

KiCad

The KiCad Team

Table of Contents

简介	2
系统要求	2
安装和升级 KiCad	3
First time setup	3
Migrating design files from previous versions	4
使用 KiCad 工程管理器	6
单机模式	7
创建一个新工程	8
从其他 EDA 工具导入工程	9
保存和加载工程档案	10
Git 集成	11
KiCad 文件和文件夹	14
工程文件	14
Schematic editor files and folders	14
PCB 编辑器的文件和文件夹	15
通用文件	15
制造和文档文件	16
储存和发送 KiCad 文件	16
Paths and libraries configuration	17
KiCad 路径变量	17
高级环境变量	18
库配置	19
作业集	20
定义作业	20
定义工作集目标	21
可用的作业类型	23
工程模板	25
使用模板	25
模板位置	26
Template contents	27
Creating new templates	29
插件和内容管理器	31
浏览软件包	31
安装软件包	33
Managing repositories	34
创建软件包和仓库	35
KiCad preferences	36
共同偏好设置	36
鼠标和触摸板偏好设置	38
快捷键偏好设置	39
操作参考	41

KiCad 10.0 参考手册

Copyright

This document is Copyright The KiCad Documentation Contributors. 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。

Contributors

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

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

Software and Documentation Version

本用户手册基于 KiCad 10.0.0 版本。其他版本的 KiCad 在功能和界面外观上可能存在差异。

文档修订版：90e4b228。

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

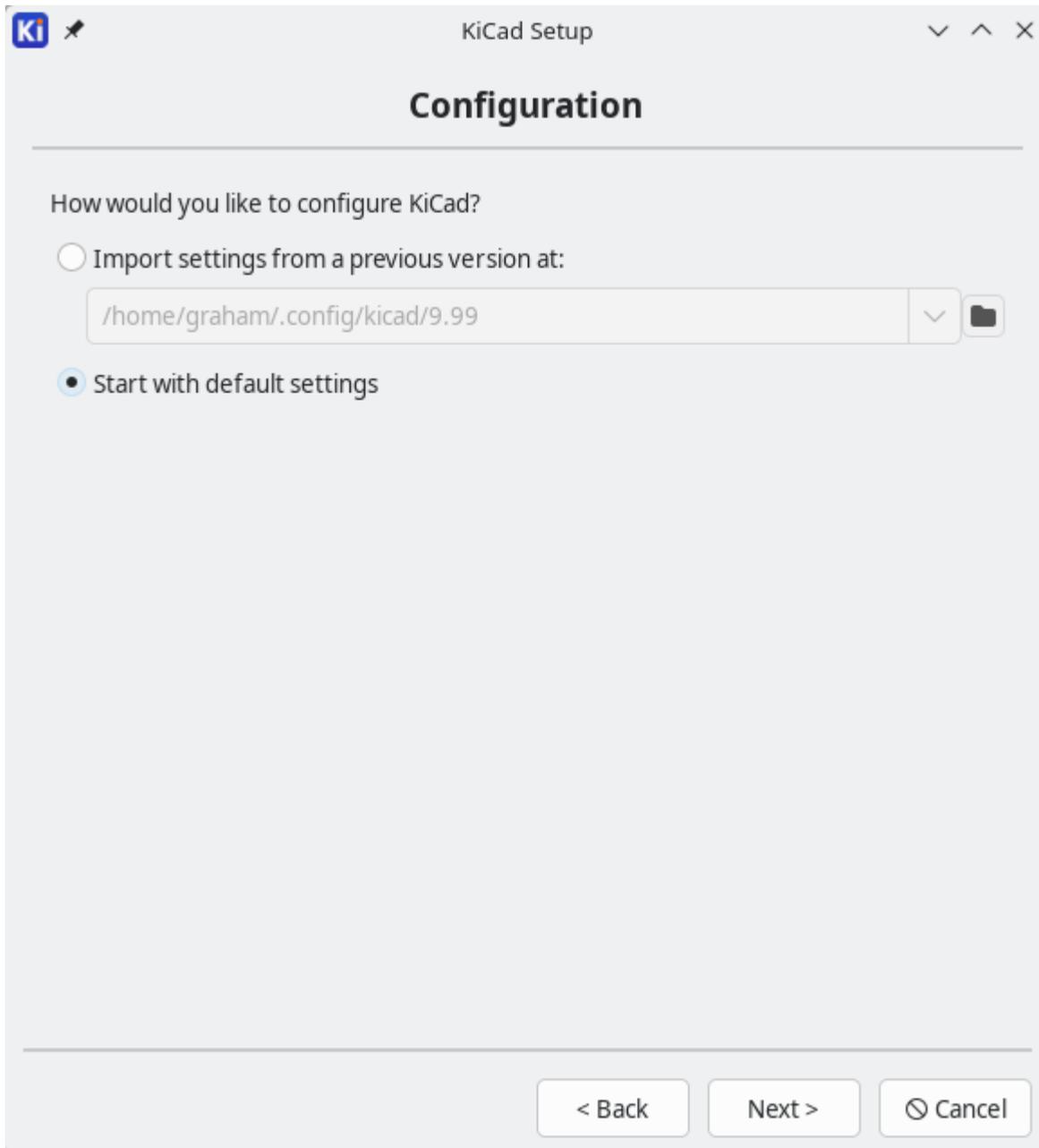
KiCad 能够在多种硬件和操作系统上运行, 但在低端硬件上执行某些任务可能会很慢或很困难。为获得最佳体验, 建议使用 1920x1080 或更高分辨率的专用显卡和显示器。

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

安装和升级 KiCad

First time setup

The first time you run a new major version of KiCad, a Welcome window will appear to ask you how to initialize your settings. Each major release of KiCad has its own configuration, so that you may run multiple KiCad versions on the same computer without the configurations interfering.



Configuration setup

You always have the option to start with default settings for the new version. If a previous version of KiCad is detected, you will have the option to import the settings from that version. The location of the previous configuration files is detected automatically, but you may override it to choose another location if desired.

NOTE

Hotkey configurations are not imported from previous versions. You can manually import hotkey configurations by using the **Import Hotkeys...** button on the **Hotkeys** page of Preferences, then browsing to the `.hotkeys` file from the old version's configuration directory. If you do so, please note that this will replace the current version's hotkey settings.

KiCad stores the settings files in a folder inside your user directory. Each KiCad version will use a different versioned subfolder. For KiCad 10, those folders are:

Windows	%APPDATA%\kicad\10.0
Linux	~/.config/kicad/10.0
macOS	/Users/<username>/Library/Preferences/kicad/10.0

Libraries setup

You also have options for how to initialize your symbol, footprint, and design block libraries. In most cases, you should choose the default option, **Start with the built-in KiCad libraries (recommended)**. This initializes your library tables to include all of the libraries that come with KiCad.

The other options are to import your library table setup from the previously installed KiCad version, or to start with no libraries and do all of the configuration yourself. Refer to the [Schematic Editor](#) and [PCB Editor](#) documentation for details on how to manually configure libraries.

Depending on your system and how KiCad was installed, some options may not be available. KiCad may also not be able to locate any libraries installed on your system. In this case, you should ensure you have installed the libraries, or you can manually configure the libraries after completing setup.

NOTE

On some systems the KiCad libraries are installed as a separate package.

The location of the default library table files depends on operating system and may vary based on installation location. Below are the defaults for each operating system:

Windows	C:\Program Files\KiCad\10.0\share\kicad\template\
Linux	/usr/share/kicad/template/
macOS	/Applications/KiCad/KiCad.app/Contents/SharedSupport/template/

Updates & Privacy setup

You can also choose whether to be notified about updates to the KiCad application and to third party packages you may have installed through KiCad's [Plugin and Content Manager](#) (plugins, themes, libraries, etc.). On Windows, you also have the option to enable anonymous crash reporting, which helps the KiCad team diagnose and fix critical issues in the application.

Migrating design files from previous versions

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。

使用 KiCad 工程管理器

The KiCad project manager is the window that opens when you start KiCad. The project manager creates and opens KiCad projects and launches the other KiCad tools:

- the [Schematic Editor](#)
- the [Symbol Editor](#)
- the [PCB Editor](#)
- the [Footprint Editor](#)
- the [Gerber Viewer](#)
- the [Image Converter](#)
- the [Calculator Tools](#)
- the [Drawing Sheet Editor](#)
- the [Plugin and Content Manager](#)

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。



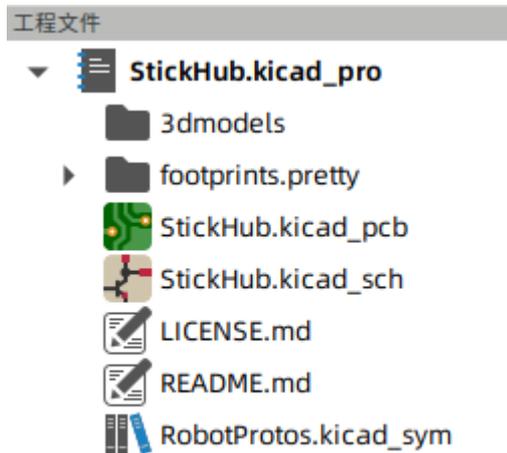
The tree view shows a list of the files inside the project folder. This includes the KiCad schematic and board design files as well as other files and folders in the project folder. If the project contains multiple schematic sheets, only the root sheet is shown in the tree view.

NOTE

仅显示 KiCad 能够打开的文件在工程工作区视图中。

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. While normally the tree view only shows the schematic root sheet, and not any subsheets, all sheets are shown when the project is in a Git repository so that modifications to each sheet can be observed individually.



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。原理图可能包含多个图纸，每个图纸都有自己的文件，但一个工程只能包含一个电路板。KiCad 要求工程文件、原理图根图纸文件和电路板文件都具有相同的名称。

