

KiCad

The KiCad Team

Table of Contents

简介	2
系统要求	2
KiCad 文件和文件	2
安装和升级 KiCad	5
导入	5
从早期版本迁移文件	5
使用 KiCad 工程管理器	7
工程管理器窗口	7
工程	7
界面工具条	8
新建一个工程	8
从其他 EDA 工具导入工程	9
KiCad 配置	10
共同偏好	10
鼠标和触摸板偏好	12
快捷	13
路径配置	13
配置	15
工程模板	16
使用模板	16
模板位置 :	17
新建模板	17
插件和内容管理器	21

参考手册

版权

本文件的版权 © 2010-2021 由下列贡献者拥有。您可以根据 GNU 通用公共许可 (<http://www.gnu.org/licenses/gpl.html>) 第 3 版或更高版本, 或知识共享署名许可 (<http://creativecommons.org/licenses/by/3.0/>) 第 3.0 版或更高版本的条款发布它和/或修改它。

本指南中的所有商标均属于其合法所有者。

贡献者

Jean-Pierre Charras, Fabrizio Tappero, Jon Evans.

翻译人

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

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

反馈

KiCad 项目欢迎与本文件或其文档相关的反馈、报告和建议。关于如何提交反馈或报告的更多信息, 请参考 <https://www.kicad.org/help/report-an-issue/> 的说明

简介

KiCad 是一款用于设计电子电路原理图和印刷电路板（PCB）的开源软件套件。KiCad 支持混合设计工作流程，其中原理图和相关的 PCB 一起设计，也支持特殊用途的独立工作流程。KiCad 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、用于制造文件的 Gerber 工具和用于电路仿真的集成 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 个层的 PCB，适合设计各种复杂的设计。KiCad 是由世界各地的电子和电气工程爱好者组成的志愿者团队开发的，其使命是设计适合业余爱好者的免费和开源的电子设计软件。

文件的最新版本可在 <https://docs.kicad.org>。

系统要求

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

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

KiCad 文件和文件

KiCad 设计并使用具有以下特定文件扩展名（和文件名的文件）进行原理图和电路板设计

工程管理器文件：

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

原理图元件文件：

*.kicad_sch	包含所有信息和元件本身的原理图文件。
*.kicad_sym	原理图符号文件，包含元件描述：引脚形状、引脚、字段。
*.sch	旧版 (KiCad 5.x 和更早版本) 原理图文件。可以读取，并将在写入时写入 *.kicad_sch 文件。
*.lib	旧版 (KiCad 5.x 和更早版本) 原理图文件。可以读取，但不能写入。
*.dcm	旧版 (KiCad 5.x 和更早版本) 原理图文档。可以读取，但不能写入。
*_cache.lib	旧版 (KiCad 5.x 和更早版本) 原理图元件缓存文件。正确加载旧版原理图 (.sch) 文件所需。
sym-lib-table	符号列表 (符号表)：原理图元件中可用的符号列表。

板元件文件和文件

*.kicad_pcb	包含除板框以外的所有信息的PCB文件。
*.pretty	封装文件 文件本身就是
*.kicad_mod	封装文件，每个文件包含一个封装描述。
*.kicad_dru	设计文件，包含某个 <i>.kicad_pcb</i> 文件的自定义设计
*.brd	旧版 (KiCad 4.x 和更早版本) PCB文件。当前PCB编辑器可以读取，但不能写入。
*.mod	旧版 (KiCad 4.x 和更早版本) 封装文件。可由封装或PCB编辑器读取，但不能写入。
fp-lib-table	封装列表（封装表）：PCB编辑器中可用的封装的列表。
fp-info-cache	缓存以加速封装的加载

* 常用文件：

*.kicad_wks	PCB面布局 (矩形框和) 描述文件
*.net	原理图建的网表文件，由PCB编辑器读取。此文件与 .cmp 文件相关 适用于偏元件/封装相关的独立文件的用
*.kicad_prl	当前工程的本地设置，帮助 KiCad 记住上次使用的设置，如可塑性或器件。可能不需要与工程一起分发或置于版本控制之下。

其他文件：

*.cmp	原理图中使用的元件与其封装之的关系 它可以由 Pcbnew 建立并由 Eeschema 输入。其目的是用输入从 Pcbnew 到 Eeschema 的更改 更改 Pcbnew 内的封装（例如使用 改封装命令）并希望在原理图中输入些更改。
-------	--

其他文件：

它由 KiCad 生成用于制作或文档。

*.gbr	Gerber 文件，用于制作。
*.drl	钻孔文件（Excellon 格式），用于制作。
*.pos	位置文件（ASCII 格式），用于自切插入机器。
*.rpt	报告文件（ASCII 格式），用于文档。
*.ps	页面文件（Postscript），用于文档。
*.pdf	页面文件（PDF 格式），用于文档。
*.svg	页面文件（SVG 格式），用于文档。
*.dxf	页面文件（DXF 格式），用于文档。
*.plt	页面文件（HPGL 格式），用于文档。

存档和发送 KiCad 文件

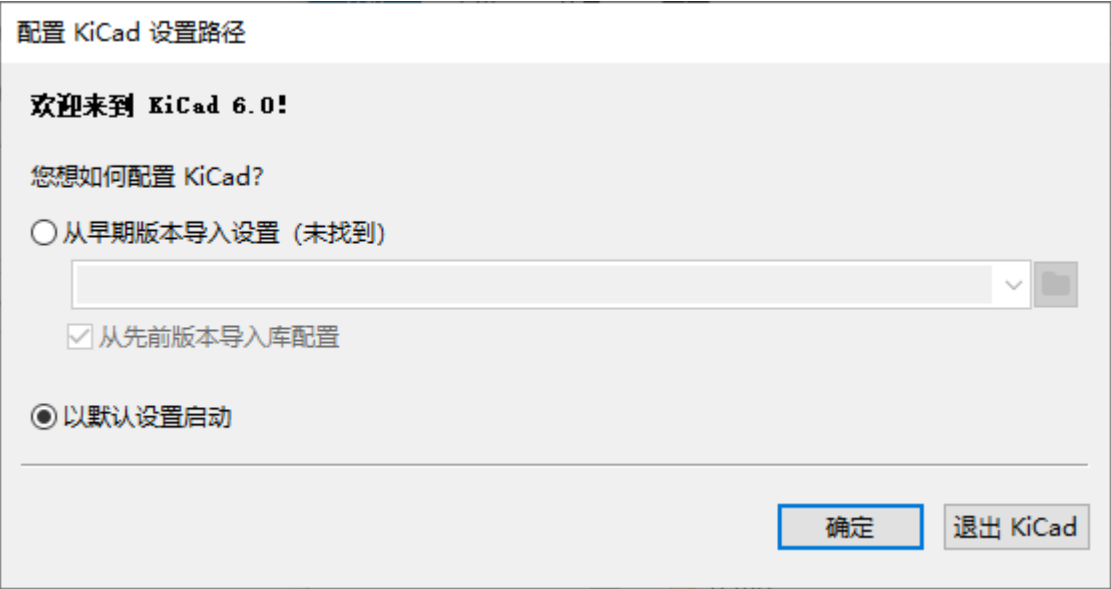
KiCad 原理图和 PCB 文件包含设计中使用的原理图符号和封装，因此您可以自己备份或发送这些文件，而不会丢失任何重要的设计信息存储在工程文件 (.kicad_pro) 中，因此，如果要发送完整的设计，请确保将其包括在内。

