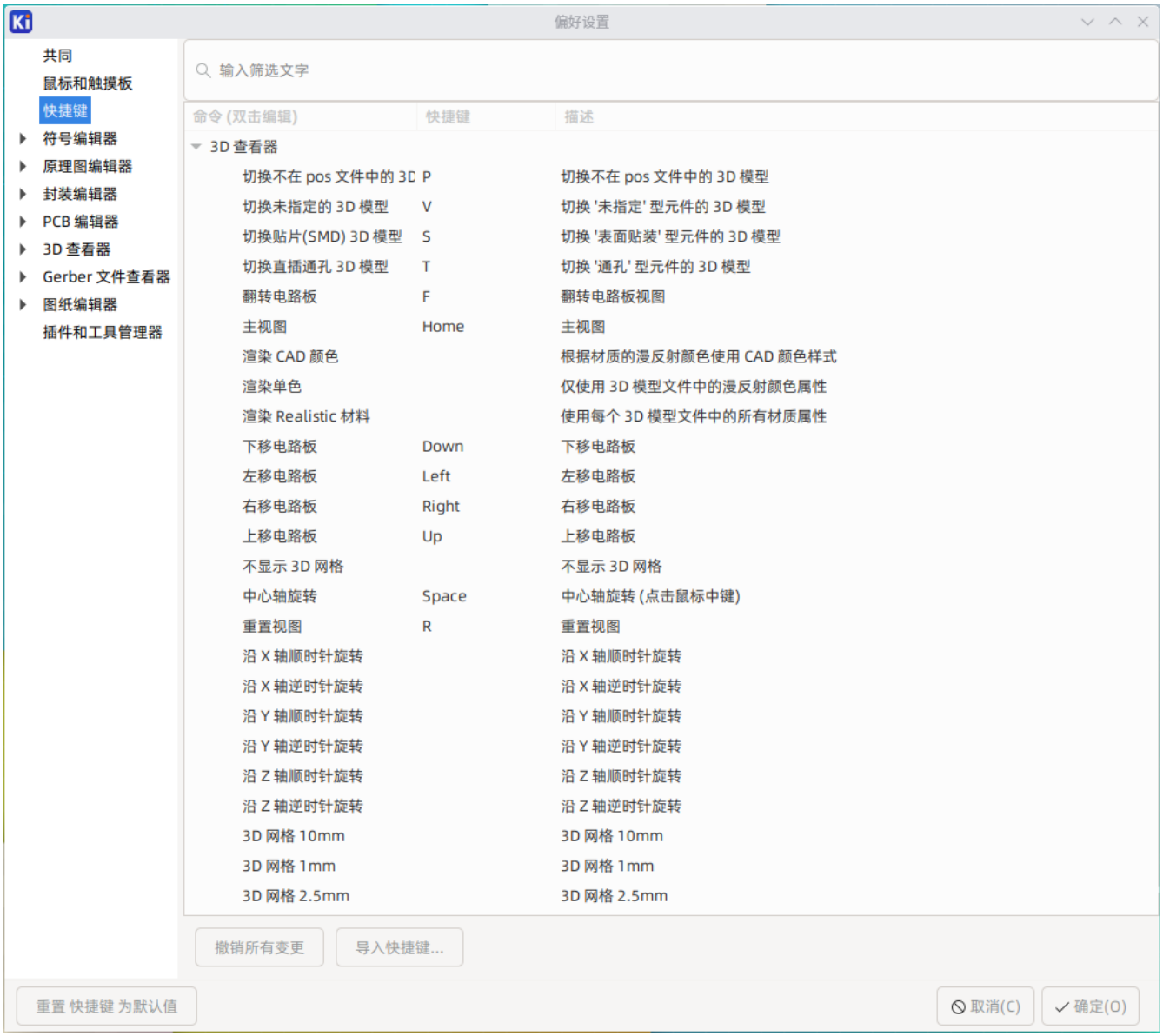
on whether or not any objects are already selected in the editing canvas.

NOTE	鼠标左键始终用于选择和操作对象。

Mouse wheel and touchpad scrolling: You can set the behavior of scrolling the mouse wheel or vertical motion of the touchpad while pressing certain modifier keys.

Pan left/right with horizontal movement: When enabled, you can pan the view using the touchpad or horizontal scroll wheel (if present on your mouse).