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。

单机模式

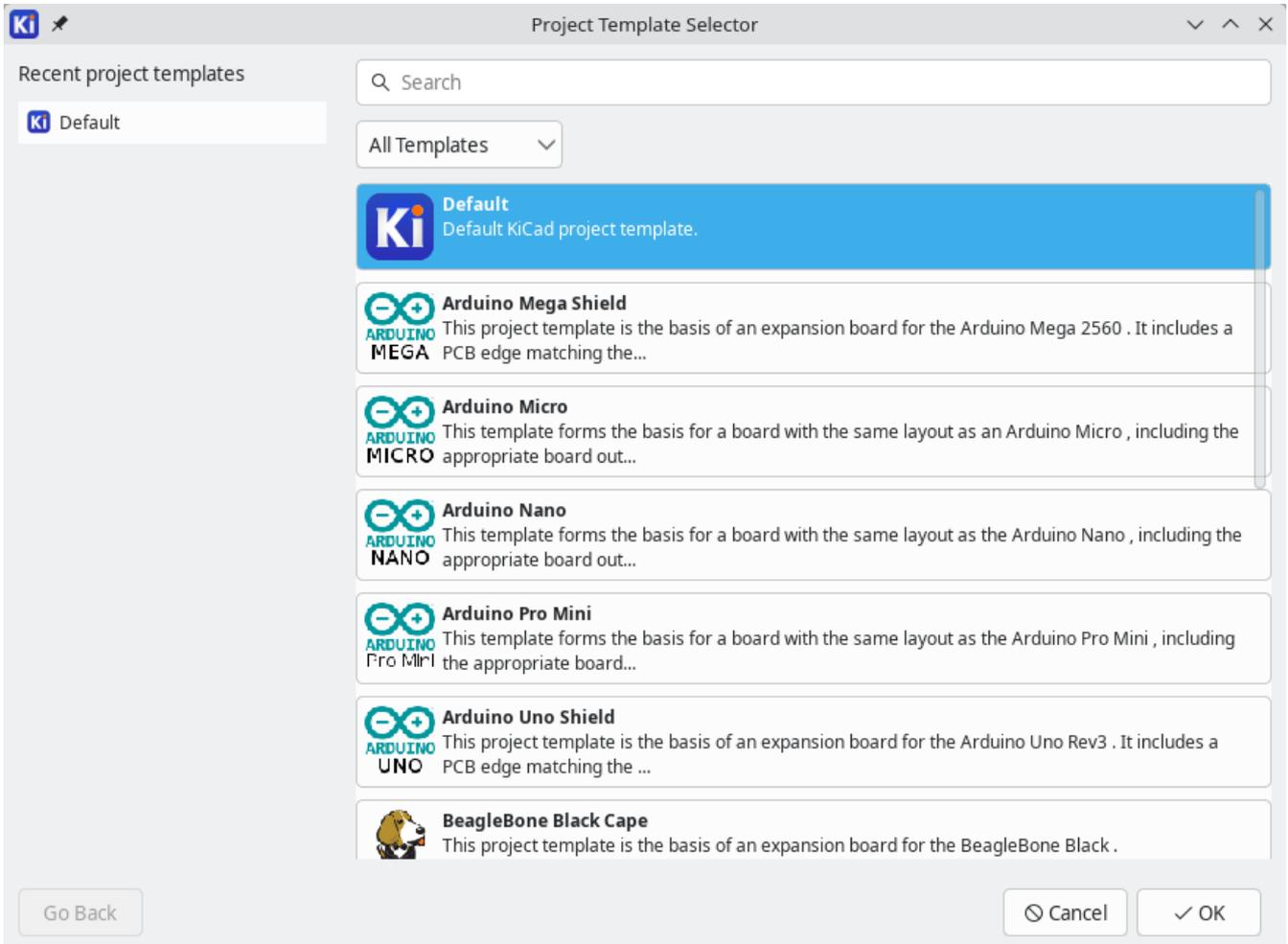
您也可以在 *单机模式* 模式下运行 KiCad 编辑器工具，方法是直接从操作系统的应用程序启动器而不是从工程管理器启动这些工具。通常 **不建议** 在单机模式下运行工具，除非在某些特殊情况下必须这样做，例如从其他 EDA 工具导入工程时。在单机模式下运行时，某些工程功能不可用，包括：

- 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

Most KiCad designs start with the creation of a project. When you create a new project, you always begin by choosing a project template, even if that template is the default, blank project template.

To create a new project, use the **New Project...** command in the **File** menu, click the **New Project** button in the top toolbar, or use the keyboard shortcut (**Ctrl** + **N** by default).



The right side of this dialog shows a list of all installed project templates. The **Default** template is always available for you to create an empty project. Clicking one of these templates will show the template's description in a new panel. The left side of the dialog shows a list of your recently used project templates.

NOTE Creating new project templates is covered in the [Project Templates](#) section.

TIP Creating your own project templates is a good way to easily create new projects that are preconfigured with your preferred drawing sheets, design rules, PCB outlines, logos, etc.

When you have chosen a template, click the **OK** button to create a new project from that template.

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`。

If you already have a directory to store your project files in, you can uncheck the **Create a new directory for the project** checkbox in the New Project dialog.

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。

Other files may also be created, depending on the contents of the template.

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, and must be manually linked together into a KiCad project. The following types of projects are supported:

Source EDA Tool	File Extension(s)
Altium Designer	.PrjPcb
CADSTAR archive format	.csa, .cpa
Eagle 6.x or newer (XML format)	.sch, .brd
EasyEDA (JLCEDA) Standard Backup	.zip
EasyEDA (JLCEDA) Pro Project	.epro, .zip
PADS ASCII Project	.asc, .txt
gEDA / Lepton EDA Project	.prj, .sch, .pcb

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。

Depending on the format, you will be prompted to select either a project, schematic, or a board file in the import file browser dialog. The imported schematic and board files should have the same base file name (e.g. project.sch and project.brd). Once the requested files are selected, you will be asked to select a directory to store the resulting KiCad project.

A greater number of formats can be imported as either schematics or boards, rather than complete projects. The following schematic formats can be imported into the Schematic Editor:

Source EDA Tool	File Extension(s)
Altium Designer	.SchDoc
Eagle (Autodesk)	.sch (XML)
LTspice	.asc
PADS Logic	.asc, .txt
CADSTAR Schematic Archive	.csa
gEDA / Lepton EDA	.sch
EasyEDA (JLCEDA) Standard	.json
EasyEDA (JLCEDA) Professional	.epro, .zip

Importing these schematic formats is explained in more detail in the [Schematic Editor documentation](#).

The following PCB formats can be imported into the PCB Editor:

Source EDA Tool	File Extension(s)
Altium Designer	.PcbDoc
Altium Circuit Maker	.CMPcbDoc
Altium Circuit Studio	.CSPcbDoc
Cadence Allegro	.brd
CADSTAR PCB Archive	.cpa
Eagle (Autodesk)	.brd (XML)
EasyEDA / JLCEDA Standard	.json, .zip
EasyEDA / JLCEDA Professional	.epro, .zip
Fabmaster	.txt, .fab
gEDA / Lepton EDA	.pcb
P-CAD 200x	.pcb (ASCII)
PADS (ASCII)	.asc
Solidworks PCB	.SWPcbDoc

Importing these PCB formats is explained in more detail in the [PCB Editor documentation](#).

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...）。

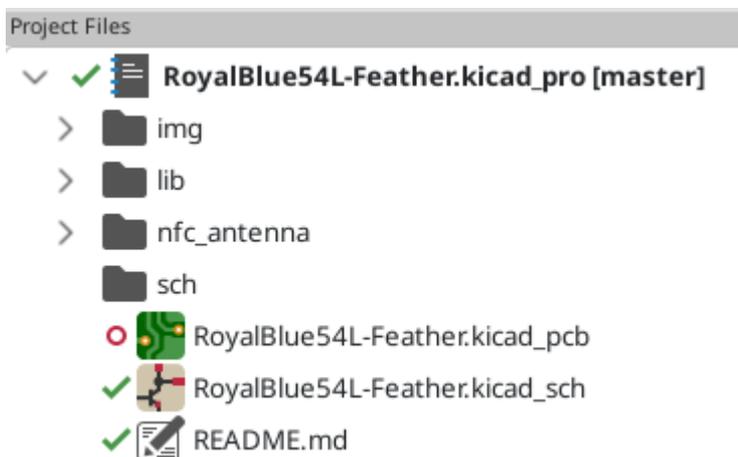
您也可以使用解压缩工具（**文件** → **解压缩项目...**）来解压缩工程。将项目解压缩到当前加载的工程目录中时，工程将自动重新加载，以反映归档版本中的任何更改。

存档工具会将工程文件夹中的以下文件保存到存档中：

*.kicad_pr1, *.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 集成

KiCad 工程管理器与 Git 版本控制工具集成，用于跟踪工程中的更改。它可以与现有的本地 Git 仓库配合使用，从远程仓库克隆工程，或在现有工程中创建新仓库。您可以使用该工具提交工程中的更改，从远程仓库推送和拉取代码，以及切换分支。



如果您打开一个已经使用 Git 进行版本控制的工程，即该工程属于现有的 Git 仓库，您可以直接使用 KiCad 的版本控制功能来跟踪工程中的更改，无需进行额外配置。当前活动分支会显示在工程名称旁边，而工程中每个文件的版本控制状态会在工程文件工作区中以图形方式显示。例如，图标  表示文件未更改， 表示文件有未提交的更改，而  表示文件未被跟踪。如果工程不属于 Git 仓库，则不会显示任何图标。

NOTE

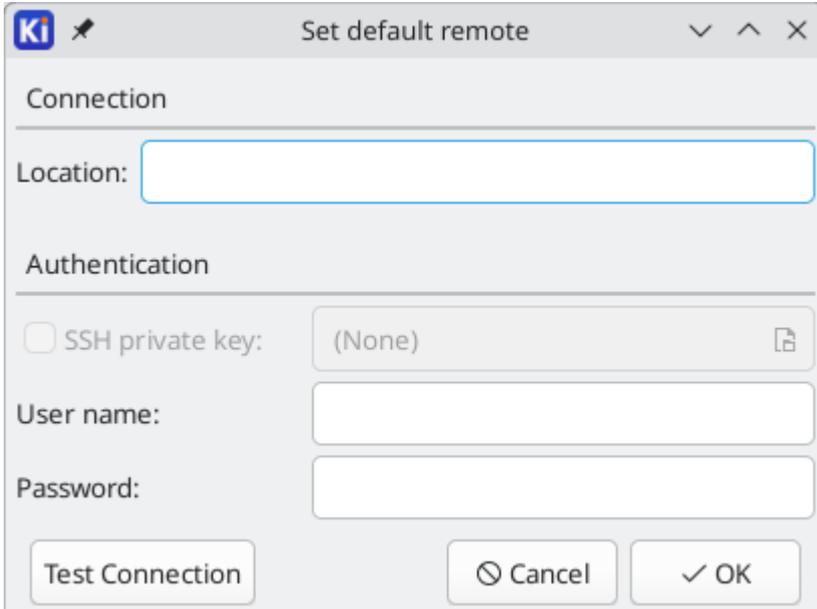
您可以在首选项对话框的 **版本控制** 页面禁用 KiCad 的 Git 集成功能。偏好设置还允许你配置远程更新间隔和默认提交作者信息。

将版本控制添加到现有工程

如果现有工程尚未处于版本控制之下，可以右键单击工程文件工作区中的某个文件，然后单击**版本控制** → **添加工程到版本控制...**，在工程中初始化一个新的 Git 仓库。以这种方式初始化版本库时，必须配置远程。配置版本库需要以下信息：

- **Location:** The URL or file path to the remote. HTTPS, SSH, and local (file) connections are supported. The format of the URL is used to automatically determine the type of connection and set the authentication options accordingly.
- **SSH private key:** The SSH private keyfile to use for authentication (SSH connections only).
- **User name:** The user name to use for authentication. For SSH connections, the user name is often `git`.
- **Password:** The password to use for authentication (HTTPS connections only).
- **SSH key password:** The passphrase for your SSH key, if you have configured one (SSH connections only).

