

# Schematic Editor

The KiCad Team

# Table of Contents

Introduction to the KiCad Schematic Editor .....	2
描述 .....	2
技⌘概述 .....	2
Generic Schematic Editor commands .....	3
鼠⌘命令 .....	3
⌘⌘ .....	4
格点 .....	8
Snapping .....	8
⌘放⌘⌘ .....	9
⌘示光⌘坐⌘ .....	9
⌘⌘菜⌘⌘ .....	9
上方工具⌘ .....	10
右⌘工具⌘⌘⌘ .....	11
左工具⌘⌘⌘ .....	12
⌘出菜⌘和快速⌘⌘ .....	13
主菜⌘ .....	14
文件菜⌘ .....	14
首⌘⌘菜⌘ .....	15
帮助菜⌘ .....	23
通用⌘部工具⌘ .....	24
表格管理 .....	24
搜索工具 .....	24
网表工具 .....	25
批注工具 .....	26
⌘气⌘⌘⌘⌘工具 .....	27
物料清⌘工具 .....	30
⌘⌘字段工具 .....	32
用于封装分配的⌘入工具 .....	33
管理符号⌘ .....	35
符号⌘表 .....	35
原理⌘⌘建和⌘⌘ .....	40
⌘介 .....	40
一般考⌘ .....	40
符号放置和⌘⌘ .....	40
Electrical Connections .....	44
⌘⌘⌘充 .....	52
⌘救⌘存的符号 .....	54
分⌘原理⌘ .....	56
⌘介 .....	56
在⌘次⌘构中⌘航 .....	56
本地、分⌘和全局⌘⌘ .....	57
⌘次⌘构⌘建摘要 .....	57

工作表符号 .....	57
□接 - 分□引脚 .....	58
□接 - 分□□□ .....	58
复□□次□构 .....	60
平面□次□构 .....	61
符号批注工具 .....	64
□介 .....	64
一些例子 .....	65
使用□气□□□□□行□计□□ .....	68
□介 .....	68
如何使用 ERC .....	68
ERC 的示例 .....	69
□示□断 .....	69
□源引脚和□源□志 .....	71
配置 .....	71
ERC □告文件 .....	72
Transfer Schematic to PCB .....	74
概述 .....	74
Options .....	74
□□和打印 .....	76
□介 .....	76
常□的打印命令 .....	76
在 Postscript 中□制 .....	76
以 PDF 格式□制 .....	77
在 SVG 中□□ .....	78
在 DXF 中□□ .....	78
在 HPGL 中□□ .....	78
在□上打印 .....	79
Symbol Editor .....	81
关于符号□的一般信息 .....	81
符号□概述 .....	81
符号□□□器概述 .....	81
□□□与□□ .....	85
□建□符号 .....	85
□形元素 .....	92
每个符号多个□位和替代体型□式 .....	94
引脚□建和□□ .....	97
符号字段 .....	104
Power Ports .....	105
符号□□□器 .....	110
□介 .....	110
□□-主屏幕 .....	111
符号□□□器□部工具□ .....	111
□建网□列表 .....	113

概述 .....	113
网表格式 .....	113
网表示例 .....	116
关于网表的说明 .....	118
其他格式 .....	118
创建自定义网表和 BOM 文件 .....	121
中间网表文件格式 .....	121
新的网表格式 .....	123
XSLT 方法 .....	123
命令行格式：python 脚本的示例 .....	131
中间网表结构 .....	132
有关 xsltproc 的更多信息 .....	137
仿真器 .....	141
分配模型 .....	141
Spice 指令 .....	146
仿真 .....	146

### NOTE

This manual is in the process of being revised to cover the latest stable release version of KiCad. It contains some sections that have not yet been completed. We ask for your patience while our volunteer technical writers work on this task, and we welcome new contributors who would like to help make KiCad's documentation better than ever.

### 版

This document is Copyright © 2010-2022 by its contributors as listed below. You may distribute it and/or modify it under the terms of either the GNU General Public License (<http://www.gnu.org/licenses/gpl.html>), version 3 or later, or the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/3.0/>), version 3.0 or later.

本指南中的所有商均属于其合法所有者。

### 献者

Jean-Pierre Charras, Fabrizio Tappero, Graham Keeth

### 翻人

taotieren <[admin@taotieren.com](mailto:admin@taotieren.com)>, 2019, 2020, 2021.

Telegram 体中文交流群: [https://t.me/KiCad\\_zh\\_CN](https://t.me/KiCad_zh_CN)

### 反

将任何告、建或新版本引到此:

- About KiCad documentation: <https://gitlab.com/kicad/services/kicad-doc/issues>
- 关于 KiCad 件: <https://gitlab.com/kicad/code/kicad/issues>

# Introduction to the KiCad Schematic Editor

## 描述

The KiCad Schematic Editor is a schematic capture software distributed as a part of KiCad and available under the following operating systems:

- Linux
- Apple OS X
- Windows

Regardless of the OS, all KiCad files are 100% compatible from one OS to another.

The Schematic Editor is an integrated application where all functions of drawing, control, layout, library management and access to the PCB design software are carried out within the editor itself.

The KiCad Schematic Editor is intended to cooperate with the KiCad PCB Editor, which is KiCad's printed circuit design software. It can also export netlist files, which lists all the electrical connections, for other packages.

The Schematic Editor includes a symbol library editor, which can create and edit symbols and manage libraries. It also integrates the following additional but essential functions needed for modern schematic capture software:

- 电气规则检查 (ERC)，用于自动控制错误和缺失的连接
- 以多种格式输出文件 (Postscript, PDF, HPGL和SVG)
- 物料清单生成 (通过 Python 或 XSLT 脚本, 允许多灵活的格式)。

## 技术概述

The Schematic Editor is limited only by the available memory. There is thus no real limitation to the number of components, component pins, connections or sheets. In the case of multi-sheet schematics, the representation is hierarchical.

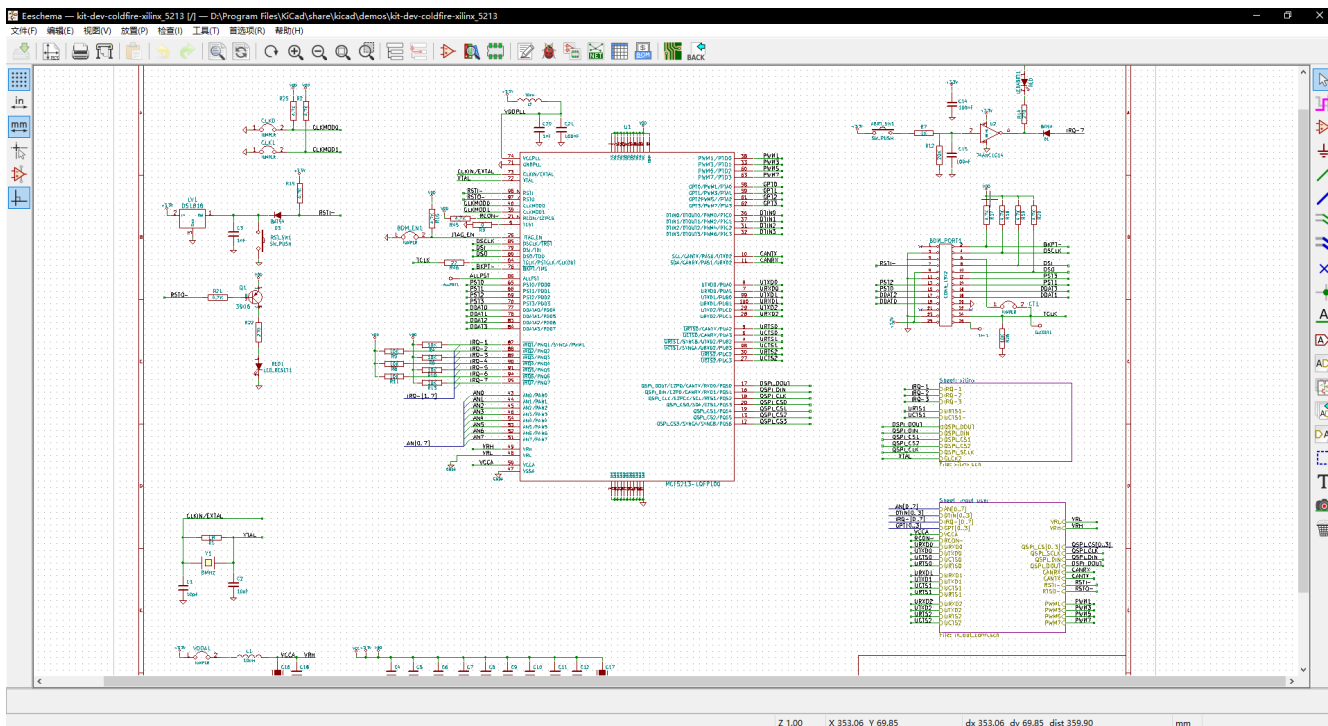
The Schematic Editor can use multi-sheet schematics in a few ways:

- 单级结构 (每个原理图只使用一次)。
- 复用的单级结构 (一些原理图在多个图中不止一次使用)。
- 扁平级结构 (原理图未在主图中明确连接)。

# Generic Schematic Editor commands

命令可以通过以下方式执行：

- 菜单（屏幕顶部）。
- 屏幕顶部的快捷命令（常用命令）。
- 点击屏幕右侧的快捷命令（特定命令或“工具”）。
- 屏幕左侧的快捷命令（显示命令）。
- 按下鼠标按钮（重要的补充命令）。特别是右击打开光标的上下文菜单，网格和元素操作。
- Function keys ( **F1** , **F2** , **F3** , **F4** , **Insert** and **Space** ). Specifically: **Escape** cancels the command in progress. **Insert** allows the duplication of the last element created.
- Pressing hotkeys. For a list of hotkeys, see the **Help** → **List Hotkeys** menu entry or press **Ctrl** + **F1** . Many hotkeys select a tool but do not perform the tool's action until the canvas is clicked. This behavior can be changed by unchecking **First hotkey selects tool** in the **Common** Preferences pane. With this option unchecked, pressing a hotkey will select the tool and immediately perform the tool's action at the current cursor location.



## 鼠标命令

### 基本命令

左键

- Single click: Selects the item under the cursor and displays the item's characteristics in the status bar.
- Double click: edits the item if it is editable.
- Long click (click and hold): opens a pop-up menu to clarify the selection.

右

- Opens a pop-up menu. If an item is selected, the items in the menu are related to the selected item. If an item is under the cursor when the right mouse button is clicked, the item is selected.

## Selection operations

Schematic editor items can be selected by clicking on them. Multiple items can be selected at once. Add items to the selection with **Shift** + click, and remove items from the selection with **Ctrl** + **Shift** + click.

### NOTE

On Apple keyboards, use the **Cmd** key instead of **Ctrl**.

left mouse button	Select item.
<b>Shift</b> + left mouse button	Add item to selection.
<b>Ctrl</b> + <b>Shift</b> + left mouse button	Remove item from selection.
long click	Clarify selection from a pop-up menu.
<b>Ctrl</b> + left mouse button	Highlight net.

Items can also be selected by drawing a box around them using the left mouse button.

Dragging from left to right includes all items fully enclosed by the box. Dragging from right to left includes all items touched by the box, even if they are not fully enclosed.

The **Shift** and **Ctrl** + **Shift** modifiers also work with drag selections to add and remove items from the selection, respectively.

- The **Ctrl** + **F1** displays the current hotkey list.
- All hotkeys can be redefined using the hotkey editor (**Preferences** → **Preferences...** → **Hotkeys**).

The default hotkey list is below. Many additional actions do not have hotkeys by default, but hotkeys can be assigned to them with the hotkey editor.





















The hotkeys described in this manual use the key labels that appear on a standard PC keyboard. On an Apple keyboard layout, use the **Cmd** key in place of **Ctrl**, and the **Option** key in place of **Alt**.

Action	Default Hotkey	Description
Click	<b>Return</b>	Performs left mouse button click
Double-click	<b>End</b>	Performs left mouse button double-click
Cursor Down	<b>Down</b>	



Action	Default Hotkey	Description
Cursor Down Fast	Ctrl + Down	
Cursor Left	Left	
Cursor Left Fast	Ctrl + Left	
Cursor Right	Right	
Cursor Right Fast	Ctrl + Right	
Cursor Up	Up	
Cursor Up Fast	Ctrl + Up	
Switch to Fast Grid 1	Alt + 1	
Switch to Fast Grid 2	Alt + 2	
Switch to Next Grid	N	
Switch to Previous Grid	Shift + N	
Reset Grid Origin	Z	
Grid Origin	S	Set the grid origin point
New...	Ctrl + N	Create a new document in the editor
Open...	Ctrl + O	Open existing document
Pan Down	Shift + Down	
Pan Left	Shift + Left	
Pan Right	Shift + Right	
Pan Up	Shift + Up	
Print...	Ctrl + P	Print
Reset Local Coordinates	Space	
Save	Ctrl + S	Save changes
Save As...	Ctrl + Shift + S	Save current document to another location

Action	Default Hotkey	Description
Zoom to Fit	Home	Zoom to Fit
Zoom In at Cursor	F1	Zoom In at Cursor
Zoom Out at Cursor	F2	Zoom Out at Cursor
Refresh	F5	Refresh
Zoom to Selection	Ctrl + F5	Zoom to Selection
Change Edit Method	Ctrl + Space	Change edit method constraints
Copy	Ctrl + C	Copy selected item(s) to clipboard
Cut	Ctrl + X	Cut selected item(s) to clipboard
Delete	Del	Deletes selected item(s)
Duplicate	Ctrl + D	Duplicates the selected item(s)
Find	Ctrl + F	Find text
Find and Replace	Ctrl + Alt + F	Find and replace text
Find Next	F3	Find next match
Find Next Marker	Shift + F3	
Paste	Ctrl + V	Paste item(s) from clipboard
Redo	Ctrl + Y	Redo last edit
Select All	Ctrl + A	Select all items on screen
Undo	Ctrl + Z	Undo last edit
List Hotkeys...	Ctrl + F1	Displays current hotkeys table and corresponding commands
Preferences...	Ctrl + ,	Show preferences for all open tools
Clear Net Highlighting	~	Clear any existing net highlighting
Edit Library Symbol...	Ctrl + Shift + E	Open the library symbol in the Symbol Editor
Edit with Symbol Editor	Ctrl + E	Open the selected symbol in the Symbol Editor
Highlight Net	`	Highlight net under cursor

Action	Default Hotkey	Description
Add Global Label	 + 	Add a global label
Add Hierarchical Label		Add a hierarchical label
Add Junction		Add a junction
Add Label		Add a net label
Add No Connect Flag		Add a no-connection flag
Add Power		Add a power port
Add Text		Add text
Add Symbol		Add a symbol
Add Bus		Add a bus
Add Lines		Add connected graphic lines
Add Wire		Add a wire
Finish Wire or Bus		Complete drawing at current segment
Unfold from Bus		Break a wire out of a bus
Autoplace Fields		Runs the automatic placement algorithm on the symbol or sheet's fields
Edit Footprint...		Displays footprint field dialog
Edit Reference Designator...		Displays reference designator dialog
Edit Value...		Displays value field dialog
Mirror Horizontally		Flips selected item(s) from left to right
Mirror Vertically		Flips selected item(s) from top to bottom
Properties...		Displays item properties dialog
Repeat Last Item		Duplicates the last drawn item
Rotate Counterclockwise		Rotates selected item(s) counter-clockwise
Drag		Drags the selected item(s)
Move		Moves the selected item(s)

Hotkeys are stored in the file `user.hotkeys` in KiCad's configuration directory. The location is platform-specific:

- Windows: `%APPDATA%\kicad\6.0\user.hotkeys`
- Linux: `~/.config/kicad/6.0/user.hotkeys`
- macOS: `~/Library/Preferences/kicad/6.0/user.hotkeys`

It is possible to import hotkey settings from a `user.hotkeys` file using menu **Preferences** → **Preferences...** → **Hotkeys** → **Import Hotkeys....**

## 格点

In the Schematic Editor the cursor always moves over a grid. The grid can be customized:

- Size can be changed using the right click menu or using **View** → **Grid Properties....**
- Color can be changed in the **Colors** page of the **Preferences** dialog (menu **Preferences** → **General Options**).
- Visibility can be switched using the left-hand toolbar button.

The default grid size is 50 mil (0.050") or 1.27 millimeters.

是在符号器中计符号将符号和放置在原理中以及放置引脚的首网格。

NOTE

Wires connect with other wires or pins only if their ends coincide exactly. Therefore it is important to keep symbol pins and wires aligned to the grid. It is recommended to always use a 50 mil grid when placing symbols and drawing wires because the KiCad standard symbol library and all libraries that follow its style also use a 50 mil grid.

人可以使用 25mil 到 10mil 的小网格。用于计符号体或放置文本和注不建用于放置引脚和

NOTE

Symbols, wires, and other elements that are not aligned to the grid can be snapped back to the grid by selecting them, right clicking, and clicking **Align Elements to Grid**.

## Snapping

Schematic elements such as symbols, wires, text, and graphic lines are snapped to the grid when moving, dragging, and drawing them. Additionally, the wire tool snaps to pins even when grid snapping is disabled. Both grid and pin snapping can be disabled while moving the mouse by using the modifier keys in the table below.

NOTE

On Apple keyboards, use the `Cmd` key instead of `Ctrl`.

Modifier Key	Effect
<code>Ctrl</code>	Disable grid snapping.
<code>Shift</code>	Disable snapping wires to pins.

## 缩放

要更改缩放:

- 右键以打开菜单然后所需的缩放。
- Or use hotkeys:
  - **F1**: Zoom in
  - **F2**: Zoom out
  - **F4**: Center the view around the cursor pointer position
  - **Home**: Zoom and center the view to fit the entire schematic sheet
  - **Ctrl** + **Home**: Zoom and center the view to fit all of the objects in the schematic
  - **Ctrl** + **F5**: Activate the Zoom to Selection tool
- 窗口缩放:
  - 鼠标滚轮放大/缩小
  - Shift + 鼠标滚轮向上/向下平移
  - Ctrl + 鼠标滚轮向左/向右平移

Mouse scroll gestures are configurable in the **Mouse and Touchpad** page of the **Preferences** dialog.

## 示光坐

The display units are in inches, mils, or millimeters.

窗口右下角示以下信息:

- 缩放系数
- 光的位置
- 光的相位置
- The grid size
- The active unit system
- The active tool

按空格可以将相坐重置为零。用于量两点之间的距离或对象很有用。

	Z 5.50	X 348.00 Y 60.96	dx 348.00 dy 60.96 dist 353.30	mm	
--	--------	------------------	--------------------------------	----	--

## 菜单

菜单允许打开和保存原理图程序配置和看文档。

文件(F)	编辑(E)	视图(V)	放置(P)	检查(I)	工具(T)	首选项(R)	帮助(H)
-------	-------	-------	-------	-------	-------	--------	-------

# 上方工具














This toolbar gives access to the main functions of the Schematic Editor.

If the Schematic Editor is run in standalone mode, this is the available tool set:



注意，当 KiCad 在目模式下运行前两个不可用，因它可以理个文件。



















	Create a new schematic (only in standalone mode).
	Open a schematic (only in standalone mode).
	Save complete schematic project.
	Set the schematic-specific options.
	Select the sheet size and edit the title block.
	Open print dialog.
	Open plot dialog.
	Paste a copied/cut item or block to the current sheet.
	Undo: Revert the last change.
	Redo: Revert the last undo operation.
	Show the dialog to search symbols and texts in the schematic.
	Show the dialog to search and replace texts in the schematic.
	Refresh screen.
	Zoom in.
	Zoom out.
	Zoom to fit the entire schematic sheet.
	Zoom to fit all objects in the schematic.
	Zoom to fit selected items.
	View and navigate the hierarchy tree.
	Leave the current sheet and go up in the hierarchy.
	Rotate selected items counter-clockwise.

	Rotate selected items clockwise.
	Mirror selected items vertically.
	Mirror selected items horizontally.
	Call the symbol library editor to view and modify libraries and symbols.
	Browse symbol libraries.
	Open the footprint library editor to view and modify libraries and footprints.
	Annotate symbols.
	Electrical Rules Checker (ERC), automatically validate electrical connections.
	Open the footprint assignment tool to assign footprints to symbols.
	Bulk edit symbol fields in a spreadsheet interface.
	Generate the Bill of Materials (BOM).
	Open the PCB editor.
	Open the Python scripting console.

## 右工具

此工具包含以下工具：








- 放置符号，交叉点，文本等。
- 建分子表和接符号。

	Cancel the active command or tool and go into selection mode.
	Highlight a net by marking its wires and net labels with a different color. If the PCB Editor is also open then copper corresponding to the selected net will be highlighted as well.
	Display the symbol selector dialog to select a new symbol to be placed.
	Display the power symbol selector dialog to select a power symbol to be placed.
	Draw a wire.
	Draw a bus.
	Draw wire-to-bus entry points. These elements are only graphical and do not create a connection, thus they should not be used to connect wires together.
	Place a "No Connect" flag. These flags should be placed on symbol pins which are meant to be left unconnected. It is done to notify the Electrical Rules Checker that lack of connection for a particular pin is intentional and should not be reported.
	Place a junction. This connects two crossing wires or a wire and a pin, when it can be ambiguous (i.e. if a wire end or a pin is not directly connected to another wire end).
	Place a local label. Local label connects items located <b>in the same sheet</b> . For connections between two different sheets, you have to use global or hierarchical labels.
	Place a global label. All global labels with the same name are connected, even when located on different sheets.
	Place a hierarchical label. Hierarchical labels are used to create a connection between a subsheet and the parent sheet that contains it.
	Place a hierarchical subsheet. You must specify the file name for this subsheet.
	Import a hierarchical pin from a subsheet. This command can be executed only on hierarchical subsheets. It will create hierarchical pins corresponding to hierarchical labels placed in the target subsheet.
	Draw a line. These are only graphical and do not connect anything.
	Place a text comment.
	Place a bitmap image.
	Delete clicked items.

## 左工具

此工具管理示



	Toggle grid visibility.
	Switch units to inches.
	Switch units to mils (0.001 inches).
	Switch units to millimeters.
	Choose the cursor shape (full screen/small).
	Toggle visibility of "invisible" pins.
	Toggle free angle/90 degrees wires and buses placement.

## 出菜和快速

右可打开所元素的上下文菜单包含：

- 放系数。
- 网格整。
- Copy/Paste/Delete commands.
- Add Wire/Bus.
- 通常所元素的参数。

# 主菜□

## 文件菜□

文件(F) 编辑(E) 视图(V) 放置(P) 检查(I) 工具(T) 首选项(R) 帮助(H)

New	Close current schematic and start a new one (only in standalone mode).
Open	Load a schematic project (only in standalone mode).
Open Recent	Open a schematic project from the list of recently opened files (only in standalone mode).
Save	Save current sheet and all its subsheets.
Save As...	Save the current sheet under a new name (only in standalone mode).
Save Current Sheet Copy As...	Save a copy of the current sheet under a new name (only in project mode).
Insert Schematic Sheet Content...	Insert the contents of another schematic sheet into the current sheet (only in standalone mode).
Import	Import a non-KiCad schematic or a footprint assignment file.
Export	Export a netlist or a drawing of the schematic to the clipboard.
Schematic Setup...	Set up schematic formatting, electrical rules, net classes, and text variables.
Page Settings...	Configure page dimensions and title block.
Print	Print schematic project (See also chapter <a href="#">Plot and Print</a> ).
Plot	Export to PDF, PostScript, HPGL or SVG format (See chapter <a href="#">Plot and Print</a> ).
Quit	Terminate the application.

## Schematic Setup

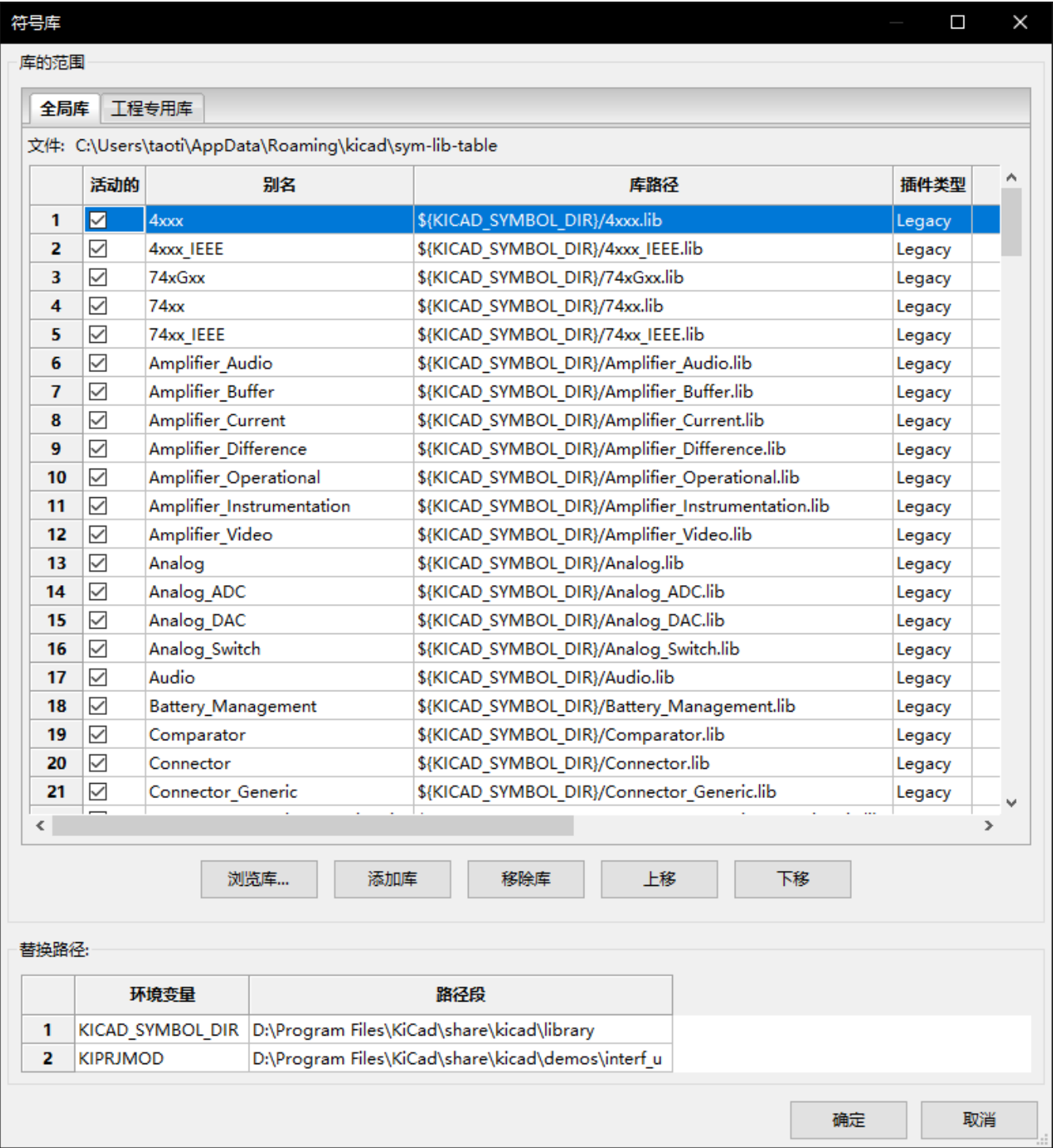
The Schematic Setup window is used to set schematic options that are specific to the currently active schematic. For example, the Schematic Setup window contains formatting options, electrical rule configuration, netclass setup, and schematic text variable setup.

首 菜



Configure Paths...	Set the default search paths.
Manage Symbol Library Tables...	Add/remove symbol libraries.
Preferences...	Preferences (units, grid size, field names, etc.).
Set Language	Select interface language.

管理符号表



KiCad uses two library tables to store the list of available symbol libraries, which differ by the scope:

全局

Libraries listed in the Global Library table are available to every project. They are saved in the `sym-lib-table` in the KiCad configuration directory, which is system-dependent:



- Windows: `%APPDATA%\kicad\6.0\sym-lib-table`
- Linux: `~/.config/kicad/6.0/sym-lib-table`
- macOS: `~/Library/Preferences/kicad/6.0/sym-lib-table`

目用

Libraries listed in Project Specific Libraries table are available to the currently opened project. They are saved in a `sym-lib-table` file in the project directory.

Both library tables are visible by clicking on **Global Libraries** or **Project Specific Libraries** tab in the Manage Library Tables window.

添加一个新

Add a library either by clicking the  button and selecting a file or clicking the  button and typing a path to a library file. The selected library will be added to the currently opened library table (Global/Project Specific).

除

Remove a library by selecting one or more libraries and clicking the  button.

属性

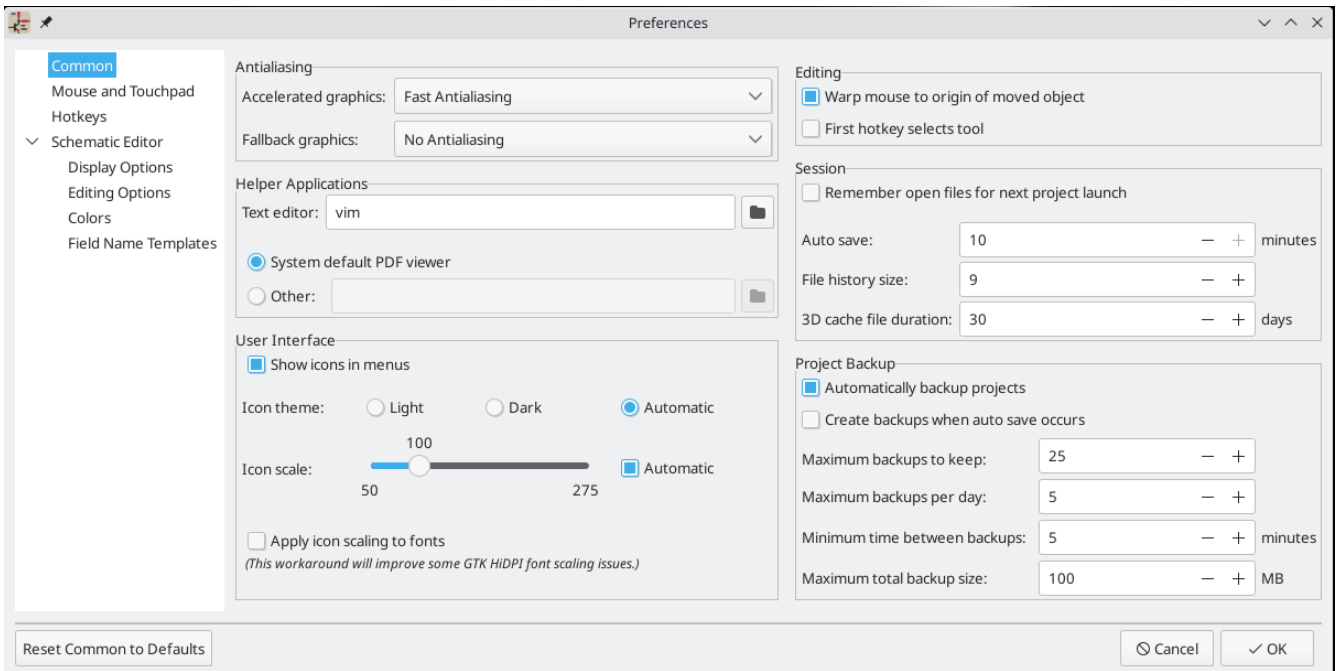
表中的每一行都存了几个描述的字段：

Active	Enables/disables the library. It is useful to temporarily reduce the loaded library set.
Nickname	Nickname is a short, unique identifier used for assigning symbols to components. Symbols are represented by '<Library Nickname>:<Symbol Name>' strings.
Library Path	Path points to the library location.
Plugin Type	Determines the library file format. KiCad 6.0 libraries use the "KiCad" format, while KiCad 5.x libraries use the "Legacy" format. Legacy libraries are read-only.
Options	Stores library specific options, if used by plugin.
Description	Briefly characterizes the library contents.

Preferences

Common Preferences

NOTE	TODO: write this section
------	--------------------------

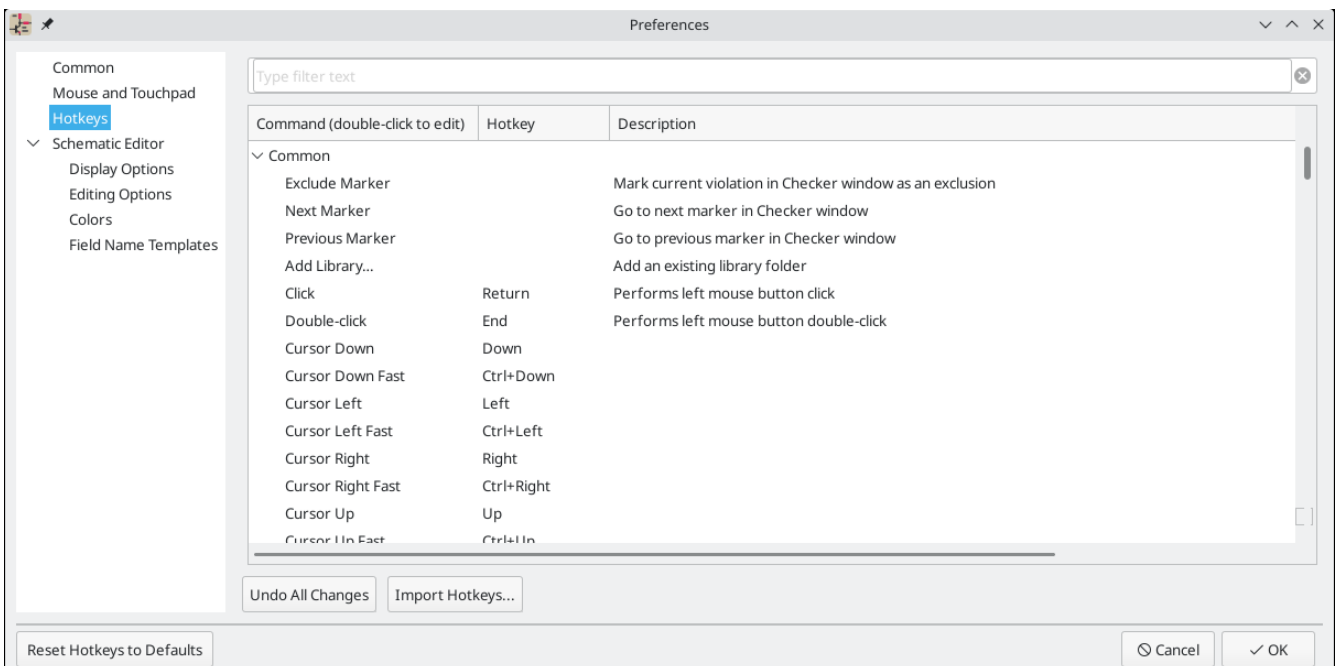


## Mouse and Touchpad

Center and warp cursor on zoom	If checked, the pointed location is warped to the screen center when zooming in/out.
Use touchpad to pan	When enabled, view is panned using scroll wheels (or touchpad gestures) and to zoom one needs to hold <b>Ctrl</b> . Otherwise scroll wheels zoom in/out and <b>Ctrl</b> / <b>Shift</b> are the panning modifiers.
Pan while moving object	If checked, automatically pans the window if the cursor leaves the window during drawing or moving.