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。

Importing hotkeys

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 **路径变量** 定义路径。 KiCad 内部定义了一些路径变量，可用于为库、3D 图形等定义路径。

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：

- 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 `KICAD9_FOOTPRINT_DIR` path variable, would be defined as `${KICAD9_FOOTPRINT_DIR}/connect.pretty`.

通过 **偏好设置** → **配置路径...** 菜单，您可以为 KiCad 的一些内置路径变量定义路径，并根据需要添加自己的路径变量来定义个人路径。

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, `${KICAD8_FOOTPRINT_DIR}` will automatically resolve to the value of `${KICAD9_FOOTPRINT_DIR}` if there is no `KICAD8_FOOTPRINT_DIR` variable defined.

KiCad 路径变量

`KICAD9_3DMODEL_DIR`	Base path of KiCad's standard 3D footprint model library files.
`KICAD9_3RD_PARTY`	Location for plugins, libraries, and color themes installed by the Plugin and Content Manager.
`KICAD9_FOOTPRINT_DIR`	Base path of KiCad's standard footprint library files.
`KICAD9_SYMBOL_DIR`	Base path of KiCad's standard symbol library files.
`KICAD9_TEMPLATE_DIR`	Location of KiCad's standard project template library files.
`KICAD_USER_TEMPLATE_DIR`	Location of personal project templates.
`SPICE_LIB_DIR`	Location of personal simulation model libraries. This variable is not defined by default.
`KIPRJMOD`	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》。

路径还可以设置为 KiCad 外部的系统环境变量，这将覆盖用户配置中的任何设置。

NOTE	使用 `配置路径` 对话框无法覆盖在 KiCad 外部设置的系统环境变量。任何外部设置的变量都将在对话框中显示为只读。

还请注意，路径变量 `KIPRJMOD` **始终** 由 KiCad 内部定义，并展开为 **当前工程的绝对路径**。 例如，`${KIPRJMOD}/connect.pretty` 始终是 **当前工程文件夹** 内的 `connect.pretty` 文件夹（封装库）。 KIPRJMOD 变量不能在 `配置路径` 对话框中更改，也不能被外部环境变量覆盖。

高级环境变量

可以设置一些高级环境变量，自定义 KiCad 预计某些文件的位置。默认情况下，这些位置是根据你的平台设置的，但它们可以被系统环境变量覆盖。这些变量不会在 `配置路径` 对话框中显示，也不能用于路径替换。

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	KiCad 配置文件的基础路径。此目录下将为每个 KiCad 次版本创建子目录，用于存储特定版本的配置文件。
KICAD_DOCUMENTS_HOME	KiCad 用户可修改文档的基础路径，如工程、模板、Python 脚本、库等。此目录下也将为每个 KiCad 次版本创建子目录。该目录作为推荐的用户数据存储位置，但并非强制使用。
KICAD_STOCK_DATA_HOME	KiCad 标配数据的基础路径，包含默认库等资源。此目录中的数据由 KiCad 安装程序或系统包管理器管理，不建议用户在此目录中写入自定义数据。