单击 **测试连接** 按钮，即可检查连接和身份验证。



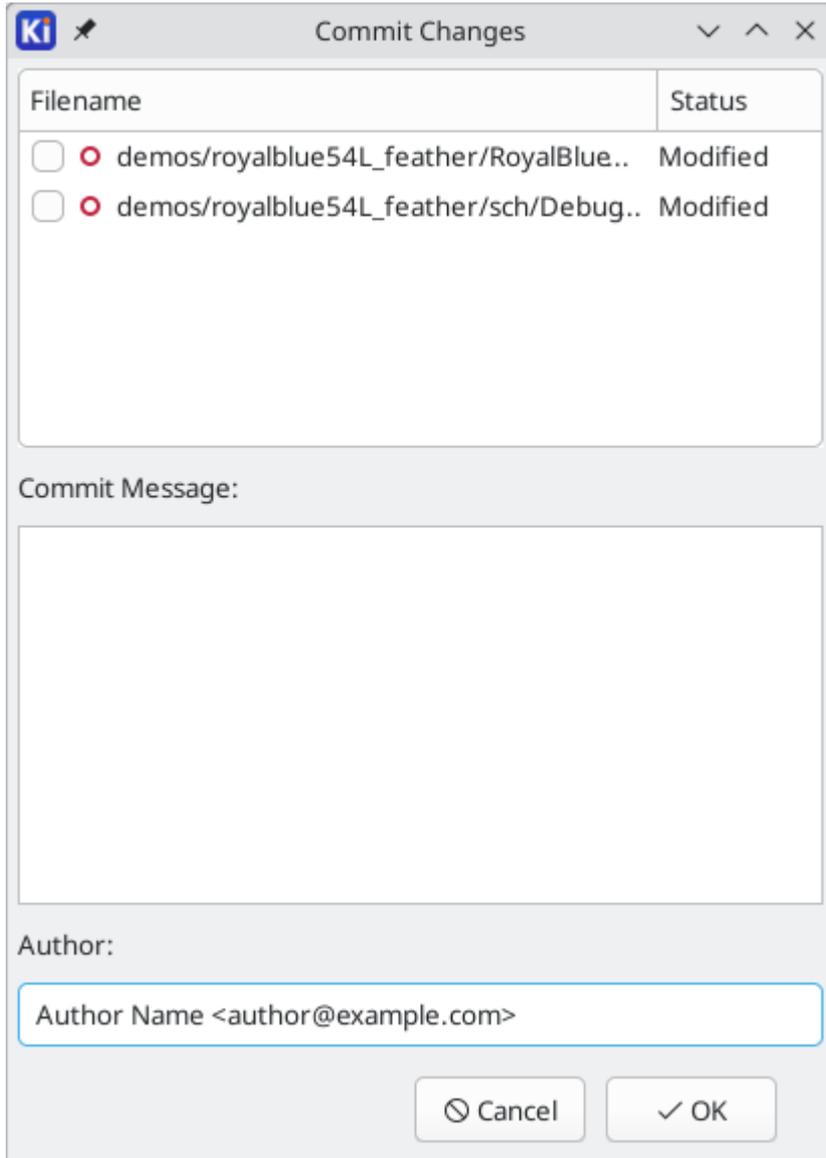
克隆现有仓库

要克隆现有仓库并打开克隆的工程，请使用 **文件** → **从仓库克隆工程...**。您可以使用 SSH 或 HTTPS 克隆远程仓库，也可以克隆本地仓库。克隆的配置设置与为现有工程配置新仓库和远程仓库的设置相同。

使用 Git 跟踪工程更改

当你做出要提交的更改时，你可以提交整个工程（右击 → **版本控制** → **提交工程...**）或某个特定文件（右击文件 → **版本控制** → **提交文件...**）。两种操作都会打开“提交更改”对话框，但“提交工程”操作会显示仓库中所有更改的文

件，而“提交文件”操作只会显示右键点击的文件。在“提交更改”对话框中，您可以选择要包含在提交中的更改文件，提供提交信息和作者，并提交更改。



要向远程推送更改，请在工程文件工作区中单击右键，然后选择 **版本控制** → **推送**。要从远程调用，请右击并选择 **版本控制** → **调用**。从 **版本控制** → **切换到分支** 菜单中选择所需的分支，即可切换分支。

Finally, you can disable version control tracking for the project by right clicking and selecting **Version Control** → **Disable Git Integration**. This disables KiCad's Git integration when editing the current project, but it does not remove the project's Git history or otherwise affect the actual Git repository on disk, so you can continue to version control the project with another Git tool. It also does not disable KiCad's Git integration with other projects. You can disable the Git integration for all projects in the **Version Control** page of the Preferences dialog.

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。

这些文件中包含大量重要设计信息，尤其是工程文件（.kicad_pro）、原理图文件（.kicad_sch）和电路板文件（.kicad_pcb）。其他文件也可能需要包含。在分发工程时，应始终包含这些文件。某些文件无需与工程一起分发，例如工程本地设置文件（.kicad_pr1）或 fp-info-cache 文件。无需分发的文件在下方表格中有所标注。

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.

Schematic editor files and folders

.kicad_sch	Schematic files, containing all symbol and connection information.
.kicad_sym	Schematic symbol library file, containing the symbol descriptions: graphic shape, pins, fields. This is a <i>packed</i> library, i.e. all of the symbols in the library are in a single file. This can also be an individual symbol in an <i>unpacked</i> library, where the library is a folder containing individual .kicad_sym files, and each file is a separate symbol. Unpacked library folders do not have a required extension, but .kicad_symdir is a common convention.
.kicad_symdir	Schematic symbol library folder. The folder itself is the library, and contains individual .kicad_sym symbol files. This is an <i>unpacked</i> alternative to a .kicad_sym <i>packed</i> library, where the library is contained in a single file. Note that .kicad_symdir is a convention for unpacked symbol libraries but is not required; any folder name can be used.
.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_pr1	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.
.kicad_blocks	Design block library folders. The folder itself is the library. Design blocks can contain both schematic and layout fragments.
.kicad_block	Design block folder for defining a reusable design. The folder is the design block, and contains a .kicad_sch file defining the design block's schematic, a .kicad_pcb file defining the design block's layout, and a .json file defining the design block's metadata.
.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 files, for fabrication.
.drl	Drill files (Excellon format), for fabrication.
.pos	Position files (ASCII format), for automatic insertion machines.
.rpt	Report files (ASCII format), for documentation.
.ps	Plot files (Postscript), for documentation.
.pdf	Plot files (PDF format), for documentation.
.svg	Plot files (SVG format), for documentation.
.dxf	Plot files (DXF format), for documentation.

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，因此您可以单独备份或发送这些文件，无需担心。一些重要的设计信息存储在工程文件（`.kicad_pro`）中，因此，如果您要发送完整的设计，请确保将该文件包括在内。

某些文件，例如工程本地设置文件（`.kicad_pr1`）和 `fp-info-cache` 文件，无需随工程一起发送。如果您使用 Git 等版本控制系统来管理 KiCad 工程，可以将这些文件添加到忽略文件列表中，以便它们不被跟踪。

Paths and libraries configuration

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 will 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 `KICAD10_FOOTPRINT_DIR` path variable, would be defined as `${KICAD10_FOOTPRINT_DIR}/connect.pretty`.

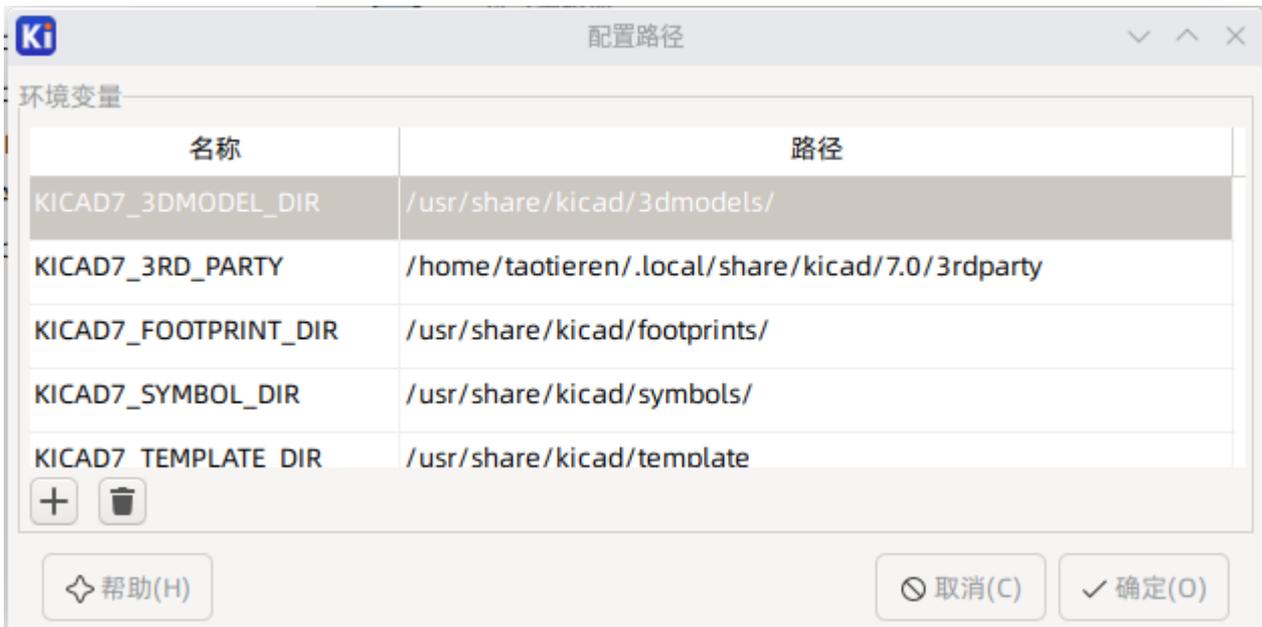
通过 **偏好设置** → **配置路径...** 菜单，可以为一些内置的 KiCad 路径变量定义路径，并根据需要添加自己的路径变量来定义个人路径。

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, `${KICAD9_FOOTPRINT_DIR}` will automatically resolve to the value of `${KICAD10_FOOTPRINT_DIR}` if there is no `KICAD9_FOOTPRINT_DIR` variable defined.

KiCad 路径变量

<code>KICAD10_3DMODEL_DIR</code>	Base path of KiCad's standard 3D footprint model library files.
<code>KICAD10_3RD_PARTY</code>	Location for plugins, libraries, and color themes installed by the Plugin and Content Manager .
<code>KICAD10_FOOTPRINT_DIR</code>	Base path of KiCad's standard footprint library files.
<code>KICAD10_SYMBOL_DIR</code>	Base path of KiCad's standard symbol library files.
<code>KICAD10_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 [KiCad 的用户配置文件](#) 章节。

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 [用户配置目录章节](#) 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。