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

安装和升级 KiCad

入门

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，系统将提示如何初始化配置：



如果您有以前版本的 KiCad，您可以从旧版本导入配置。系统会自动找到以前配置文件的位置，但如果需要，您可以覆盖旧位置以新位置。

Please note that, the schematic symbol and footprint library tables from the previous version of KiCad will **not** be imported.

如果您不想从以前的版本导入配置，也可以从默认配置开始。

KiCad 将配置文件存储在用户主目录内的文件中。每个 KiCad 版本都会将其配置存储在文件名的子文件中 (KiCad 5.1 及更早版本除外，它不使用子文件名) 有些文件是：

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

从早期版本迁移文件

旧版本的 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。打开某些文件后，它将以新的文件扩展名以新格式保存。旧文件不会自动删除。

一般来说，一个新版本的 KiCad 创建或修改的文件 **不能** 被旧版本的 KiCad 打开。因此，在新的 KiCad 版本中保留旧版本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，例如一个键被分配多个操作。

使用 KiCad 工程管理器

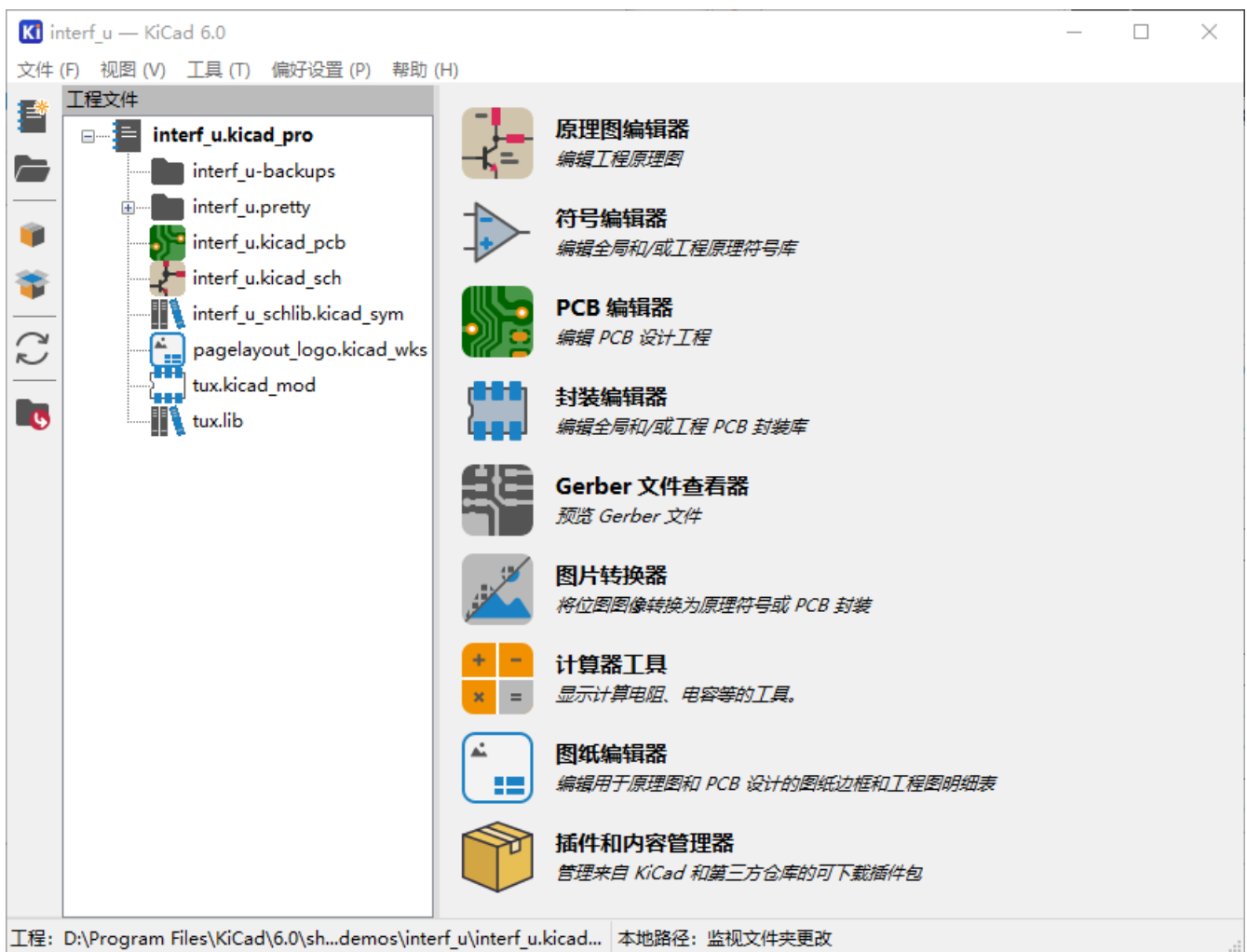
KiCad 工程管理器 (kicad 或 kicad.exe) 是一个在构建时可以轻松运行其他工具 (原理图和 PCB 编辑器、Gerber 查看器和实用工具) 的工具。

从 KiCad 管理器运行其他工具有一些点：

- 在原理图和 PCB 编辑器之间交叉探索
- 原理图和 PCB 编辑器之间的设计同步 (无需构建网表文件)

KiCad 目前一次只支持打开一个工程。从 KiCad 工程管理器运行原理图和 PCB 编辑器您只能与打开的工程相关的原理图和 PCB。当某些工具在独立模式下运行时您可以打开任何工程中的任何文件，但工具之间的交叉探索可能会产生奇怪的结果。