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

通过 **偏好设置** → **管理符号库...** 菜单，您可以管理符号库列表（符号库表）。

同样，使用 **偏好设置** → **管理封装库表...** 菜单来管理封装库列表（封装库表）。

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。

Jobsets

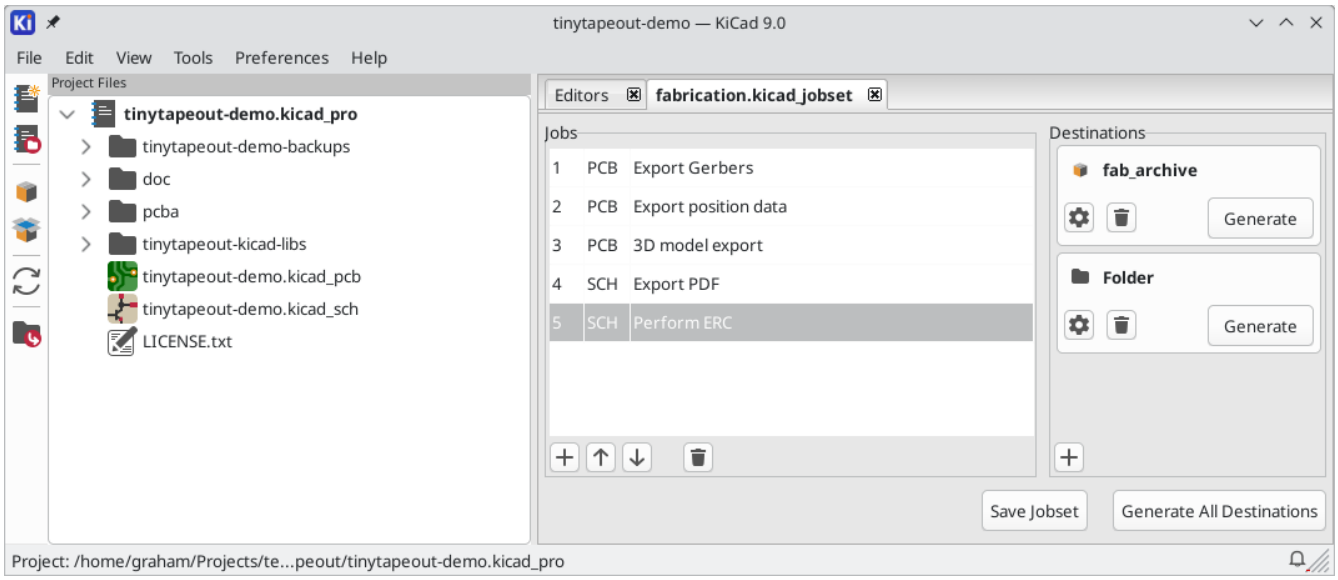
KiCad lets you configure a list of outputs that are all generated with a single click. The list of output jobs and the destinations where they will be saved is called a *jobset*. For example, a jobset might contain jobs to generate Gerber files, assembly data, a bill of materials, PDF plots of the schematic and PCB, while also running ERC and DRC checks, with all of the outputs saved to a compressed archive. The full list of available jobs is given below.

Each *job* in a jobset defines a single type of generated output, such as a bill of materials or a set of Gerbers. A job can be configured in the same way as if the output was manually generated from the schematic or board editor. The configuration for each job is stored in the jobset and remembered when you load the jobset later. Jobs are configured individually, so if you include the same type of job multiple times in a single jobset, each job will have its own independent configuration. For example, this lets you generate PDF outputs in color as well as black and white.

In addition to the jobs, jobsets also contain *destinations*, which define a list of jobs to run and how to store their outputs. A jobset destination can simply store the chosen jobs' output files in a specified location, or it can add the output files to a compressed archive. Each jobset destination can select a different subset of jobs from the full list of jobs in the jobset. You can run each jobset destination individually or run all jobset destinations at once. As an example, you could set up one jobset destination that generates PDFs of the board and schematic and copies them to an external location, while another destination generates the fabrication files and compresses them in a zip archive to send to the board manufacturer.

Projects can have multiple jobsets, with each jobset defining a different list of jobs and output configurations. Each jobset is stored in a `.kicad_jobset` file, which can be specific to a single project, copied between projects, or even stored in a central location and shared between projects.

To use a jobset, first create a new jobset file in the KiCad project manager (**File → New Jobset File...**) and choose a name and location for it. Alternatively, you can open an existing jobset file with **File → Open Jobset File...**. Jobset files that are stored in the project directory are considered part of the project and are displayed in the project file tree. You can open a jobset file in the project file tree by double clicking on it.