作业集

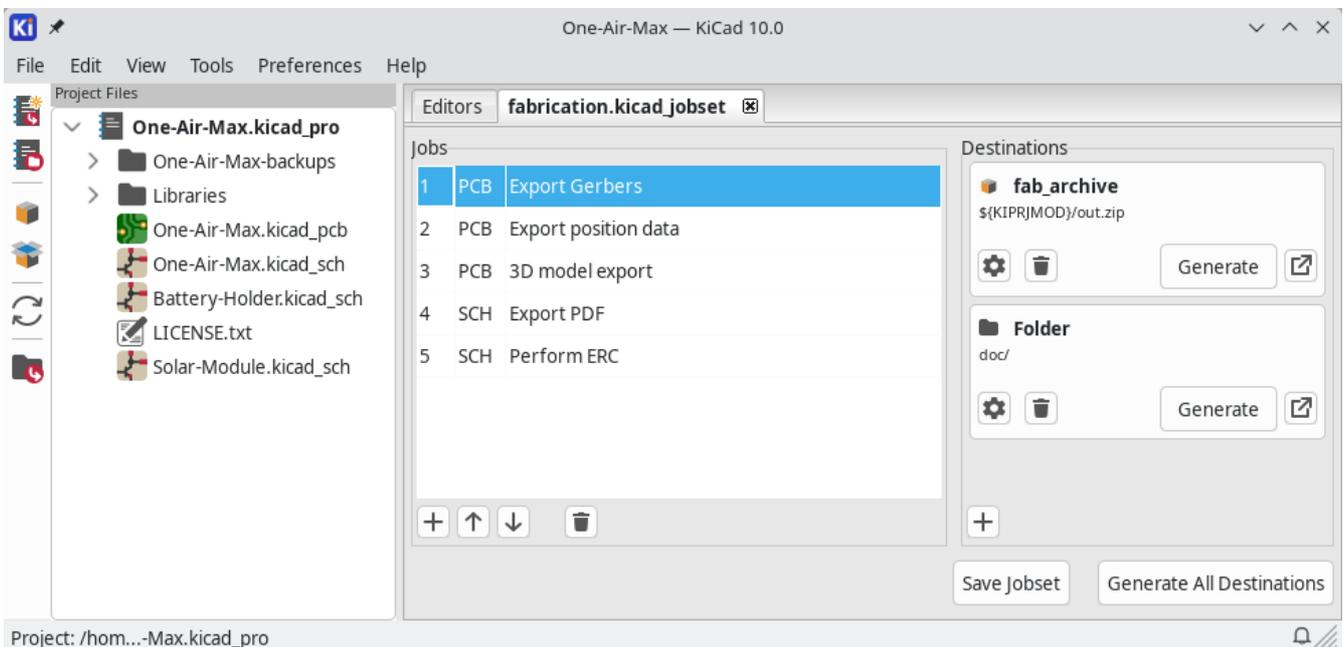
KiCad 可让您配置输出作业列表，只需单击一下即可生成所有输出作业。输出作业列表及其保存目的地称为 **作业集**。例如，一个作业集可能包含生成 Gerber 文件、装配数据、BOM、原理图和 PCB 的 PDF 图的作业，同时还运行 ERC 和 DRC 检查，并将所有输出保存到压缩文档中。可用工作的完整列表 [如下章节](#)。

作业集中的每个 **作业** 都定义了单一类型的生成输出，如 BOM 或一组 Gerbers。作业的配置方式与从原理图或电路板编辑器手动生成输出的方式相同。每个作业的配置都存储在作业集中，并在以后加载作业集时被记住。作业是单独配置的，因此如果在一个作业集中多次包含相同类型的作业，每个作业都将有自己独立的配置。例如，这样就可以生成彩色和黑白 PDF 输出。

除作业外，作业集还包含 **目标**，它定义了要运行的作业列表以及如何存储其输出。作业集目标可以简单地将所选作业的输出文件存储在指定位置，也可以将输出文件添加到压缩档案中。每个作业集目标都可以从作业集的完整作业列表中选择不同的作业子集。您可以单独运行每个作业集目标，也可以一次性运行所有作业集目标。例如，您可以设置一个作业集目标，生成电路板和原理图的 PDF 文件并将其复制到外部位置，而另一个目标则生成制作文件并将其压缩到 zip 压缩包中发送给电路板制造商。

工程可以有多个作业集，每个作业集定义不同的作业列表和输出配置。每个作业集都存储在一个 `.kicad_jobset` 文件中，该文件可以特定于单个工程，也可以在工程之间复制，甚至可以存储在一个中心位置并在工程之间共享。

要使用作业集，首先要在 KiCad 工程管理器中创建一个新的作业集文件（**文件** → **新建作业集文件...**），并为其选择名称和位置。或者，您也可以使用 **文件** → **打开作业集文件...** 打开现有的作业集文件。存储在工程目录中的作业集文件被视为工程的一部分，并显示在工程文件工作区中。双击工程文件工作区中的作业集文件，即可打开该文件。



创建或打开作业集后，作业集就会显示在工程管理器的新标签页中。作业列表显示在中间，作业集目标列表显示在右侧。新作业集不包含任何作业，但会自动创建一个目标，将输出保存到文件夹中。对作业集进行更改时，可单击 **保存作业集** 按钮保存更改。

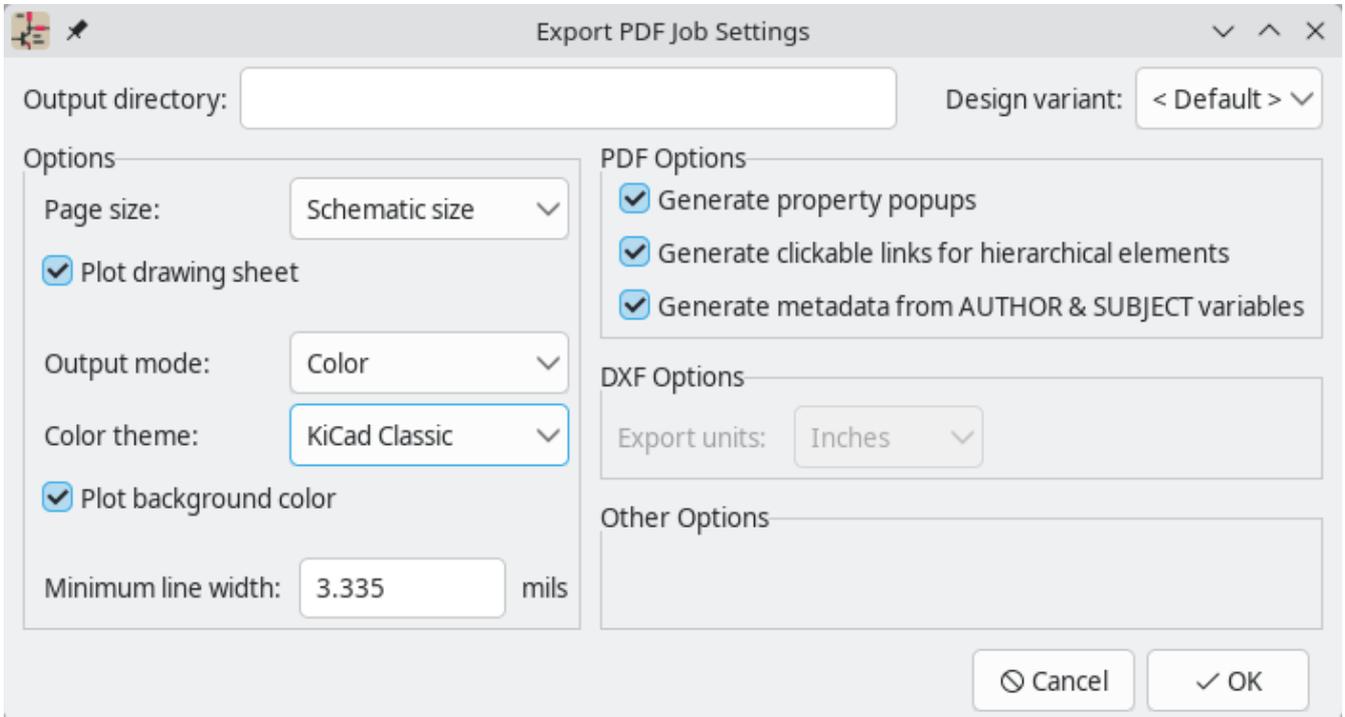
定义作业

要添加新任务，请单击任务列表下的 **+** 按钮。在出现的添加新任务对话框中，选择所需的任务类型。您可以在底部的 **筛选器** 文本框中输入内容，过滤列表中显示的工作类型。

选择作业并按下 **确定** 后，将出现该类型输出的设置对话框。每个作业设置对话框提供的选项与从原理图或电路板编辑器手动生成该类型输出时相同。

NOTE

工作设置中指定的输出文件名和路径是相对于 [作业集目标章节](#) 文件夹或存档根目录的。您可以使用某些 [文本变量](#)，如 `${PROJECTNAME}`、`${CURRENT_DATE}` 和 [工程文本变量](#)。



接受作业设置对话框后，作业将被添加到作业列表中，您可以选择更改新作业的默认描述。要稍后更改作业描述或设置，请右键单击列表中的作业并选择 **编辑作业描述** 或 **编辑作业设置...**。双击一项作业也可编辑其设置。要删除作业，请选择作业并单击 **删除** 按钮。要对列表重新排序，请选择一项作业，然后使用 **上** 或 **下** 按钮将其上移或下移。

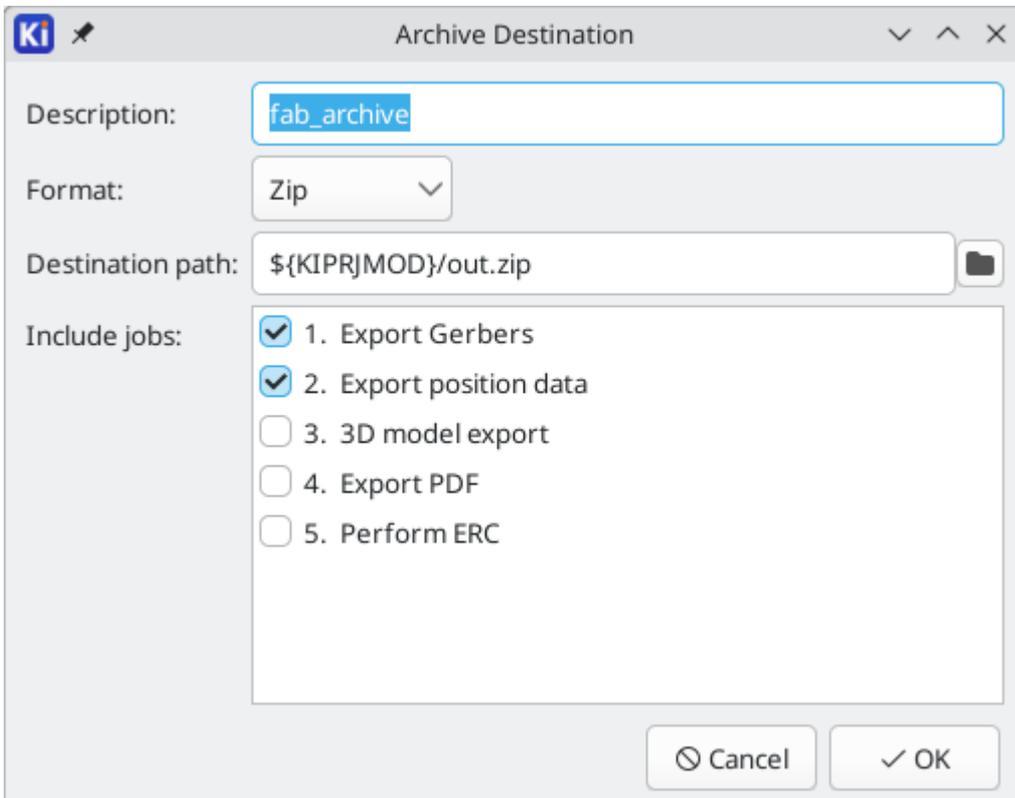
定义工作集目标

在添加作业集目标之前，无法从作业集生成任何输出。创建作业集时会自动创建一个目标，但您可以根据需要添加任意多个目标。

要添加作业集目标，请单击目的地列表下的 **+** 按钮。出现添加新目标对话框后，选择目标类型：

- **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.