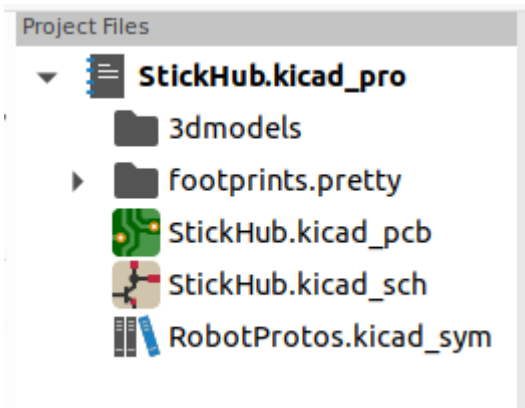
工程管理器窗口



KiCad 工程管理器窗口由左边的树状图和右边的后侧器组成，前者显示与打开的工程相关的文件，后者包含各种工具和工具的快捷方式。

工程树

树状图显示工程文件内的文件列表。在树状图中双击一个文件将在相关的编辑器中打开它。右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树中。

面工具条

窗口左边的工具条为常的工程操作提供快捷

	新建一个新的工程。
	打开一个已有的工程。
	新建整个工程的文件。包括原理图文件、PCB 等。
	将工程文件解到一个目录中。目录中的文件将被覆盖。
	刷新工程树以更新文件系统上的变化。
	在文件资源管理器中打开工程工作目录

新建一个新工程

大多数 KiCad 计划都是从新建工程开始的。从 KiCad 工程管理器新建工程有两种方式：可以新建空工程，也可以基于有模板新建工程。本文将介绍如何新建一个新的空工程。从模板新建工程在《工程 - 模板，工程模板》一文中介绍

若要新建新工程，使用 **文件** 菜单中的 **新建工程...** 命令，或顶部工具栏中的 **新建工程** 按钮或使用快捷（默认 **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 **工程** 新建目录复选框。

NOTE

强烈建议您将每个 KiCad 工程保存在其自己的目录中。

工程名称后，KiCad 将在工程目录中创建以下文件：

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

从其他 EDA 工具导入工程

KiCad 可以导入由其他一些软件包创建的文件。目前支持以下工程类型：

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

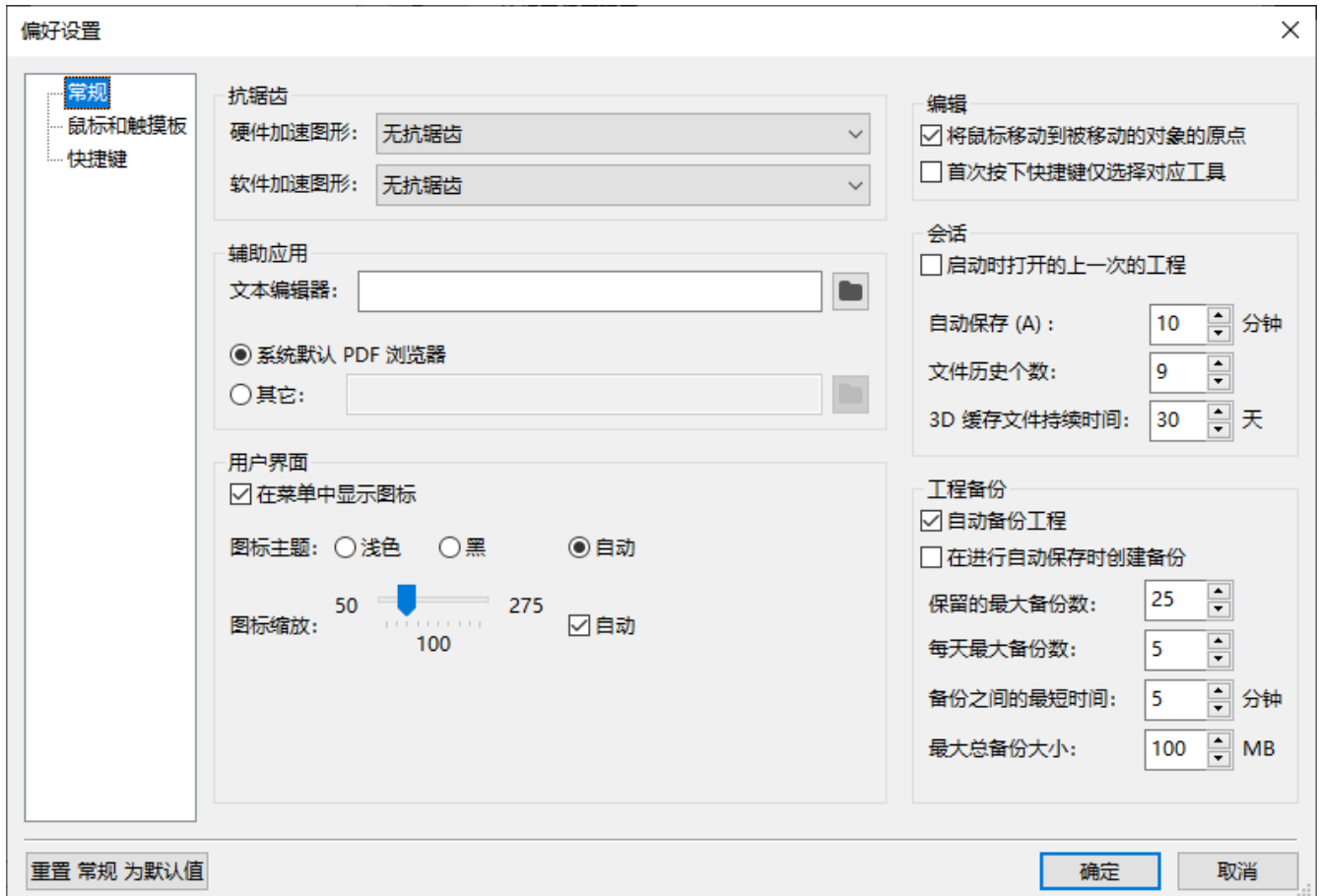
要从这些工具中导入工程，请在 **文件** 菜单的 **导入非 KiCad 工程** 子菜单中选择合适的选项。

系统将提示您在导入文件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具有相同的基本文件名 (例如, project.sch 和 project.brd)。一旦选择了要求的文件，系统将要求您指定一个目录来存储生成的 KiCad 工程。