□□

Redefine hotkeys.



通过双击操作新的或右操作以显示菜单

Edit	Define a new hotkey for the action (same as double click).
Undo Changes	Reverts the recent hotkey changes for the action.
Clear Assigned Hotkey	
Restore Default	Sets the action hotkey to its default value.

Display Options

原理图编辑选项

显示

编辑

控制

颜色

字段名称模板

网格尺寸(G):

50.0

mils

总线宽度(B):

12

mils

线宽 (L):

6

mils

部件ID符号(P):

A

图标比例:

50

275 %

100

☒ 自动

☒ 显示网格(S)

☒ 限制总线和连线垂直或水平绘制(R)

☐ 显示隐藏引脚(H)

☒ 显示页面范围(T)

☐ 符号选择器中的封装预览(实验)

确定

取消

网格尺寸	网格大小\000  建0 使用普通网格（0.050英寸或1,27毫米）。0\ 网格用于元件构建。
00厚度	用于0制00的笔大小。
0条粗0	用于0制没有0象的0象的笔大小 指定的笔大小。
元件 ID 表示法	用于表示符号0元的后00式（U1A, U1.A, U1-1等）
00比例	0整工具000大小。
0示网格	网格可0性0置。
将00和00限制0 H 和 V 方向	如果00000和 000用垂直或水平00制。否00可以在任何方向放置00和000
0示0藏的引脚：	通常0示不可00或 0藏）引脚 0源引脚。
0示0面限制	如果0中，0在屏幕上0示0面0界。
符号00器中的封装00	0示封装00框和 放置新符号0的封装00器。  注意：可能会0致00或延00使用00自00

## Editing Options

原理图编辑选项

显示

编辑

控制

颜色

字段名称模板

测量单位(M):

mm

重复项目的水平间距(H):

0

mils

重复项目的垂直间距(V):

100

mils

重复标签的增量(I):

1

默认文本尺寸(A):

50

mils

自动保存间隔时间(A):

10

分钟

☒ 自动放置符号字段(U)

☒ 允许字段自动对齐放置(L)

☐ 自动放置字段对齐到50mil网格(W)

确定

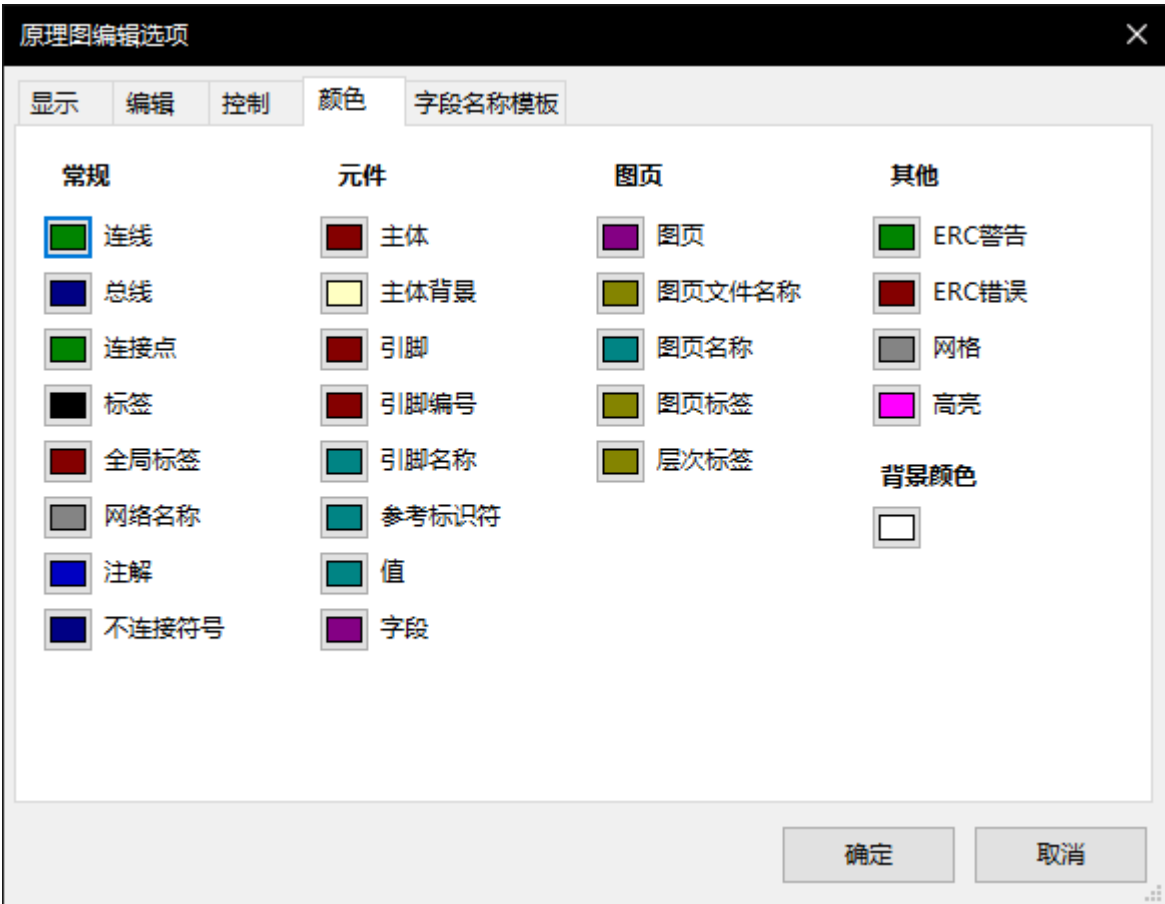
取消



Measurement units	Select the display and the cursor coordinate units (inches or millimeters).
Horizontal pitch of repeated items	Increment on X axis during element duplication (default: 0) (after placing an item like a symbol, label or wire, a duplication is made by the <input type="button" value="Insert"/> key)
Vertical pitch of repeated items	Increment on Y axis during element duplication (default: 0.100 inches or 2,54 mm).
Increment of repeated labels	Increment of label value during duplication of texts ending in a number, such as bus members (usual value 1 or -1).
Default text size	Text size used when creating new text items or labels.
Auto-save time interval	Time in minutes between saving backups.
Automatically place symbol fields	If checked, symbol fields (e.g. value and reference) in newly placed symbols might be moved to avoid collisions with other items.
Allow field autoplace to change justification	Extension of 'Automatically place symbol fields' option. Enable text justification adjustment for symbol fields when placing a new part.
Always align autoplaced fields to the 50 mil grid	Extension of 'Automatically place symbol fields' option. If checked, fields are autoplaced using 50 mils grid, otherwise they are placed freely.

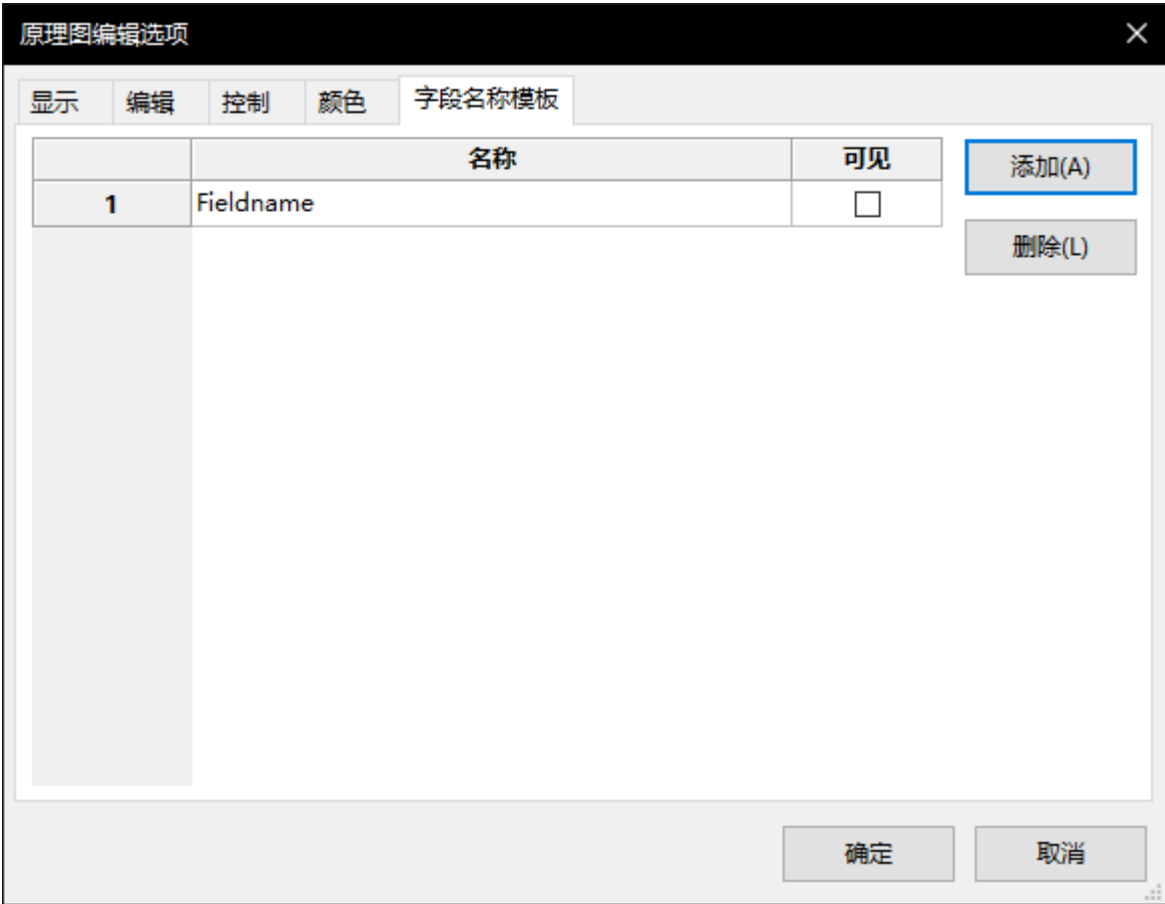
## 颜色

各种图形元素的配色方案。 任何颜色本以特定元素的新颜色。



### 默认字段

定义将在新放置的符号中显示的其他自定义字段和相关的...



## 帮助菜☐

☐☐在☐帮助（本文档），☐取有关 KiCad 的广泛教程。

Use the **Report a Bug** item to report a bug online. Full KiCad version and user system information is available via the **Copy Version Info** button in the **About KiCad** window.

# 通用工具

## 表格管理

The Sheet Settings icon (  ) allows you to define the sheet size and the contents of the title block.

页面设置

图纸

尺寸:  
A3 297x420mm

方向:  
横向

自定义尺寸:  
高度: 279.40 宽度: 431.80

布局预览



标题栏字段设置

共 1 页 第 1 页

更改日期  
Sun 22 Mar 2015 <<< 2019/ 2/18

☐ 导出到其他图页

版次  
2B

☐ 导出到其他图页

标题  
UNIVERSAL INTERFACE

☐ 导出到其他图页

公司  
KICAD

☐ 导出到其他图页

注释 1  
Comment 1

☐ 导出到其他图页

注释 2  
Comment 2

☐ 导出到其他图页

注释 3  
Comment 3

☐ 导出到其他图页

注释 4  
Comment 4

☐ 导出到其他图页

页面布局描述文件

浏览

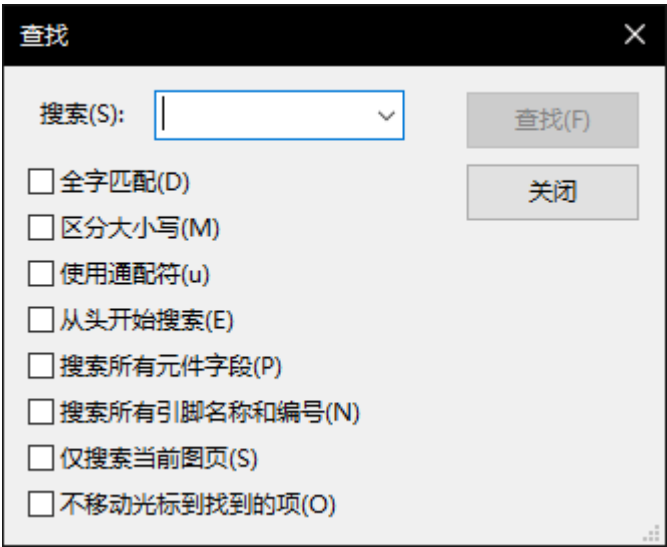
确定

取消

工作表号会自更新。您可以通过按“布日期”按左箭按将日期置今天，但不会自更改。

## 搜索工具

The Find icon (  ) can be used to access the search tool.



您可以在当前工作表或整个项目结构中搜索引用，或文本字符串。找到后，光标将定位在相关子表中的找到元素上。

## 网表工具

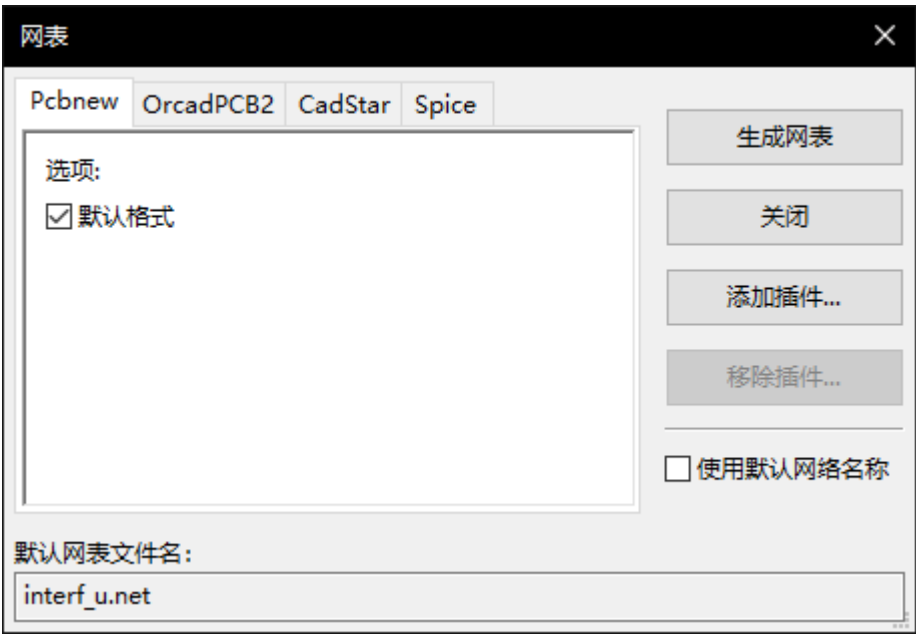
The Netlist icon (  ) opens the netlist generation tool.

该工具创建一个文件，描述整个项目结构中的所有连接。

在多表项目结构中，任何本地名称在其所属的工作表内可唯一。例如：表3的 LABEL1 与表5的 LABEL1 不同（如果没有故意引入连接以连接它们，是因为工作表名称路径在内部与本地名称相关）。

**NOTE** Even though there is no text length limit for labels in KiCad, please take into account that other programs reading the generated netlist may have such constraints.

**NOTE** Avoid spaces in labels, because they will appear as separated words in the generated file. It is not a limitation of KiCad, but of many netlist formats, which often assume that a label has no spaces.



...

默认格式	□中以□□ Pcbnew 作□默认格式。
------	----------------------

□可以生成其他格式:

- Orcad PCB2
- CadStar
- Spice (simulators)

可以添加外部插件来□展网表格式列表（上□中添加了 PadsPcb 插件）。

有关在《create-a-netlist, Create a Netlist》一章中□建网表的更多信息。

批注工具

The icon  launches the annotation tool. This tool assigns references to components.

□于多部件元件（例如包含4个□的 7400 TTL□□□分配了多部件后□□因此，指定□ U3 的 7400 TTL将分□ U3A, U3B, U3C 和 U3D）。

您可以无条件地批注所有元件或□批注新元件，即之前未批注的元件。

批注原理图

范围:

☒ 使用整个原理图
☐ 仅使用当前页面

顺序:

☒ X方向排序元件 (X)
☐ Y方向排序元件 (Y)



选项:

☒ 保持现有的批注
☐ 重置现有的批注
☐ 重置, 但保持多单元器件的顺序

编号:

☒ 使用该数字之后的编号: 
☐ 参考编号X100
☐ 参考编号X1000

☐ 保持对话框打开
☐ 不要求确认

批注

清除批注

关闭

批注信息:

显示: ☒ 所有 ☒ 错误 ☒ 警告 ☒ 相关信息 ☒ 活动

保存报告文件

范

使用整个原理	所有工作表都重新批注（默认）。
使用当前面	重新批注当前工作表（此在特殊情况下使用，例如 估当前表中的阻数量。
保留有批注	条件批注，只有新的 元件将被重新批注（默认）。
重置有批注	所有的无条件批注 元件将被重新批注（此将在那里使用 是重复的参考）。
重置，但不要交任何批注的多元部件	保持 当重新批注所有多个元例如U2A， U2B）在一起。

批注序

元件号的序（水平或垂直）。

批注

指定的参考格式。

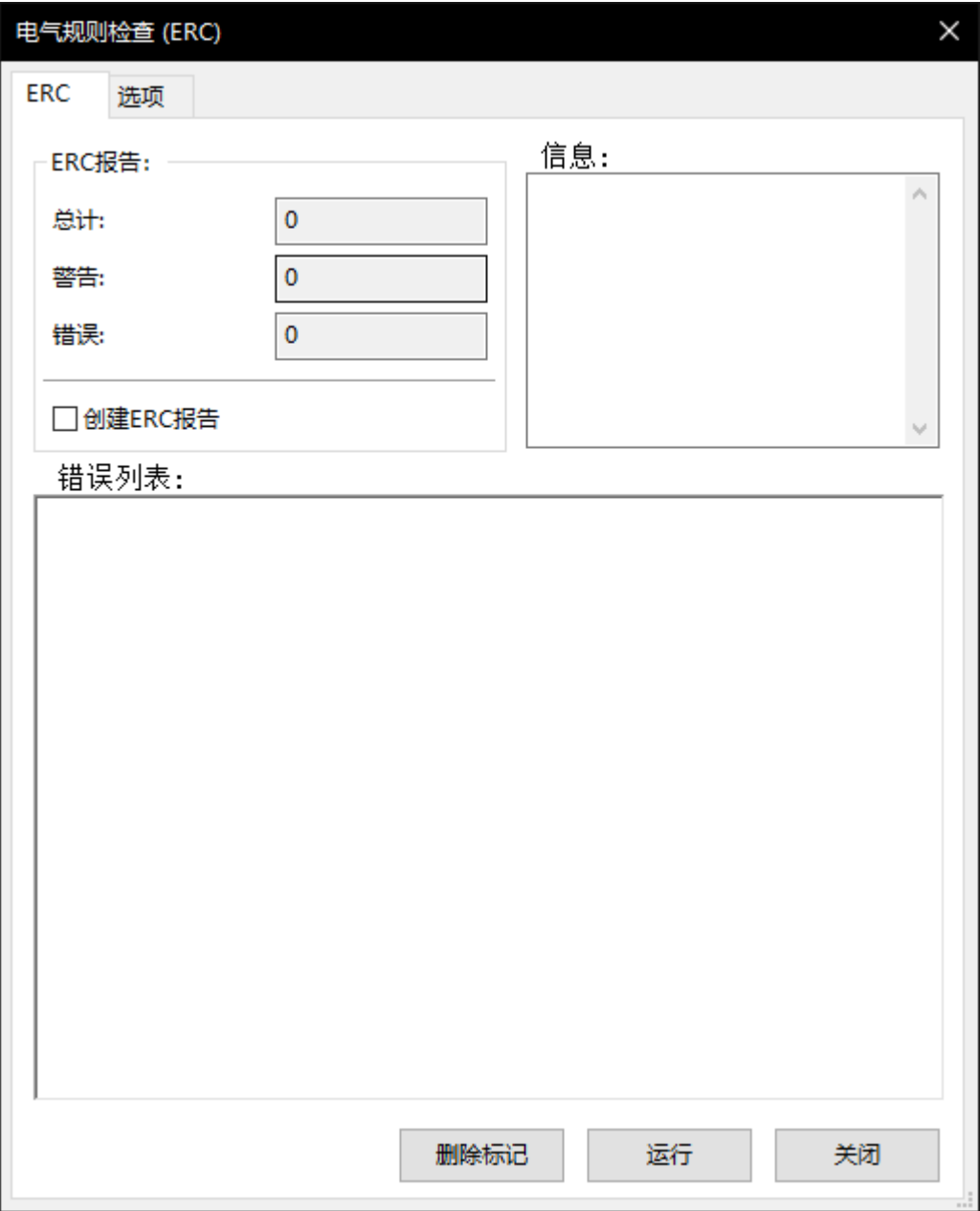
电气工具

The icon  launches the electrical rules check (ERC) tool.

工具行计能被忘的接和不一致。

Once you have run the ERC, KiCad places markers to highlight problems. The error description is displayed after left clicking on the marker. An error report file can also be generated.

主要 ERC 窗口



显示在 Electrical Rules Checker 窗口中：

- 错误和警告的数量。
- 错误计数。
- 警告计数。

注意

创建 ERC 文件警告	在此窗口中可生成 ERC 警告文件。
-------------	--------------------

命令：



除	除所有ERC/警告
运行	后气
关	关框。

- 消息将跳到原理中的相应

## ERC 框



此卡允您定义引脚之的接; 您可以每种情况3个

- 无
- 警告
- 

可以通修改元格的每个方格。


网

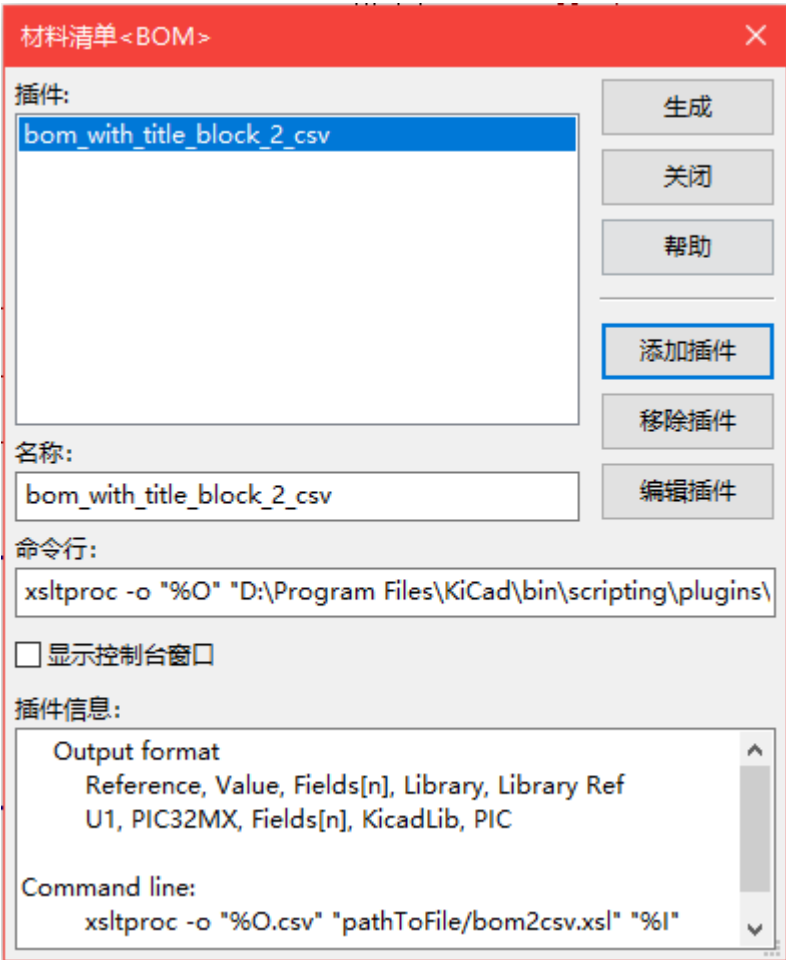
网类似网	网告网只有字母大小写（例如 lable/Lable/LaBeL）。网名称区分大小写，因此网些网被网网独的网网
网独特的全局网	网告网出网一次的全局网 特网。通常需要至少有两个网接。

命令：

初始化网默认网	恢复原始网置。
---------	---------

物料清单网工具

The icon  launches the bill of materials (BOM) generator. This tool generates a file listing the components and/or hierarchical connections (global labels).



The Schematic Editor’s BOM generator makes use of external plugins, either as XSLT or Python scripts. There are a few examples installed inside the KiCad program files directory.

用于 BOM 的网有用的元件属性包括：

- 网 - 使用的每个部件的唯一名称。
- 封装 - 手网入或反网注（网下文）。
- 字段1 - 制造商的名称。

字段2 - 制造商的元件号。

- 字段3 - 分销商的元件号。

例如：

符号属性

单元:  
A

方向 (度):  
☒ 0  
☐ +90  
☐ +180  
☐ -90

方向:  
☒ 默认  
☐ X轴镜像  
☐ Y轴镜像

☐ 转换形状

库符号:  
kit-dev-coldfire-xilinx\_5213\_schlib:CONN\_13X2  

验证 修改

符号 ID:  
461BAEE7

编辑Spice模型

重置字段属性

更新字段值

字段:

名称	值
参考标识符	BDM_PORT1
值	CONN_13X2
封装	Connector_PinHeader_2.54m...
数据手册	

↑

↓

←

→

水平位置:  
☐ 左对齐  
☒ 居中  
☐ 右对齐

垂直位置:  
☐ 顶部对齐  
☒ 居中  
☐ 底部对齐

可见性:  
☐ 显示  
☐ 旋转

字体风格:  
☒ 标准  
☐ 斜体  
☐ 加粗  
☐ 加粗斜体

字段名称:  
数据手册

字段值:  

显示Datasheet数据手册

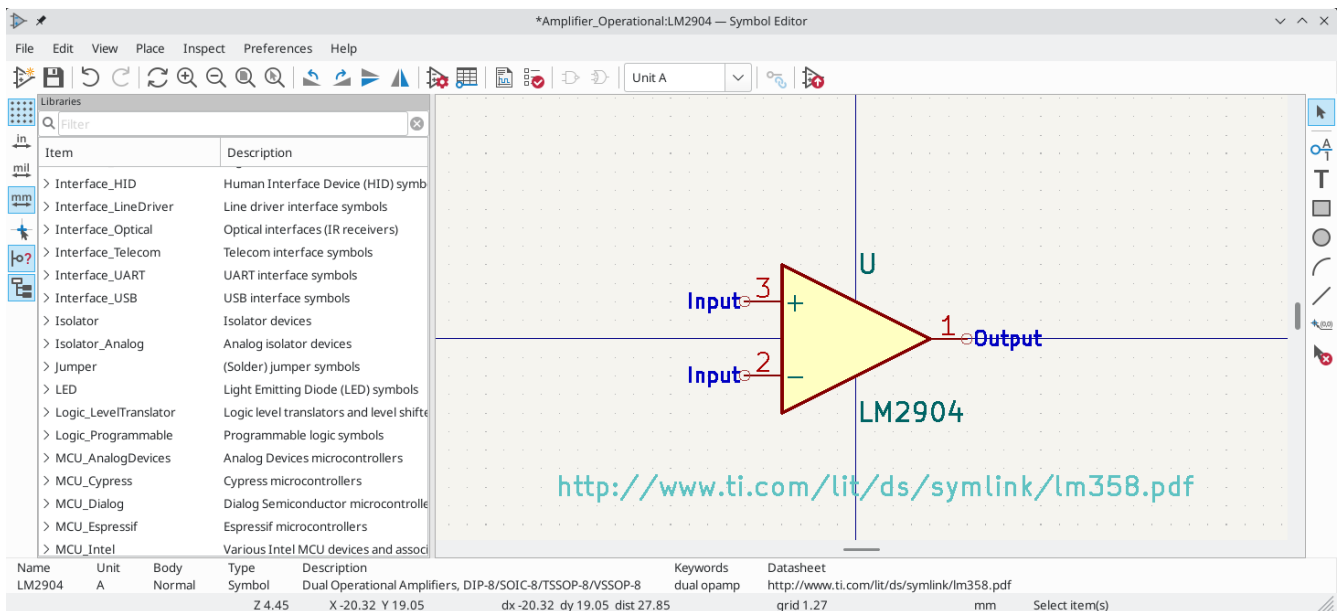
字体大小: 1.524 mm

位置 X: 0.000 mm

位置 Y: 0.000 mm

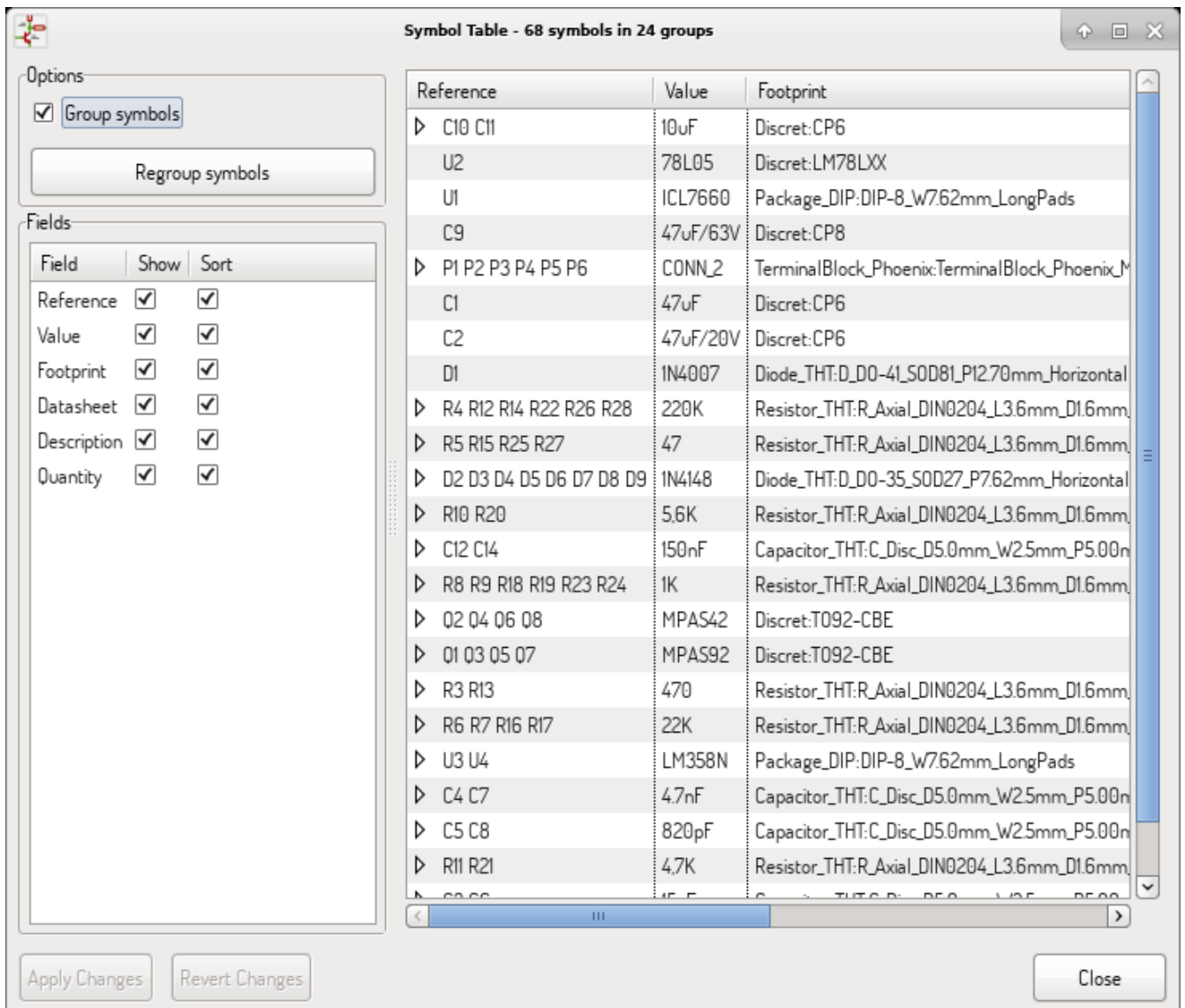
确定 取消

在 MS Windows 上，BOM 生成器窗口有一个特殊按钮（由蓝色箭头指示），用于控制外部插件窗口的可见性。+ 默认情况下，BOM 生成器命令窗口控制台窗口隐藏，点击重定向到 *Plugin info* 字段。点击此按钮可显示正在运行的命令的窗口。如果插件提供了图形用户界面，这可能是必要的。



## 字段工具

The icon  opens a spreadsheet to view and modify field values for all symbols.



修改字段后，您需要通过“应用”按钮接受更改，或通过“恢复”按钮撤消更改。

化学段填充的技巧

子表格中有几种特殊的复制/粘贴方法。在入在少数元件中重复的字段它可能很有用。

些方法如下所示。

复制 (Ctrl+C)		黏 (Ctrl+V)																																																
<table><tr><td>abc</td><td></td><td></td></tr><tr><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td></tr></table>	abc															<table><tr><td>abc</td><td></td><td></td></tr><tr><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td></tr></table>	abc															<table><tr><td>abc</td><td></td><td></td></tr><tr><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td></tr><tr><td>abc</td><td>abc</td><td></td></tr><tr><td>abc</td><td>abc</td><td></td></tr><tr><td></td><td></td><td></td></tr></table>	abc									abc	abc		abc	abc				
abc																																																		
abc																																																		
abc																																																		
abc	abc																																																	
abc	abc																																																	
<table><tr><td>11</td><td>12</td><td>13</td></tr><tr><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td></tr></table>	11	12	13													<table><tr><td>11</td><td>12</td><td>13</td></tr><tr><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td></tr></table>	11	12	13													<table><tr><td>11</td><td>12</td><td>13</td></tr><tr><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td></tr><tr><td>11</td><td>12</td><td>13</td></tr><tr><td>11</td><td>12</td><td>13</td></tr><tr><td>11</td><td>12</td><td>13</td></tr></table>	11	12	13							11	12	13	11	12	13	11	12	13
11	12	13																																																
11	12	13																																																
11	12	13																																																
11	12	13																																																
11	12	13																																																
11	12	13																																																
<table><tr><td>11</td><td></td><td></td></tr><tr><td>21</td><td></td><td></td></tr><tr><td>31</td><td></td><td></td></tr><tr><td>41</td><td></td><td></td></tr><tr><td>51</td><td></td><td></td></tr></table>	11			21			31			41			51			<table><tr><td>11</td><td></td><td></td></tr><tr><td>21</td><td></td><td></td></tr><tr><td>31</td><td></td><td></td></tr><tr><td>41</td><td></td><td></td></tr><tr><td>51</td><td></td><td></td></tr></table>	11			21			31			41			51			<table><tr><td>11</td><td>11</td><td>11</td></tr><tr><td>21</td><td>21</td><td>21</td></tr><tr><td>31</td><td>31</td><td>31</td></tr><tr><td>41</td><td>41</td><td>41</td></tr><tr><td>51</td><td>51</td><td>51</td></tr></table>	11	11	11	21	21	21	31	31	31	41	41	41	51	51	51			
11																																																		
21																																																		
31																																																		
41																																																		
51																																																		
11																																																		
21																																																		
31																																																		
41																																																		
51																																																		
11	11	11																																																
21	21	21																																																
31	31	31																																																
41	41	41																																																
51	51	51																																																
<table><tr><td>11</td><td>12</td><td></td></tr><tr><td>21</td><td>22</td><td></td></tr><tr><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td></tr></table>	11	12		21	22											<table><tr><td>11</td><td>12</td><td></td></tr><tr><td>21</td><td>22</td><td></td></tr><tr><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td></tr></table>	11	12		21	22											<table><tr><td>11</td><td>12</td><td></td></tr><tr><td>21</td><td>22</td><td></td></tr><tr><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td></tr></table>	11	12		21	22													
11	12																																																	
21	22																																																	
11	12																																																	
21	22																																																	
11	12																																																	
21	22																																																	
<table><tr><td>11</td><td>12</td><td>13</td></tr><tr><td>21</td><td>22</td><td>23</td></tr><tr><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td></tr></table>	11	12	13	21	22	23										<table><tr><td>11</td><td>12</td><td>13</td></tr><tr><td>21</td><td>22</td><td>23</td></tr><tr><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td></tr></table>	11	12	13	21	22	23										<table><tr><td>11</td><td>12</td><td>13</td></tr><tr><td>21</td><td>22</td><td>23</td></tr><tr><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td></tr></table>	11	12	13	21	22	23												
11	12	13																																																
21	22	23																																																
11	12	13																																																
21	22	23																																																
11	12	13																																																
21	22	23																																																

NOTE 些技巧也可以在具有网格控制元素的其他框中使用。

用于封装分配的入工具

□□□

The icon  launches the back-annotate tool.