选择输出类型后，将出现“目标”选项对话框。

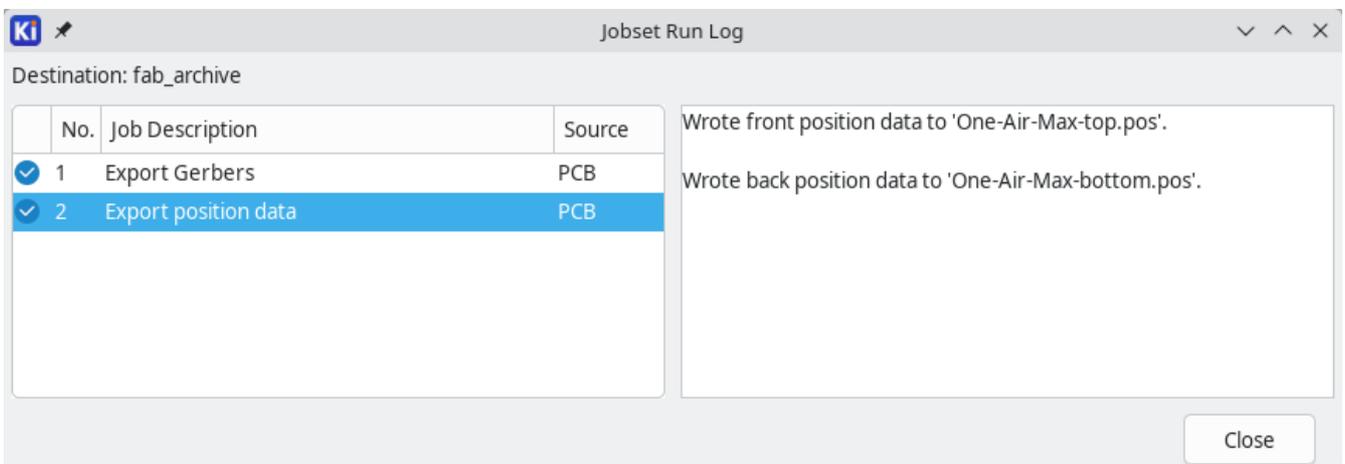


在这里，您可以选择哪些作业将作为此作业集目标的一部分运行，以及用于存储这些作业的文件夹或存档名称。默认情况下，所有作业都将启用。您还可以为目标设置一个描述，以便显示在“目标”列表中。输出路径控制作业生成的文件的保存位置。此处的路径可以是工程目录的绝对路径或相对路径，可以使用 [路径变量章节](#) 或某些文本变量（`${PROJECTNAME}`、`${CURRENT_DATE}` 和 [工程文本变量](#)）。作业配置中定义的文件名是相对于作业集目标目录或归档根目录而言的。

单击对话框中的 **确定** 后，新的作业集目标就会添加到目的地列表中。您可以单击现有作业集目标的  按钮修改该目标，或单击其  按钮删除该目标。

配置好作业和目标后，您可以单击所需目标的 **生成** 按钮，生成一组单独的输出。单击 **生成所有目标** 按钮，可以同时运行所有目标。

如果作业集目标成功运行并生成其输出，则会显示一个蓝色检查符，表示上次运行成功。如果作业集目标未成功完成，则会显示红色感叹号，表示运行不成功。单击成功/失败指示符将显示作业集运行日志对话框，其中显示作业集目标中每个作业的状态。单击特定作业将显示该作业的日志输出（如果有的话）。



After a jobset destination runs successfully, you can click the destination's **Open Output** button to open a file browser in the destination location. This button is disabled if the jobset destination has not run or hasn't successfully finished.

Jobset destination details

When jobs run, output files are initially generated in a temporary folder. After all jobs in a jobset destination are completed, the output files are moved from the temporary folder to the folder or archive specified by the jobset destination. None of the outputs are moved until all of the jobs finish. Therefore, if a job needs to access files that are generated by another job in the same jobset destination, these files will be located in the temporary folder, not the final jobset destination.

KiCad defines the `#{JOBSET_OUTPUT_WORK_PATH}` environment variable while jobs are running and sets it to the temporary path for the current jobset destination. You can use this environment variable if a job needs to be aware of the temporary folder's actual location. For example, if an Execute Command job runs a script that renames the outputs of another job, the script needs to know the temporary location of the outputs being renamed, before they are moved to the jobset destination. `#{JOBSET_OUTPUT_WORK_PATH}` provides this location.

可用的作业类型

可提供以下类型的作业：

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, STL, STPZ, U3D, or PDF.
PCB: Export Board Statistics	Generates a Board Statistics report in either JSON or text report format.
PCB: Export Drill Data	Exports a drill file from the board.
PCB: Export DXF	Exports the board design to a DXF file .
PCB: Export Gencad	Exports the board design in GenCAD format .
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 PostScript	Exports the board design to a PostScript file .
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 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	<p>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. The command is executed in the platform's shell (<code>/bin/sh -c</code> on Linux and macOS, <code>cmd.exe \c</code> on Windows), so platform-shell-specific features can be used in the command, such as globbing and pipes.</p> <p>Note: Job output files are generated in a temporary folder, then moved to the location specified by the jobset destination after all jobs for that destination are executed. In other words, when an Execute Command job runs, the output from other jobs in the same jobset destination are still in a temporary location and not yet in the ultimate destination folder or archive. You can use the <code> \${JOBSET_OUTPUT_WORK_PATH}</code> environment variable if you need to refer to the temporary location in an Execute Command job (for example, in a script that renames files generated by another job).</p>

工程模板

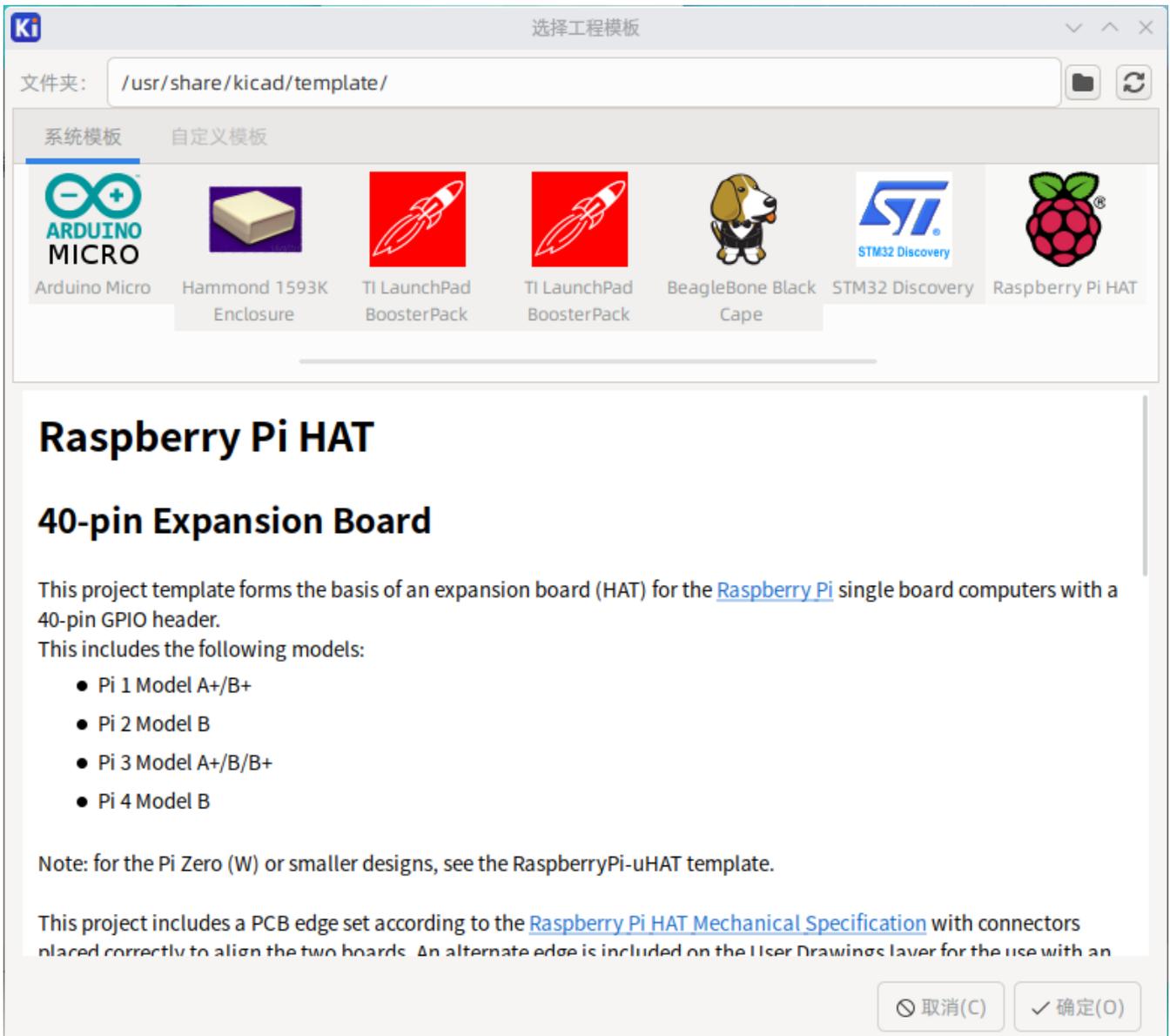
Using a project template facilitates setting up a new project with predefined settings. Templates may contain pre-defined sheet sizes, drawing sheets, board outlines, connector positions, schematic elements, design rules, logos, etc. Complete schematics and/or PCBs used as seed files for the new project may even be included.

使用模板

Creating a new project (**File** → **New Project...**) opens the Project Template Selector dialog. By default, all installed templates are shown in the list of templates, but you can limit the list to system templates or user templates using the dropdown menu above the list.

- *System templates* are installed with KiCad and include templates for designs in some common form factors.
- *User templates* are any templates created by you. The default, blank KiCad template is also considered a user template.

When you click on a template, information about the template is shown in the right panel.



A further click on the **OK** button creates a new project based on the selected template. You can pick the name of the new project and where to create it. The template files will be copied to the new project location and renamed to reflect the new project's name.

模板位置

When you create a new project, the Project Template Selector dialog shows templates from two locations:

- System Templates are templates in the path defined by the `KICAD10_TEMPLATE_DIR` path variable. These templates are installed with KiCad's default libraries and generally should not be modified by users.
- User templates are templates in the path defined by the `KICAD_USER_TEMPLATE_DIR` path variable. When you create your own project templates, store them in this location so you can use them when you create a new project.

To change the location of the user template folder, edit the `KICAD_USER_TEMPLATE_DIR` path variable in your [paths configuration](#).

Template contents

A KiCad template is 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. This means that a KiCad template folder is simply a KiCad project folder with an additional `meta` subdirectory.

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:

- 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。
- The `meta` directory is not copied.

NOTE

Other than the two exceptions above, all files in the template are copied, even if they are not KiCad design files. This means you can use templates to create projects that are already set up with non-KiCad files you need to include, such as documentation templates, license files, logos, `.gitignore` files, etc.

Template metadata

A template's `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.

`<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>树莓派 (Raspberry Pi) - 扩展板开发模板</TITLE>
</HEAD>
<BODY LANG="fr-FR" DIR="LTR">
<P>本设计模板用于开发树莓派扩展电路板, 兼容
<A HREF="http://www.raspberrypi.org/" TARGET="blank">树莓派 $25 ARM 开发板</A> <BR><BR>基础设计
包含与树莓派 PCB 等尺寸的板框定义, 并通过精确定位连接器实现双板对齐。
树莓派开发板的所有 I/O 接口均通过 0.1 英寸 (2.54mm) 扩展排针连接至本设计。
<BR><BR>电路板外形如下图所示 :
</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 开发团队</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

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 dotfiles. Files and directories containing the template directory name will be renamed with the new project file name. Files and directories that don't contain the template name keep their original name.