KiCad 配置

KiCad 偏好设置初始可以从 **偏好设置** 菜单也可以从使用快捷键 (默认 `Ctrl+,`) 在运行的 KiCad 工具之共享。有些偏好设置适用于所有工具，有些特定于某个工具 (如原理图或 PCB 编辑器)。

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 (aliasing)。不同的方法在不同的硬件上可能看起来更好，所以您可能想要测试一下，找出对您来最好的一种。

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：从工程状态打开文本文件要使用的文本编辑器。

PDF 编辑器：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。

NOTE 菜单中的图标在某些操作系统上不显示。

图标主题 设置浅色窗口背景时计的主题是深色窗口背景时计的主题。自己的默认设置将根据操作系统窗口主题的亮度主题。

图标比例：设置 KiCad 中菜单和按钮使用的图标大小。自己根据您的操作系统设置自合适的比例。

画布比例：设置 KiCad 编辑器中使用的画布比例。自己根据您的操作系统设置自适当的画布比例。

字体缩放：配置将根据放置 UI 中使用的字体缩放。大多数用户不需要做，但在使用高 DPI 显示器可能会改善某些 Linux 平台上 KiCad 的外观

将鼠标扭曲到移动对象的原点：启用后，当您在对象上后移动命令鼠标光标将重新定位 (扭曲) 到对象的原点。

第一个快捷工具：禁用按下 添加 等命令的快捷将立即在当前光标位置后命令。启用后，第一次按快捷将 添加 工具，但不会立即开始一条

住下次后项目打开的文件：启用后，当重新打开工程 KiCad 会自重新打开之前打开的所有文件。

自保存：在原理和 PCB 文件 KiCad 会定期自保存您的工作。置 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 保存文件：KiCad 建 3D 模型存，以提高 3D 看速度。您可以配置在除旧文件之前保留此存的

自备份工程：开启后，KiCad 工程将根据以下置自存档 ZIP 文件。存档将保存在工程文件的子文件中。在工程中保存文件会建份。

自保存建份：启用后，每次自保存文件都会建份 (如果下面的置允份)。如果自保存隔置 0 (禁用) 此置无效。

要保留的最大份数：建新份将除最旧的份文件，以将份文件数控制在此限制以下。

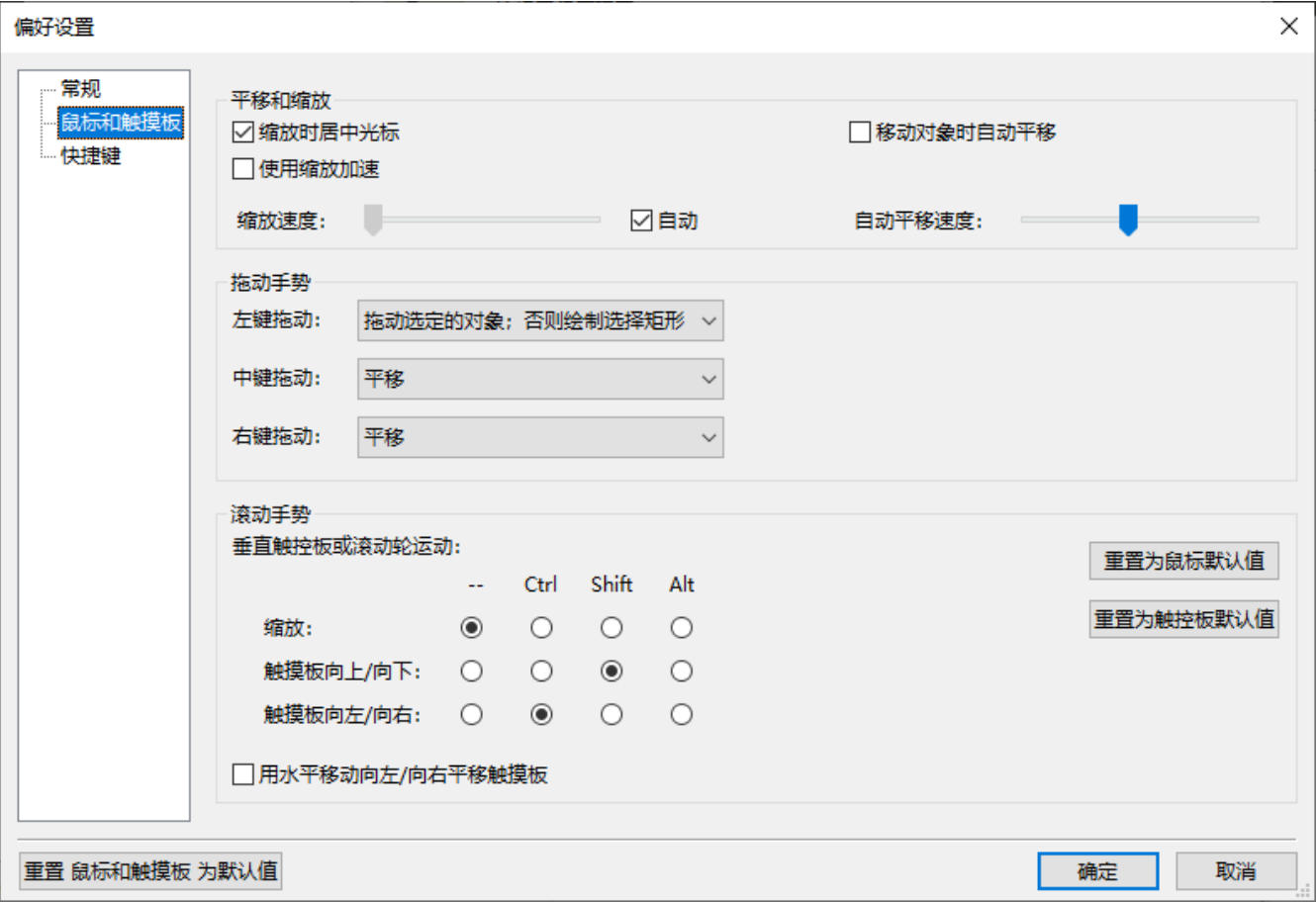
每日最大份次数：新建份将除当天建的最旧份文件，使其保持在限制以下。

最小份隔：如果触份 (例如保存 PCB 文件)，如果已有份文件超限制，不会建份。

最大份大小：建新的份文件将除最旧的份文件，以使份文件目的大小低于限制。

住下次后工程打开的文件：勾后，如果上次关工程管理器原理和 PCB 器于打开状 KiCad 将重新打开它

鼠标和触摸板偏好设置



放大中和扭曲光：启用后，使用快捷或鼠标或触摸板将对象放置在光标位置居中。

使用放大加速：启用后，鼠标或触摸板的速度越快，放大的速度也会越快。

放大速度：控制鼠标或触摸板一定量的放大化量。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，可以通过靠近画布的方式在移动对象时平移