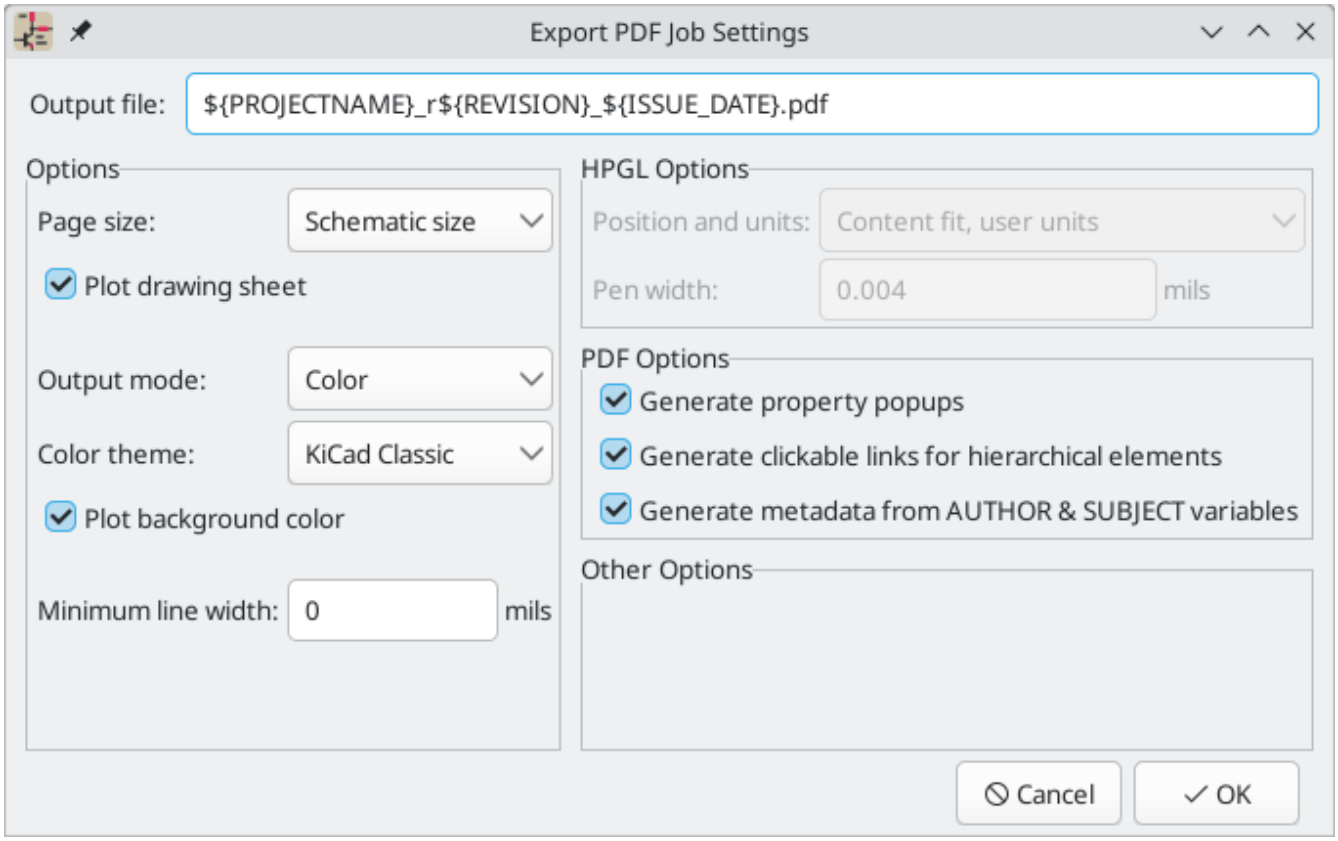
Once you create or open a jobset, it is displayed in a new tab in the project manager. The list of jobs is shown in the middle and the list of jobset destinations is shown on the right. New jobsets will not contain any jobs, but a destination is automatically created to save outputs to a folder. When you make changes to a jobset, you can save the changes by clicking the **Save Jobset** button.

Defining jobs

To add a new job, click the + button under the Jobs list. In the Add New Job dialog that appears, select the desired type of job. You can filter which types of jobs are shown in the list by typing in the **Filter** textbox at the bottom.

When you select a job and press **OK**, the settings dialog for that type of output will appear. Each job settings dialog provides the same options you would have if you manually generated that type of output from the schematic or board editor.

NOTE	Output filenames and paths specified in job settings are relative to the jobset destination folder or archive root. You can use certain text variables, like `${PROJECTNAME}`, `${CURRENT_DATE}`, and project text variables.



When you accept the job settings dialog, the job is added to the list of jobs, where you can optionally change the new job's description from its default. To change a job's description or settings later, right click the job in the list and select **Edit Job Description** or **Edit Job Settings...**. Double clicking on a job also edits its settings. To remove a job, select the job and click the 🗑 button. To reorder the list, select a job and move it up or down using the ↑ or ↓ buttons.