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

模板不需要包含完整的工程。如果缺少必需的工程文件, KiCad 将使用默认的创建工程行为来生成该文件 :

Files in template example directory	Files created in newproject directory
example.kicad_sch	newproject.kicad_sch
first-example.kicad_sch	first-newproject.kicad_sch
first-example.kicad_pcb	first-newproject.kicad_pcb
second-example.kicad_sch	second-newproject.kicad_sch
second-example.kicad_pcb	second-newproject.kicad_pcb
	newproject.kicad_pro (default)
	newproject.kicad_pcb (default)

作为模板名称重命名规则的一个例外，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 将根据该工程文件的名称进行重命名：

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

WARNING | 不建议创建包含多个工程文件的模板。

Creating new templates

Because a KiCad template is just a Kicad project with an additional metadata folder, the easiest way to create a new template is to start from a project. If you save a project in your [user template folder](#) and add a `meta` folder to it, you can use it as a template.

TIP

To find out where your user template folder is, check the value of the `KICAD_USER_TEMPLATE_DIR` path variable in the Configure Paths dialog (**Preferences** → **Configure Paths...**).

If you are creating a template from scratch, you can create a new project in the user template folder, then edit the project until you have created your desired template. This new project will be your template, so when you create it you should choose the name based on what you want the template to be called.

If you have an existing project that you want to turn into a template, you can save a copy of that project into your user template folder (**File** → **Save As...** from the Project Manager). The name you choose for the re-saved project will be the template name. After saving in the template folder, you can edit the template project so that it exactly matches your desired template.

In either case, templates are required to have a [metadata directory](#), so you must add a `meta` folder inside the template folder. The metadata folder must also contain an `info.html` file, which is intended to contain a description of the template. You can copy the [sample info.html](#) [above](#) as a starting point.

In general, you will also want to remove any files and folders that you don't want to include each time you use the template. Some examples of things to remove include the `*-backups` folder, the `.history` snapshot folder, any unneeded version control files (`.git`, etc.), and anything else you don't want to include in new projects.

插件和内容管理器

KiCad 提供了一个插件和内容管理器，允许您浏览、安装和管理由其他用户和组织提交的软件包。这些软件包可以是提供特定功能的插件、用于从特定制造商订购 PCB 的工具、符号和封装库，或是新的编辑器配色方案。您可以通过从 KiCad 工程管理器主窗口启动来访问 PCM。

PCM 从互联网上的软件包仓库下载软件包。每个仓库都是由个人或组织管理的软件包集合。默认情况下，PCM 使用由 KiCad 组织管理的单一仓库。用户和组织可以自由创建自己的仓库，其他用户随后可以将这些仓库添加为自己 KiCad 安装中的额外仓库。第三方仓库可以是公开的，也可以是私有的。

NOTE

软件包可能会安装运行在您计算机上的代码。这些软件包并非由 KiCad 开发人员开发。KiCad 组织不对通过 PCM 安装的软件包的质量或安全性做出任何保证。请确保您仅安装您信任的软件包。

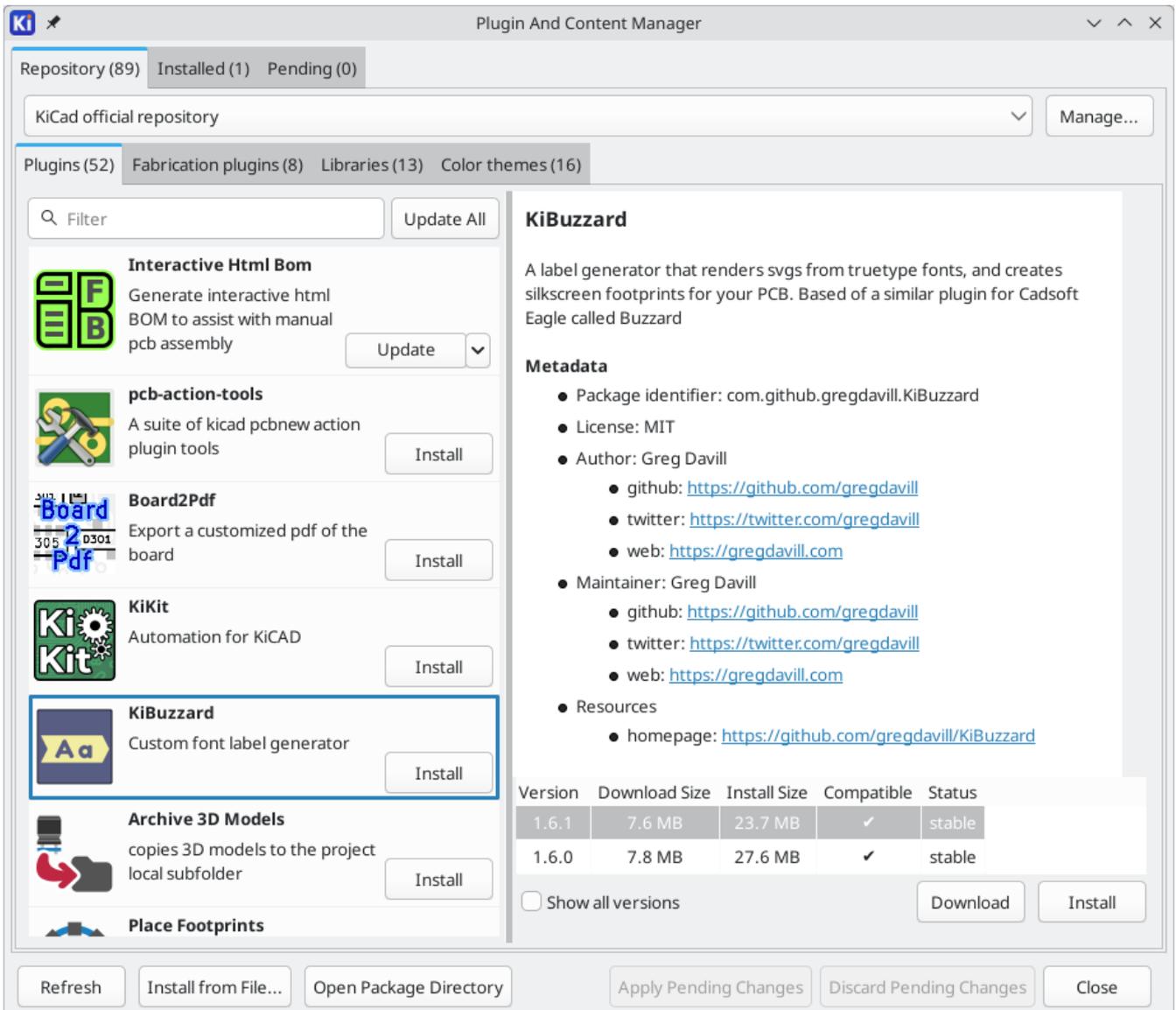
NOTE

如果您对某个软件包有反馈意见，请将其提交给该软件包的开发者，而不是 KiCad 团队。每个软件包的联系信息显示在软件包的描述中。

要分享您开发的软件包，您可以将其提交到 KiCad 官方仓库。如果被接受，其他用户在浏览 KiCad 仓库时就能看到并使用它。您也可以创建自己的仓库，或将软件包提交到由他人或团队管理的仓库，但在这种情况下，除非用户在自己的安装中配置了该仓库，否则他们将无法看到您的软件包。有关创建自己的软件包和仓库的更多信息，请参见 [下文章节](#)。

浏览软件包

仓库中的所有包都显示在 PCM 的 **仓库** 选项卡中。您可以通过选择选项卡顶部的下拉列表中的仓库来选择要浏览的仓库。默认情况下，仅启用 KiCad 官方仓库。



包被分为四个类别，每个类别在 **仓库** 标签下都有其专属的标签页：

- **Plugins** are additional tools that can be launched from the PCB Editor. Plugins can have many purposes, for example modifying a board design or generating specific outputs. Footprint wizards can also be distributed as plugins.
- **Fabrication plugins** are a sub-category of plugins for ordering your PCBs from specific fabricators. These plugins may be a convenient way to order from a manufacturer, but they are typically not required; you can usually provide manufacturers with normal fabrication outputs instead. Consult with your manufacturer to find out the best way to order from them.
- **Libraries** contain symbols, footprints, and/or 3D models. By default, libraries installed by a library package are automatically added to the appropriate symbol and/or footprint library tables when the package is installed, and removed from the table when the package is uninstalled. Libraries installed by a package have configurable library name prefix (`PCM_` by default). These settings are configurable in the **Packages and Updates** section of Preferences.
- **Color themes** are color themes for the Schematic, Symbol, Board, and Footprint editors. You can select an installed theme in the **Colors** section of the Preferences for each editor.