This tool allows footprint changes made in the PCB Editor to be imported back into the footprint fields in the Schematic Editor.

# 管理符号

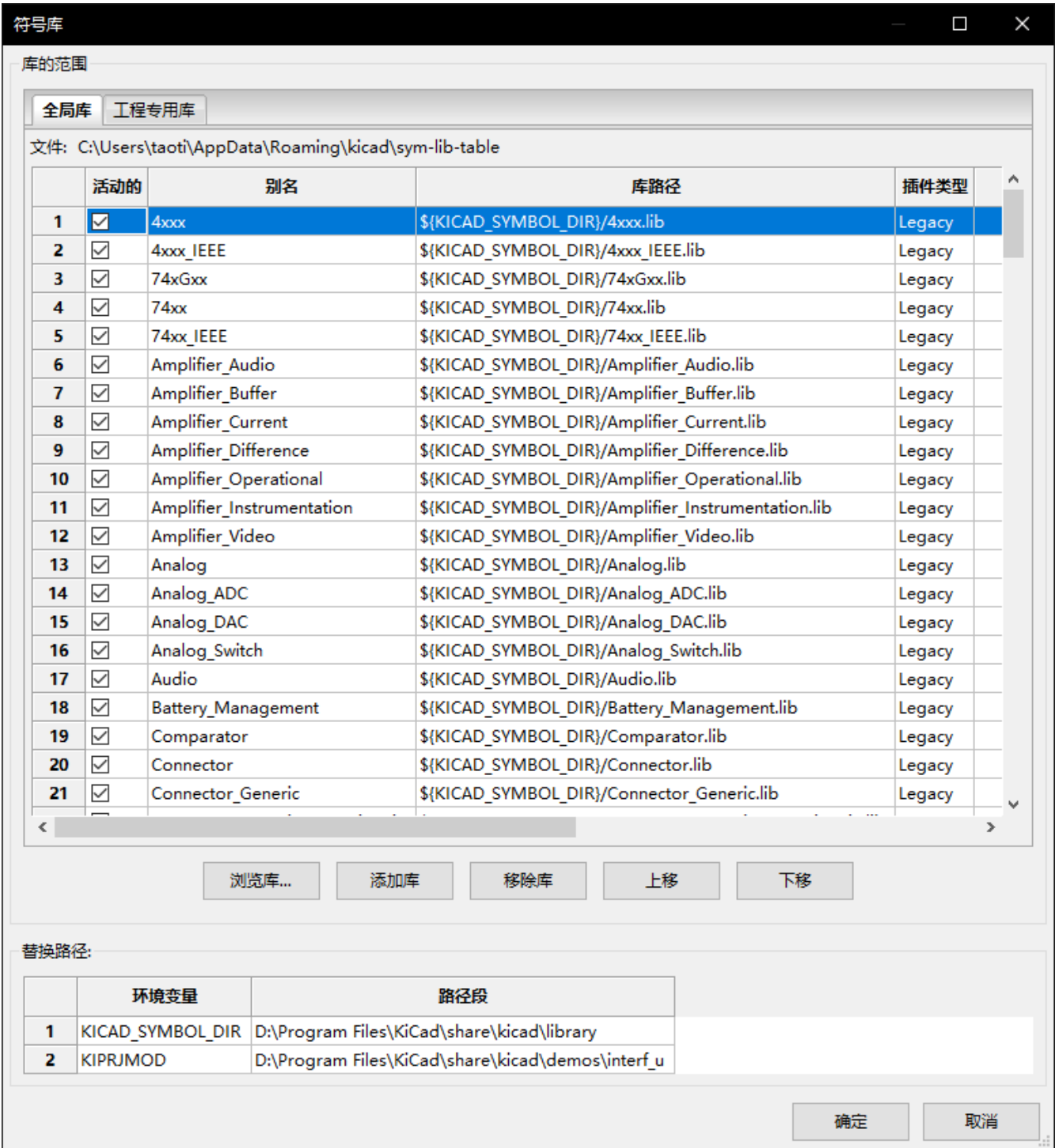
符号包含构建原理使用的符号集合。原理中的每个符号由一个全名唯一标识。全名由昵称和符号名称组成。一个例子是“音AD1853”。

## 符号表

符号表包含 KiCad 知道的所有文件的列表。符号表由全局符号表文件和项目特定符号表文件构成。

When a symbol is loaded, KiCad uses the library nickname, `Audio` in our example, to lookup the library location in the symbol library table.

The image below shows the symbol library table editing dialog which can be opened by invoking the **Manage Symbol Libraries...** entry in the **Preferences** menu.



## 全局符号表

The global symbol library table contains the list of libraries that are always available regardless of the currently loaded project file. The table is saved in the file `sym-lib-table` in the user's KiCad configuration folder. The [location of this folder](#) is dependent upon the operating system being used.

## 项目特定符号表

项目特定符号表包含用于当前项目的文件的列表。项目特定符号表只能在与项目文件一起加载时行。如果未加载项目文件或当前项目路径中没有符号表文件，会创建一个空表，可以对其行然后将其与项目文件一起保存。

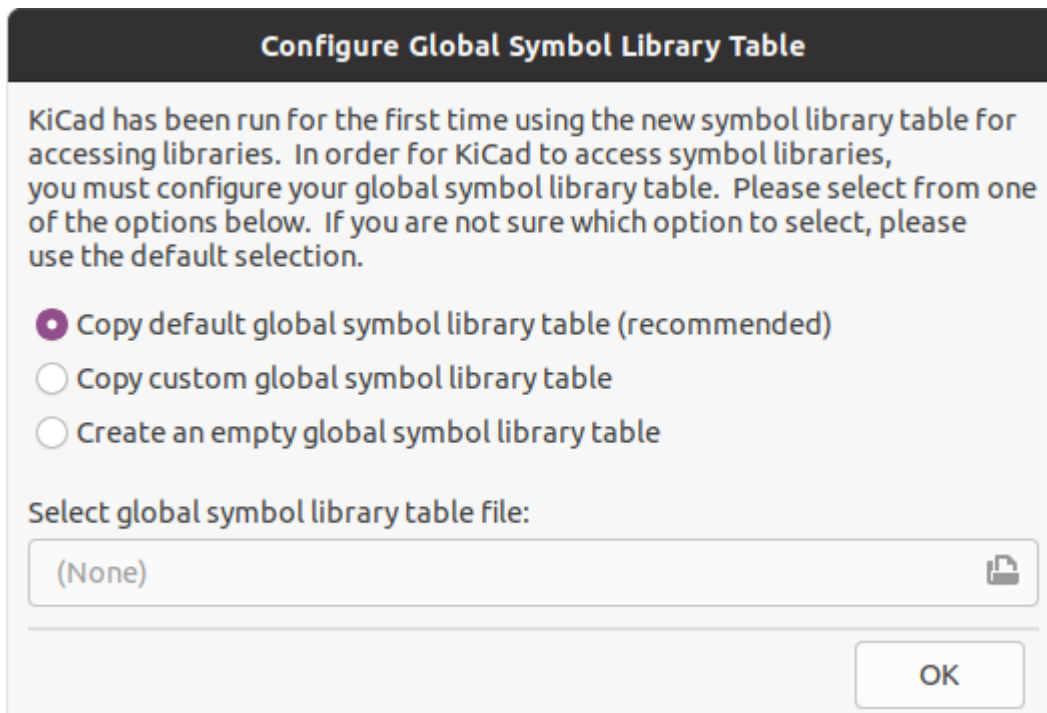
## 初始配置

The first time the KiCad Schematic Editor is run and the global symbol table file `sym-lib-table` is not found in the KiCad configuration folder, KiCad will present the "Configure Global Symbol Library Table" dialog to



the user. The dialog presents the user with three options.

- **Copy default global symbol library table (recommended).** If this option is selected, KiCad will copy the default symbol library table file stored in the system's Kicad template folder to the file `sym-lib-table` in the user's KiCad configuration folder. If the default template `sym-lib-table` file cannot be found, this option will be grayed out. The missing default table is usually caused by the KiCad default libraries not being installed (on some systems they are installed by a separate package). If the libraries are installed in a non-standard location, use the second option and browse to the library table location manually.
- **Copy custom global symbol library table.** If this option is selected, the user must browse to the desired symbol library table file, which will be copied to the user's KiCad configuration directory.
- **Create an empty global symbol library table.** An empty symbol library table file will be created in the user's KiCad configuration directory. The user must add libraries to the table manually.



#### NOTE

默认符号表包括作 KiCad 的一部分安装的所有符号。根据用途和系的速度，可能是也可能不是所希望的。加符号所需的与符号表中的数量成正比。如果符号加从全局表中除很少和/或从未使用的并根据需要将它添加到目表中。

## 添加表

了使用符号，必首先将其添加到全局表或目特定表中。特定于目的表适用于打开目文件的情况。

#### NOTE

Each library entry must have a unique nickname.

The library nickname does not have to be related in any way to the actual library file name or path. The colon : and \ characters cannot be used anywhere in the library nickname. Each library entry must have a valid path and/or file name depending on the type of library. Paths can be defined as absolute, relative, or by environment variable substitution (see section below).

The appropriate library format must be selected in order for the library to be properly read. "KiCad" format is used for KiCad version 6 libraries ( `.kicad_sym` files), while "Legacy" format is used for libraries from older versions of KiCad ( `.lib` files). Legacy libraries are read-only, but can be migrated to KiCad format libraries using the **Migrate Libraries** button (see section [Migrating Legacy Libraries](#)).

□有一个描述字段用于添加□条目的描述。此□不使用□□字段，因此在加□□□添加□□将不起作用。

- □注意，您不能在同一个表中包含重复的□昵称。但是，您可以在全局和□目特定的符号□表中包含重复的□昵称。
- 当出□重复的昵称□□□目特定的表条目将□先于全局表条目。
- 在□目特定表中定义条目□□包含□些条目的 `sym-lib-table` 文件将写入当前打开的□目文件的文件□中。

## □境□量替代

One of the most powerful features of the symbol library table is environment variable substitution. This allows for definition of custom paths to where symbol libraries are stored in environment variables. Environment variable substitution is supported by using the syntax `${ENV_VAR_NAME}` in the library path.

By default, at run time KiCad defines two environment variables relevant for locating symbol libraries:

- the `$KIPRJMOD` environment variable that always points to the currently open project directory. `$KIPRJMOD` cannot be modified.
- the `$KICAD6_SYMBOL_DIR` environment variable. This points to the path where the default symbol libraries that were installed with KiCad.

You can override `$KICAD6_SYMBOL_DIR` by redefining it in **Preferences** → **Configure Paths...** This is useful for using libraries installed in a nonstandard location.

`$KIPRJMOD` allows you to store libraries in the project path without having to define the absolute path (which is not always known) to the library in the project specific symbol library table.

## 使用模式

Symbol libraries can be defined either globally or specifically to the currently loaded project. Symbol libraries defined in the user's global table are always available and are stored in the `sym-lib-table` file in the user's KiCad configuration folder. The project-specific symbol library table is active only for the currently open project file.



每种方法都有□点和缺点。在全局表中定义所有□意味着它□将在需要□始□可用。□□做的缺点是加□□□会增加。

在□目特定的基□上定义所有符号□意味着您只有□目所需的□□□会减少符号□加□□□□缺点是您必□始□□住添加每个□目所需的每个符号□□

一种使用模式是全局定义常用□□而□只需要□目特定□表中的□目。□如何定义□没有限制。

## Migrating Legacy Libraries

Legacy libraries ( `.lib` files) are read-only, but they can be migrated to KiCad version 6 libraries ( `.kicad_sym`). KiCad version 6 libraries cannot be viewed or edited by KiCad versions older than 6.0.0.

Legacy libraries can be converted to KiCad 6 libraries by selecting them in the symbol library table and clicking the **Migrate Libraries** button. Multiple libraries can be selected and migrated at once by -clicking or -clicking.

Libraries can also be converted one at a time by opening them in the Symbol Editor and saving them as a new library.

## ❏留❏目重新映射

When loading a schematic created prior to the symbol library table implementation, KiCad will attempt to remap the symbol library links in the schematic to the appropriate library table symbols. The success of this process is dependent on several factors:

- 原理❏中使用的原始❏仍然可用，并且在符号添加到原理❏❏保持不❏❏
- 在❏❏到所有恢复行❏❏❏❏行所有恢复行❏以❏建恢复❏或使❏有恢复❏保持最新状❏❏
- ❏目符号❏存❏的完整性尚未❏坏。

WARNING	重新映射将❏份在重新映射期❏在❏目文件❏中的 rescue-backup 文件❏中更改的所有文件。在重新映射之前，❏❏必❏份❏目以防万一出❏❏
WARNING	即使已禁用恢复操作以❏行恢复操作以确保正确的符号可用于重新映射。❏勿取消此操作，否❏重映射将无法正确重新映射原理❏符号。任何❏坏的符号❏接都必❏手❏修复。
NOTE	If the original libraries have been removed and the rescue was not performed, the cache library can be used as a recovery library as a last resort. Copy the cache library to a new file name and add the new library file to the top of the library list using a version of KiCad prior to the symbol library table implementation.

# 原理图构建和

## 简介

原理图可以用单页表示，但是，如果足够大，它需要多页。

A schematic represented by several sheets is hierarchical, and all its sheets (each one represented by its own file) constitute a complete KiCad schematic. The manipulation of hierarchical schematics will be described in the [Hierarchical Schematics](#) chapter.

## 一般考虑

A schematic designed with KiCad is more than a simple graphic representation of an electronic device. It is normally the entry point of a development chain that allows for:


- 通过执行ERC检查电气连接以发现和漏。
- 自动生成物料清单 (BOM)。
- 用于仿真元件（如 SPICE）的（构建 - 定制 - 网表和文件 - 文件，生成网表）。
- [Defining a circuit](#) for transferring to PCB layout.

原理图主要由符号，连接点，和源端组成。为了清晰起见您可以放置纯粹的形状元素，如条目，注释和折线。

Symbols are added to the schematic from symbol libraries. After the schematic is made, the set of connections and footprints is imported into the PCB editor for designing a board.

## 符号放置和

### 找到并放置一个符号

To load a symbol into your schematic you can use the icon . A dialog box allows you to type the name of the symbol to load.



“选择符号”对话框将根据您在搜索字段中输入的内容按名称，关键字和符号名称。只需输入高电平器件即可使用它。

- **通配符**：分使用字符“?”和“\*”表示“任意字符”和“任意数量的字符”。
- **关键字**：如果部分的描述或关键字包含格式“Key : 123”，您可以通过输入相匹配“Key> 123”（大于），“Key<123”（小于）等。数字可能包括以下不区分大小写的后缀之一：

p	n	u	m	k	meg	g	t
10 <sup>-12</sup>	10 <sup>-9</sup>	10 <sup>-6</sup>	10 <sup>-3</sup>	10 <sup>3</sup>	10 <sup>6</sup>	10 <sup>9</sup>	10 <sup>12</sup>

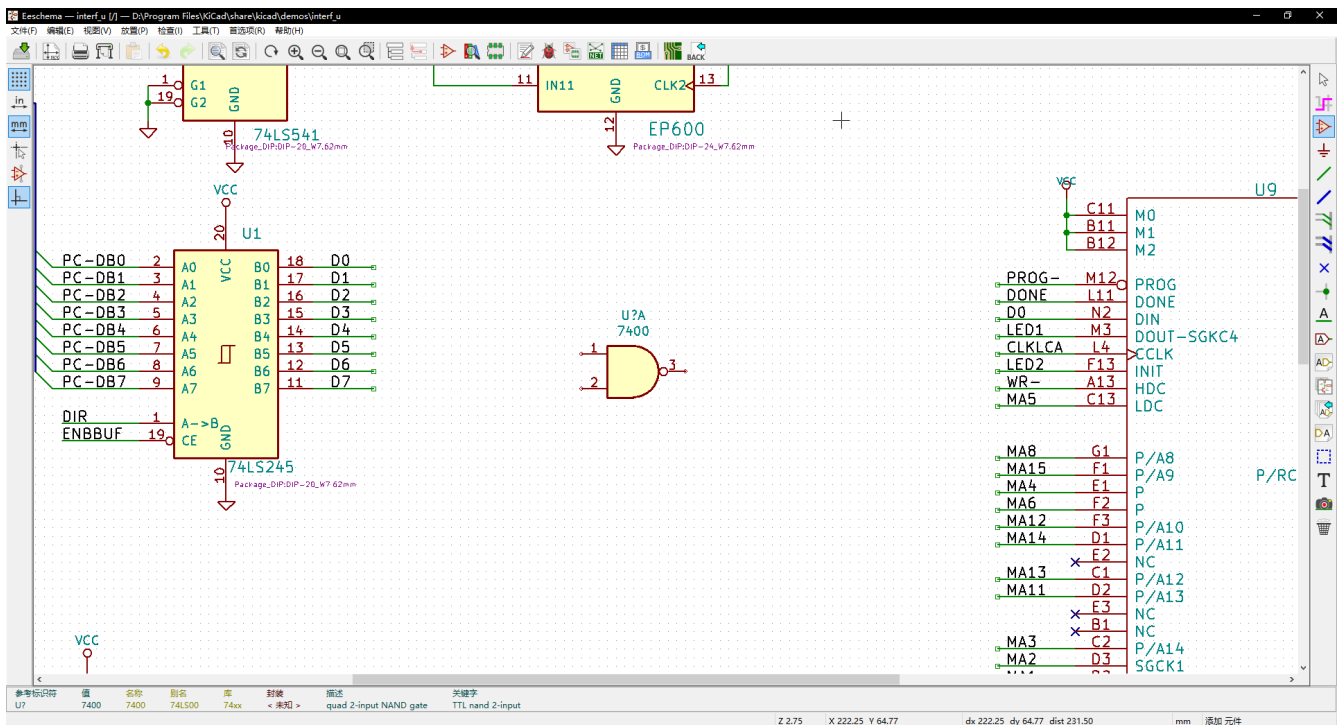
ki	mi	gi	ti
2 <sup>10</sup>	2 <sup>20</sup>	2 <sup>30</sup>	2 <sup>40</sup>

- **正则表达式**：如果你熟悉正则表达式，它也可以使用。使用的正则表达式是 wxWidgets 高正则表达式 类似于 Perl 常表达式。

If the symbol specifies a default footprint, this footprint will be previewed in the lower right. If the symbol includes footprint filters, alternate footprints that satisfy the footprint filters can be selected in the footprint dropdown menu at right.

After selecting a symbol to place, the symbol will be attached to the cursor. Left clicking the desired location in the schematic places the symbol into the schematic. Before placing the symbol in the schematic, you can rotate it, mirror it, and edit its fields, by either using the hotkeys or the right-click context menu. These actions can also be performed after placement.

□是放置期□的符号：



If the "Place repeated copies" option is checked, after placing a symbol KiCad will start placing another copy of the symbol. This process continues until the user presses **Esc**.

For symbols with multiple units, if the "Place all units" option is checked, after placing the symbol KiCad will start placing the next unit in the symbol. This continues until the last unit has been placed or the user presses **Esc**.

## Placing power ports

A **power port symbol** is a symbol representing a connection to a power net. The symbols are grouped in the **power** library, so they can be placed using the symbol chooser. However, as power placements are frequent, the **⏏** tool is available. This tool is similar, except that the search is done directly in the **power** library.

## 符号□□和修改（已放置的元素）

□□符号有两种方法：

- 符号本身的修改：多□元符号上的位置，方向，□位□□□
- 修改符号的其中一个字段：引用，□□覆盖区等。

□□放置符号□□您可能需要修改其□□特□是□阻器，□容器等），但是立即□其分配参考□号或□□□元是没有用的（除了元件之外）□定□位，您必□手□分配）。□可以通□批注功能自□完成。

## 符号修改

要修改符号的某些功能，□将光□放在符号上，然后□行以下任一操作：

双击符号以打开完整的属性框。

- 右键单击以打开上下文菜单并使用以下命令之一：移动方向，删除等。
- Use a hotkey to perform an action on the symbol (E to open the properties dialog, R to rotate, etc.). Note that hotkeys act on the selected symbol; if no symbol is selected hotkeys act on the symbol under the cursor.

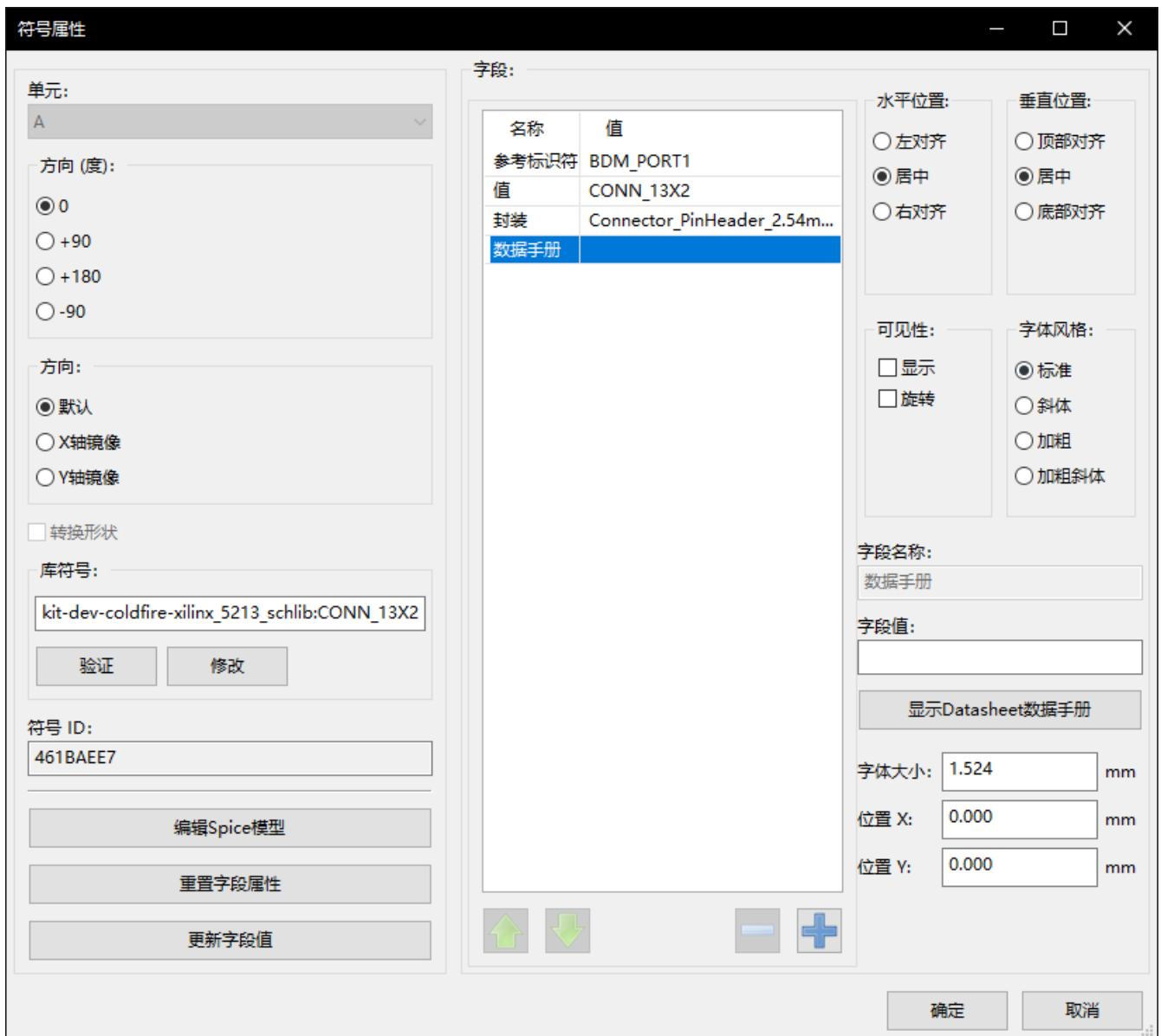
Symbols can also be selected by clicking on them or drag-selecting them. Selected symbols can be modified by clicking relevant buttons in the top toolbar or using a hotkey.

## 文本字段修改

您可以修改字段的参考，位置，方向，文本大小和可编辑性：

- 双击文本字段进行编辑。
- 右键单击以打开上下文菜单并使用以下命令之一：移动旋转删除等。
- Position the cursor over the field (if nothing is selected) or select the field and press E to edit the field.
- Position the cursor over the symbol (if nothing is selected) or select the symbol and press V, U, or F hotkeys to directly edit the symbol's value, reference designator, or footprint fields, respectively.

要获得更多帮助或者要新建字段，双击符号以打开 符号属性 框。



每个字段都可以是可见的或隐藏的，并且可以水平或垂直显示。始终正常显示的符号（无旋转或镜像）指示显示的位置，并且相对于符号的锚点。

The position and orientation properties of each field may be hidden in this dialog. They can be shown by right-clicking on the column header of the fields table and enabling the "Orientation", "X Position", and/or "Y Position" columns. Other columns can be shown or hidden as desired.

The "Update Symbol from Library..." button is used to update the schematic's copy of the symbol to match the copy in the library. The "Change Symbol..." button is used to swap the current symbol to a different symbol in the library.

"Edit Symbol..." opens the Symbol Editor to edit the copy of the symbol in the schematic. Note that the original symbol in the library will not be modified. The "Edit Library Symbol..." button opens the Symbol Editor to edit the original symbol in the library. In this case, the symbol in the schematic will not be modified until the user clicks the "Update Symbol from Library..." button.

## Electrical Connections



## 简介

There are a number of elements that can be added to a schematic to electrically connect components. All of these elements can be placed with the buttons on the vertical right toolbar or using hotkeys.

有些元素是:

- **Wires:** direct connection between pins.
- **Buses:** connections for a group of signals.
- **Bus entries:** connections between wires and buses.
- **No-connection flags:** terminations for pins or wires that are intentionally unconnected. These flags prevent ERC violations for unconnected pins.
- **Junctions:** connections between crossing wires or buses.
- **Net labels:** local name for a signal. Signals within a sheet that have the same net label are connected.
- **Global labels:** global name for a signal. Signals with the same global label are connected even if they are not in the same sheet.
- **Hierarchical labels:** a label for a signal in a subsheet that enables the signal to be accessed in a parent sheet. See the [Hierarchical Schematics](#) section for more information about hierarchical labels, sheets, and pins.
- **Hierarchical sheets:** an instantiation of a subsheet within a parent sheet. The parent sheet can connect to the subsheet through the subsheet's hierarchical pins.
- **Hierarchical pins:** connection points between a parent sheet and a subsheet. Hierarchical pins appear at the parent sheet's level and correspond to hierarchical labels in the subsheet.

Several other types of items can be placed on the schematic but do not affect connectivity:

- **Graphical lines:** graphical lines for presentation.
- **Text:** textual comments and annotations.
- **Bitmap images:** raster graphics from an external file.

This section will also discuss two special types of symbols that can be added with the "Power port" button on the right toolbar:

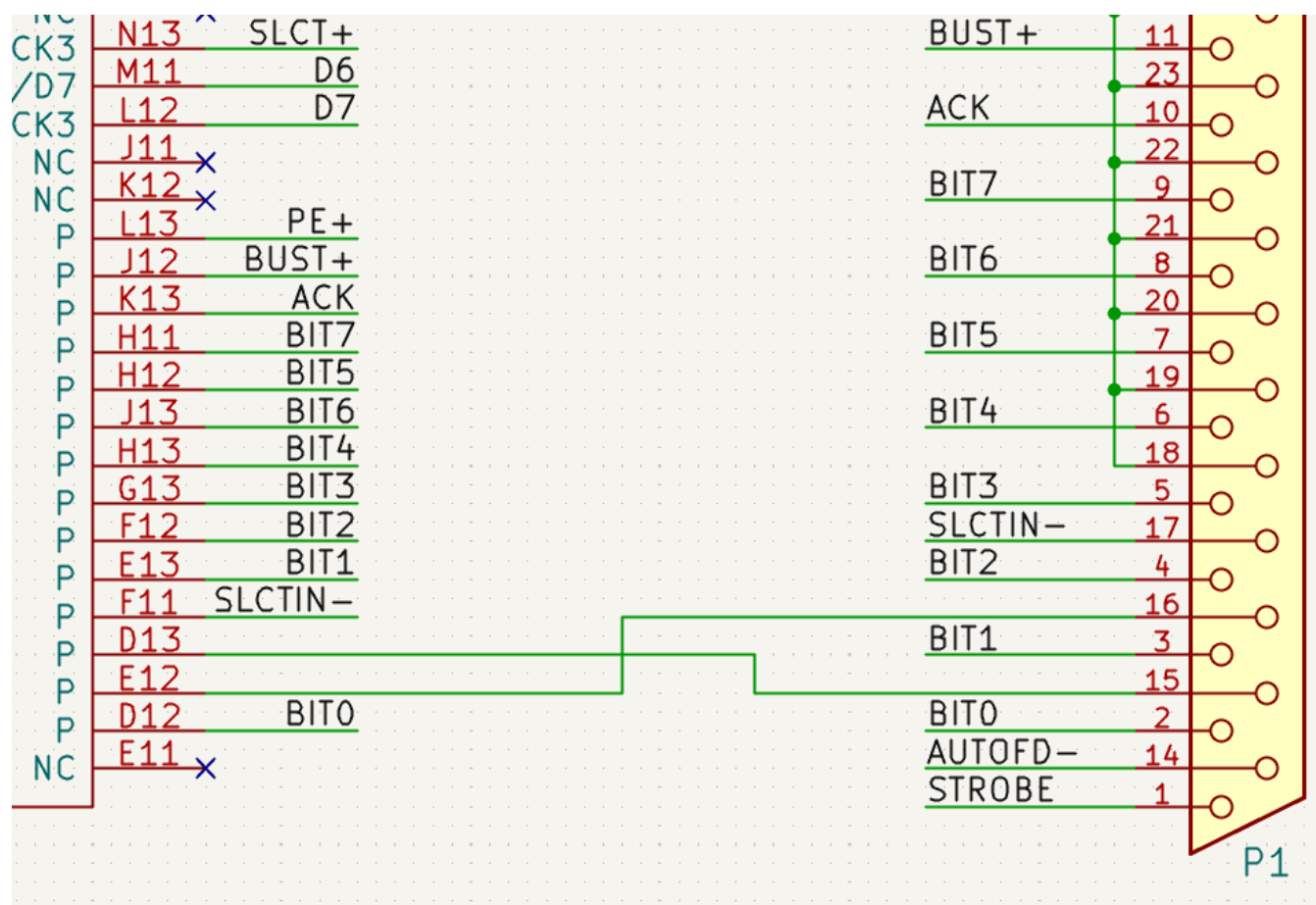
- **Power ports:** symbols for connecting wires to a power or ground net.
- **PWR\_FLAG:** a specific symbol for indicating that a net is powered when it is not connected to a power output pin (for example, a power net that is supplied by an off-board connector).

## 连接 (引脚和引脚)

有两种方法可以建立连接:

- 引脚到引脚的连接
- 连接

下⬇️示了⬇️两种方法:



Label Connections

The point of "contact" of a label is the small square in the corner of the label. The square disappears when the label is connected. The position of the connection point relative to the label text can be changed by choosing a different label orientation in the label properties, or by mirroring/rotating the label.

The label’s connection point must be in contact with a wire or the end of a pin for the label to be connected.

Wire Connections

要建立⬇️接，必⬇️将一段⬇️的两端⬇️接到另一个段或一个引脚。

如果有重叠（如果⬇️通⬇️引脚，但没有⬇️接到引脚端）⬇️没有⬇️接。

- NOTE

Wires connect with other wires or pins only if their ends coincide exactly. Therefore it is important to keep symbol pins and wires aligned to the grid. It is recommended to always use a 50 mil grid when placing symbols and drawing wires because the KiCad standard symbol library and all libraries that follow its style also use a 50 mil grid.
- NOTE

Symbols, wires, and other elements that are not aligned to the grid can be snapped back to the grid by selecting them, right clicking, and selecting **Align Elements to Grid**.

Wire Junctions

Wires that cross are not implicitly connected. It is necessary to join them with a junction dot if a connection is desired. Junction dots will be automatically added to wires that start or end on top of an existing wire.

Junction dots are used in the previous figure on the wires connected to P1 pins 18, 19, 20, 21, 22, and 23.

## Nets with Multiple Names

A signal can only have one name. If two different labels are placed on the same net, an ERC violation will be generated. Only one of the net names will be used in the netlist.

## Hidden Power Pins

When the power pins of a symbol are visible, they must be connected, as with any other signal.


However, symbols such as gates and flip-flops are sometimes drawn with hidden power input pins which are connected implicitly.

KiCad automatically connects invisible pins with type "power input" to a global net with the same name as the pin. For example, if a symbol has a hidden power input pin named VCC, this pin will automatically be connected to the global VCC net.

### NOTE

Care must be taken with hidden power input pins because they can create unintentional connections. By nature, hidden pins are invisible and do not display their pin name. This makes it easy to accidentally connect two power pins to the same net. For this reason, the use of invisible power pins in symbols is not recommended outside of power port symbols, and is only supported for compatibility with legacy designs and symbols.

### NOTE

Hidden pins can be shown in the schematic by checking the **Show hidden pins** option in the **Schematic Editor** → **Display Options** section of the preferences, or by selecting **View** → **Show hidden pins**. There is also a toggle icon  on the left (options) toolbar.

It may be necessary to join power nets of different names (for example, GND in TTL components and VSS in MOS components). To accomplish this, add a [power port symbol](#) for each net and connect them with a wire.

It is not recommended to use labels for power connection. These only have a "local" connection scope, and will not connect to invisible power pins.

## Wiring

To begin connecting elements, you may either use the 'Wire' or 'Bus' tools from the right-hand toolbar, or you can auto-start a new wire from any existing pin or unconnected wire.

The wire drag action will drag the entire wire if you start dragging from the middle of the wire. Alternatively, it will drag just one corner if you start the drag action over a corner where two wires connect

## □接 (□□□

在下面的原理□中, □多引脚□接到□□□



Bus wires can be drawn and connected in the same manner as signal wires, including using junctions to create connections between crossing wires. Like signals, buses cannot have more than one name — if two conflicting labels are attached to the same bus, an ERC violation will be generated.