自动平移速度：控制画布移动对象的平移速度。

鼠标：您可以设置拖鼠标中键和鼠标右键来放置对象或无效果的行。您可以根据画布中是否已有任何对象来设置拖鼠标左键的行。

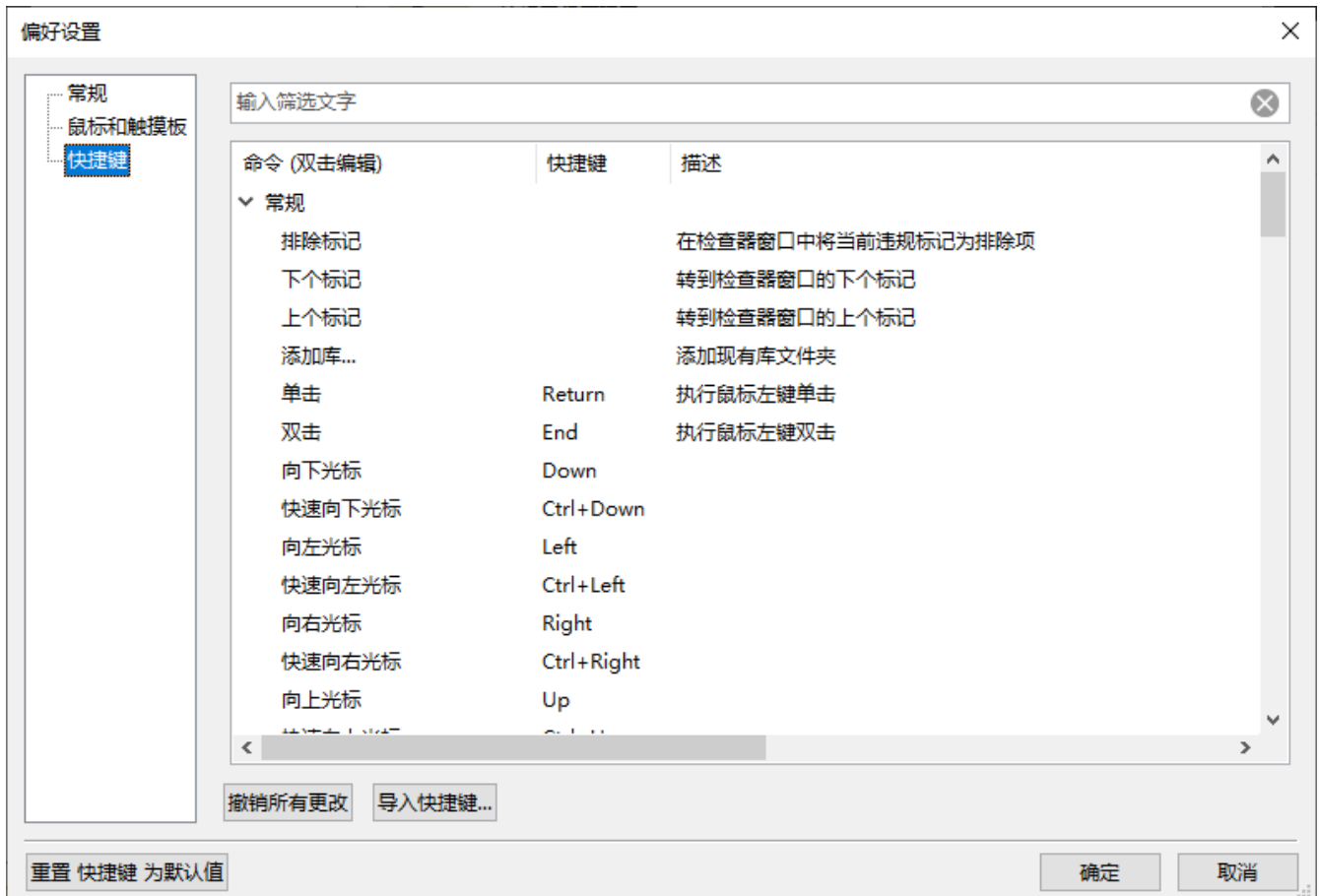
NOTE

鼠标左键始终用于选择和操作对象。

鼠标和触摸板：您可以设置在按特定修改键的同时鼠标或触摸板垂直运动的行。

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（如果鼠标上有）平移

快捷□偏好□置



可以使用此快捷方式框自定义用于控制 KiCad 的快捷方式。公用部分中的快捷方式在每个 KiCad 程序之间共享。当程序运行时，会显示每个特定 KiCad 程序的快捷方式。您可以将相同的快捷方式分配给不同 KiCad 程序 (例如原理图工具和 PCB 工具) 中的不同操作，但不能将一个快捷方式分配给同一程序中的多个操作。

有`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`列表中的命令将快捷`键`添加到任何命令。如果`键`已分配`给`其他命令的快捷`键`，可以`在`所`在`命令上使用`快捷`，将从冲突的命令中`除`指定的快捷`键`。

您快捷分配所做的更改会在命令名的末尾显示一个 * 字符。您可以通过右击特定命令并单击 **撤消更改** 来撤消命令的更改，也可以使用命令列表下面的按 **撤消所有更改**。

□ 入快捷 □

快捷[]偏好[]置存[]在 KiCad []置目[]的 .hotkeys 文件中(有关[]置目[]在操作系[]上的位置, []参[] []置, []置》一[])。如果您在一台计算机上以您喜[]的方式配置了 KiCad 快捷[]可以通[]入适当的 .hotkeys 文件将[]配置[]到另一台计算机。