窗口左侧的列表显示了每个类别中的软件包。您可以通过在软件包列表顶部的筛选框中输入内容来过滤软件包列表。按下 **刷新** 按钮可从在线仓库重新加载软件包列表。

当您在列表中选择一个软件包时，右侧会显示该软件包的相关信息。其中包括软件包的描述、许可证信息以及开发者的联系方式，还提供了报告错误和提交其他反馈的途径。

软件包信息还包括一个表格，列出了当前可用的软件包版本。对于每个版本，表格中显示了软件包的大小、其状态以及是否与您使用的 KiCad 版本兼容。

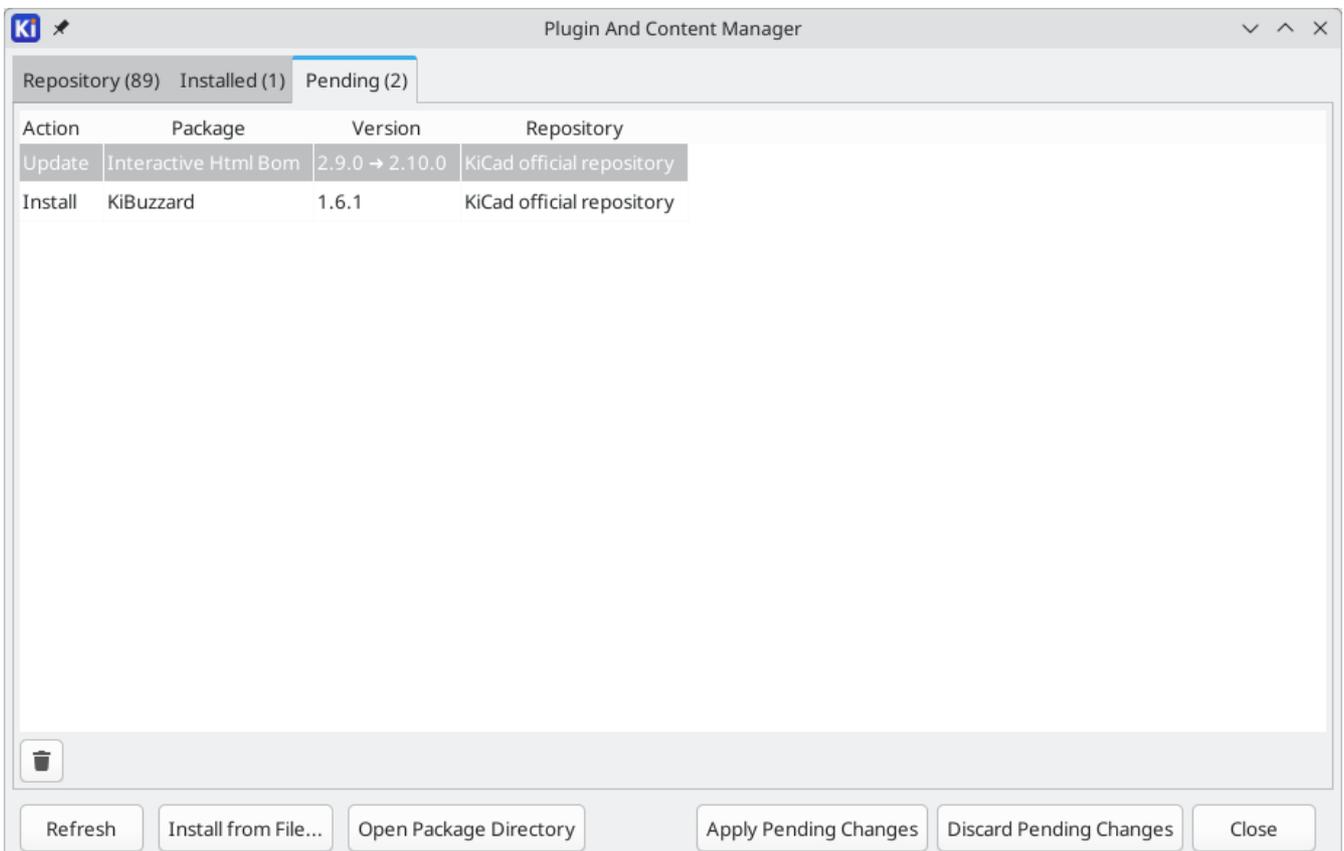
一个软件包的状态可以是以下几种之一，具体由软件包的开发者标明：

- 稳定版：该软件包版本适合一般使用。
- 测试版：该软件包版本处于测试阶段；用户应谨慎使用，并将遇到的任何问题报告给软件包的开发者。
- 开发版：该软件包版本处于开发阶段；用户不应期望其完全正常运行。
- 已弃用：该软件包不再维护。

默认情况下，版本表仅显示与您当前使用的 KiCad 版本兼容的软件包版本。您可以通过勾选 **显示所有版本** 复选框，查看软件包的所有版本，包括那些与您的 KiCad 版本不兼容的版本。如果某个软件包有多个版本可供选择，您可以自行决定安装哪个版本。

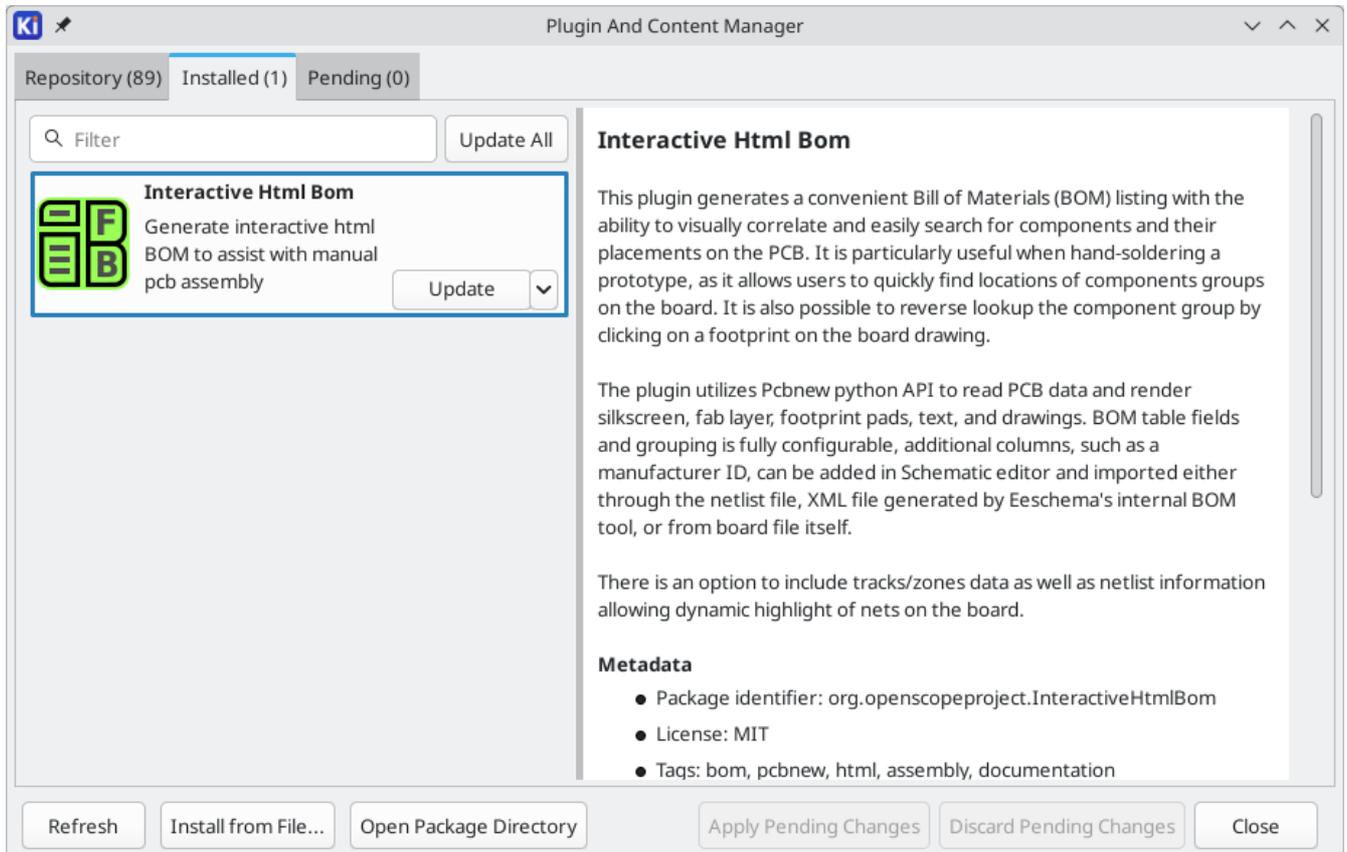
安装软件包

当您决定安装一个软件包时，请在软件包的描述页面点击 **安装** 按钮。如果有多个版本的软件包可供选择，请在安装前在版本表中选择所需的版本。您还可以在软件包列表中点击 **安装** 按钮来安装最新版本。这并不会立即安装软件包；相反，软件包会被排队等待稍后安装，并显示在 **待处理** 标签页中。直到您点击 **应用待处理更改** 按钮时，软件包才会被安装，该按钮会一次性安装所有待处理的软件包。

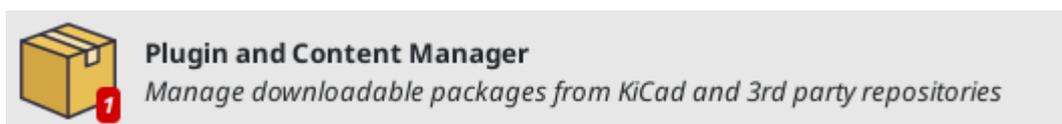


要从待安装列表中移除单个软件包，请在 **待处理** 选项卡中选择该软件包，然后按下  按钮。若要取消所有待安装软件包的安装，请点击 **放弃待处理更改** 按钮。

一旦安装了一个包，它就会列在 **已安装** 标签页中。在此标签页中，您可以查看已安装包的列表，并更新或卸载其中的任何一个。



如果有任何软件包有新版本可用，工程管理器中的 PCM 图标会显示一个指示器，显示可用更新的数量。



要将某个包更新到不同版本，请在其版本表中选择新版本，然后点击 **更新** 按钮。您还可以在已安装包列表中，点击其 **卸载** 按钮旁边的下拉菜单中的 **更新**，将包更新到最新版本。要更新所有已安装的包，请点击已安装包列表顶部的 **全部更新** 按钮。如果在其下拉菜单中选择了 **锁定包**，则该包将不会被更新。

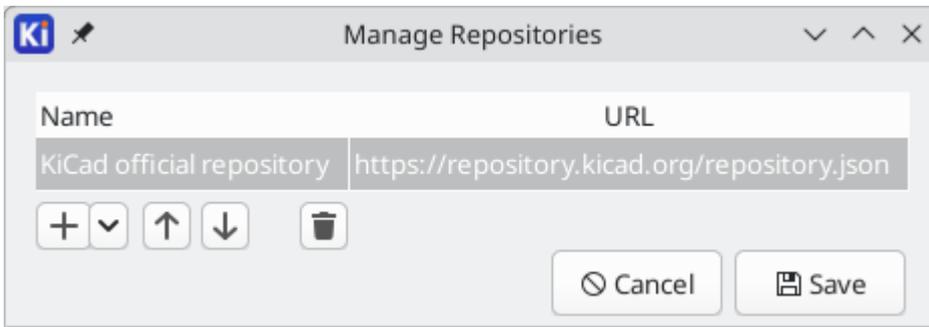
要卸载一个软件包，请在已安装软件包列表中或软件包的描述页面上点击其 **卸载** 按钮。

与安装过程类似，当您点击 **更新** 或 **卸载** 按钮时，软件包并不会立即更新或卸载。相反，该操作会被排入 **待处理** 标签页中，直到您单独应用每个更改或点击 **应用待处理更改** 按钮。

最后，您可以通过点击包描述底部的 **下载** 按钮并选择保存包的位置来下载包而不安装它。这使您可以在安装之前检查包中的文件。要安装已下载但尚未安装的包，请点击 **从文件安装...** 按钮并选择该包。**打开包目录** 按钮将在 KiCad 安装包的文件夹中打开文件浏览器。

Managing repositories

默认情况下，PCM 仅使用 KiCad 官方仓库。您可以通过点击 **仓库** 选项卡顶部的 **管理...** 按钮来添加第三方仓库或删除现有仓库。



要添加仓库，请点击 **+** 按钮并指定仓库的完整 URL。要移除仓库，请选中它并按下 **🗑** 按钮。使用 **↑** 和 **↓** 按钮可以重新排列列表中的仓库顺序。