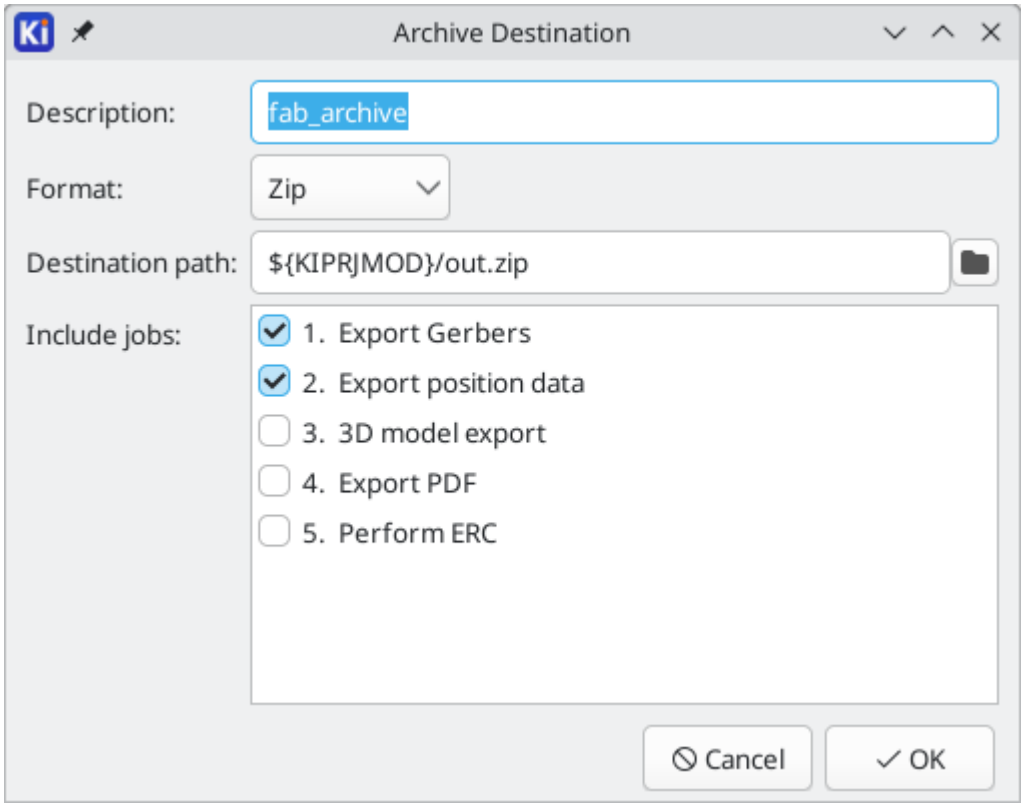
Defining jobset destinations

You cannot generate any outputs from a jobset until you add a jobset destination. One destination is created automatically when a jobset is created, but you can add as many destinations as you need.

To add a jobset destination, click the + button under the Destinations list. When the Add New Destination dialog appears, select a type of destination:

- **Archive** saves the outputs generated by the jobs in a compressed zip archive.
- **Folder** saves the outputs generated by the jobs uncompressed in a folder.

Once you have selected a type of output, the Destination options dialog appears.