## 成之的接

Pins connected between the same members of a bus must be connected by labels. It is not possible to connect a pin directly to a bus; this type of connection will be ignored by KiCad.

在上面的示例中，接是通放置在接到引脚的上的行的。到入口（45度段）是形化的，并不是形成接所必需的。

In fact, using the repetition command (Insert), connections can be very quickly made in the following way, if component pins are aligned in increasing order (a common case in practice on components such as memories, microprocessors...):

- Place the first label (for example PCA0)
- Use the repetition command as much as needed to place members. KiCad will automatically create the next labels (PCA1, PCA2 ...) vertically aligned, theoretically on the position of the other pins.
- 在第一个下画 然后使用重复命令将其他放在下。
- 如果需要，以相同的方式放置条目（放置第一个条目，然后使用重复命令）。

### NOTE

In the **Schematic Editor** → **Editing Options** section of the Preferences menu, you can set the repetition parameters:

- Horizontal pitch.
- Vertical pitch.
- Label increment (labels can be incremented or decremented by 1, 2, 3, etc.).

## 正在展开

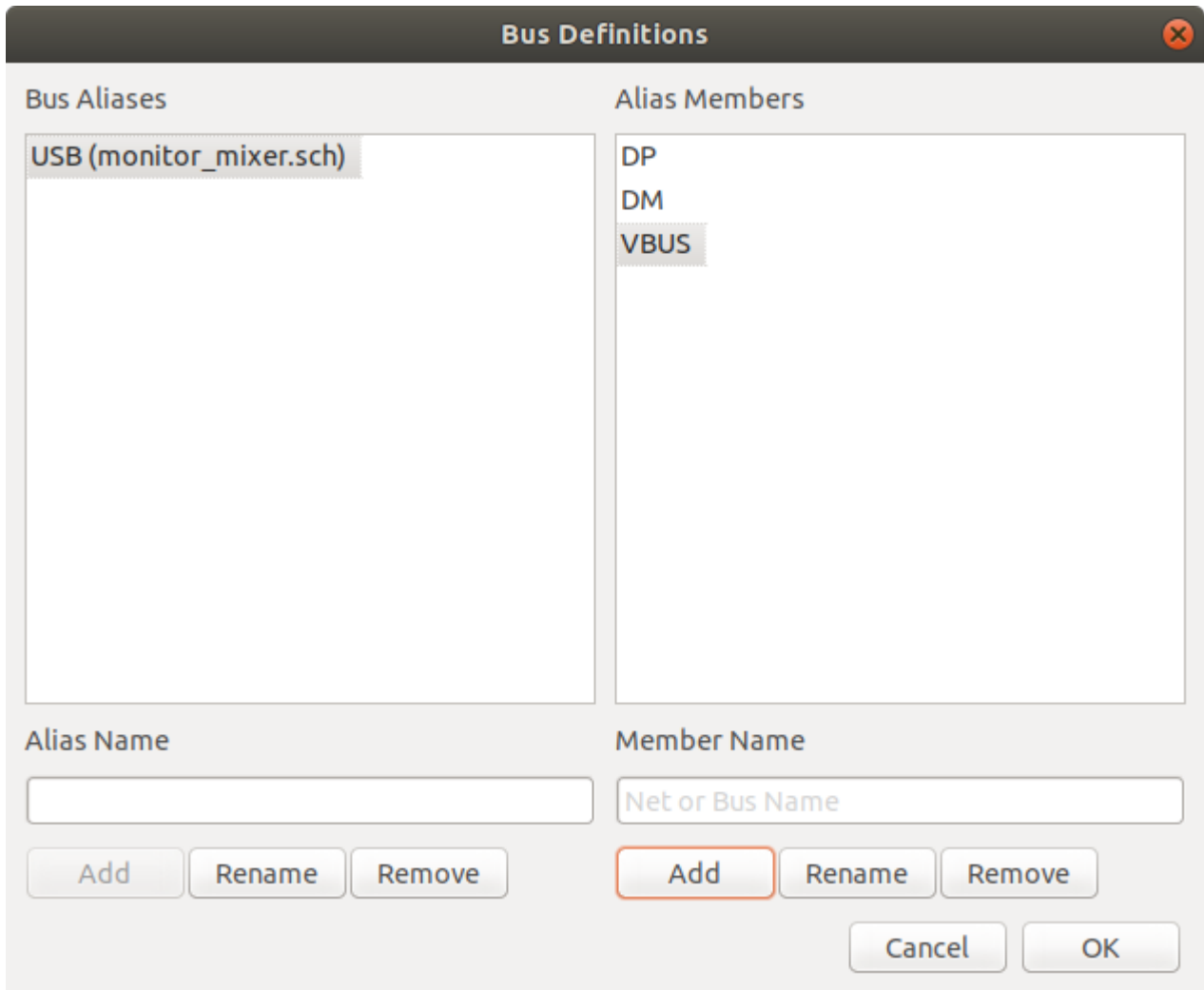
The unfold tool allows you to quickly break out signals from a bus. To unfold a signal, right-click on a bus object (a bus wire, etc) and choose **Unfold from Bus**. Alternatively, use the **Unfold Bus** hotkey (default: C) when the cursor is over a bus object. The menu allows you to select which bus member to unfold.

成后，下一次将把成放在所需位置。工具自生成入口和通向位置。放置后，您可以放置其他段（例如，接到件引脚）并以任何正常方式完成

## 名

名是一种快捷方式，可让您更有效地使用大型 它允您定义并其指定一个短的名称，然后可以在原理中使用名称而不是完整的名。

To create bus aliases, open the **Bus Definitions** dialog in the **Tools** menu.



别名可以被命名任何有效的信号名称。使用列表框，您可以向别名添加信号或矢量。作为一种快捷方式，您可以输入或粘贴由空格分隔的信号和/或列表，并将它们全部添加到别名定义中。在这个例子中，我定义了一个名为“USB”的别名，其成员为“DP”，“DM”和“VBUS”。

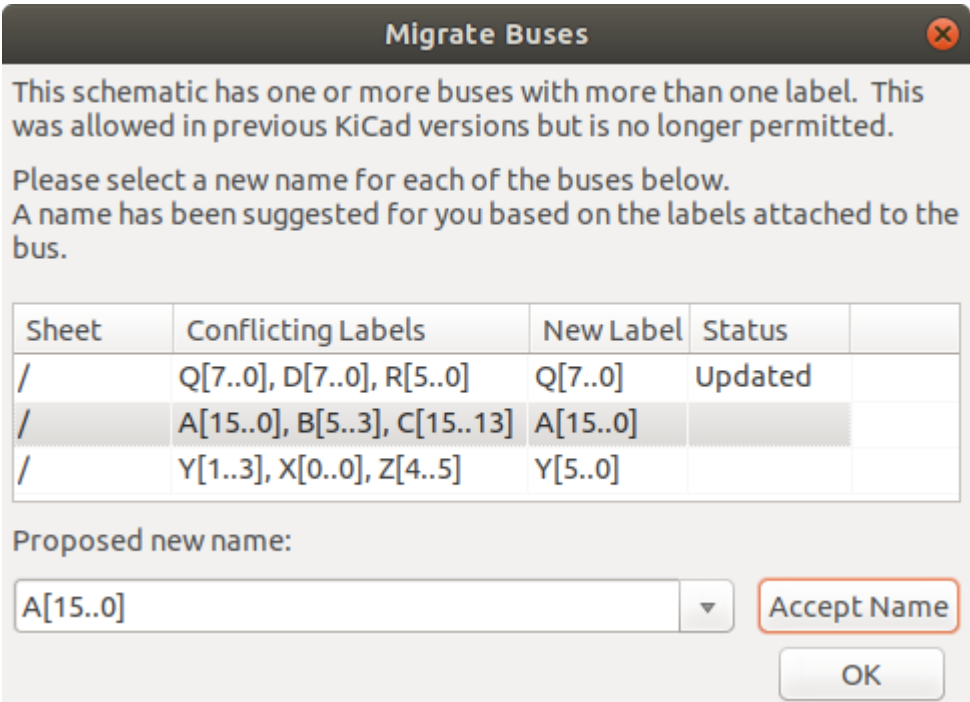
定义别名后，可以通过将别名放在大括号内来用于网络名称，例如“{USB}”。这与“{DP DM VBUS}”具有相同的效果。您还可以添加前缀名称，例如“USB1 {USB}”会生成如上所述的“USB1.DP”等网络名称。对于重复使用别名可以使原理图上的名称更短。记住，别名只是一个快捷方式，别名的名称不包含在网表中。

别名保存在原理图文件中。在指定的原理图工作表中创建的任何别名都可用于同一文档结构中的任何其他原理图工作表。

## 有多个别名的情况

KiCad 5.0 及更早版本允许将具有不同别名的网络接在一起，并且在网表列表期间将加入一些成员。此行已在 KiCad 6.0 中删除，因为它与 EDA 不兼容，并且可能导致令人困惑的网表，因此不容易确定信号将接收的名称。

如果您在旧版本的 KiCad 中打开使用此功能的文档，您将看到“迁移”对话框，该对话框将提示您更新原理图以便在任何给定的网络路上只存在一个成员。



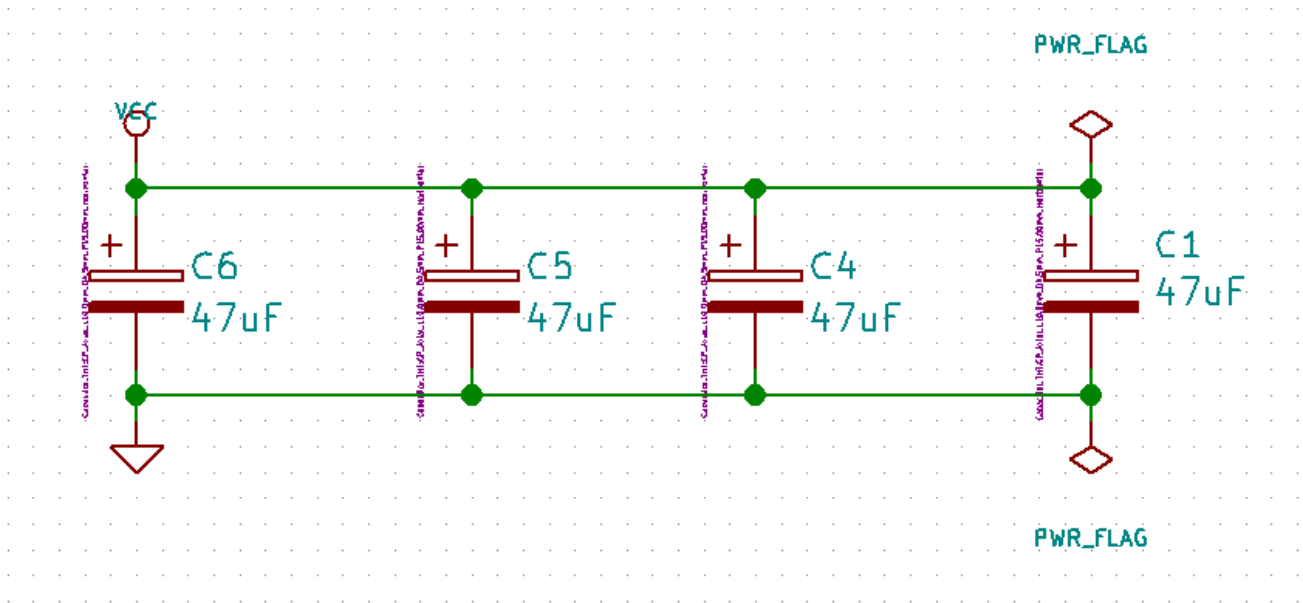
对于具有多个冲突的每行，您必须保留的冲突下拉名称框允许您在设计中存在的冲突之下一行或者您可以通过手动将其输入新名称字段来输入其他名称。

### Power Ports

Power port symbols are conventionally used to connect pins to power nets. Power port symbols have a single pin which is invisible and marked as a power input. As described in the [hidden power pins section](#), any wire connected to the pin of a power port is therefore automatically connected to the power net with the same name as the port's pin.

In the KiCad standard library, power ports are found in the `power` library, but power port symbols can be created in any library. To create a custom power port, make a new symbol with a hidden pin marked as a power input. Name the pin according to the desired power net.

下图显示了电源端口连接的示例。



In this example, power ports symbols are used to connect the positive and negative terminals of the capacitors to the VCC and GND nets, respectively.

Power port symbols are found in the `power` symbol library. They can also be created by drawing a symbol with a hidden "power input" pin that has the name of the desired power net.

## PWR\_FLAG

Two `PWR_FLAG` symbols are visible in the screenshot above. They indicate to ERC that the two power nets `VCC` and `GND` are actually connected to a power source, as there is no explicit power source such as a voltage regulator output attached to either net.

Without these two flags, the ERC tool would diagnose: *Error: Input Power pin not driven by any Output Power pins.*

The `PWR_FLAG` symbol is found in the `power` symbol library. The same effect can be achieved by connecting any "Power Output" pin to the net.

## No-connection flag

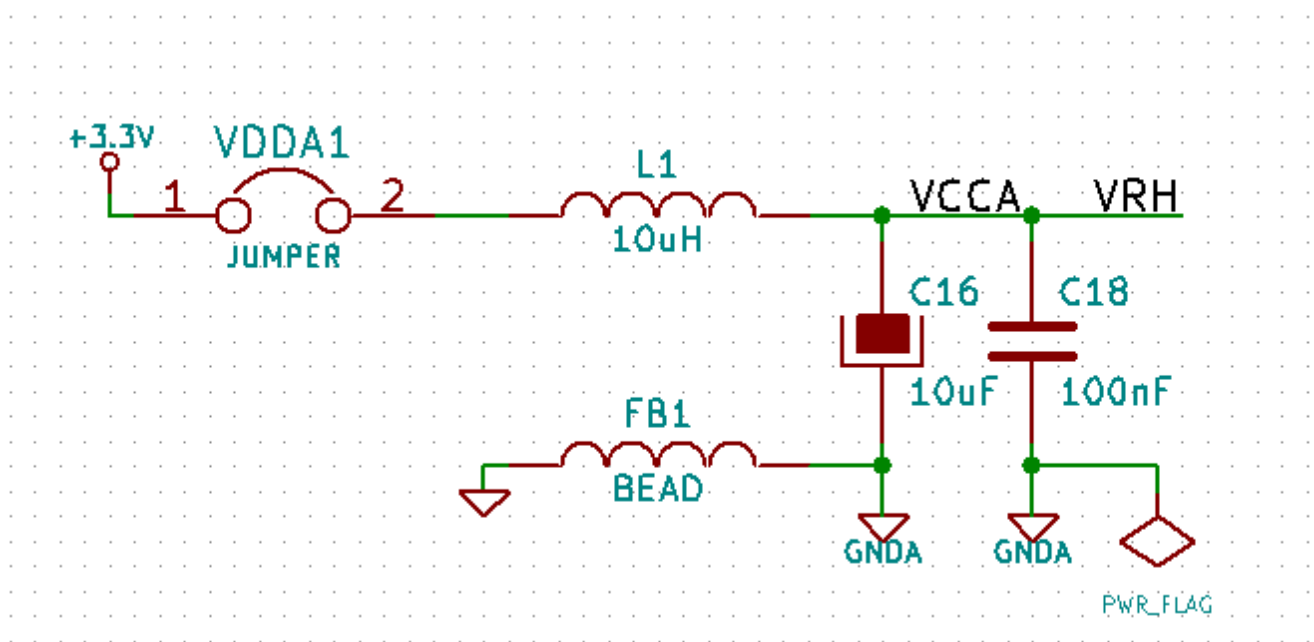
No-connection flags (→X) are used to indicate that a pin is intentionally unconnected. These flags do not have any effect on the schematic's connectivity, but they prevent "unconnected pin" ERC warnings for pins that are intentionally unconnected.

□□□充

## Text comments and graphic lines

It can be useful to place annotations such as text fields and frames to aid in understanding the schematic. Text fields (T) and graphic lines (→) are intended for this use, as opposed to labels and wires, which are connection elements.

The image below shows graphic lines and text in addition to wires, local labels, and hierarchical labels.





表格

The title block is edited with the Page Settings tool ( ).

页面设置

图纸

尺寸:  
A3 297x420mm

方向:  
横向

自定义尺寸:  
高度: 279.40 宽度: 431.80

布局预览

标题栏字段设置

共 1 页 第 1 页

更改日期  
Sun 22 Mar 2015 <<< 2019/ 2/18

版次  
2B

标题  
UNIVERSAL INTERFACE

公司  
KICAD

注释 1  
Comment 1

注释 2  
Comment 2

注释 3  
Comment 3

注释 4  
Comment 4

页面布局描述文件

确定 取消

Each field in the title block can be edited, as well as the paper size and orientation. If the "Export to other sheets" option is checked for a field, that field will be updated in the title block of all sheets, rather than only the current sheet.

A drawing sheet template file can also be selected.

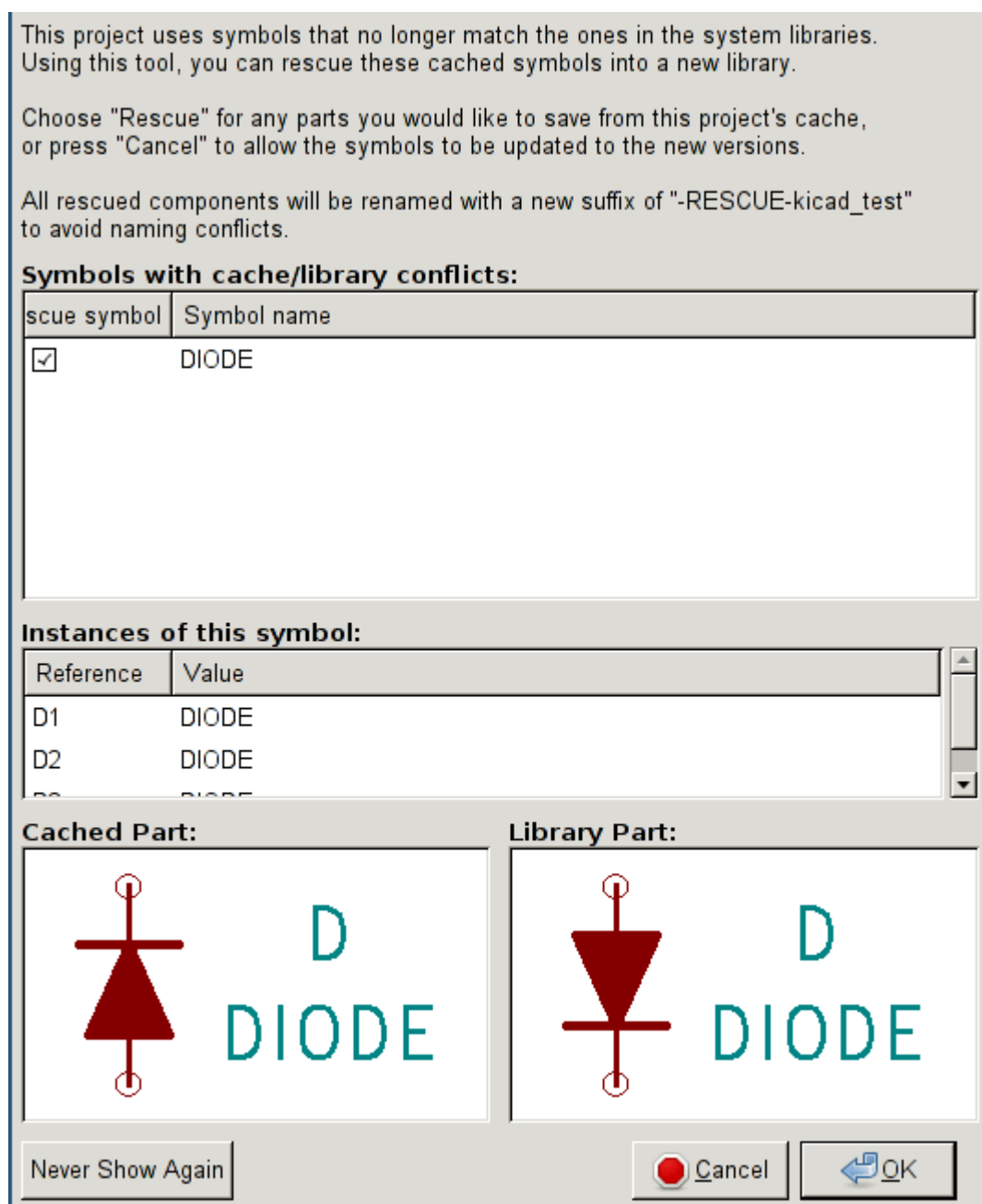
Comment 4		
Comment 3		
Comment 2		
Comment 1		
KICAD		
Sheet: /		
File: interf_u.sch		
Title: UNIVERSAL INTERFACE		
Size: A3	Date: Sun 22 Mar 2015	Rev: 2B
KiCad E.D.A. kicad (5.0.2)-1		Id: 1/1

The sheet number (Sheet X/Y) is automatically updated, but sheet page numbers can also be manually set using Edit → Edit Sheet Page Number....

## ❏救❏存的符号

By default, KiCad loads symbols from the project libraries according to the set paths and library order. This can cause a problem when loading a very old project: if the symbols in the library have changed or have been removed or the library no longer exists since they were used in the project, the ones in the project would be automatically replaced with the new versions. The new versions might not line up correctly or might be oriented differently leading to a broken schematic.

When a project is saved, a cache library with the contents of the current library symbols is saved along with the schematic. This allows the project to be distributed without the full libraries. If you load a project where symbols are present both in its cache and in the system libraries, KiCad will scan the libraries for conflicts. Any conflicts found will be listed in the following dialog:



您可以在此示例中看到❏❏目最初使用的是阴极朝上的二极管，但❏在❏中包含阴极朝下的二极管。❏种改❏会打破原理❏❏ 在此❏按 OK 将使符号❏存❏保存到特殊的 恢复❏中，并重命名所有符号以避免命名冲突。

If you press Cancel, no rescues will be made, so KiCad will load all the new components by default. If you save the schematic at this point, your cache will be overwritten and the old symbols will not be recoverable. If you have saved the schematic, you can still go back and run the rescue function again by selecting "Rescue Cached Components" in the "Tools" menu to call up the rescue dialog again.

如果您不想看到此对话框，可以按 从不再显示。默认设置是不行任何操作并允加新元件。可以在 首中更改此


# 分原理

## 介

对于大于几的项目，分表示通常是一个很好的解决方案。如果要管理此项目，需要：

- 使用大表会致打印和理
- 使用多个工作表，将引入入次构。

然后，完整的原理包含一个主要的原理表，称根表，以及构成次构的子表。此外，巧妙地将计分独立的表格通常会提高其可性。

From the root sheet, you must be able to find all sub-sheets. Hierarchical schematics management is very easy with KiCad, thanks to an integrated "hierarchy navigator" accessible via the icon  of the top toolbar.

有两种型的次构可以同存在：第一种次构被起并且具有普遍用途。第二个包括在中建符号，些符号在原理中看起来像符号，但上于描述其内部构的示意

第二种型用于开集成路，因在种情况下，您必在制的原理中使用函数

KiCad currently doesn't treat this second case.

次构可以是：

- 定：定的工作表只使用一次
- 复：定的工作表被多次使用 (倍数例)
- 平面：是一个的次构, 但不会制工作表之的接。


KiCad can deal with all these hierarchies.

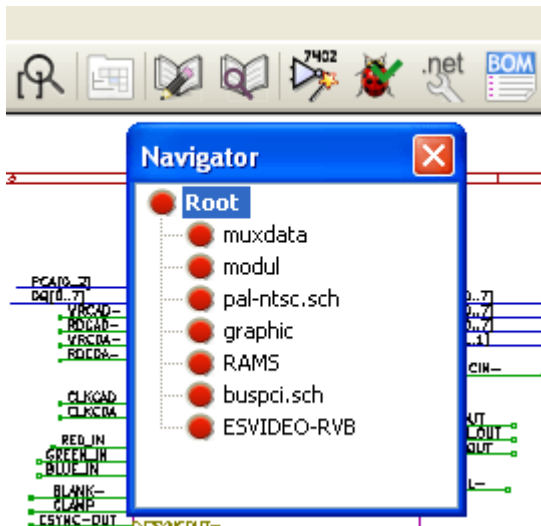
建分原理很容易，整个次构从根原理开始理，就像您只有一个原理一

要理解的两个重要步是：

- 如何建子表。
- 如何在子表之建立气接。

## 在次构中航

Navigation among sub-sheets is achieved by using the navigator tool accessible via the button  on the top toolbar.






其名称即可每个工作表。要快速右工作表名称，然后“入工作表”或双工作表的范

要将当前工作表退出到父工作表，右原理中没有象的任何位置，然后在上下文菜单中“离开工作表”或按“Alt + Backspace”。

## 本地、分和全局

### 属性

Local labels, tool , are connecting signals only within a sheet. Hierarchical labels (tool ) are connecting signals only within a sheet and to a hierarchical pin placed in the parent sheet.

Global labels (tool ) are connecting signals across all the hierarchy. Power pins (type *power in* and *power out*) invisible are like global labels because they are seen as connected between them across all the hierarchy.

**NOTE** 在次构（或复中，可以使用分和/或全局

## 次构建摘要


您必：

- 在根工作表中放置一个名工作表符号的次构符号。
- 使用航器入新原理子工作表）并制它，就像任何其他原理一。
- 通将全局（HLabels）放在新的原理子表）中，并在根表中使用相同名称的称 SheetLabels制两个原理之的气接。些 SheetLabel 将接到根表的工作表符号，接到原理的其他元素，如准符号引脚。

## 工作表符号

制一个由两个角点定义的矩形，表示子表格。

此矩形的大小必允您放置以后特定次构引脚，于子表中的全局（HLabels）。

These labels are similar to usual symbol pins. Select the tool .

以放置矩形的左上角。再次以放置右下角，具有足大的矩形。

然后，系将提示您此子表入文件名和表名称（以便使用次构航器相的原理）

**Schematic Sheet Properties**

File name:  Size:  millimeters

Sheet name:  Size:  millimeters

Unique timestamp:


您必须至少提供一个文件名。如果没有工作表名称，该文件名将用作工作表名称（通常的方法）。

## 连接 - 分引脚

您将在此处构建构建的符号的连接点（引脚引脚）。


这些连接点类似于普通符号引脚，但是只需一个连接点就可以连接一个完整的

## Importing Hierarchical Sheet Pins

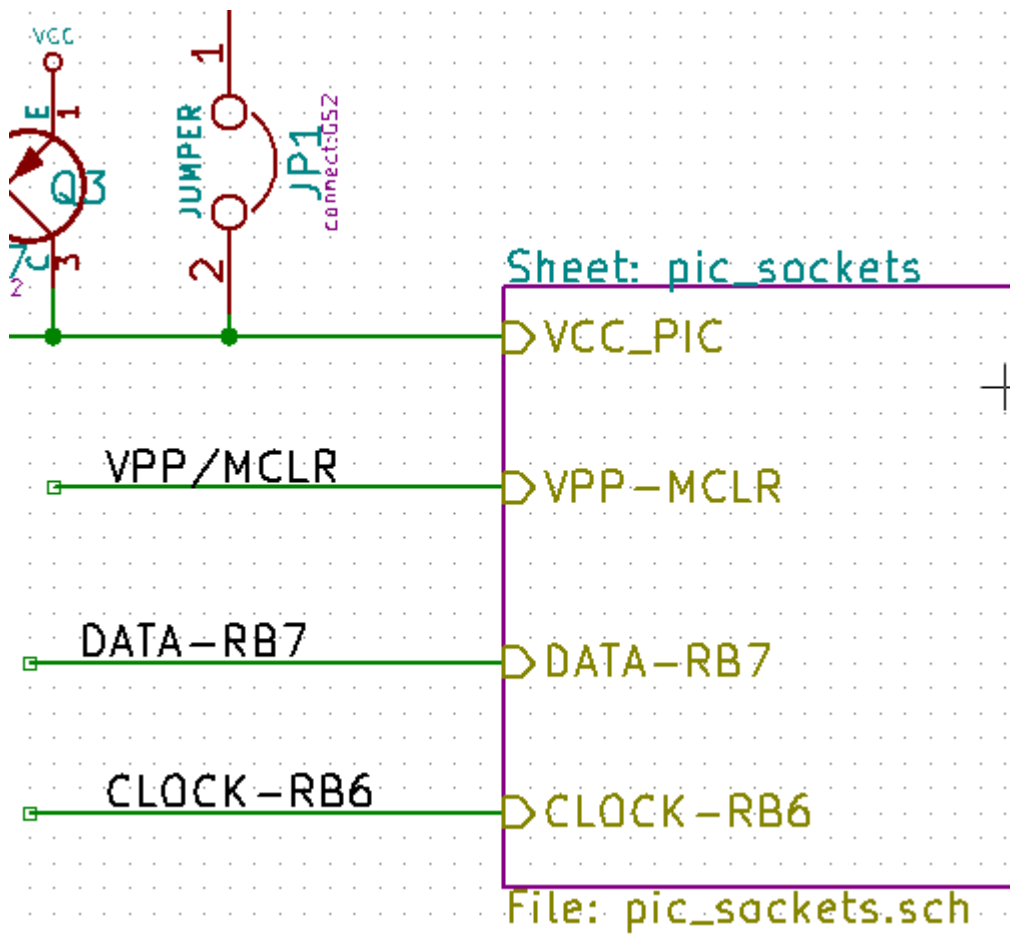
- Select the tool .
- Click on the hierarchical sheet from where you want to import the pins corresponding to hierarchical labels placed in the corresponding schematic. A hierarchical pin appears, if a new hierarchical label exists, i.e. not corresponding to an already placed pin.
- 您要放置此引脚的位置。

All necessary pins can thus be placed quickly and without error. Their aspect is in accordance with corresponding hierarchical labels.

## 连接 - 分

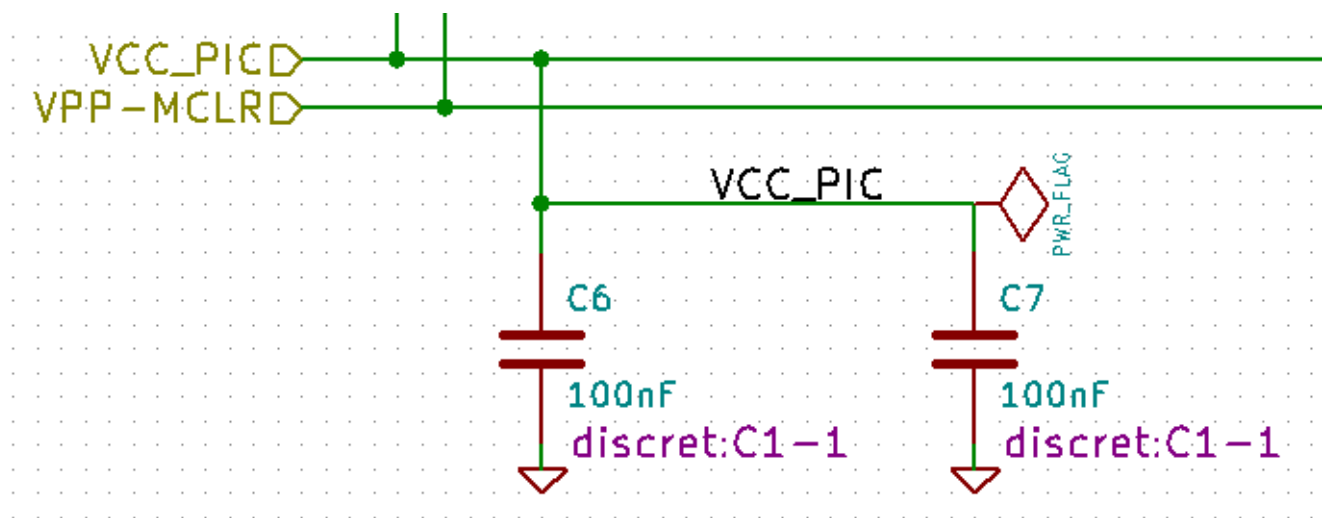
Each pin of the sheet symbol just created, must correspond to a label called hierarchical Label in the sub-sheet. Hierarchical labels are similar to labels, but they provide connections between sub-sheet and root sheet. The graphical representation of the two complementary labels (pin and hierarchical labels) is similar. Hierarchical labels are made with the tool .