如果您移除了默认的 KiCad 官方仓库，可以通过点击图片旁边的下拉菜单中的 **添加默认仓库** 按钮轻松重新添加它 **+**。

创建软件包和仓库

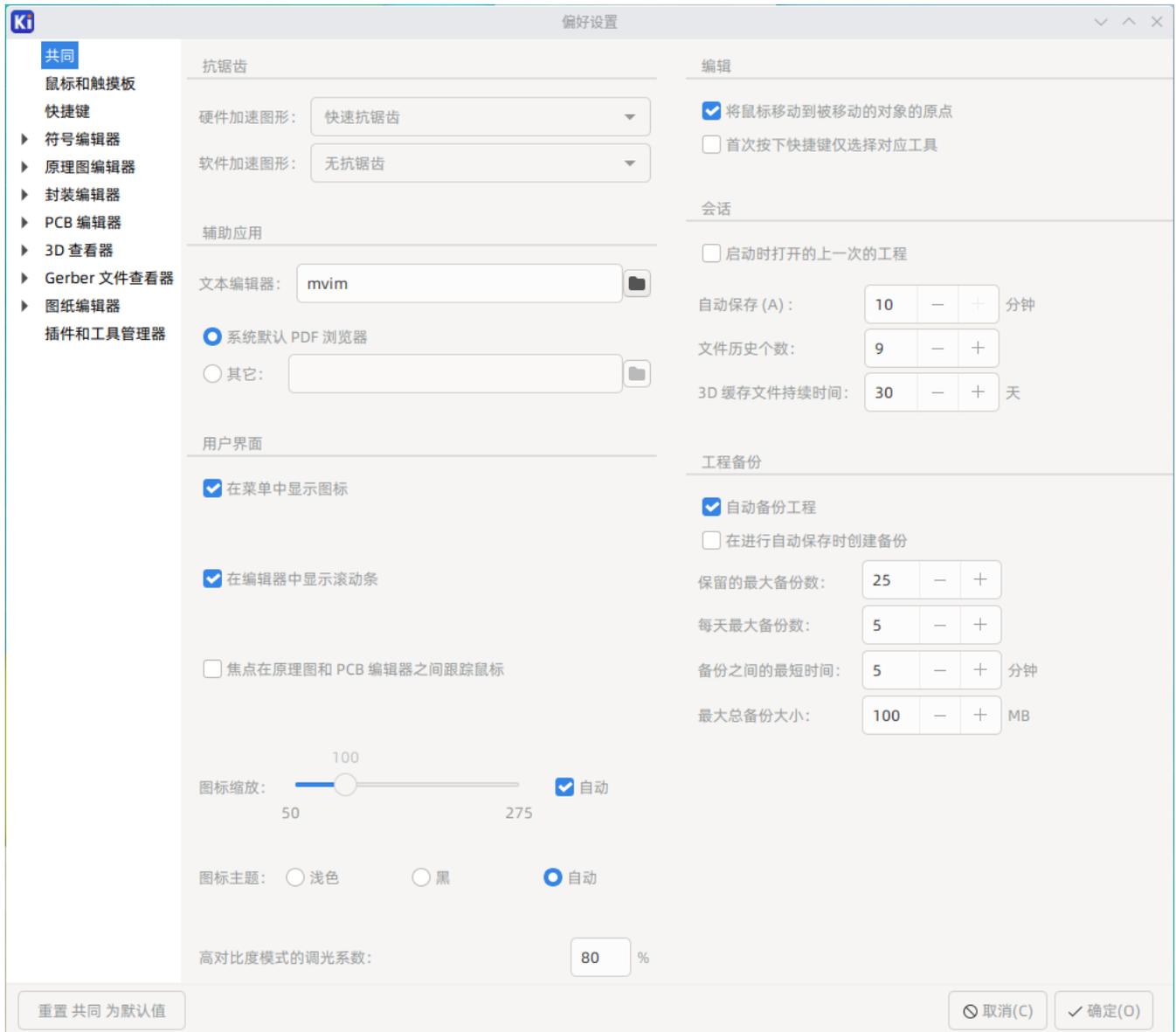
要为 PCM 创建软件包，请按照以下链接中的说明操作：<https://dev-docs.kicad.org/en/addons/index.html>。这些说明解释了如何为任何仓库创建 PCM 软件包，包括但不限于 KiCad 官方仓库。它们还解释了 KiCad 仓库中包含的软件包的规则，以及如何向该仓库提交软件包。第三方仓库使用相同的软件包格式，但可能有不同的提交规则和流程。

要创建您自己的仓库，请按照在 <https://go.kicad.org/pcm/schemas/v1> 发布的模式发布一个仓库 JSON 文件。

KiCad preferences

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 antialiasing: 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.

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).

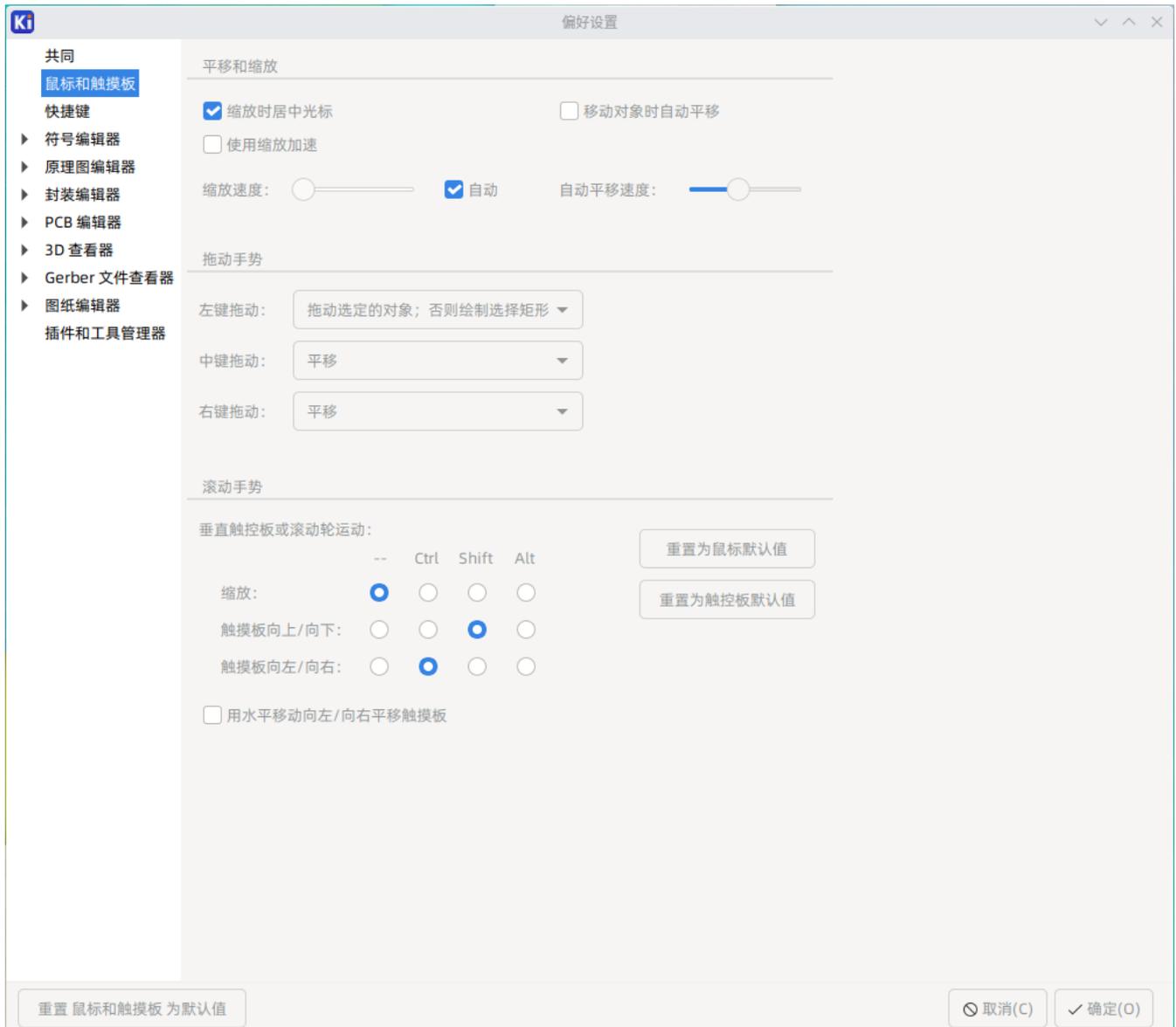
Maximum backups to keep: When creating a new backup, the oldest backup file will be deleted to keep the total number of backup files below this limit.

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

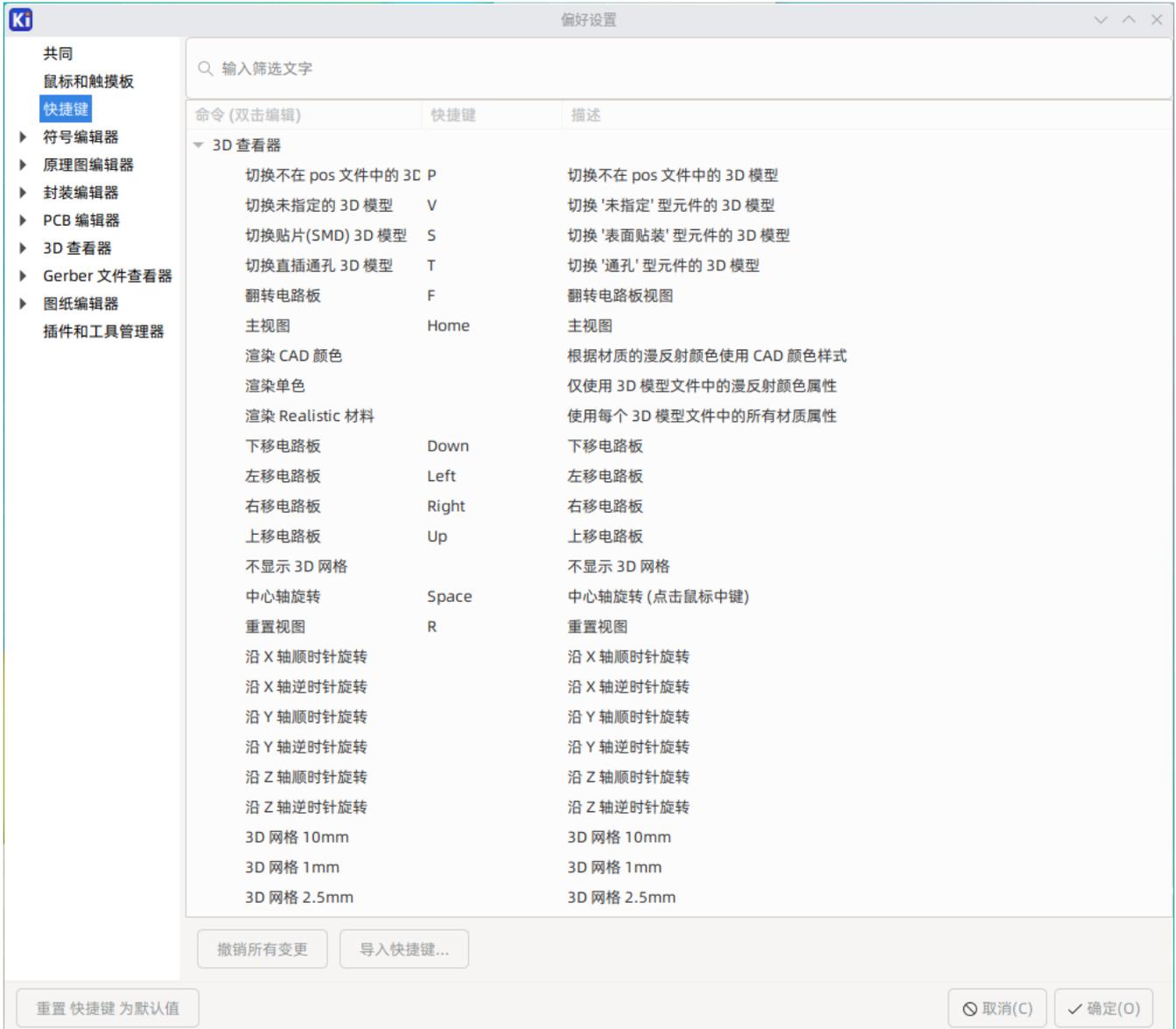
on whether or not any objects are already selected in the editing canvas.

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](#) 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。

操作参考

以下是 KiCad 工程管理器中每个可用 **操作** 的列表：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