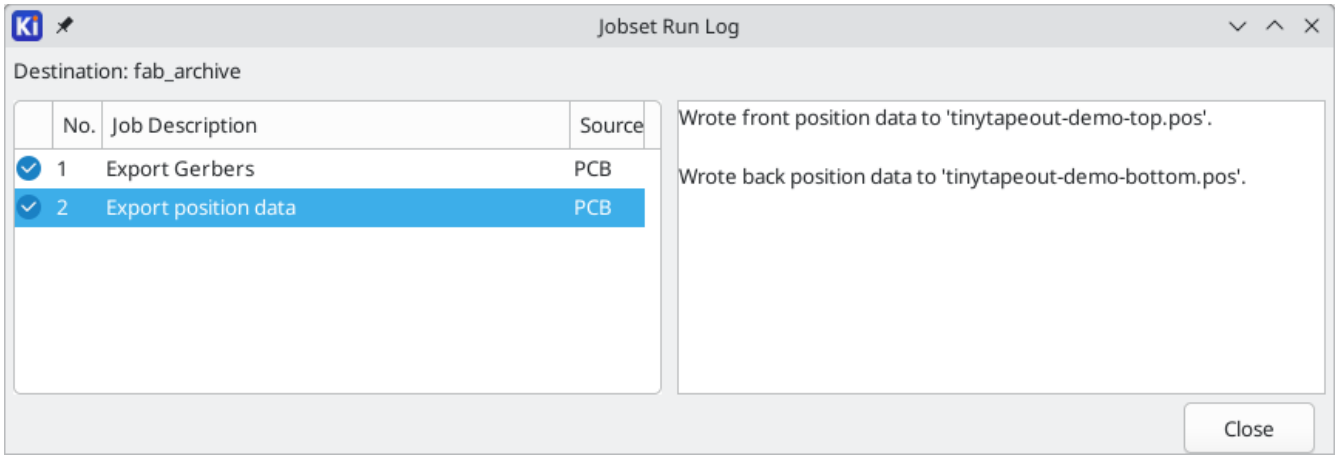


Here you can select which jobs will be run as part of this jobset destination, as well as the folder or archive name that will be used to store them. By default all jobs are enabled. You can also set a description for the destination to be displayed in the Destinations list. The output path controls where the files generated by the jobs will be saved. The path here can be absolute or relative to the project directory, and it can use path variables or certain text variables (`${PROJECTNAME}`, `${CURRENT_DATE}`, and project text variables). Filenames defined in job configurations are relative to the jobset destination directory or archive root.

When you click **OK** in this dialog, the new jobset destination is added to the Destinations list. You can modify an existing jobset destination by clicking its ⚙ button, or remove it by clicking its 🗑 button.

After configuring your jobs and destinations, you can generate an individual set of outputs by clicking the **Generate** button for the desired destination. You can run all destinations at once by clicking the **Generate All Destinations** button.

If a jobset destination runs and generates its outputs successfully, a blue check is shown that indicates the last run was successful. If a jobset destination fails to complete successfully, a red exclamation point is shown to indicate the run was not successful. Clicking on the success/failure indicator will display the Jobset Run Log dialog, which displays the status of each job in the jobset destination. Clicking on a specific job will display the logged output from that job, if there is any.



Available job types

The following types of jobs are available:

Job	Description
PCB: Export 3D Model	Exports a 3D model of the board. The model format can be STEP, GLB (binary glTF), XAO, BREP (OCCT), PLY, or STL.
PCB: Export Drill Data	Exports a drill file from the board.
PCB: Export DXF	Exports the board design to a DXF file.
PCB: Export Gerbers	Exports the board design to Gerber files, with one file per selected layer.
PCB: Export IPC-2581	Exports the board design in IPC-2581 format.
PCB: Export ODB++	Exports the board design in ODB++ format.
PCB: Export PDF	Exports the board design to PDF files, with one file per selected board layer. You can also generate a single PDF with multiple layers depending on the plot configuration.
PCB: Export Position Data	Exports a position (component placement) file from the board.
PCB: Export SVG	Exports the board design to a SVG file.
PCB: Perform DRC	Performs a Design Rule Check on the board and generates a report. If DRC violations are found, this job can optionally report a job failure.
PCB: Render	Generates a raytraced rendering of the 3D model of the board as a PNG or JPG file.
Schematic: Export DXF	Exports the schematic to a DXF file.
Schematic: Export HPGL	Exports the schematic to a HPGL file.
Schematic: Export Netlist	Exports a netlist from the schematic, with various formats available.
Schematic: Export PDF	Exports the schematic to a PDF file.
Schematic: Export Postscript	Exports the schematic to a PostScript file.
Schematic: Export SVG	Exports the schematic to a SVG file.
Schematic: Generate Bill of Materials	Exports a bill of materials from the schematic.
Schematic: Perform ERC	Performs an Electrical Rule Check on the schematic and generates a report. If ERC violations are found, this job can optionally report a job failure.
Special: Copy Files	Copies the specified file to the specified location. A failure to copy the files can optionally cause the output job to fail. You can control whether files in the output location should be overwritten or not.
Special: Execute Command	Executes an arbitrary command. Output from the command can optionally be logged to a file. You can either ignore non-zero output codes or cause them to fail the output job.