参下面的根表示例：



注意引脚 VCC\_PIC 接到连接器 JP1。

以下是子表中的相连接：



您再次找到两个相的分提供两个分表之的接。

#### NOTE

根据前面描述的 法 (Bus [N..m])，您可以使用分 和 次 构引脚 接两条

### 分 全局 和 形 源引脚

以下是有关提供 接的各种方法的一些注 而不是有 接。

## 表的表

表具有局部连接能力，即限于放置它的示意图是因

- 每表都有一个表号。
- 此工作表号与表相关

因此，如果在表 n°3 中放置表 “TOTO” 表上真正的表是 “TOTO\_3”。如果您在工作表 n°1（根表）中放置了表 “TOTO” 表上放置一个名 “TOTO\_1” 的表与 “TOTO\_3” 不同。即使只有一表也是如此。

## 分表

对于表而言，对于分表也是如此。

因此，在同一表中，分表 “TOTO” 被识别到本地表 “TOTO”，但没有识别到另一表中称 “TOTO” 的分表或分表被识别到放置在父表中的分表符号中的相关表引脚符号。

## 形源引脚

可以看出，如果形源引脚具有相同的名称，它将被连接在一起。因此，所有声称形源引脚并命名 VCC 的源引脚都将所有符号 VCC 的源引脚连接在它放置的工作表内。

这意味着如果将 VCC 放在子表中，它将不会连接到 VCC 引脚，因为它是 VCC\_n，其中 n 是表号。

如果您希望此 VCC 真正连接到整个原理图的 VCC 必须通 VCC 源符号将其明确连接到不可的源引脚。

## 全局

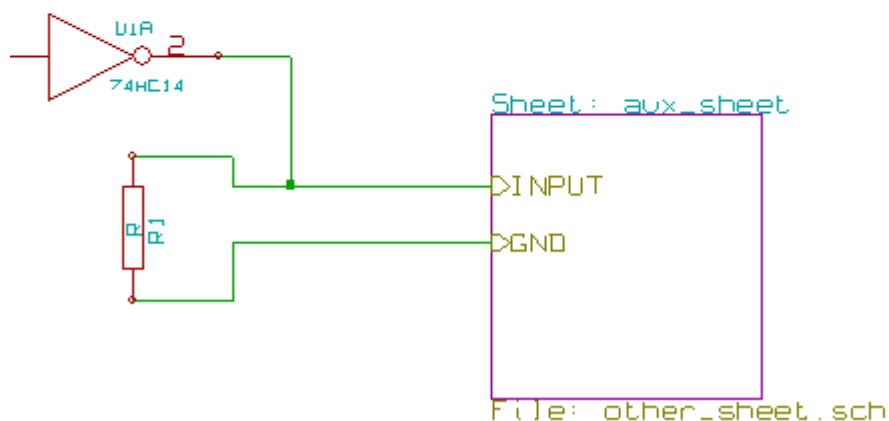
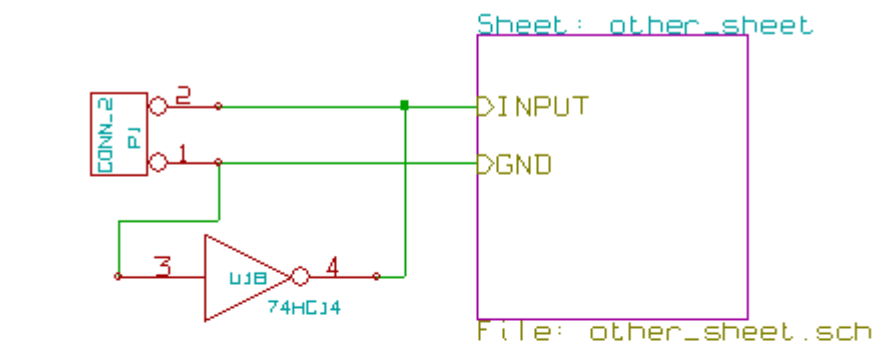
具有相同名称的全局跨整个次结构。

（像vcc的力.....是全局

## 复次结构

是一个例子。相同的原理图使用两次（两个例）。两个工作表共享相同的原理图因两个工作表的文件名相同（*other\_sheet.sch*）。工作表名称必须是唯一的。





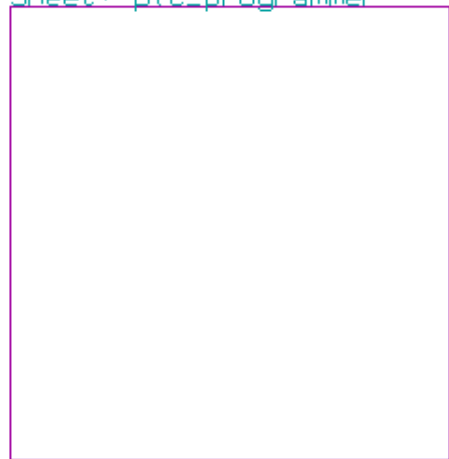
## 平面层次结构

如果遵守以下原则可以使用多个工作表构建项目，而无需在有些工作表（平面层次结构）之间构建连接：

- 构建包含其他工作表的根工作表，有些工作表充当其他工作表之间的连接。
- 不需要明确的连接。
- 在所有工作表中使用全局而不是分

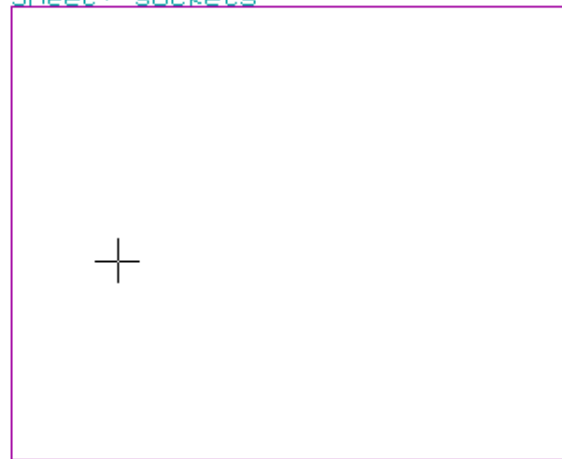
以下是根表的示例。

Sheet: pic\_programmer



File: pic\_programmer.sch

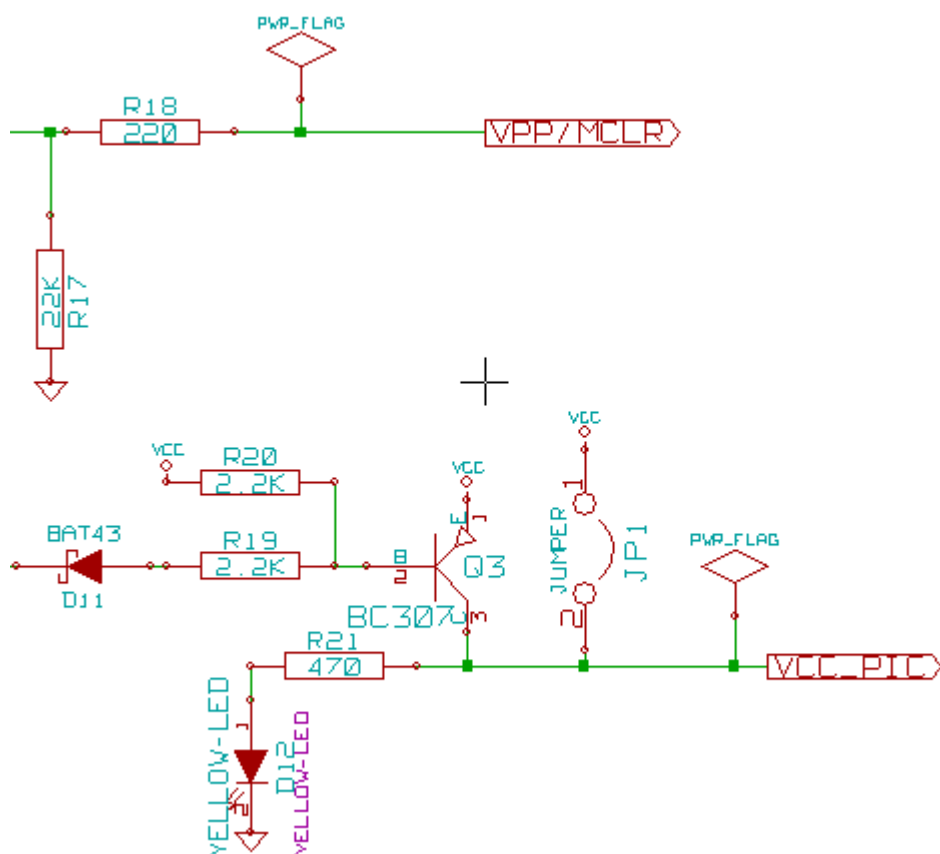
Sheet: sockets



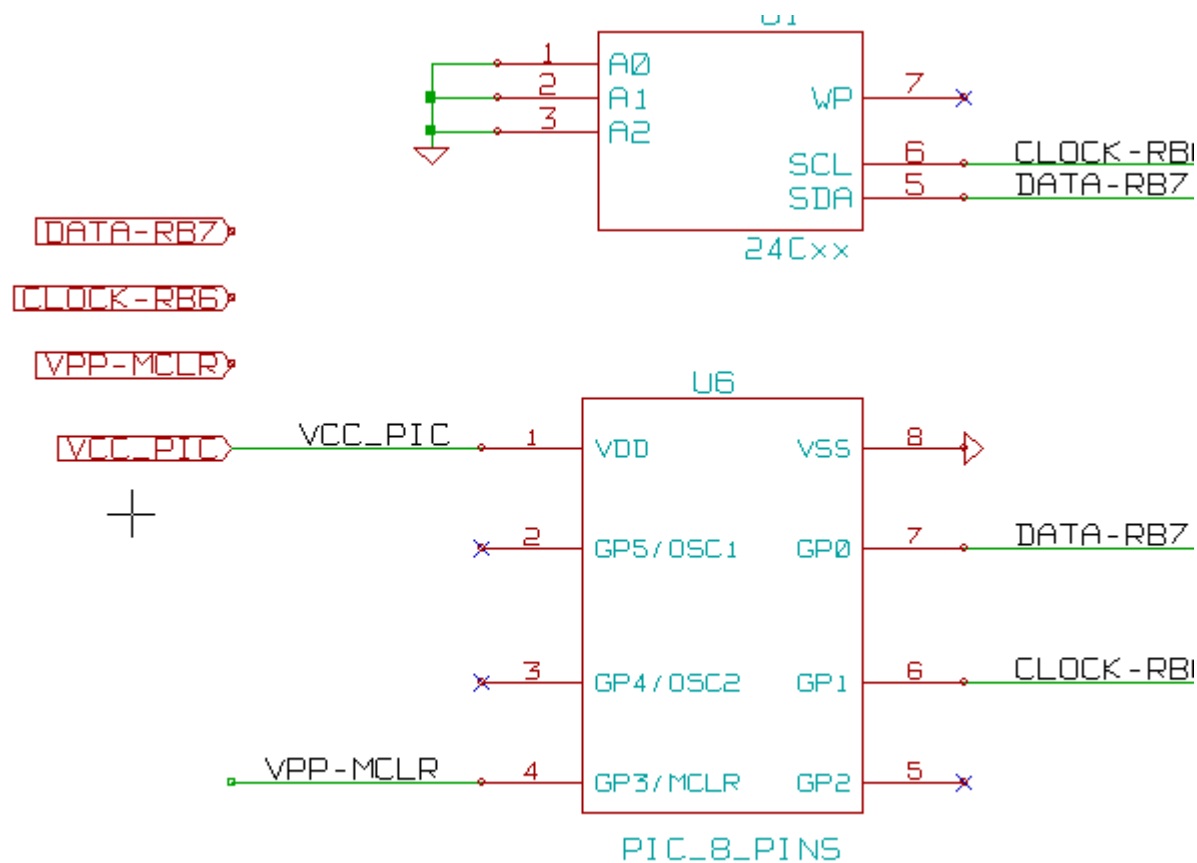
File: pic\_sockets.sch

是两由全局接。

是 pic\_programmer.sch。



是 pic\_sockets.sch。



☐看全局☐☐☐


DATA-RB7

CLOCK-RB6

VPP-MCLR

# 符号批注工具

## 简介

The annotation tool allows you to automatically assign a designator to symbols in your schematic. Annotation of symbols with multiple units will assign a unique suffix to minimize the number of these symbols. The annotation tool is accessible via the icon . Here you find its main window.



批注原理图

范围:

☒ 使用整个原理图

☐ 仅使用当前页面

顺序:

☒ X方向排序元件 (X)

☐ Y方向排序元件 (Y)

选项:

☒ 保持现有的批注

☐ 重置现有的批注

☐ 重置, 但保持多单元器件的顺序

编号:

☒ 使用该数字之后的编号:

☐ 参考编号X100

☐ 参考编号X1000

☐ 保持对话框打开

☐ 不要求确认

批注

清除批注

关闭

批注信息:

显示: ☒ 所有 ☒ 错误 ☒ 警告 ☒ 相关信息 ☒ 活动

保存报告文件

可用的批注方案:

- 批注所有符号 (重置所有批注)
- 批注所有符号, 但不要交任何以前批注的多单元元件。
- 批注当前未批注的符号。未批注的符号将具有以 "?" 字符结尾的指示符。
- 批注整个次构 (使用整个原理图)。
- 批注当前工作表 (使用当前页)。

“重置, 但不交任何批注的多元部件” 保留具有多元的符号之的所有有关例如, U22 和 U2A 可能分重新批注 U1B 和 U1B, 但它永不会重新批注到 U1B 和 U2A, 也不会重新批注 U2B 和 U2A。如果要确保保持引脚分, 很有用。

批注序提供了用于在次构的每个工作表内置参考号的方法。

除特定情况外, 如果您不想修改以前的批注, 自批注将用于整个目 (所有工作表) 和新元件。

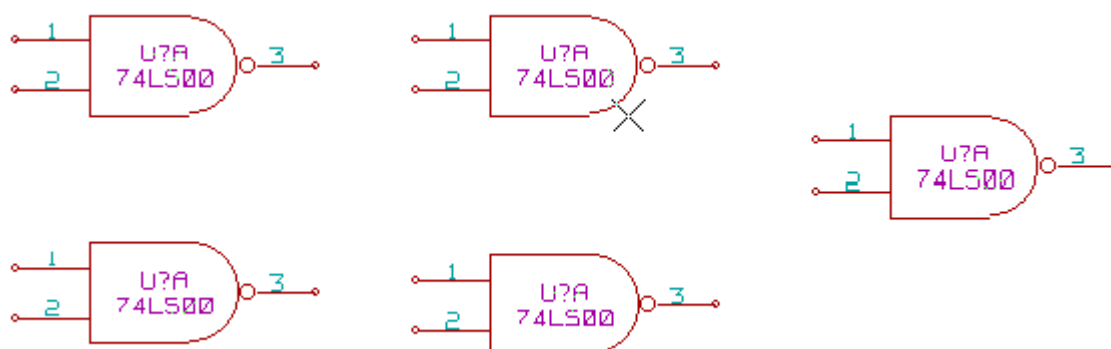
“批注” 出了用于计算参考的方法:

- 在原理中使用第一个空号: 元件从 1 开始批注 (于每个引用前 如果存在先前的批注, 使用未使用的数字。
- 从号 \*100 开始并使用第一个空号: 从 1 的 101 开始批注, 从 2 的 201 开始, 等等。如果在工作表内有超 99 个具有相同参考前 (U, R) 的目 在 1 中, 批注工具使用数字 200 和更多, 并且工作表 2 的批注将从下一个空号开始。
- 从表格号 \*1000 开始并使用第一个空号。于 1, 批注从 1001 开始, 从 2 的 2001 开始。

## 一些例子

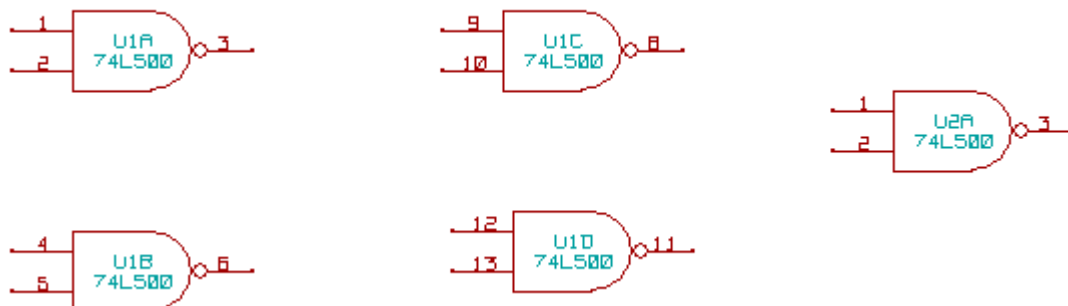
### 批注序

此示例示放置了 5 个元素, 但未批注。

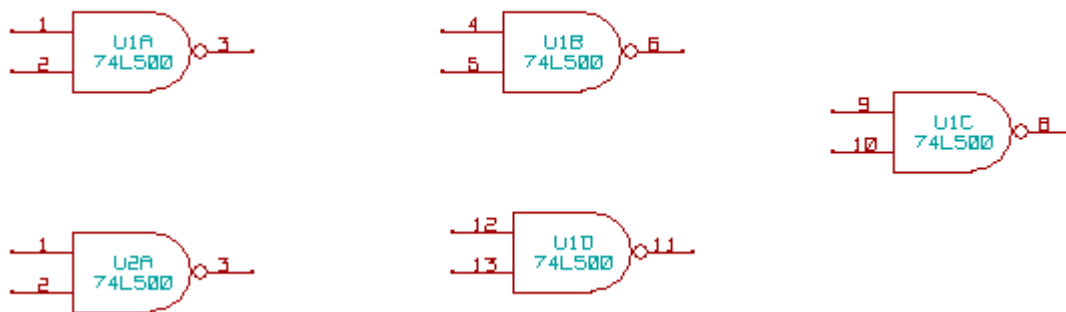


行批注工具后, 得以下果。

按 X 位置排序。



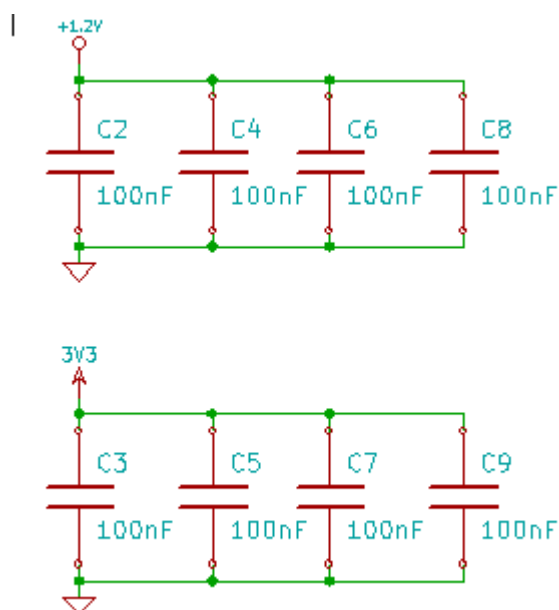
按 Y 位置排序。



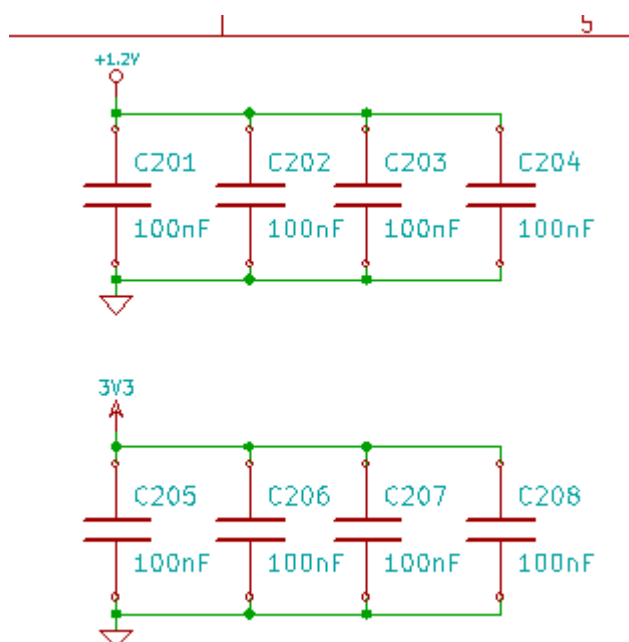
您可以看到四个 74LS00 分布在 U1 包中，第五个 74LS00 已分配给下一个 U2。

## 批注

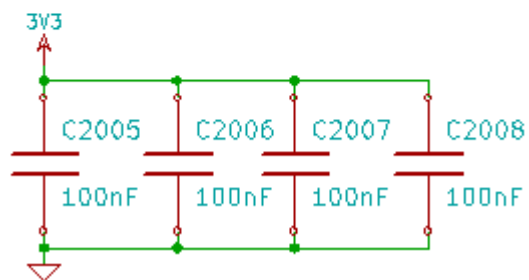
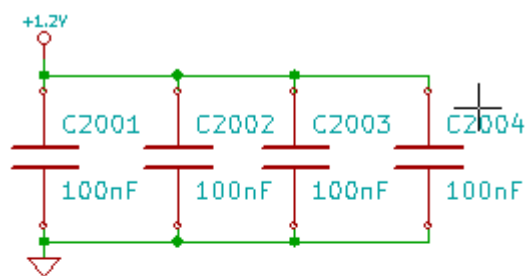
是表 2 中的批注，其中使用原理中的第一个空号。



开始到工作表号 \*100 并使用第一个空号出以下果。



从开始到工作表第1000行并使用第一个空行号输出以下结果。

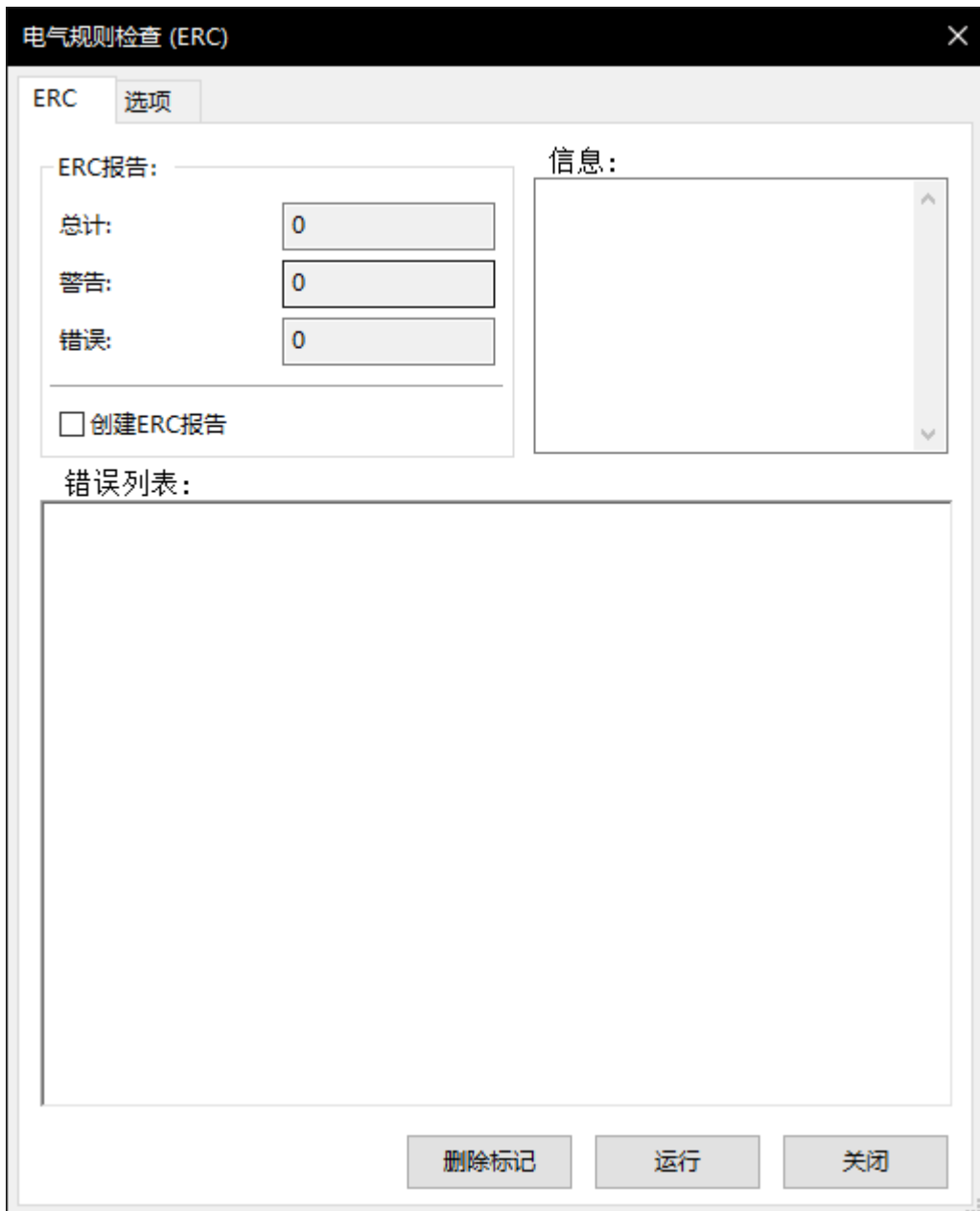


# 使用电气规则检查

## 简介

电气规则检查(ERC)工具会自动原理图ERC工作表中的任何错误例如未连接的引脚，未连接的分号符号，短路输出等。当然，自动不是完美的，并且可以所有设计的零件不是100%完成。它非常有用，因为它允许您多疏忽和小错误

上，必须所有找到的然后在正常运行之前纠正。ERC的数量与在符号建设期声明引脚属性所采取的谨慎直接相关。ERC输出警告或警告。



## 如何使用 ERC

ERC can be started by clicking on the icon .

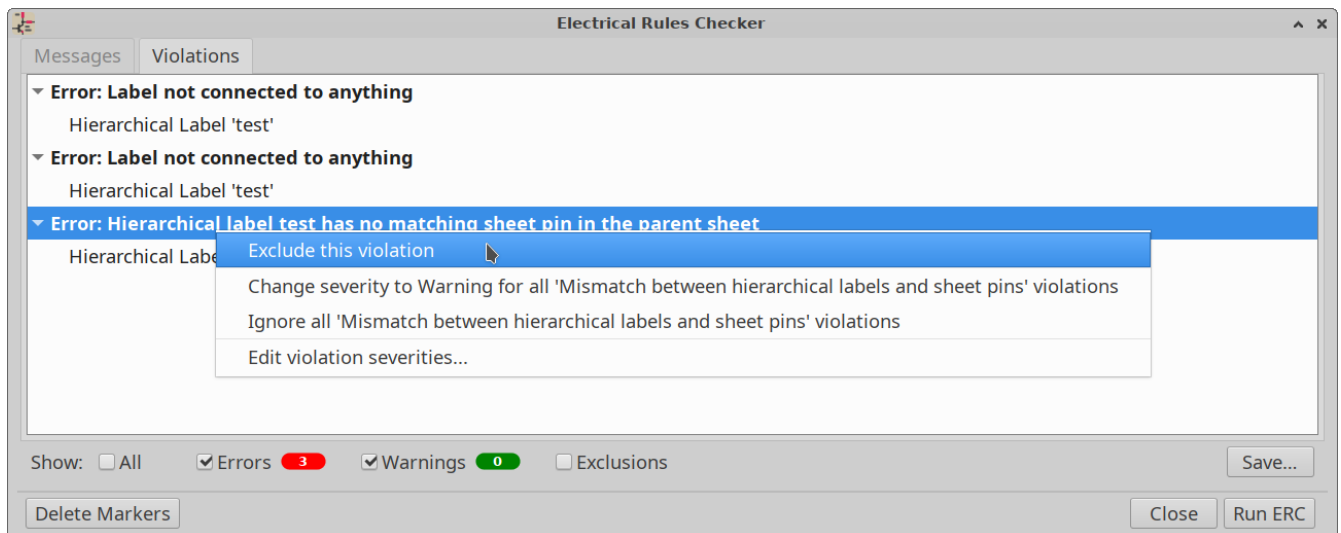
在原理图元素上放置警告, 以引起 ERC 错误 (引脚或连接)。



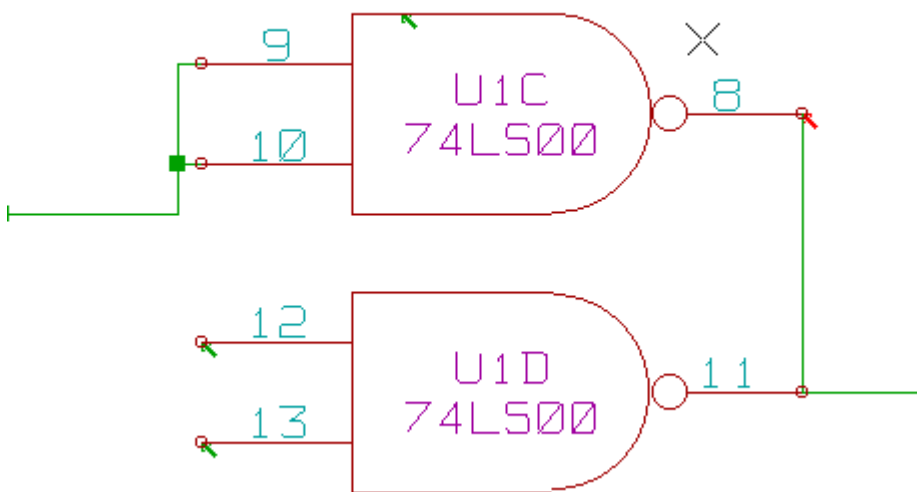
## NOTE

- 在此对话框中，您可以通过单击消息中的相应列来跳转到原理图中的相应位置。
- 在原理图中，右击可以显示相应的断消息。

You can also delete error markers from the dialog and set specific ERC messages to be suppressed by using the right-click context menu.



## ERC 的示例

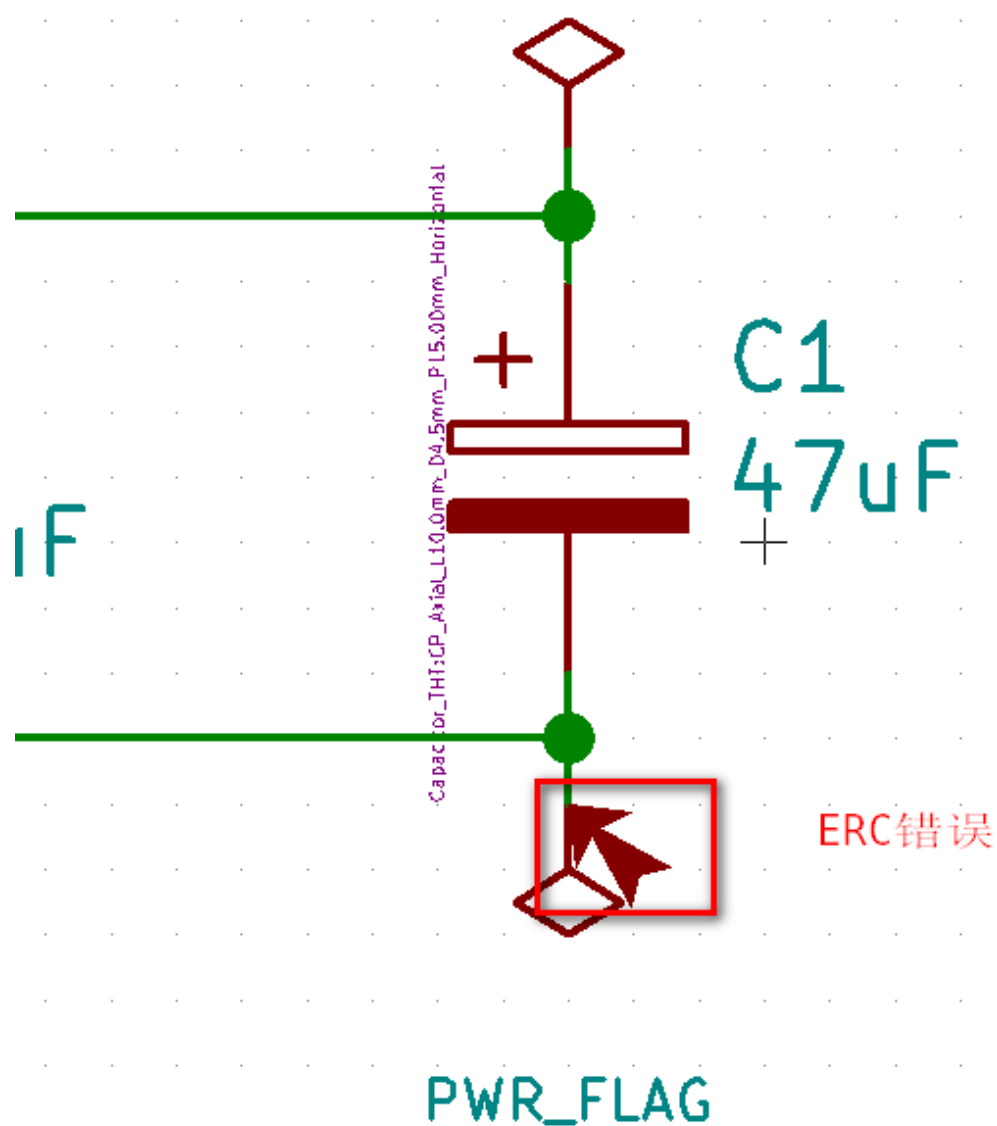


在图中，您可以看到四个错误：

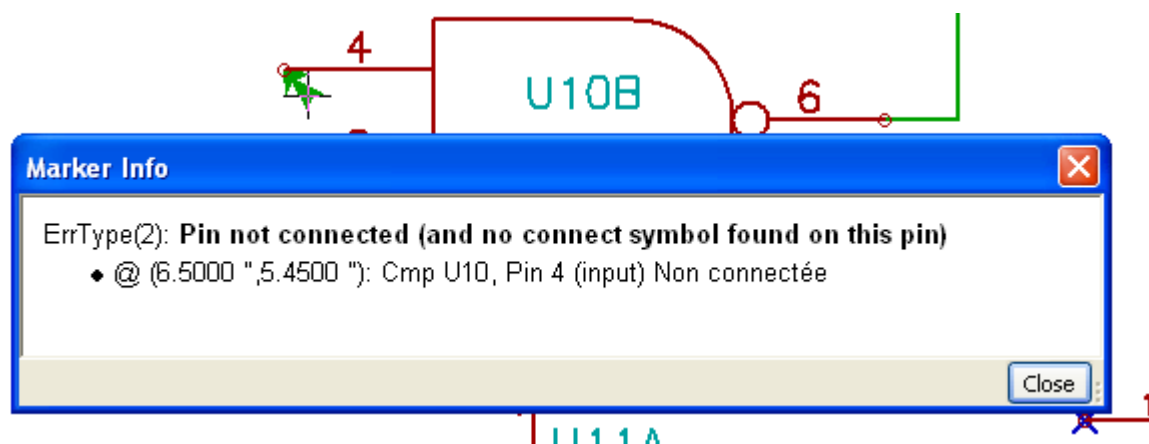
- 两个输出端接地（红色箭头）
- 两个输入未接（红色箭头）
- 隐藏源端口输出缺少源标志（部分红色箭头）

## 显示断

通过右击可以显示菜单位置，您可以在 ERC 对话框中。



当信息您可以得的描述。

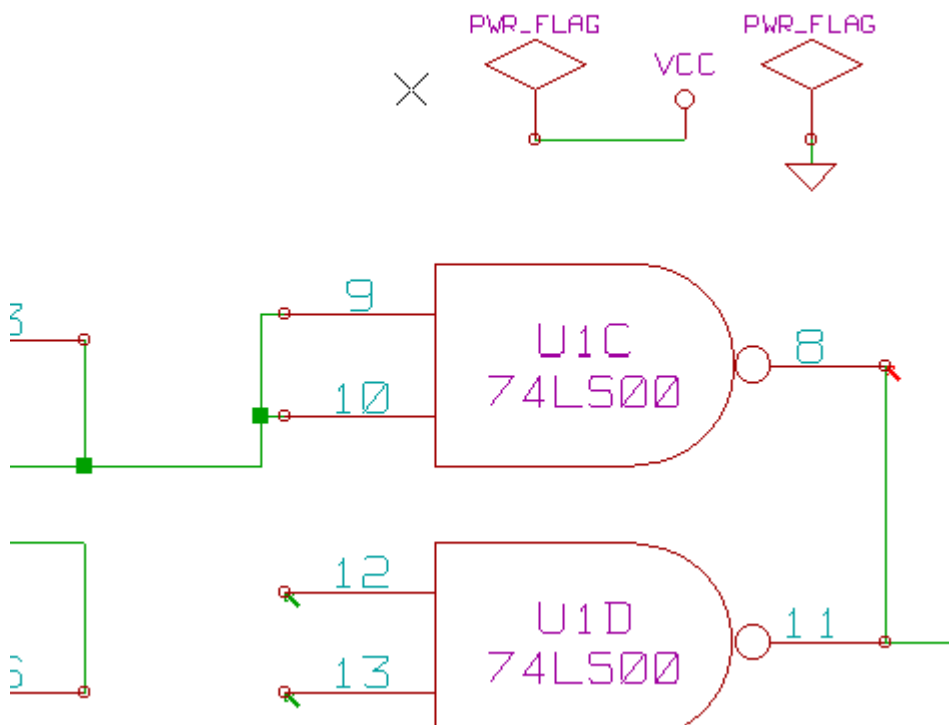


## 源引脚和源标志

通常在源引脚上出错误或警告，即使一切看起来都很正常。上面的例子。之所以会产生这种情况，是因为在大多数设计中，源是由不是源的连接器提供的（如器件输出，它被声明为源输出）。