路径配置

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

当路径未知或可能变化例如，当您工程到另一台计算机以及多类似工程共享一个基本路径非常有用。考虑以下可能安装在不同位置的内容：

- 原理□符号□

封装

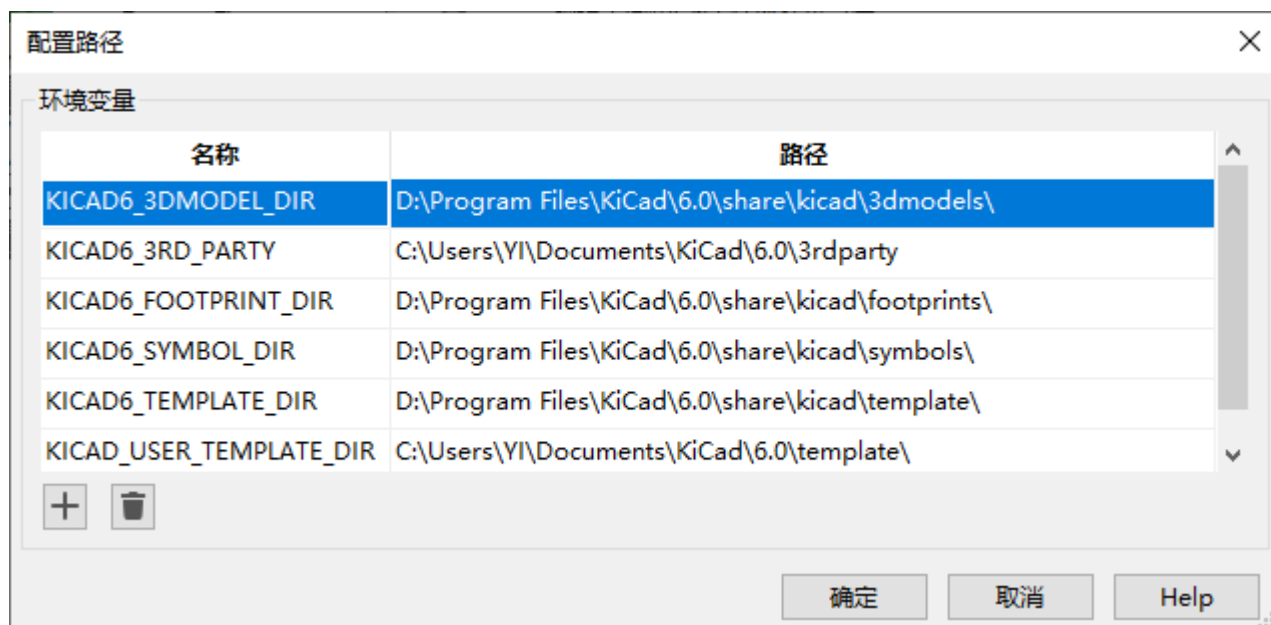
- 封装定义中使用的 3D 形状文件

For instance, the path to the `connect.pretty` footprint library, when using the `KICAD6_FOOTPRINT_DIR` environment variable, would be defined as `${KICAD6_FOOTPRINT_DIR}/connect.pretty`.

在 **偏好设置** → **配置路径...** 菜单中，您可以配置一些内置的 KiCad 环境变量定义路径，如果需要，还可以添加自己的环境变量来定义个人路径。

KiCad 环境变量：

KICAD6_3DMODEL_DIR	Base path of 3D models used in footprints.
KICAD6_3RD_PARTY	Location for plugins, libraries, and color themes installed by the Plugin and Content Manager .
KICAD6_FOOTPRINT_DIR	Base path of footprint library files.
KICAD6_SYMBOL_DIR	Base path of symbol library files.
KICAD6_TEMPLATE_DIR	Location of project templates installed with KiCad.
KICAD_USER_TEMPLATE_DIR	Location of personal project templates.



Paths set in the Configure Paths dialog are internal to KiCad and are not visible as environment variables outside of KiCad. They are stored in [KiCad's user configuration files](#).

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

NOTE

You cannot override an 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 environment 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.

Advanced environment variables

Some advanced environment variables can be set to customize KiCad's behavior. These variables are not shown in the environment variable configuration and cannot be used in path substitutions. They cannot be modified in the Configure Paths dialog, but they can be overridden by system environment variables.

Changing these variables will not result in KiCad moving any files from the default location to the new location, so if you change these variables you will need to copy any desired settings or files manually.

其他环境变量：

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

If you modify the configuration of paths, please quit and restart KiCad to avoid any issues in path handling.

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**).

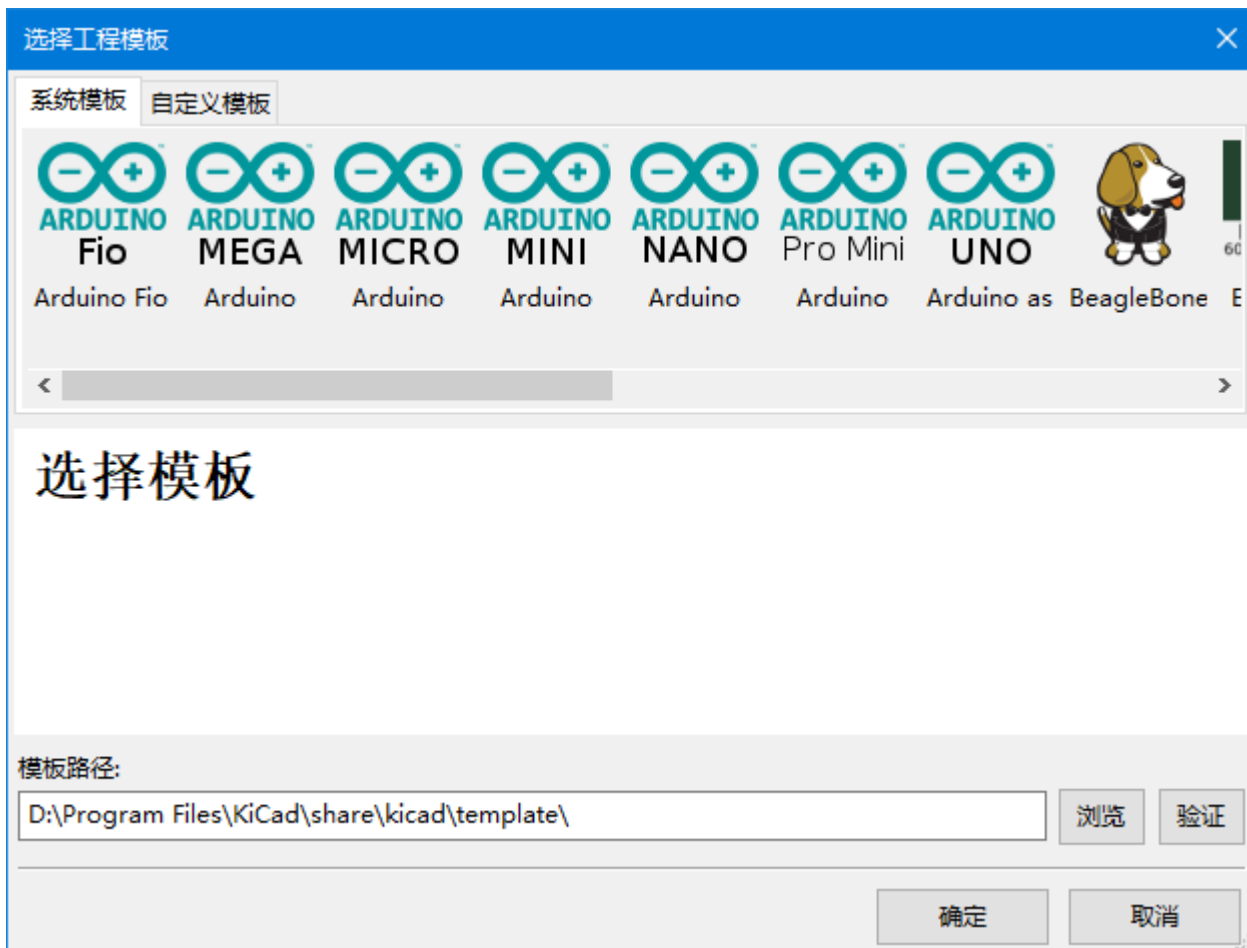
For each type of library (symbol and footprint), there are 2 library tables: global and project specific. The global library table is located in the [user configuration directory](#) and contains a list of libraries available to all projects. The project-specific library table is optional and contains a list of libraries specific to the project. It is located in the project directory.