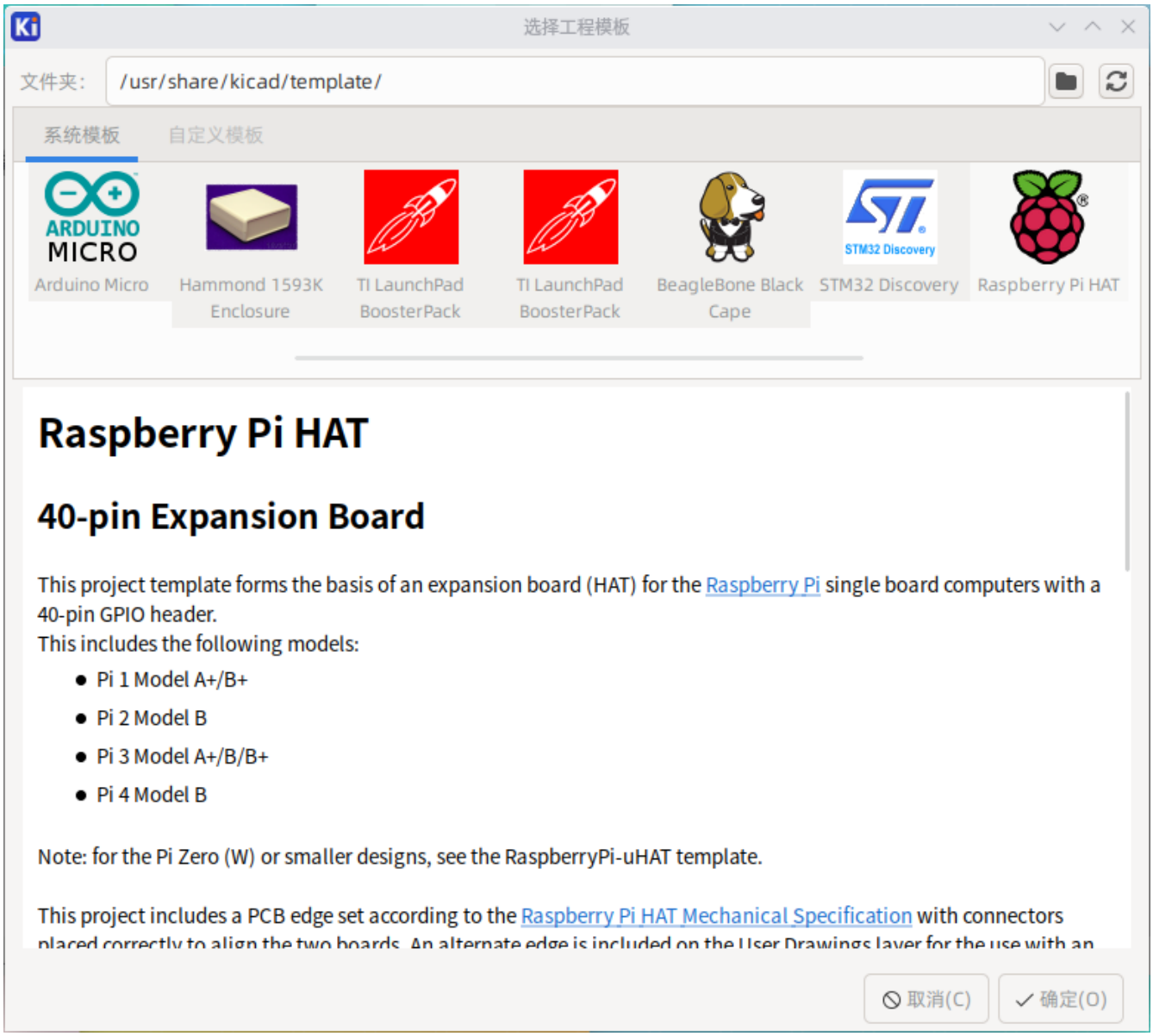
工程模板

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 `KICAD9_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

KiCad 模板只是一个包含模板工程文件的目录，以及子目录 `meta` 中模板所需的一些元数据。包含模板文件的目录名称决定了模板的名称。根据模板创建工程时，KiCad 会将模板文件复制到新的工程目录，并重新命名这些文件以匹配新工程名称，如下所述。