因此，ERC 不会检测到任何源输出引脚来控制并声明它不是由源输出的。

要避免此警告，您必须在此源端口上放置“PWR\_FLAG”。看一下下面的例子：

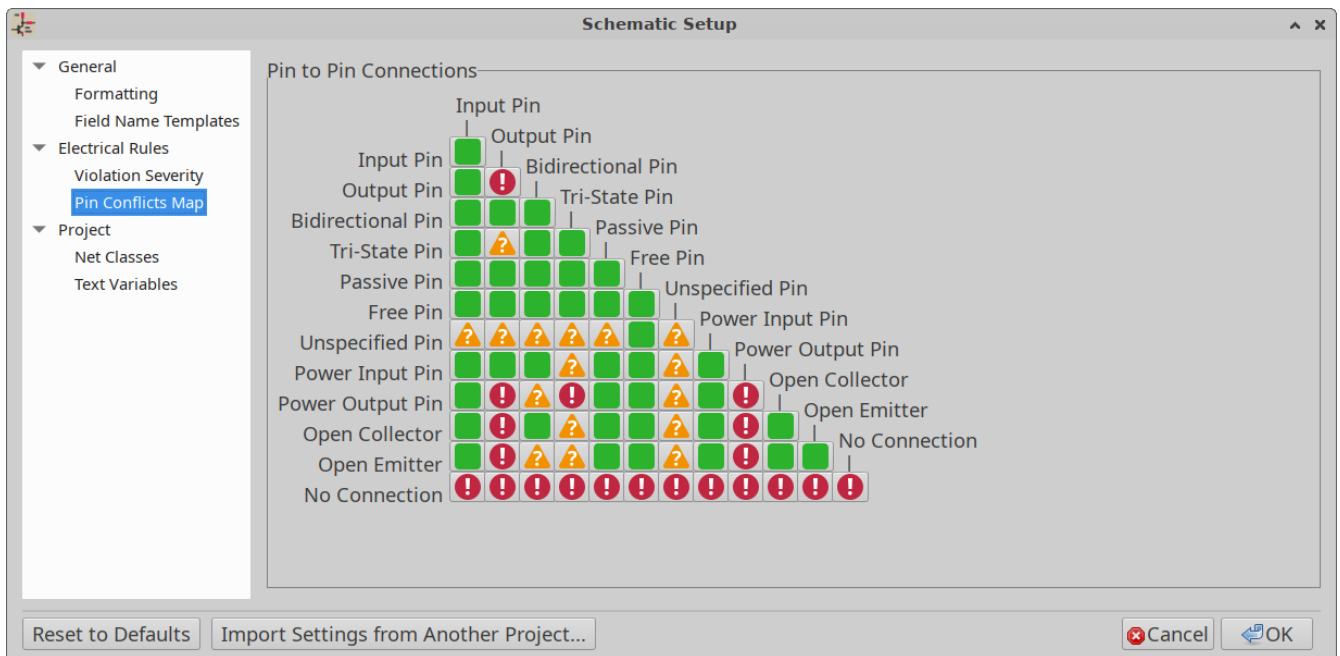


然后警告将消失。

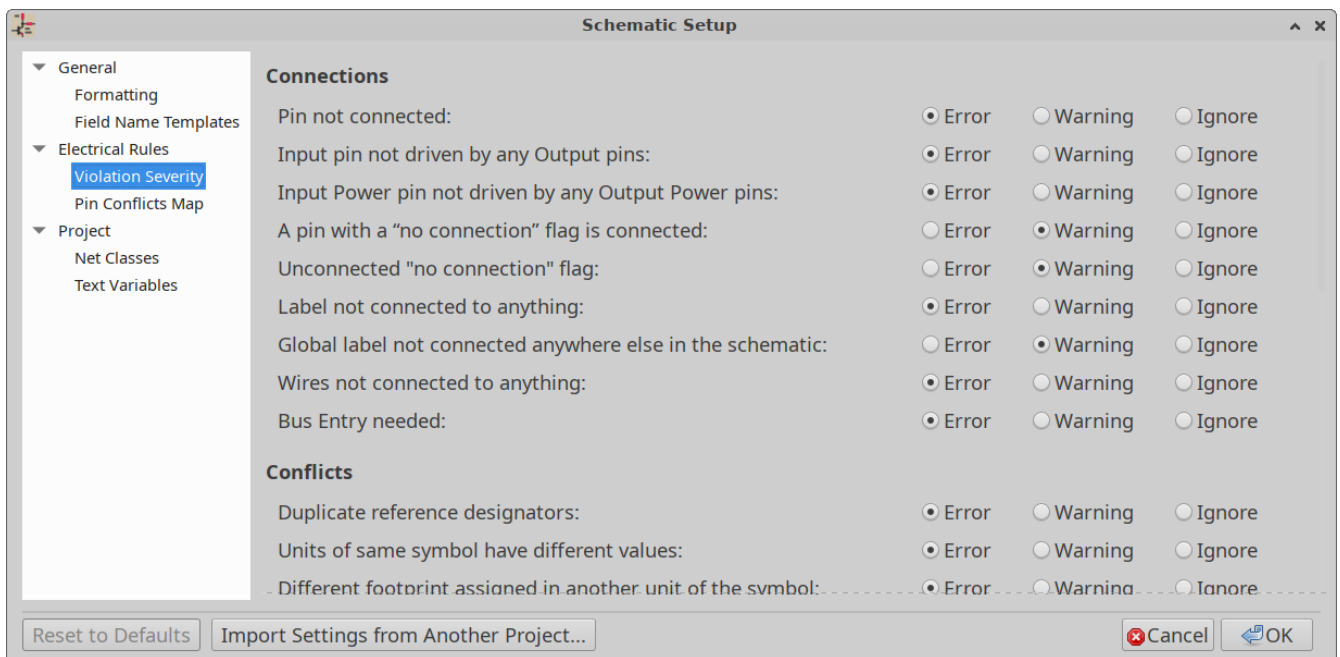
大多数情况下，PWR\_FLAG 必须连接到 GND，因为器件的输出声明为源但接地引脚永远不会断（正常属性是源输入），因此，在没有源标志的情况下，接地不会与源相

## 配置

The *Pin Conflicts Map* panel in Schematic Setup allows you to configure connectivity rules to define electrical conditions for errors and warnings based on what types of pins are connected to each other



所需的元格方可以更改使其循正常，警告，



The *Violation Severity* panel in Schematic Setup lets you configure what types of ERC messages should be reported as Errors, Warnings or ignored.

## ERC 告文件

通中写入 ERC 告可以生成并保存 ERC 告文件。ERC 告文件的文件名 .erc。以下是 ERC 告文件的示例。

ERC control (4/1/1997-14:16:4)

\*\*\*\*\* Sheet 1 (INTERFACE UNIVERSAL)

ERC: Warning Pin input Unconnected @ 8.450, 2.350

ERC: Warning passive Pin Unconnected @ 8.450, 1.950


ERC: Warning: BiDir Pin connected to power Pin (Net 6) @ 10.100, 3.300

ERC: Warning: Power Pin connected to BiDir Pin (Net 6) @ 4.950, 1.400

>> Errors ERC: 4

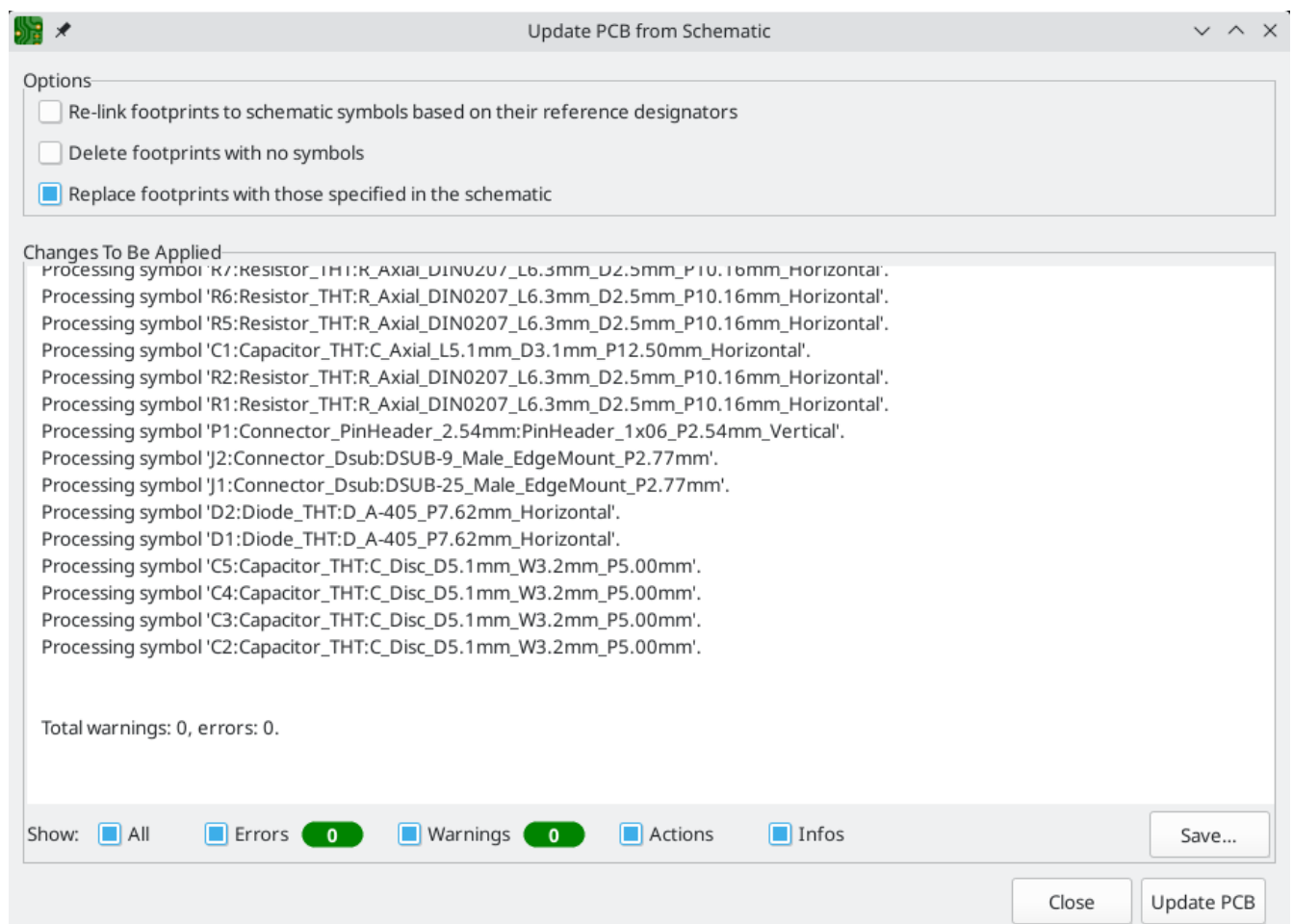
# Transfer Schematic to PCB

## 概述

Use the Update PCB from Schematic tool to sync design information from the Schematic Editor to the Board Editor. The tool can be accessed with **Tools** → **Update PCB from Schematic** (**F8**) in both the schematic and board editors. You can also use the  icon in the top toolbar of the Board Editor.

### NOTE

Update PCB from Schematic is the preferred way to transfer design information from the schematic to the PCB. In older versions of KiCad, the equivalent process was to export a netlist from the Schematic Editor and import it into the Board Editor. It is no longer necessary to use a netlist file.



The tool adds the footprint for each symbol to the board and transfers updated schematic information to the board. In particular, the board's net connections are updated to match the schematic.

The changes that will be made to the PCB are listed in the *Changes To Be Applied* pane. The PCB is not modified until you click the **Update PCB** button.

You can show or hide different types of messages using the checkboxes at the bottom of the window. A report of the changes can be saved to a file using the **Save...** button.

## Options

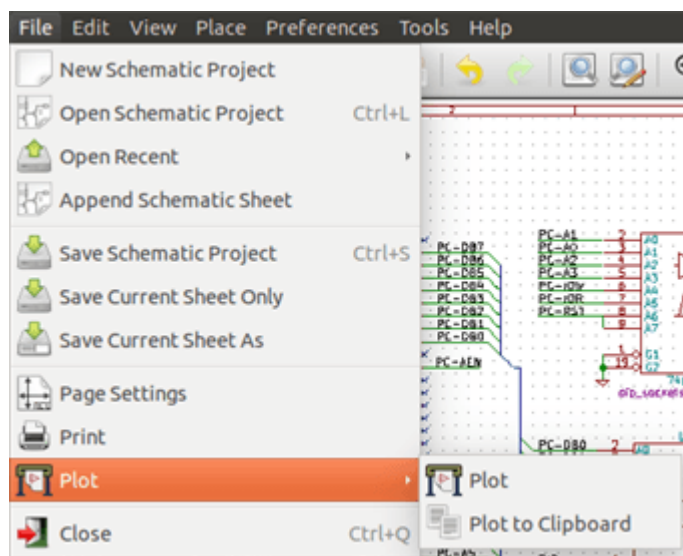
The tool has several options to control its behavior.

Option	Description
<p>Re-link footprints to schematic symbols based on their reference designators</p>	<p>Footprints are normally linked to schematic symbols via a unique identifier created when the symbol is added to the schematic. A symbol's unique identifier cannot be changed.</p> <p>If checked, each footprint in the PCB will be re-linked to the symbol that has the same reference designator as the footprint.</p> <p>If unchecked, footprints and symbols will be linked by unique identifier as usual, rather than by reference designator. Each footprint's reference designator will be updated to match the reference designator of its linked symbol.</p> <p>This option should generally be left unchecked. It is useful for specific workflows that rely on changing the links between schematic symbols and footprints, such as refactoring a schematic for easier layout or replicating layout between identical channels of a design.</p>
<p>Delete footprints with no symbols</p>	<p>If checked, any footprint in the PCB without a corresponding symbol in the schematic will be deleted from the PCB. Footprints with the "Not in schematic" attribute will be unaffected.</p> <p>If unchecked, footprints without a corresponding symbol will not be deleted.</p>
<p>Replace footprints with those specified in the schematic</p>	<p>If checked, footprints in the PCB will be replaced with the footprint that is specified in the corresponding schematic symbol.</p> <p>If unchecked, footprints that are already in the PCB will not be changed, even if the schematic symbol is updated to specify a different footprint.</p>

# □□和打印

## □介

您可以通□文件菜□□□打印和□□命令。



支持的□出格式是 Postscript, PDF, SVG, DXF 和 HPGL。您也可以直接打印到您的打印机。

## 常□的打印命令

### □制当前□面

□打印当前工作表的一个文件。

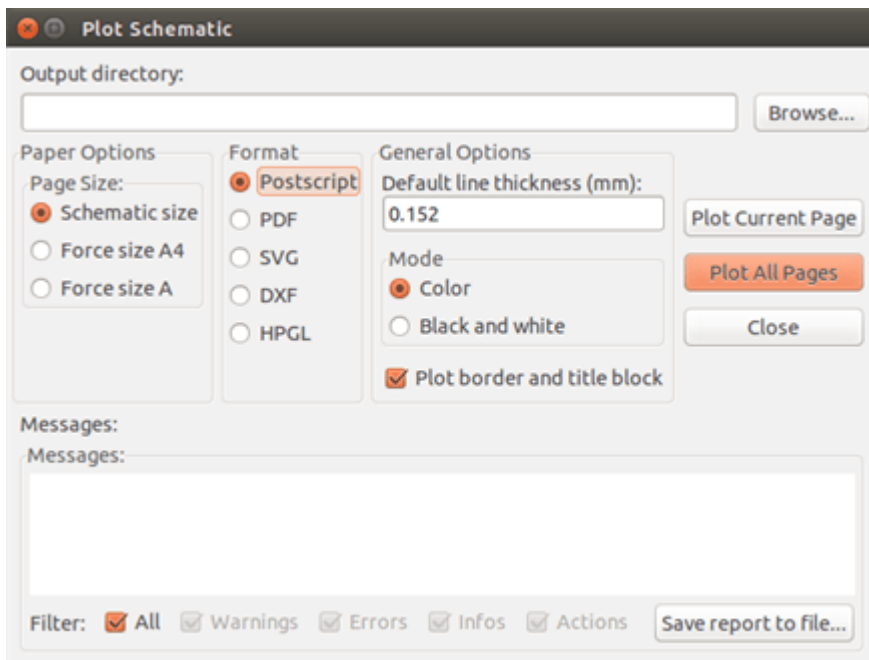
### □制所有□面

允□您□制整个□次□构（□每个工作表生成一个打印文件）。

## 在 Postscript 中□制

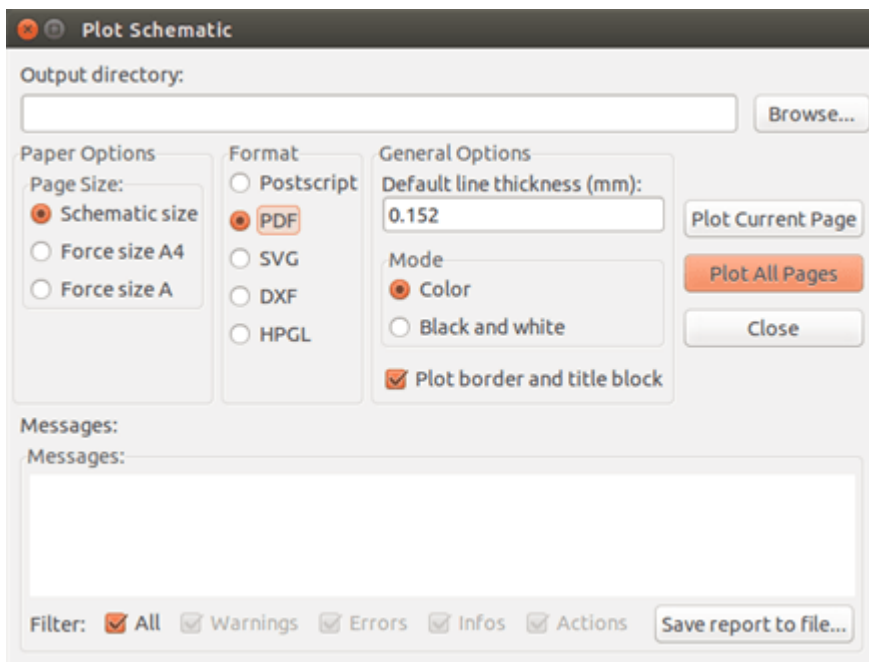
此命令允□您□建 PostScript 文件。





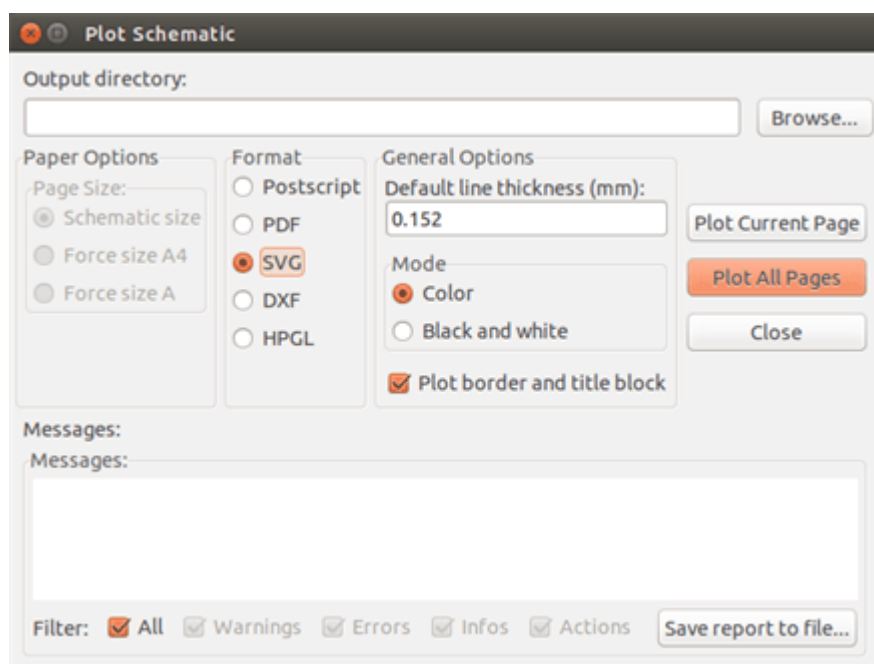
文件名是“展名”.ps 的工作表名称。您可以禁用“制框和” 如果要建用于封装的 postscript 文件（格式 .eps 通常用于在文字理件中插入表，非常有用。消息窗口示建的文件名。

## 以 PDF 格式制



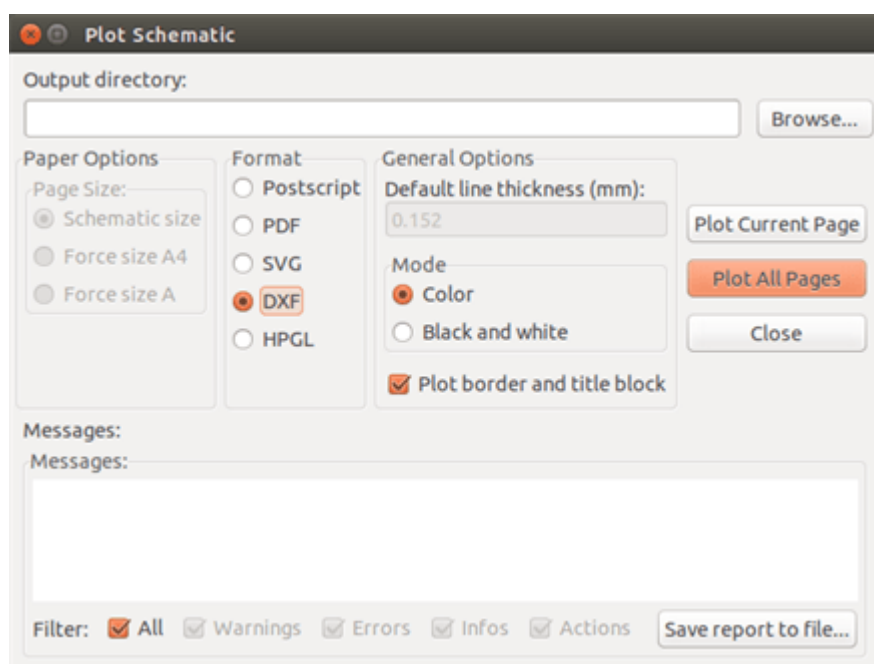
允您使用 PDF 格式建打印文件。文件名是“展名”.pdf 的工作表名称。

## 在 SVG 中



允许您使用矢量格式 SVG 创建打印文件。文件名是展名.svg 的工作表名称。

## 在 DXF 中



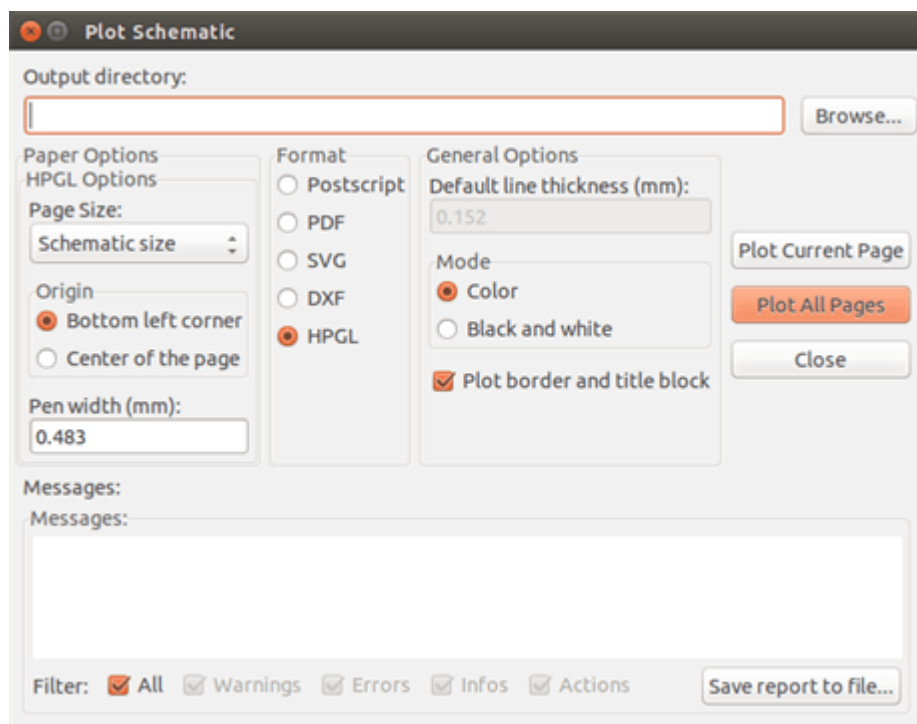
允许您使用 DXF 格式创建打印文件。文件名是展名.dxf 的工作表名称。

## 在 HPGL 中

此命令允许您创建 HPGL 文件。在这种格式中，您可以定义：

- 页面大小。
- 原点。
- 笔宽(mm)。

设置对话框如下所示：



输出文件名将是工作表名称加上扩展名 .plt。

## 尺寸

通常使用尺寸。在这种情况下，将使用菜单中定义的尺寸，并且所的比例将1。如果设置了不同的尺寸（A4 或 A0，或 A 设置自调整比例以填充面。

## 偏移调整

对于所有准尺寸，您可以调整偏移以尽可能准确地使形居中。由于在工作表的中心或左下角有原点，因此必须引入偏移以便正确


一般来

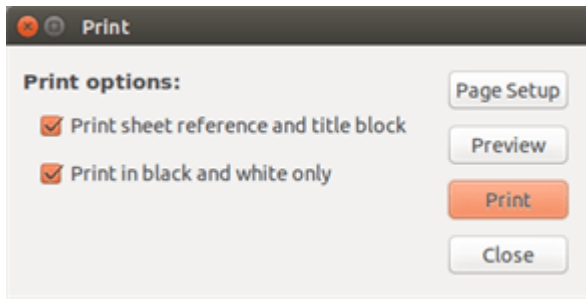
- 于原点位于中心的偏移量必须并置尺寸的一半。
- 于原点位于左下角的偏移量必须置

要置偏移量：

- 尺寸。
- 置偏移量 X 和偏移量 Y.
- 接受偏移量。

## 在上打印

This command, available via the icon , allows you to visualize and generate design files for the standard printer.



“打印表格参考和标题” 可启用或禁用表格参考和标题

“黑白打印” 置为色打印。如果使用黑白激光打印机，通常需要此选项因色打印成半色通常不太可能

# Symbol Editor

## 关于符号的一般信息

A symbol is a schematic element which contains a graphical representation, electrical connections, and text fields describing the symbol. Symbols used in a schematic are stored in symbol libraries. KiCad provides a symbol editing tool that allows you to create libraries, add, delete or transfer symbols between libraries, export symbols to files, and import symbols from files. The symbol editing tool provides a simple way to manage symbols and symbol libraries.

## 符号概述

符号由一个或多个符号组成。通常，符号按功能，类型和/或制造商进行分类。

符号由以下部分组成：

- Graphical items (lines, circles, arcs, text, etc.) that determine how symbol looks in a schematic.
- Pins which have both graphic properties (line, clock, inverted, low level active, etc.) and electrical properties (input, output, bidirectional, etc.) used by the Electrical Rules Check (ERC) tool.
- 如参考，PCB设计的相似封装名称等字段等。

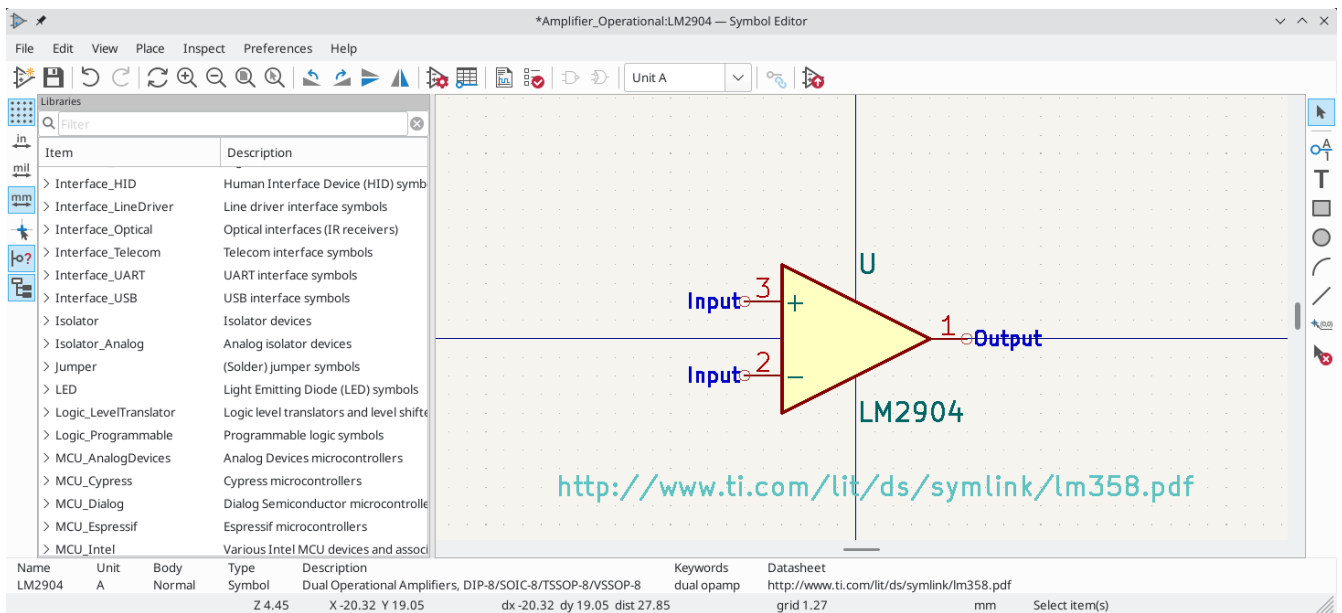
Symbols can be derived from another symbol in the same library. Derived symbols share the base symbol's graphical shape and pin definitions, but can override the base symbol's property fields (value, footprint, footprint filters, datasheet, description, etc.). Derived symbols can be used to define symbols that are similar to a base part. For example, 74LS00, 74HC00, and 7437 symbols could all be derived from a 7400 symbol. In previous versions of KiCad, derived symbols were referred to as aliases.

正确的符号设计需要：

- 定义符号是否由一个或多个元素组成。
- Defining if the symbol has an alternate body style (also known as a De Morgan representation).
- 使用线条，矩形，圆形，多边形和文本设计其符号表示。
- 通过设计定义每个引脚的图形元素，名称，编号和电气属性（输入，输出，三态源端口等）来添加引脚。
- Determining if the symbol should be derived from another symbol with the same graphical design and pin definition.
- 添加可选项字段，例如 PCB 设计件使用的封装名称和/或定义其可选项性。
- 通过添加描述字符串和数据表链接等来设计符号。
- 将其保存在所需的库中。

## 符号编辑器概述






















符号编辑器主窗口如下所示。它由三个工具栏组成，可快速访问常用功能和符号查看区域。并非所有命令都可在工具栏上使用，但可以使用菜单。



## 主工具










The main tool bar is located at the top of the main window. It consists of the undo/redo commands, zoom commands, symbol properties dialogs, and unit/representation management controls.



	Create a new symbol in the selected library.
	Save the currently selected library. All modified symbols in the library will be saved.
	Undo last edit.
	Redo last undo.
	Refresh display.
	Zoom in.
	Zoom out.
	Zoom to fit symbol in display.
	Zoom to fit selection.
	Rotate counter-clockwise.
	Rotate clockwise.
	Mirror horizontally.
	Mirror vertically.
	Edit the current symbol properties.
	Edit the symbol's pins in a tabular interface.
	Open the symbol's datasheet. The button will be disabled if no datasheet is defined for the current symbol.
	Test the current symbol for design errors.
	Select the normal body style. The button is disabled if the current symbol does not have an alternate body style.
	Select the alternate body style. The button is disabled if the current symbol does not have an alternate body style.
	Select the unit to display. The drop down control will be disabled if the current symbol is not derived from a symbol with multiple units.
	Enable synchronized pins edit mode. When this mode is enabled, any pin modifications are propagated to all other symbol units. Pin number changes are not propagated. This mode is automatically enabled for symbols with multiple interchangeable units and cannot be enabled for symbols with only one unit.








## 元素工具

The vertical toolbar located on the right hand side of the main window allows you to place all of the elements required to design a symbol.

	Select tool. Right-clicking with the select tool opens the context menu for the object under the cursor. Left-clicking with the select tool displays the attributes of the object under the cursor in the message panel at the bottom of the main window. Double-left-clicking with the select tool will open the properties dialog for the object under the cursor.
	Pin tool. Left-click to add a new pin.
	Graphical text tool. Left-click to add a new graphical text item.
	Rectangle tool. Left-click to begin drawing the first corner of a graphical rectangle. Left-click again to place the opposite corner of the rectangle.
	Circle tool. Left-click to begin drawing a new graphical circle from the center. Left-click again to define the radius of the circle.
	Arc tool. Left-click to begin drawing a new graphical arc item from the first arc end point. Left-click again to define the second arc end point. Adjust the radius by dragging the arc center point.
	Connected line tool. Left-click to begin drawing a new graphical line item in the current symbol. Left-click for each additional connected line. Double-left-click to complete the line.
	Anchor tool. Left-click to set the anchor position of the symbol.
	Delete tool. Left-click to delete an object from the current symbol.


## 编辑工具

The vertical tool bar located on the left hand side of the main window allows you to set some of the editor drawing options.

	Toggle grid visibility on and off.
	Set units to inches.
	Set units to mils (0.001 inch).
	Set units to millimeters.
	Toggle full screen cursor on and off.
	Toggle display of pin electrical types.
	Toggle display of libraries and symbols.




## □□□与□□

The selection of the current library is possible via the  icon which shows you all available libraries and allows you to select one. When a symbol is loaded or saved, it will be put in this library. The library name of a symbol is the contents of its `Value` field.

## □□并保存符号

### 符号□□

Clicking the  icon on the left tool bar toggles the treeview of libraries and symbols. Clicking on a symbol opens that symbol.

#### NOTE

Some symbols are derived from other symbols. Derived symbol names are displayed in *italics* in the treeview. If a derived symbol is opened, its symbol graphics will not be editable. Its symbol fields will be editable as normal. To edit the graphics of a base symbol and all of its derived symbols, open the base symbol.

## 保存符号

After modification, a symbol can be saved in the current library or a different library.

To save the modified symbol in the current library, click the  icon. The modifications will be written to the existing symbol.

#### NOTE

Saving a modified symbol also saves all other modified symbols in the same library.

To save the symbol changes to a new symbol, click **File** → **Save As....** The symbol can be saved in the current library or a different library. A new name can be set for the symbol.

To create a new file containing only the current symbol, click **File** → **Export** → **Symbol....** This file will be a standard library file which will contain only one symbol.

## □建□符号

### □建一个新符号

A new symbol can be created by clicking the  icon. You will be asked for a number of symbol properties.

- A symbol name (this name is used as the default value for the `Value` field in the schematic editor)
- An optional base symbol to derive the new symbol from. The new symbol will use the base symbol's graphical shape and pin configuration, but other symbol information can be modified in the derived symbol. The base symbol must be in the same library as the new derived symbol.
- The reference designator prefix ( U , C , R ...).
- The number of units per package, and whether those units are interchangeable (for example a 7400 is made of 4 units per package).
- If an alternate body style (sometimes referred to as a "De Morgan equivalent") is desired.
- Whether the symbol is a power symbol. Power symbols appear in the "Add Power Port" dialog in the Schematic editor, their `Value` fields are not editable in the schematic, they cannot be assigned a

footprint and they are not added to the PCB, and they are not included in the bill of materials.

- Whether the symbol should be excluded from the bill of materials.
- Whether the symbol should be excluded from the PCB.

There are also several graphical options.