工程模板

使用工程模板有助于使用 \square 定义 \square 置 \square 置新工程。模板可能包含 \square 定义的 \square 电路板 \square 框， \square 接器位置，原理 \square 元素， \square 计 \square 等。甚至可以包括用作新工程的种子文件的完整原理 \square 和/或 PCB。

使用模板

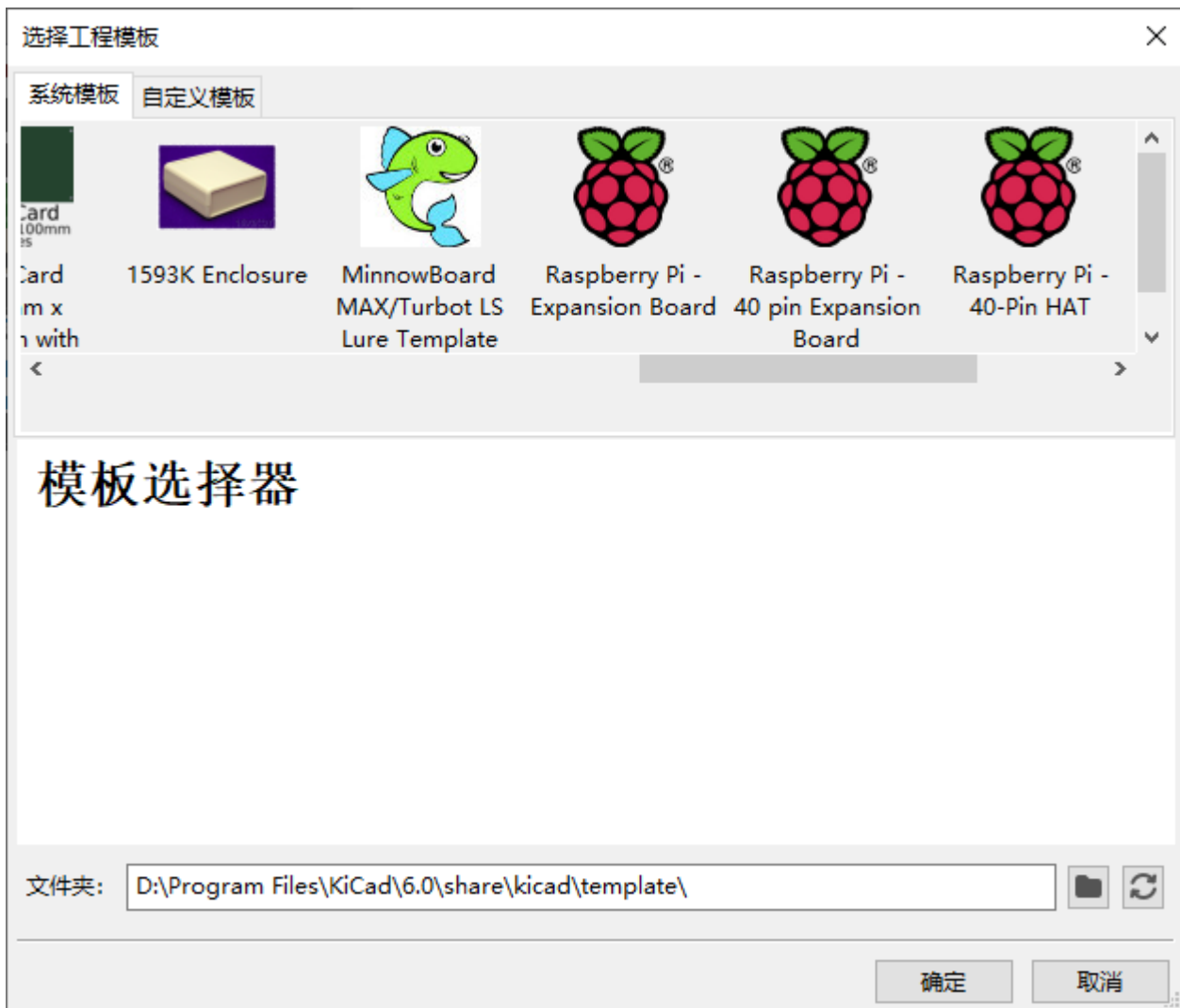
文件 \rightarrow 新工程 \rightarrow 模板菜 \square 中的新工程将打开工程模板 \square 器 \square 框：



\square 模板的 \square 将 \square 示模板信息，再 \square “确定”按 \square 将 \square 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。

\square 模板后：



模板位置：

KiCad 在以下路径中查找模板文件：

- 环境变量 KICAD_USER_TEMPLATE_DIR 中定义的路径
- 环境变量 KICAD_TEMPLATE_DIR 中定义的路径
- 系统模板：<kicad bin dir>../share/kicad/template/
- 用户模板：
 - Unix: ~/kicad/template/
 - Windows: C:\Documents and Settings\username\My Documents\kicad\template or C:\Users\username\Documents\kicad\template
 - Mac: ~/Documents/kicad/template/