模板中的所有文件都会被复制，但有两个例外：

- 名称以 `.` 开头的文件（隐藏文件）不会被复制。 名为 `.gitignore` 或 `.gitattributes` 的文件有一个特殊情况，如果它们存在，就会被复制。
- 不复制 `meta` 目录

`meta` 目录必须包含一个名为 `info.html` 的 HTML 文件，该文件将显示在 KiCad 模板浏览器中，并应包含描述模板的基本信息。支持基本的 HTML 功能，包括图像。 由 `info.html` 引用的任何图像也应存储在 `meta` 目录中。

`<title>` 标记决定模板选择时显示的模板名称。请注意，如果工程模板名称太长，将会被截断。显示名称不必与模板目录名称相同。

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&amp;quot; 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>
```

最后，`meta` 可选择包含一张名为 `icon.png` 的图片，该图片将在模板选择对话框中用作模板的图标。图标应为 64 x 64 像素的 PNG 图像。

模板文件重命名

使用模板创建工程时，模板中的所有文件和目录都会复制到新的工程路径中，`meta` 和任何隐藏文件除外。 包含模板目录名的文件和目录将以新工程文件名重命名。

例如，使用名为 `example`（左）的模板创建名为 `newproject`（右）的工程，重命名后的文件以 **粗体** 显示：

模板 `example` 目录下的文件	**在工程 `newproject` 目录下创建的文件**

`example.kicad_pro`
`example.kicad_sch`
`example.kicad_pcb`
`example-first.kicad_sch`
`second-example.kicad_sch`
`third.kicad_sch`
`third.kicad_pcb`
	`newproject.kicad_pro`
	`newproject.kicad_sch`
	`newproject.kicad_pcb`
	`newproject-first.kicad_sch`
	`second-newproject.kicad_sch`
	`third.kicad_sch`
	`third.kicad_pcb`

模板无需包含完整的工程。如果缺少所需的项目文件，KiCad 将使用默认的创建工程行为来创建文件：

模板 `example` 目录下的文件	**在 `newproject` 目录下创建的文件**

`example.kicad_sch`
`first-example.kicad_sch`
`first-example.kicad_pcb`
`second-example.kicad_sch`
`second-example.kicad_pcb`
	`newproject.kicad_sch`
	`first-newproject.kicad_sch`
	`first-newproject.kicad_pcb`
	`second-newproject.kicad_sch`
	`second-newproject.kicad_pcb`
	`newproject.kicad_pro` (default)
	`newproject.kicad_pcb` (default)

作为模板名称重命名规则的例外情况，如果模板包含一个工程文件（`.kicad_pro`），且其名称与模板名称不匹配，KiCad 将根据该工程文件名称进行重命名：

模板 `example` 目录下的文件	**在 `newproject` 目录下创建的文件**

`example.kicad_sch`
`example.kicad_pcb`
`first-example.kicad_pro`
`first-example.kicad_sch`
`first-example.kicad_pcb`
`second-example.kicad_sch`
`second-example.kicad_pcb`
	`example.kicad_sch`
	`example.kicad_pcb`
	`newproject.kicad_pro`
	`newproject.kicad_sch`
	`newproject.kicad_pcb`
	`second-example.kicad_sch`
	`second-example.kicad_pcb`

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	本节 KiCad 文档尚未编写完成。 我们的志愿文档编写小组正在努力更新和扩充文档，请您耐心等待。

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
关闭工程		关闭当前工程
图像转换	Ctrl + B	将位图图像转换为原理图或 PCB 元件
图框编辑器	Ctrl + Y	编辑图框页边框和标题栏
封装编辑器	Ctrl + F	编辑 PCB 封装
PCB 编辑器	Ctrl + P	编辑 PCB 布局
原理图编辑器	Ctrl + E	编辑电路原理图
符号编辑器	Ctrl + L	编辑原理图符号
从仓库克隆工程...		从现有仓库克隆工程
从模板新建工程...	Ctrl + T	从模板创建新工程
新建空白工程...	Ctrl + N	创建新的空白工程
打开演示工程...		打开演示工程
打开工程...	Ctrl + O	打开已有工程
打开文本编辑器		启动偏好设置的文本编辑器
插件与内容管理器	Ctrl + M	运行插件与内容管理器
计算器工具		运行元件计算、走线宽度计算等
Gerber 查看器	Ctrl + G	预览 Gerber 输出文件