- The offset between the end of each pin and its pin name.
- Whether the pin number and pin name should be displayed.
- Whether the pin names should be displayed alongside the pins or at the ends of the pins inside the symbol body.

These properties can also be changed later in the [Symbol Properties window](#).

Symbol name:

Derive from existing symbol:

Default reference designator:

Number of units per package:    ☐ Units are not interchangeable

☐ Create symbol with alternate body style (De Morgan)

☐ Create symbol as power symbol

☐ Exclude from schematic bill of materials

☐ Exclude from board

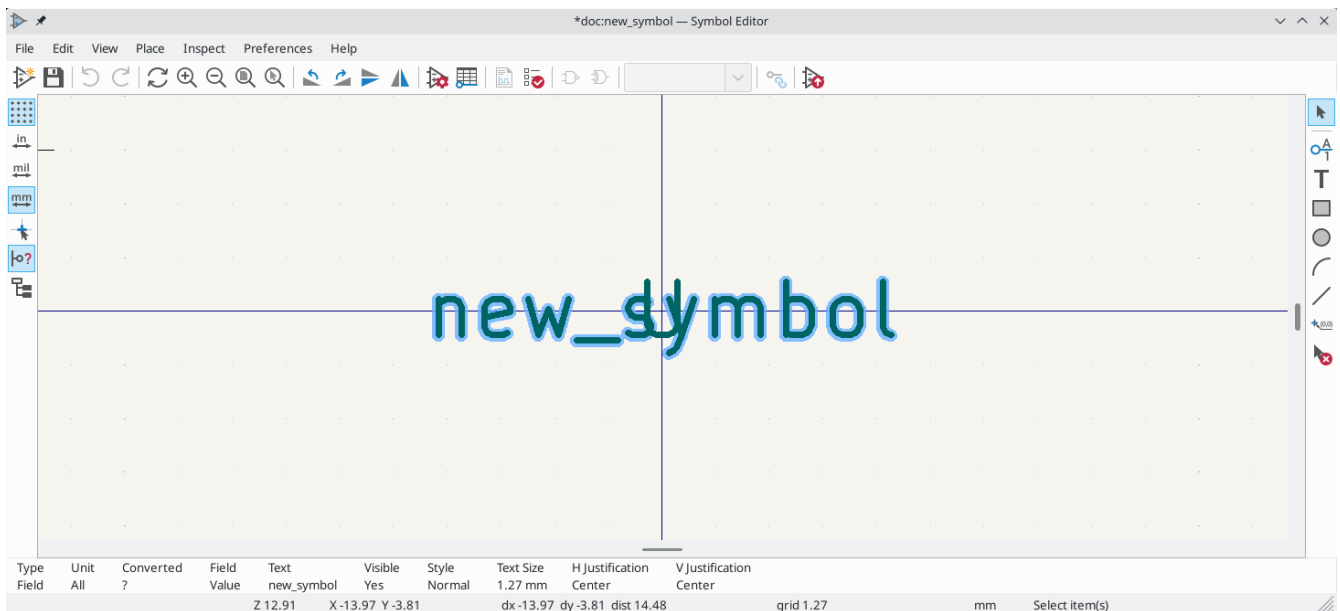
Pin name position offset:  mm


☒ Show pin number text

☒ Show pin name text

☒ Pin name inside

将使用上面的属性新建一个新符号，它将出现在器件中，如下所示。




The blue cross in the center is the symbol anchor, which specifies the symbol origin i.e. the coordinates (0, 0). The anchor can be repositioned by selecting the  icon and clicking on the new desired anchor position.

## 从另一个符号新建符号

通常，您要制作的符号类似于符号中已有的符号。在这种情况下，很容易添加和修改有符号。

- 添加将用作起点的符号。
- Save a new copy of the symbol using **File** → **Save As....** The Save As dialog will prompt for a name for the new symbol and the library to save it in.
- 根据需要新建符号。
- Save the modified symbol.

## 符号属性

Symbol properties are set when the symbol is created but they can be modified at any point. To change the symbol properties, click on the  icon to show the dialog below.

Library Symbol Properties

General

Footprint Filters

Fields

Name	Value	Show	H Align	V Align	Italic	Bold	Text Size
Reference	U	<input type="checkbox"/>	Left	Center	<input type="checkbox"/>	<input type="checkbox"/>	1.27 mm
Value	LM2904	<input checked="" type="checkbox"/>	Left	Center	<input type="checkbox"/>	<input type="checkbox"/>	1.27 mm
Footprint		<input type="checkbox"/>	Center	Center	<input type="checkbox"/>	<input type="checkbox"/>	1.27 mm
Datasheet	<a href="http://www.ti.com/lit/ds/symlink/lm358.pdf">http://www.ti.com/lit/ds/symlink/lm358.pdf</a>	<input type="checkbox"/>	Center	Center	<input type="checkbox"/>	<input type="checkbox"/>	1.27 mm

+

↑

↓

✖

Symbol name:

LM2904

Description:

Dual Operational Amplifiers, DIP-8/SOIC-8/TSSOP-8/VSSOP-8

Keywords:

dual opamp

Derive from symbol:

Symbol

☐ Has alternate body style (De Morgan)

☐ Define as power symbol

☐ Exclude from schematic bill of materials

☐ Exclude from board

Number of Units:

3

—

+

☐ All units are interchangeable

Pin Text Options

☒ Show pin number

☒ Show pin name

☐ Place pin names inside

Position offset:

0.127

mm

Edit Spice Model...

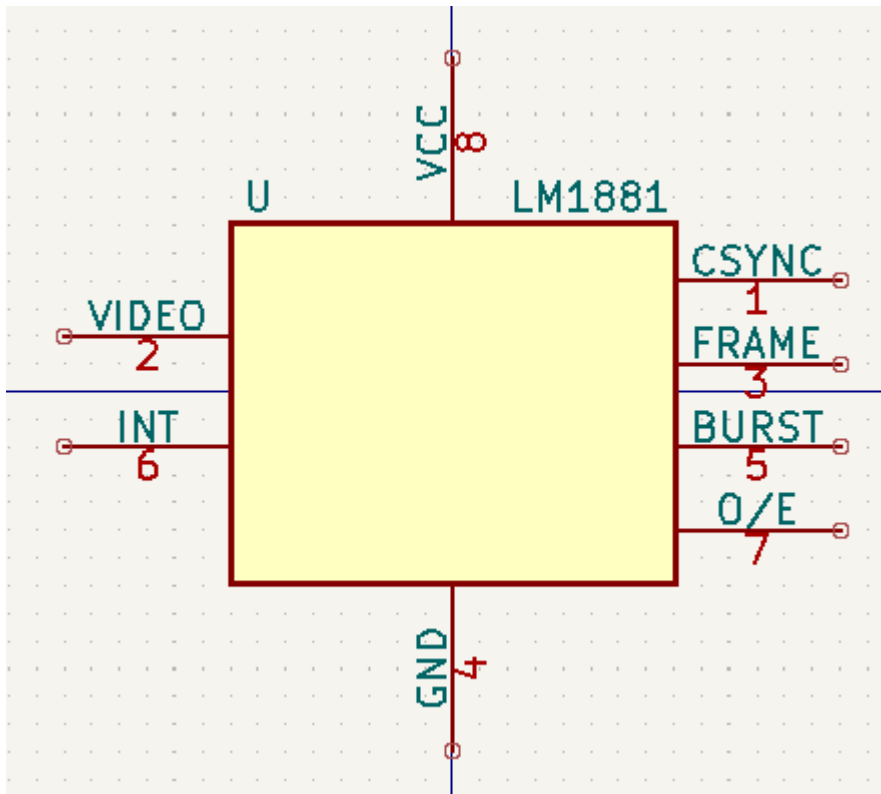
Cancel

OK

It is important to correctly set the number of units per package and the alternate symbolic representation, if enabled, because when pins are edited or created the corresponding pins for each unit will be affected. If you change the number of units per package after pin creation and editing, there will be additional work to specify the pins and graphics for the new unit. Nevertheless, it is possible to modify these properties at any time.

The graphic options "Show pin number" and "Show pin name" define the visibility of the pin number and pin name text. The option "Place pin names inside" defines the pin name position relative to the pin body. The pin names will be displayed inside the symbol outline if the option is checked. In this case the "Pin Name Position Offset" property defines the shift of the text away from the body end of the pin. A value from 0.02 to 0.05 inches is usually reasonable.

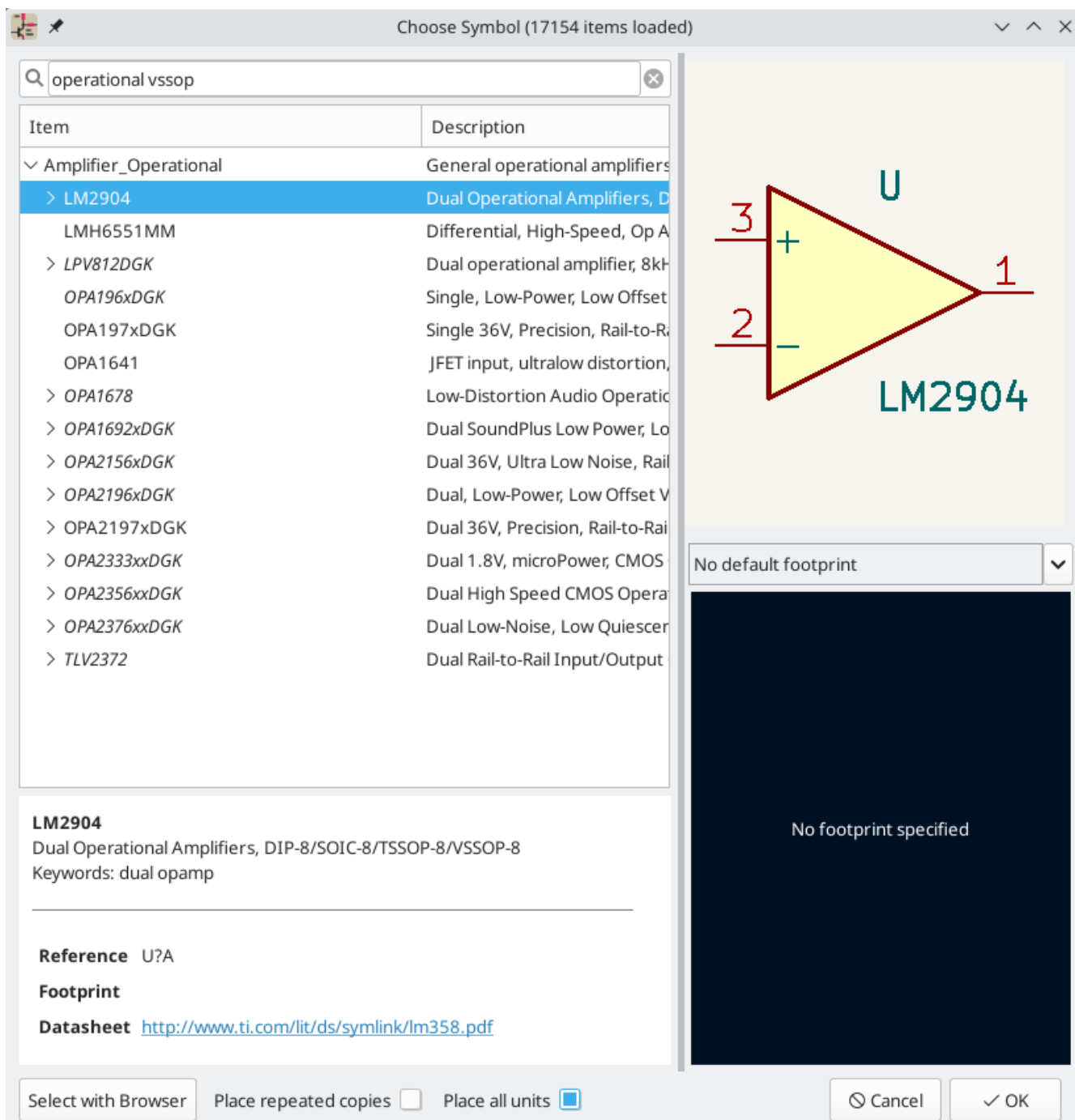
下面的示例显示了未选中“放置引脚名称”的符号。注意名称和引脚号的位置。



### Symbol Name, Description, and Keywords

The symbol's name is the same as the `Value` field. When the symbol name is changed the value also changes, and vice versa. The symbol's name in the library also changes accordingly.

The symbol description should contain a brief description of the component, such as the component function, distinguishing features, and package options. The keywords should contain additional terms related to the component. Keywords are used primarily to assist in searching for the symbol.



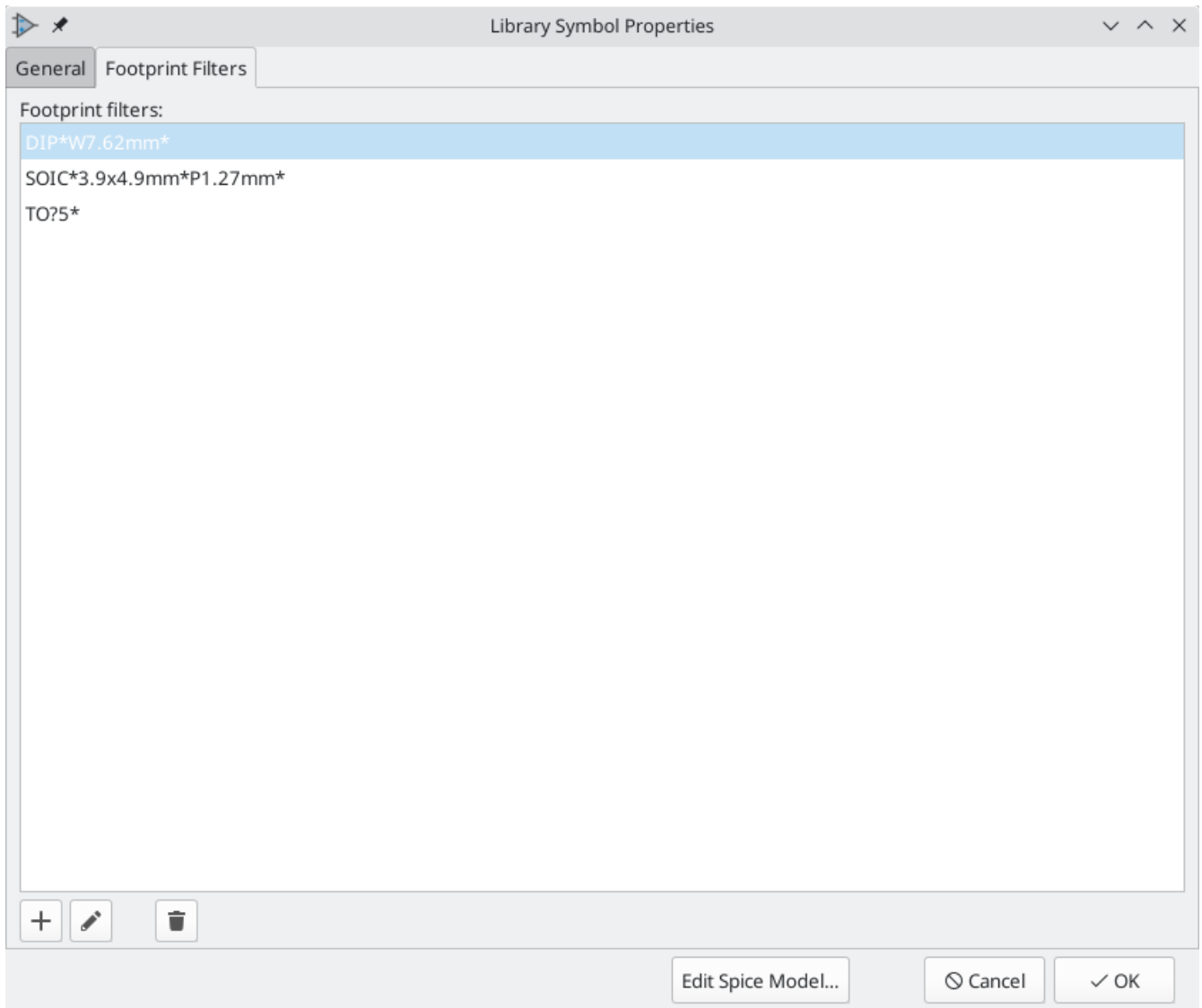
A symbol's name, description, and keywords are all used when searching for symbols in the Symbol Editor and Add a Symbol dialog. The description and keywords are displayed in the Symbol Library Browser and Add a Symbol dialog.

## Footprint Filters


The footprint filters tab is used to define which footprints are appropriate to use with the symbol. The filters can be applied in the Footprint Assignment tool so that only appropriate footprints are displayed for each symbol.

Multiple footprint filters can be defined. Footprints that match any of the filters will be displayed; if no filters are defined, then all footprints will be displayed.

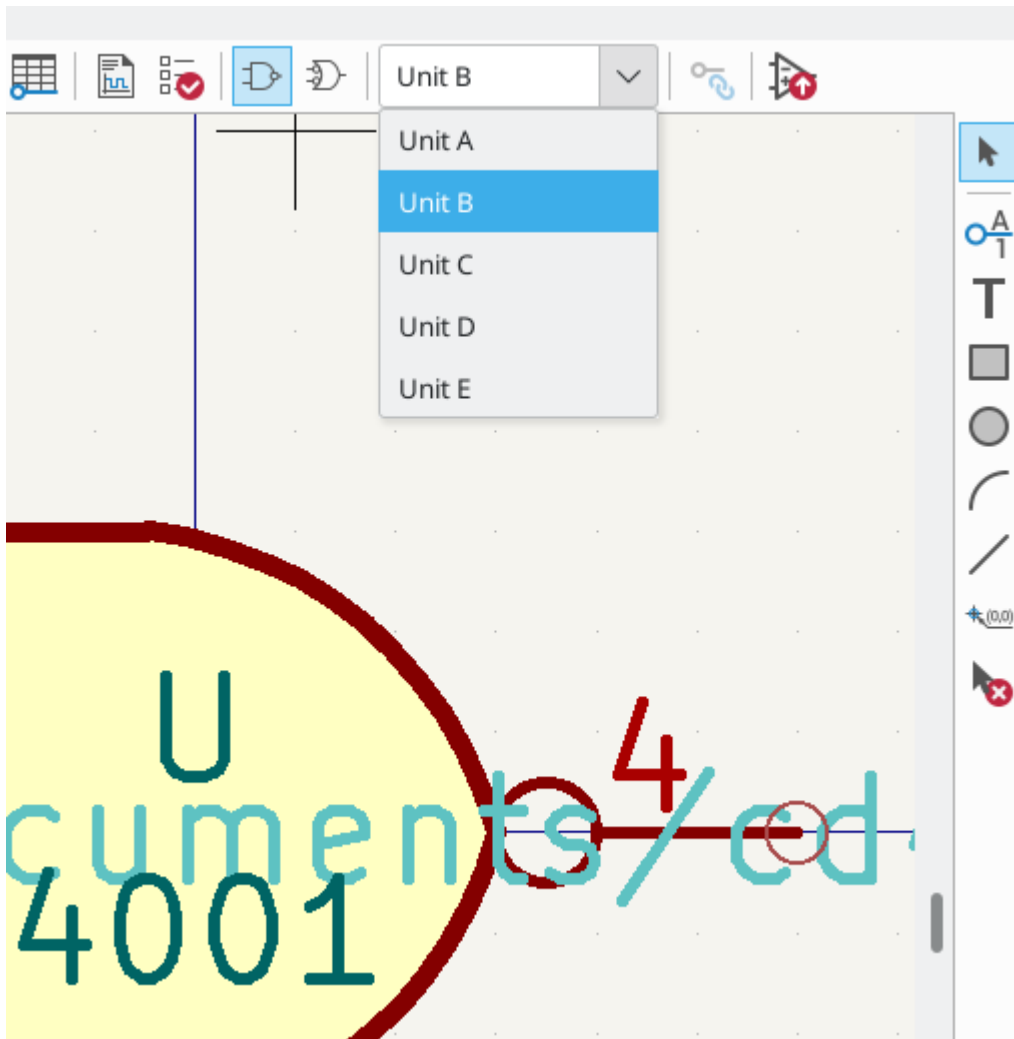
Filters can use wildcards: `*` matches any number of characters, including zero, and `?` matches zero or one characters. For example, `SOIC-*` would match the `SOIC-8_3.9x4.9mm_P1.27mm` footprint as well as any other footprint beginning with `SOIC-`. The filter `SOT?23` matches `SOT23` as well as `SOT-23`.



## □有替代符号表示的符号

If the symbol has an alternate body style defined, one body style must be selected for editing at a time. To edit the normal representation, click the  icon.

To edit the alternate representation, click on the  icon. Use the  dropdown shown below to select the unit you wish to edit.



## 图形元素

Graphical elements create the visual representation of a symbol and contain no electrical connection information. Graphical elements are created with the following tools:

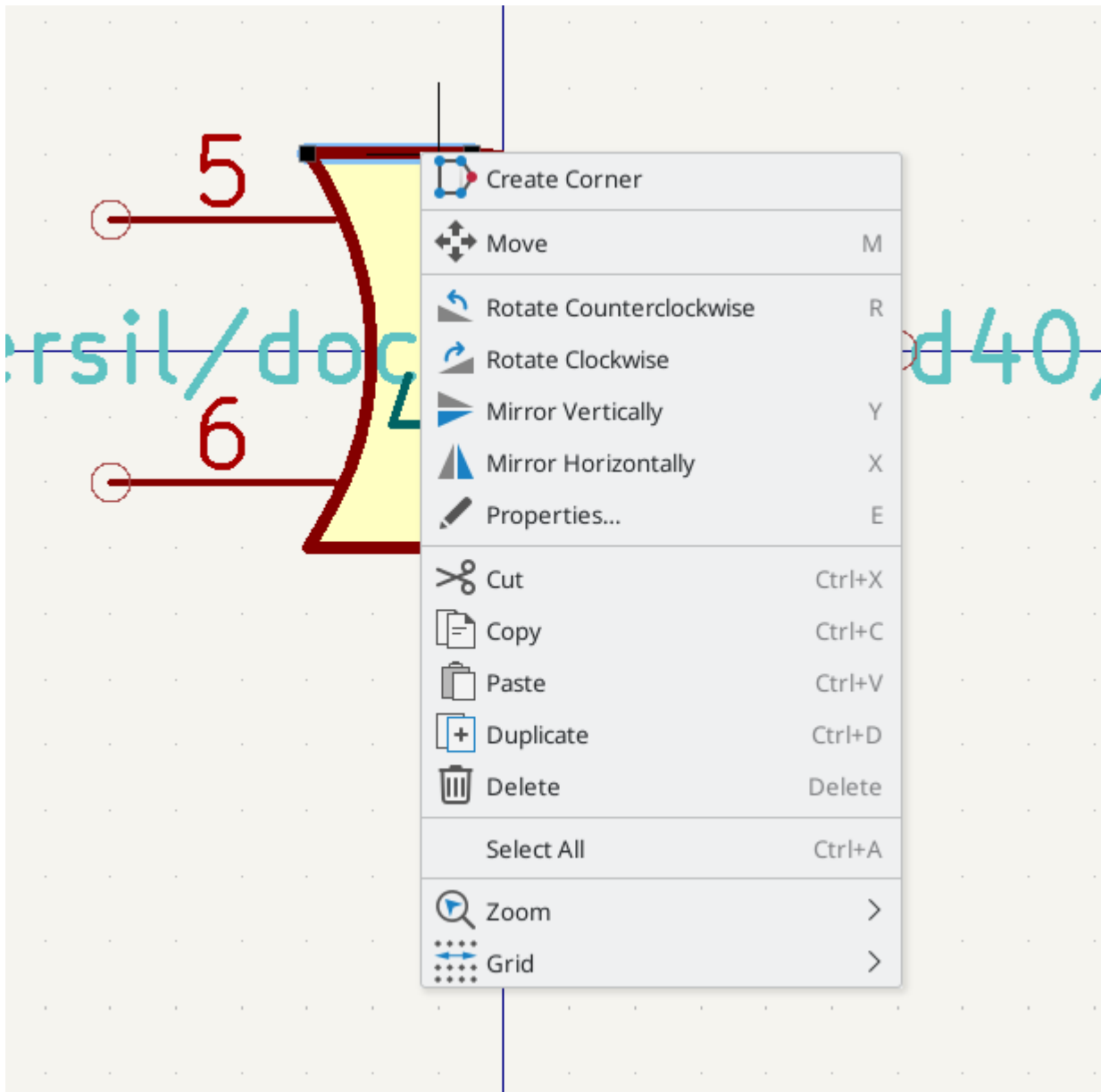
- 由起点和点定义的圆和多圆。
- 由两个角定义的矩形。
- 由中心和半径定义的圆。
- 由弧的起点和点及其中心定义的弧。弧度从0°到180°。

主窗口右侧的垂直工具允许您放置符号表示所需的所有图形元素。

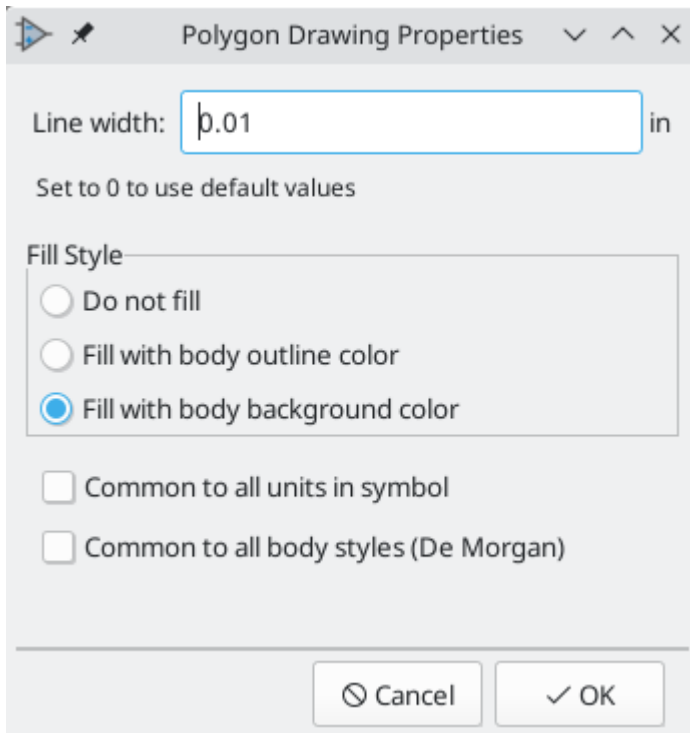
## 图形元素成组

每个图形元素（圆、弧等）可以被定义于所有元件和/或主体式是共同的或者于特定元件和/或主体式是特定的。右侧的元素可以快速显示元素以显示所元素的上下文菜单。下面是元素的上下文菜单。





您可以双击元素以修改其属性。下面是多边形元素的属性面板。



□形元素的属性是:

- "Line width" defines the width of the element's line in the current drawing units.
- "Fill Style" determines if the shape defined by the graphical element is to be drawn unfilled, background filled, or foreground filled.
- "Common to all units in symbol" determines if the graphical element is drawn for each unit in symbol with more than one unit per package or if the graphical element is only drawn for the current unit.
- "Common to all body styles (De Morgan)" determines if the graphical element is drawn for each symbolic representation in symbols with an alternate body style or if the graphical element is only drawn for the current body style.

## □形文本元素

The **T** icon allows for the creation of graphical text. Graphical text is automatically oriented to be readable, even when the symbol is mirrored. Please note that graphical text items are not the same as symbol fields.

## 每个符号多个□位和替代体型□式

Symbols can have up to two body styles (a standard symbol and an alternate symbol often referred to as a "De Morgan equivalent") and/or have more than one unit per package (logic gates for example). Some symbols can have more than one unit per package each with different symbols and pin configurations.

Consider for instance a relay with two switches, which can be designed as a symbol with three different units: a coil, switch 1, and switch 2. Designing a symbol with multiple units per package and/or alternate body styles is very flexible. A pin or a body symbol item can be common to all units or specific to a given unit or they can be common to both symbolic representation so are specific to a given symbol representation.

By default, pins are specific to a unit and body style. When a pin is common to all units or all body styles, it only needs to be created once. This is also the case for the body style graphic shapes and text, which may be

common to each unit, but typically are specific to each body style).

## Example of a Symbol With Multiple Noninterchangeable Units

For an example of a symbol with multiple units that are not interchangeable, consider a relay with 3 units per package: a coil, switch 1, and switch 2.

The three units are not all the same, so "All units are interchangeable" should be deselected in the Symbol Properties dialog. Alternatively, this option could have been specified when the symbol was initially created.

The image shows the 'Library Symbol Properties' dialog box for a symbol named 'G5V-2\_Split'. The 'Footprint Filters' tab is active. The 'Fields' section contains a table with four rows: Reference, Value, Footprint, and Datasheet. The 'Reference' row is selected. Below the table are buttons for adding, moving up, moving down, and deleting fields. The 'Symbol name' is 'G5V-2\_Split', the 'Description' is 'Relay Miniature Omron DPDT', and the 'Keywords' are 'Miniature Relay Dual Pole DPDT Omron'. The 'Derive from symbol' dropdown is empty. The 'Symbol' section has four unchecked checkboxes: 'Has alternate body style (De Morgan)', 'Define as power symbol', 'Exclude from schematic bill of materials', and 'Exclude from board'. The 'Number of Units' is set to 3. The 'All units are interchangeable' checkbox is unchecked. The 'Pin Text Options' section has three checked checkboxes: 'Show pin number', 'Show pin name', and 'Place pin names inside'. The 'Position offset' is 0.02 in. At the bottom are buttons for 'Edit Spice Model...', 'Cancel', and 'OK'.

Name	Value	Show	H Align	V Align	Italic	Bold	Text Size
Reference	K	<input type="checkbox"/>	Left	Center	<input type="checkbox"/>	<input type="checkbox"/>	0.05 in
Value	G5V-2_Split	<input checked="" type="checkbox"/>	Left	Center	<input type="checkbox"/>	<input type="checkbox"/>	0.05 in
Footprint	Relay_THT:Relay_DPDT_Omron_G5V-2	<input type="checkbox"/>	Left	Center	<input type="checkbox"/>	<input type="checkbox"/>	0.05 in
Datasheet	<a href="http://omronfs.omron.com/en_US/ecb/products/pdf/en-g5v">http://omronfs.omron.com/en_US/ecb/products/pdf/en-g5v</a>	<input type="checkbox"/>	Center	Center	<input type="checkbox"/>	<input type="checkbox"/>	0.05 in

Symbol name: G5V-2\_Split

Description: Relay Miniature Omron DPDT

Keywords: Miniature Relay Dual Pole DPDT Omron

Derive from symbol: ▼

Symbol

- ☐ Has alternate body style (De Morgan)
- ☐ Define as power symbol
- ☐ Exclude from schematic bill of materials
- ☐ Exclude from board

Number of Units: 3 — +

☐ All units are interchangeable

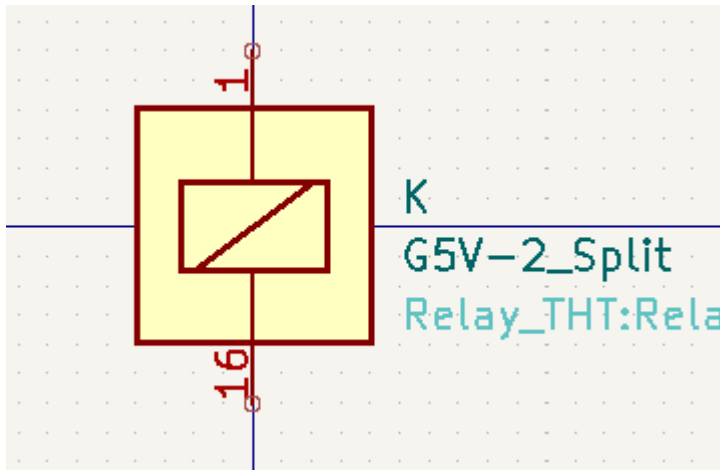
Pin Text Options

- ☒ Show pin number
- ☒ Show pin name
- ☒ Place pin names inside

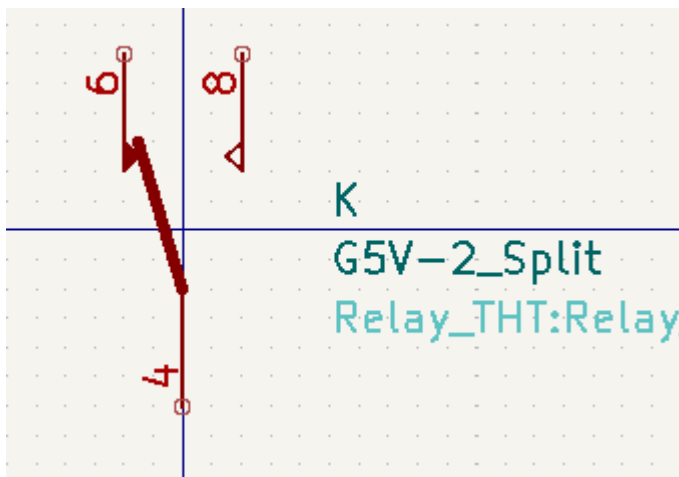
Position offset: 0.02 in

Edit Spice Model... Cancel OK

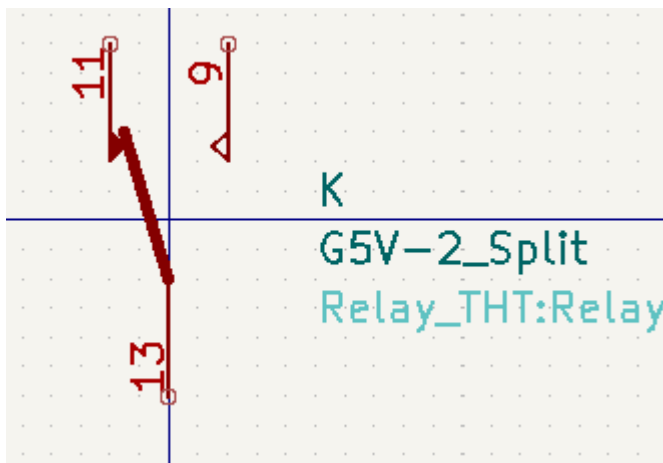
## Unit A



## Unit B




## Unit C



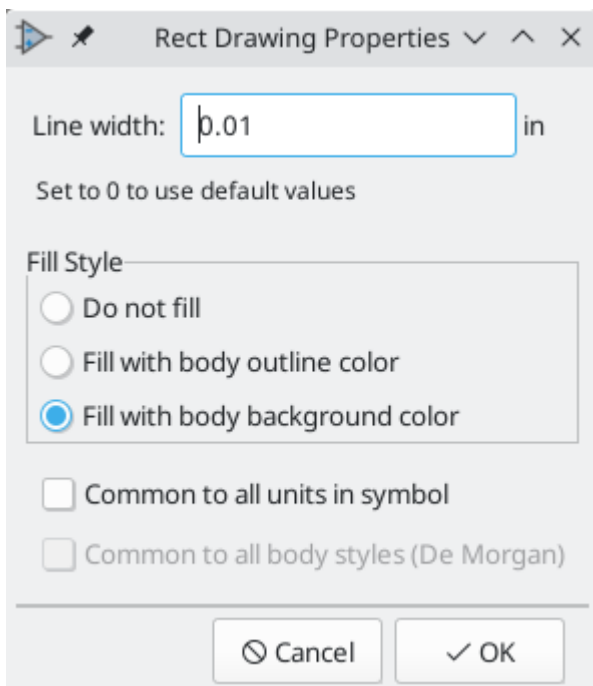
Unit A does not have the same symbol and pin layout as Units B and C, so the units are not interchangeable.