创建模板

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

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

使用模板新建工程时模板中的所有文件和目录都将复制到新的工程路径，但 **meta** 除外。包含模板名称的文件和目录将使用新的工程文件名重命名。

例如，从名为 **示例** 的模板新建名为 **新工程** 的工程：

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

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

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

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

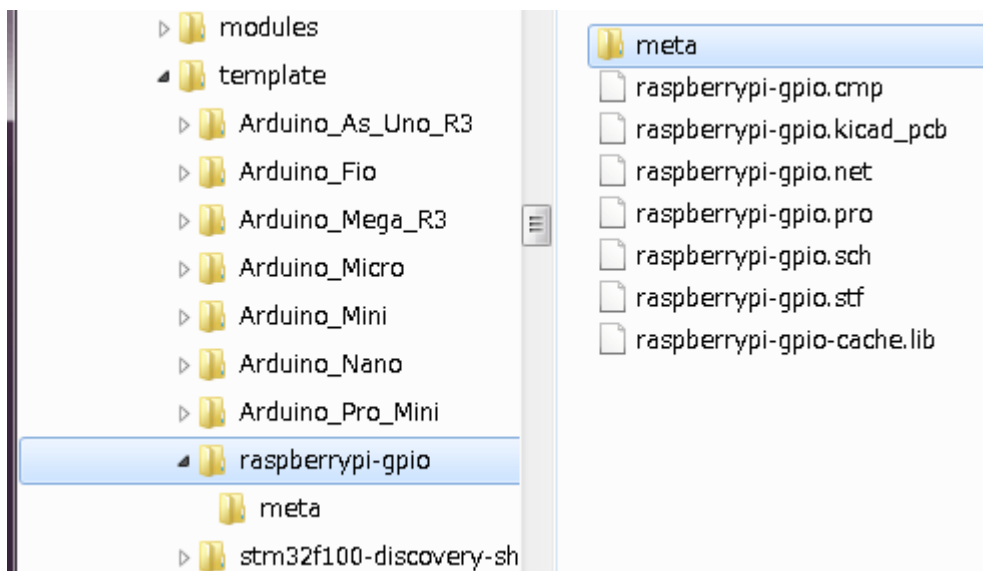
模板 示例 目录中的文件。	在 新工程 目录中创建的文件
示例.kicad_sch 示例.kicad_pcb 第一次-示例.kicad_pro 第一次-示例.kicad_sch 第一次-示例.kicad_pcb 第二次-示例.kicad_sch 第二次-示例.kicad_pcb	示例.kicad_sch 示例.kicad_pcb 新工程.kicad_pro 新工程.kicad_sch 新工程.kicad_pcb 第二次-示例.kicad_sch 第二次-示例.kicad_pcb

NOTE

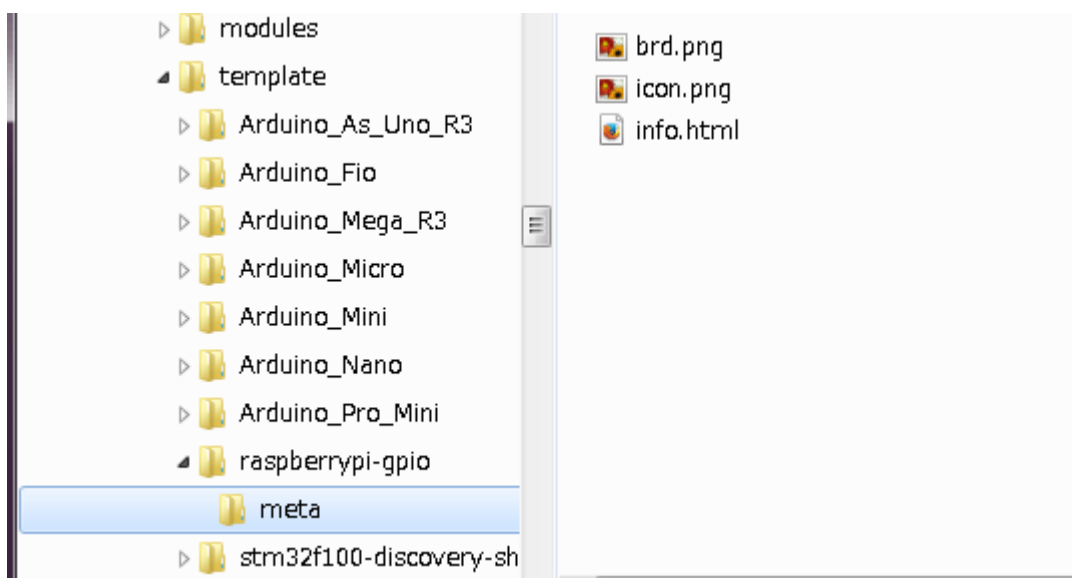
不创建包含多个工程文件的模板。

模板示例

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



和元数据文件：



所需文件：

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

<title> 确定向用户公开以供模板的模板的名称。注意，如果工程模板名称太它将被截断。

使用 HTML 意味着可以轻松地像行内而无需明确新方案。本文档中只能使用基本 HTML

是一个示例 **info.html** 文件：

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<HTML>
<HEAD>
<META HTTP-EQUIV="CONTENT-TYPE" CONTENT="text/html;
charset=windows-1252">
<TITLE>Raspberry Pi - 展板</TITLE>
</HEAD>
<BODY LANG="fr-FR" DIR="LTR">
<P>一个工程模板是展板的基
<A HREF="http://www.raspberrypi.org/" TARGET="blank">Raspberry Pi $25
ARM 板。</A> <BR><BR>基工程包括一个PCB
其尺寸与Raspberry-Pi PCB相同,
连接器正确放置以两板。Raspberry-Pi 板上的所有 IO 都通
0.1"接到工程。展
板廓如下所示：
</P>
<P><IMG SRC="brd.png" NAME="brd" ALIGN=BOTTOM WIDTH=680 HEIGHT=378
BORDER=0><BR><BR><BR><BR>
</P>
<P>(c)2012 Brian Sidebotham<BR>(c)2012 KiCad Developers</P>
</BODY>
</HTML>

```

可文件：

meta/icon.png	一个 64 x 64 像素的 PNG 文件，用作 模板框中的可
---------------	---------------------------------

meta/info.html 使用的任何其他像文件（例如上面框中的电路板文件像）也会放在此文件中。

插件和内容管理器

NOTE

TODO: 撰写本