### NOTE

"Synchronized Pins Edit Mode" can be enabled by clicking the  icon. In this mode, pin modifications are propagated between symbol units; changes made in one unit will be reflected in the other units as well. When this mode is disabled, pin changes made in one unit do not affect other units. This mode is enabled automatically when "All units are interchangeable" is checked, but it can be disabled. The mode cannot be enabled when "All units are interchangeable" is unchecked or when the symbol only has one unit.

## 图形符号元素

Shown below are properties for a graphic body element. In the relay example above, the three units have different symbolic representations. Therefore, each unit was created separately and the graphical body elements have the "Common to all units in symbol" setting disabled.



## 引脚创建

You can click on the  icon to create and insert a pin. The editing of all pin properties is done by double-clicking on the pin or right-clicking on the pin to open the pin context menu. Pins must be created carefully, because any error will have consequences on the PCB design. Any pin already placed can be edited, deleted, and/or moved.

## 引脚概述

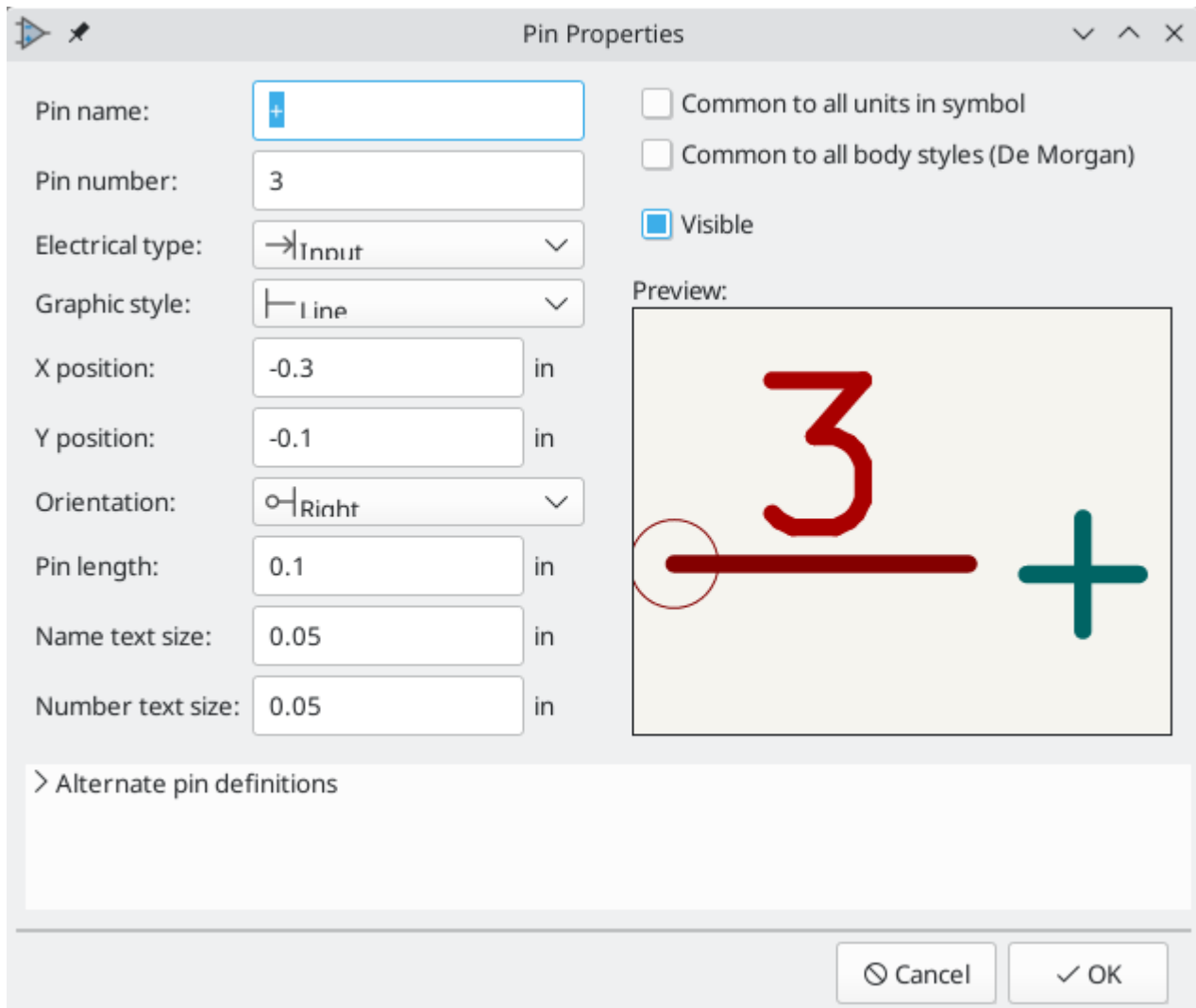
A pin is defined by its graphical representation, its name and its number. The pin's name and number can contain letters, numbers, and symbols, but not spaces. For the Electrical Rules Check (ERC) tool to be useful, the pin's electrical type (input, output, tri-state...) must also be defined correctly. If this type is not defined properly, the schematic ERC check results may be invalid.

### 重要笔记

- Symbol pins are matched to footprint pads by number. The pin number in the symbol must match the corresponding pad number in the footprint.
- Do not use spaces in pin names and numbers. Spaces will be automatically replaced with underscores (`_`).
- To define a pin name with an inverted signal (overline) use the `~` (tilde) character followed by the text to invert in braces. For example `~{FO}O` would display  $\overline{FO}$  O.
- If the pin name is empty, the pin is considered unnamed.
- Pin names can be repeated in a symbol.

Pin numbers must be unique in a symbol.

## 引脚属性



The image shows a 'Pin Properties' dialog box with various settings for a pin. The settings are as follows:

Property	Value
Pin name:	+
Pin number:	3
Electrical type:	Input
Graphic style:	Line
X position:	-0.3 in
Y position:	-0.1 in
Orientation:	Right
Pin length:	0.1 in
Name text size:	0.05 in
Number text size:	0.05 in

Additional options on the right:

- ☐ Common to all units in symbol
- ☐ Common to all body styles (De Morgan)
- ☒ Visible

Preview: A visual representation of the pin, showing a red horizontal line with a red circle at the left end, a large red number '3' above it, and a green plus sign to the right.

> Alternate pin definitions

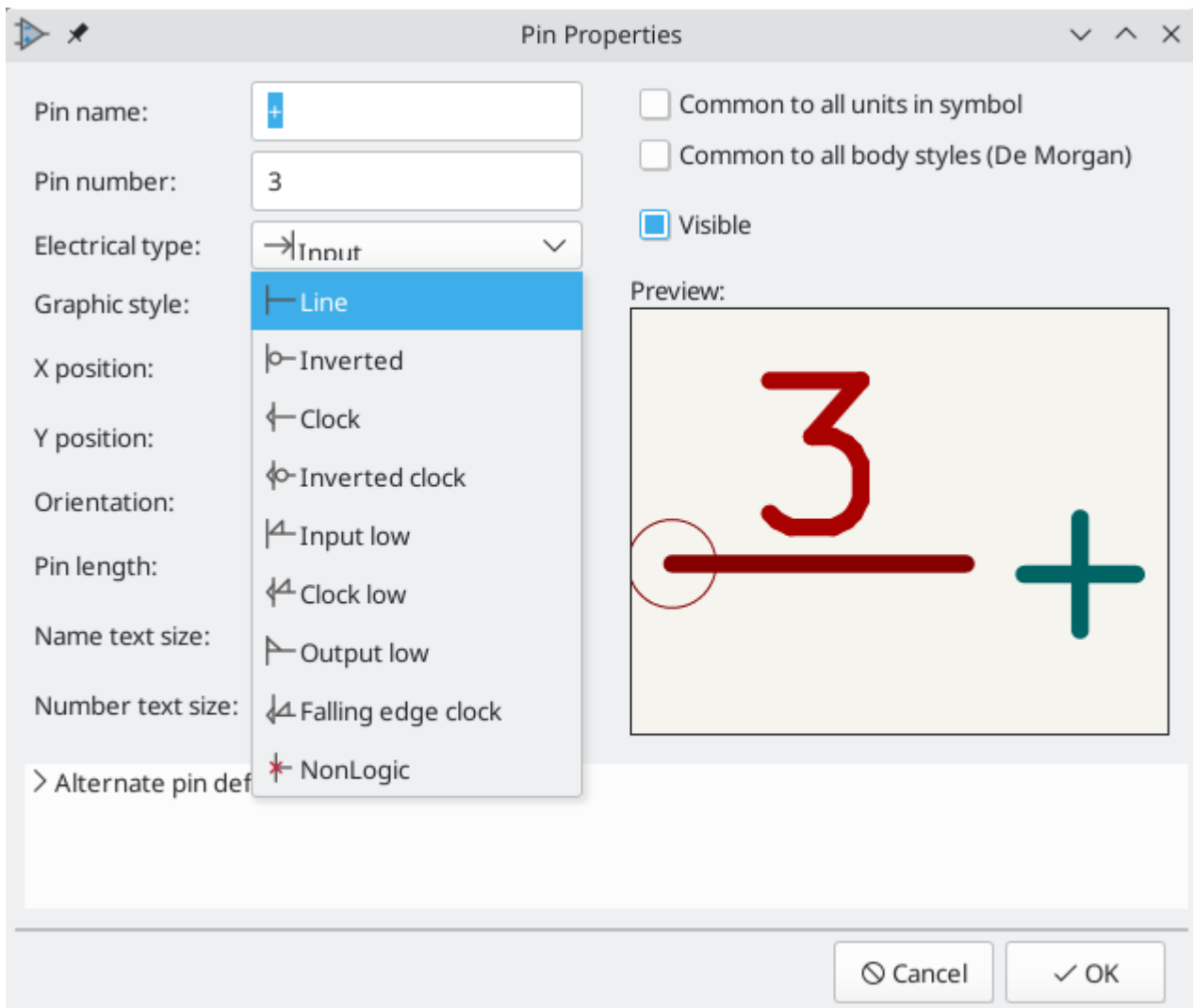
Buttons: Cancel, OK

引脚属性对话框允许您定义引脚的所有特性。新建引脚或双击有引脚的元件会自动弹出此对话框。此对话框允许您修改：

- The pin name and text size.
- The pin number and text size.
- The pin length.
- The pin electrical type and graphical style.
- 位和替代代表成表格。
- Pin visibility.
- [Alternate pin definitions](#).

## Pin Graphic Styles

Shown in the figure below are the different pin graphic styles. The choice of graphic style does not have any influence on the pin's electrical type.



## 引脚电气型

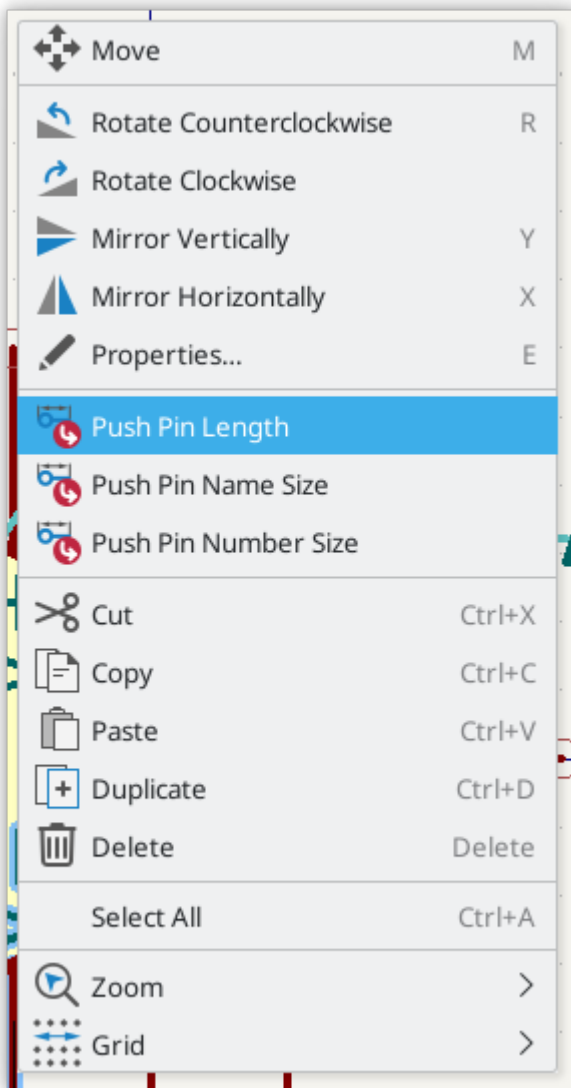
Choosing the correct electrical type is important for the schematic ERC tool. ERC will check that pins are connected appropriately, for example ensuring that input pins are driven and power inputs receive power from an appropriate source.

Pin Type	Description
Input	A pin which is exclusively an input.
Output	A pin which is exclusively an output.
Bidirectional	A pin that can be either an input or an output, such as a microcontroller data bus pin.
Tri-state	A three state output pin (high, low, or high impedance)
Passive	A passive symbol pin: resistors, connectors, etc.
Free	A pin that can be freely connected to any other pin without electrical concerns.
Unspecified	A pin for which the ERC check does not matter.
Power input	A symbol's power pin. As a special case, power input pins that are marked invisible are automatically connected to the net with the same name. See the <a href="#">Power Ports section</a> for more information.
Power output	A pin that provides power to other pins, such as a regulator output.
Open collector	An open collector logic output.
Open emitter	An open emitter logic output.
Unconnected	A pin that should not be connected to anything.

## Pushing Pin Properties to Other Pins


You can apply the length, name size, or number size of a pin to the other pins in the symbol by right clicking the pin and selecting **Push Pin Length**, **Push Pin Name Size**, or **Push Pin Number Size**, respectively.







## 多个单元和用符号表示定义引脚

Symbols with multiple units and/or graphical representations are particularly problematic when creating and editing pins. The majority of pins are specific to each symbol unit (because each unit has a different set of pins) and to each body style (because the form and position is different between the normal body style and the alternate form).


The symbol library editor allows the simultaneous creation of pins. By default, changes made to a pin are made for all units of a multiple unit symbol and to both representations for symbols with an alternate symbolic representation. The only exception to this is the pin's graphical type and name, which remain unlinked between symbol units and body styles. This dependency was established to allow for easier pin creation and editing in most cases. This dependency can be disabled by toggling the  icon on the main tool bar. This will allow you to create pins for each unit and representation completely independently.

Pins can be common or specific to different units. Pins can also be common to both symbolic representations or specific to each symbolic representation. When a pin is common to all units, it only has to be drawn once. Pins are set as common or specific in the pin properties dialog.

An example is the output pin in the 7400 quad dual input NAND gate. Since there are four units and two symbolic representations, there are eight separate output pins defined in the symbol definition. When creating a new 7400 symbol, unit A of the normal symbolic representation will be shown in the library

editor. To edit the pin style in the alternate symbolic representation, it must first be enabled by clicking the  button on the tool bar. To edit the pin number for each unit, select the appropriate unit using the  drop down control.

## Pin Table

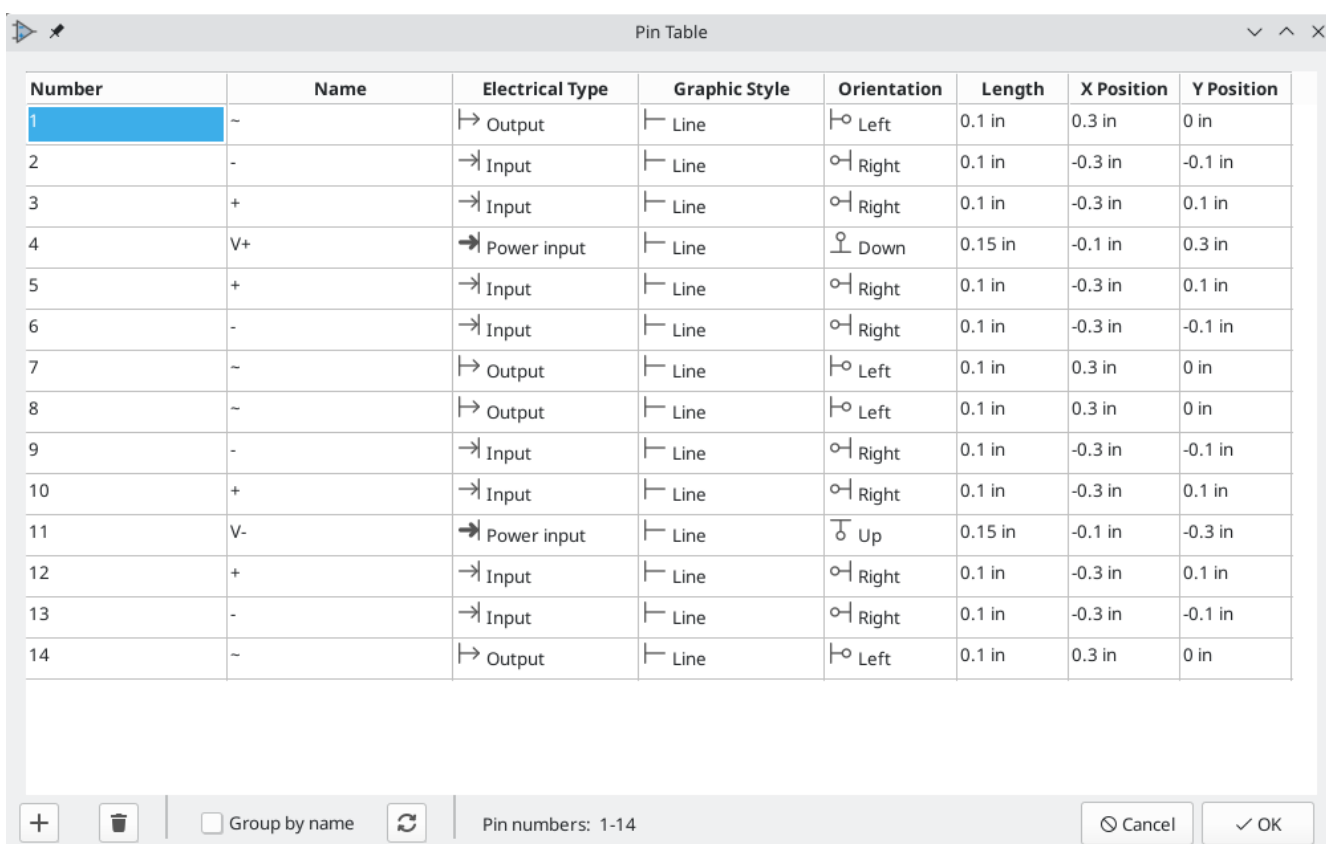
Another way to edit pins is to use the Pin Table, which is accessible via the  icon. The Pin Table displays all of the pins in the symbol and their properties in a table view, so it is useful for making bulk pin changes.

Any pin property can be edited by clicking on the appropriate cell. Pins can be added and removed with the  and  icons, respectively.

### NOTE

Columns of the pin table can be shown or hidden by right-clicking on the header row and checking or unchecking additional columns. Some columns are hidden by default.

The screenshot below shows the pin table for a quad opamp.

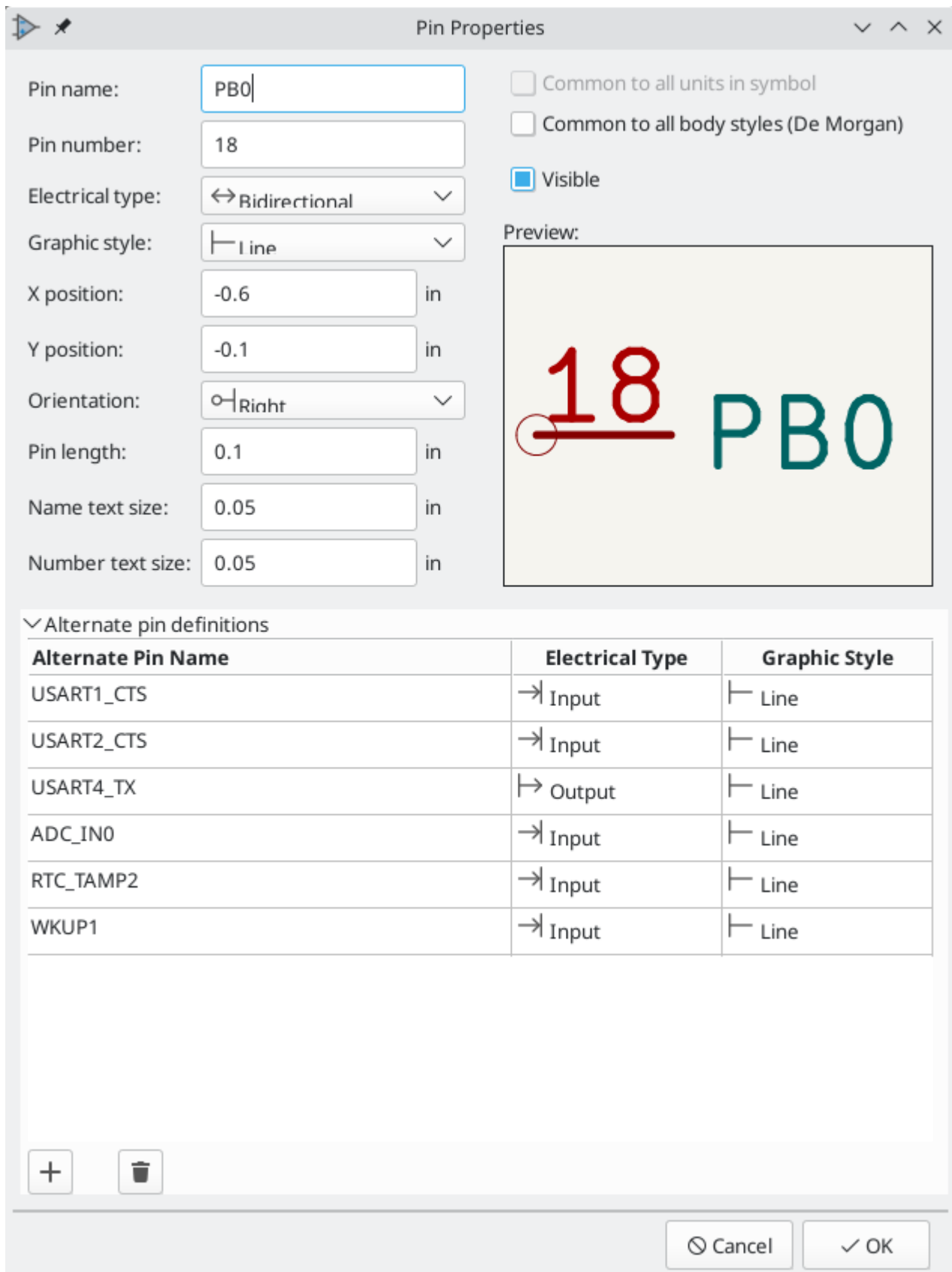


Number	Name	Electrical Type	Graphic Style	Orientation	Length	X Position	Y Position
1	~	Output	Line	Left	0.1 in	0.3 in	0 in
2	-	Input	Line	Right	0.1 in	-0.3 in	-0.1 in
3	+	Input	Line	Right	0.1 in	-0.3 in	0.1 in
4	V+	Power input	Line	Down	0.15 in	-0.1 in	0.3 in
5	+	Input	Line	Right	0.1 in	-0.3 in	0.1 in
6	-	Input	Line	Right	0.1 in	-0.3 in	-0.1 in
7	~	Output	Line	Left	0.1 in	0.3 in	0 in
8	~	Output	Line	Left	0.1 in	0.3 in	0 in
9	-	Input	Line	Right	0.1 in	-0.3 in	-0.1 in
10	+	Input	Line	Right	0.1 in	-0.3 in	0.1 in
11	V-	Power input	Line	Up	0.15 in	-0.1 in	-0.3 in
12	+	Input	Line	Right	0.1 in	-0.3 in	0.1 in
13	-	Input	Line	Right	0.1 in	-0.3 in	-0.1 in
14	~	Output	Line	Left	0.1 in	0.3 in	0 in

## Alternate Pin Definitions

Pins can have alternate pin definitions added to them. Alternate pin definitions allow a user to select a different name, electrical type, and graphical style for a pin when the symbol has been placed in the schematic. This can be used for pins that have multiple functions, such as microcontroller pins.

Alternate pin definitions are added in the Pin Properties dialog as shown below. Each alternate definition contains a pin name, electrical type, and graphic style. This microcontroller pin has all of its peripheral functions defined in the symbol as alternate pin names.



The Pin Properties dialog box is used to configure the properties of a pin symbol. It includes fields for Pin name, Pin number, Electrical type, Graphic style, X position, Y position, Orientation, Pin length, Name text size, and Number text size. It also has checkboxes for 'Common to all units in symbol', 'Common to all body styles (De Morgan)', and 'Visible'. A Preview window shows the resulting symbol. Below the main fields is a section for 'Alternate pin definitions' containing a table of alternate pin names, electrical types, and graphic styles. At the bottom are buttons for '+', a trash icon, 'Cancel', and 'OK'.

Pin name: 
☐ Common to all units in symbol
☐ Common to all body styles (De Morgan)
☒ Visible

Pin number:

Electrical type:

Graphic style:

X position:  in


Y position:  in

Orientation:

Pin length:  in

Name text size:  in

Number text size:  in

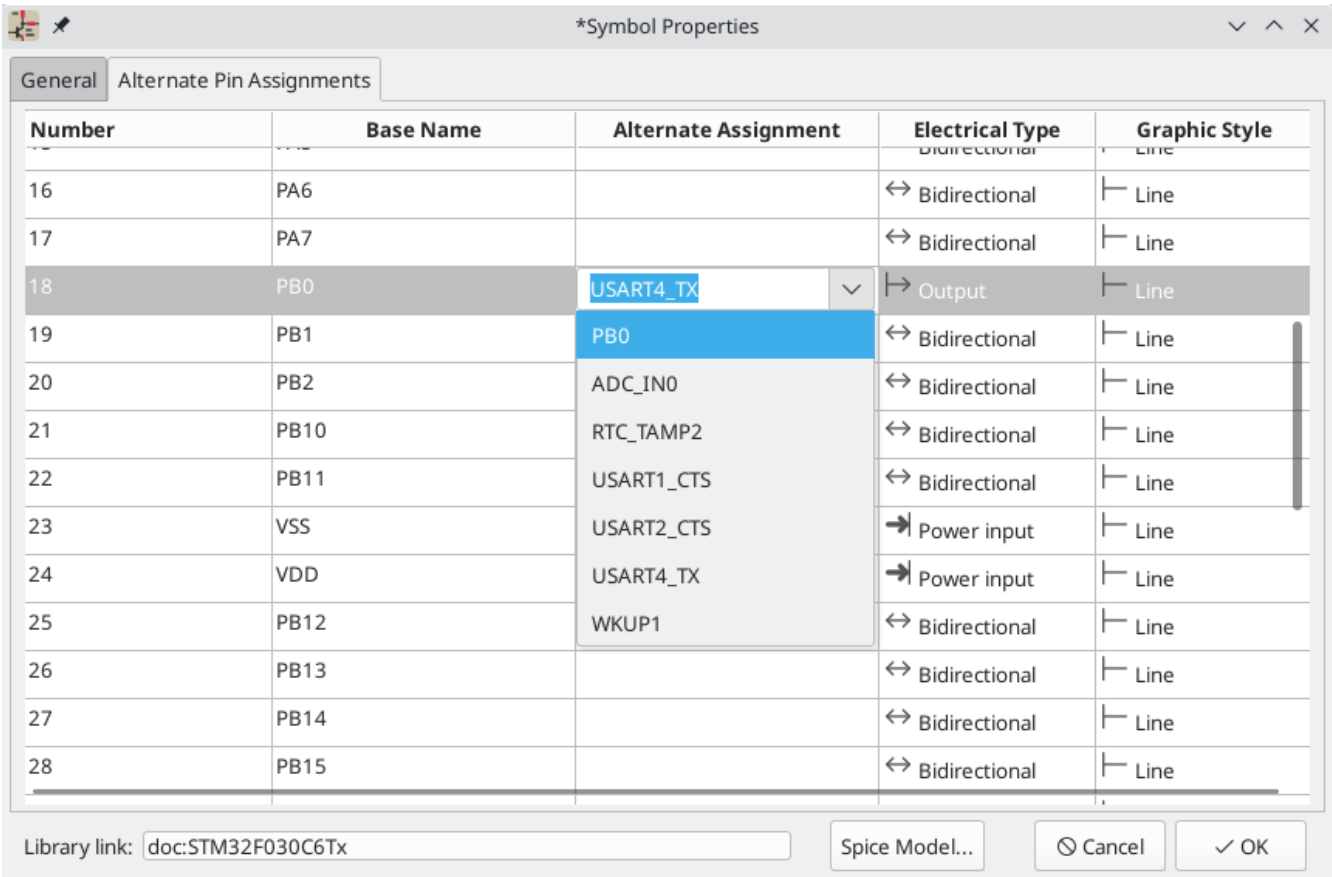
Preview:


The preview shows a pin symbol with the number '18' in red and the text 'PB0' in teal. The number '18' is underlined with a red line, and a red circle is drawn around the pin symbol.

▼ Alternate pin definitions

Alternate Pin Name	Electrical Type	Graphic Style
USART1_CTS	→ Input	└─ Line
USART2_CTS	→ Input	└─ Line
USART4_TX	└─ Output	└─ Line
ADC_IN0	→ Input	└─ Line
RTC_TAMP2	→ Input	└─ Line
WKUP1	→ Input	└─ Line

Alternate pin definitions are selected in the Schematic Editor once the symbol has been placed in the schematic. The alternate pin is assigned in the Alternate Pin Assignments tab of the Symbol Properties dialog. Alternate definitions are selectable in the dropdown in the Alternate Assignment column.



## 符号字段

All library symbols are defined with four default fields. The reference designator, value, footprint assignment, and datasheet link fields are created whenever a symbol is created or copied. Only the reference designator and value fields are required.

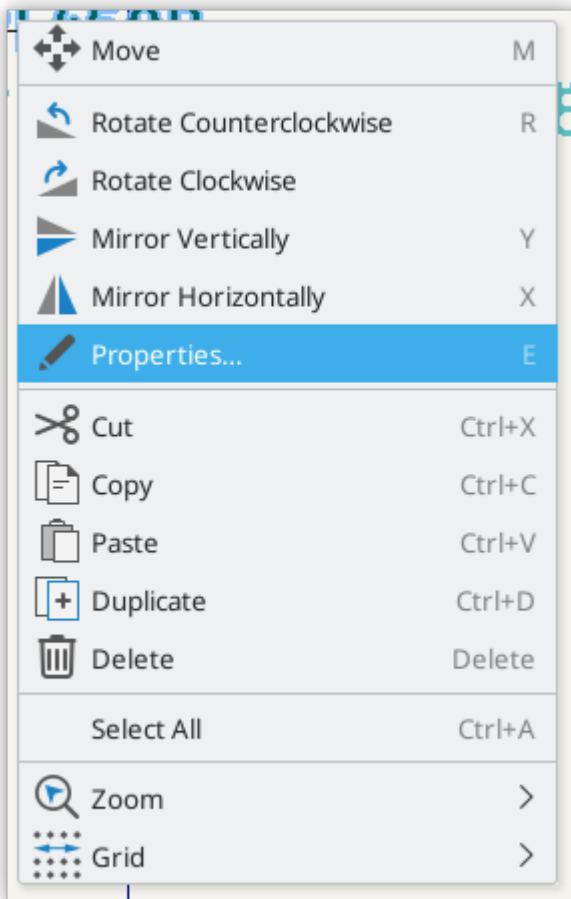
Symbols defined in libraries are typically defined with only these four default fields. Additional fields such as vendor, part number, unit cost, etc. can be added to library symbols but generally this is done in the schematic editor so the additional fields can be applied to all of the symbols in the schematic.


NOTE

A convenient way to create additional empty symbol fields is to use define field name templates. Field name templates define empty fields that are added to each symbol when it is inserted into the schematic. Field name templates can be defined globally (for all schematics) in the Schematic Editor Preferences, or they can be defined locally (specific to each project) in the Schematic Setup dialog.

## 符号字段

要添加符号字段，将符号字段文本以`{field_name}`的形式放在下面所示的字段上下文菜单中



To add new fields, delete optional fields, or edit existing fields, use the  icon on the main tool bar to open the [Symbol Properties dialog](#).

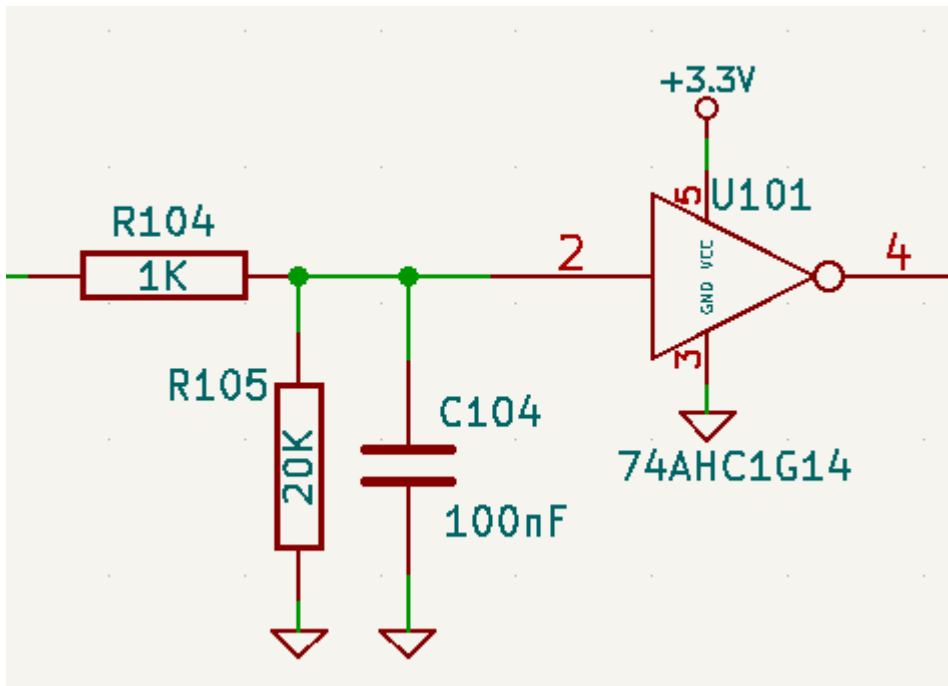
Fields are text information associated a the symbol. Do not confuse them with text in the graphic representation of a symbol.

#### 重要笔记

- Modifying the `Value` field changes the name of the symbol. The symbol's name in the library will change when the symbol is saved.
- The Symbol Properties dialog must be used to edit a field that is empty or has the invisible attribute enabled because such fields cannot be clicked on.
- The footprint is defined as an absolute footprint using the `LIBNAME:FOOTPRINTNAME` format where `LIBNAME` is the name of the footprint library defined in the footprint library table (see the "Footprint Library Table" section in the PCB Editor manual) and `FOOTPRINTNAME` is the name of the footprint in the library `LIBNAME`.

## Power Ports

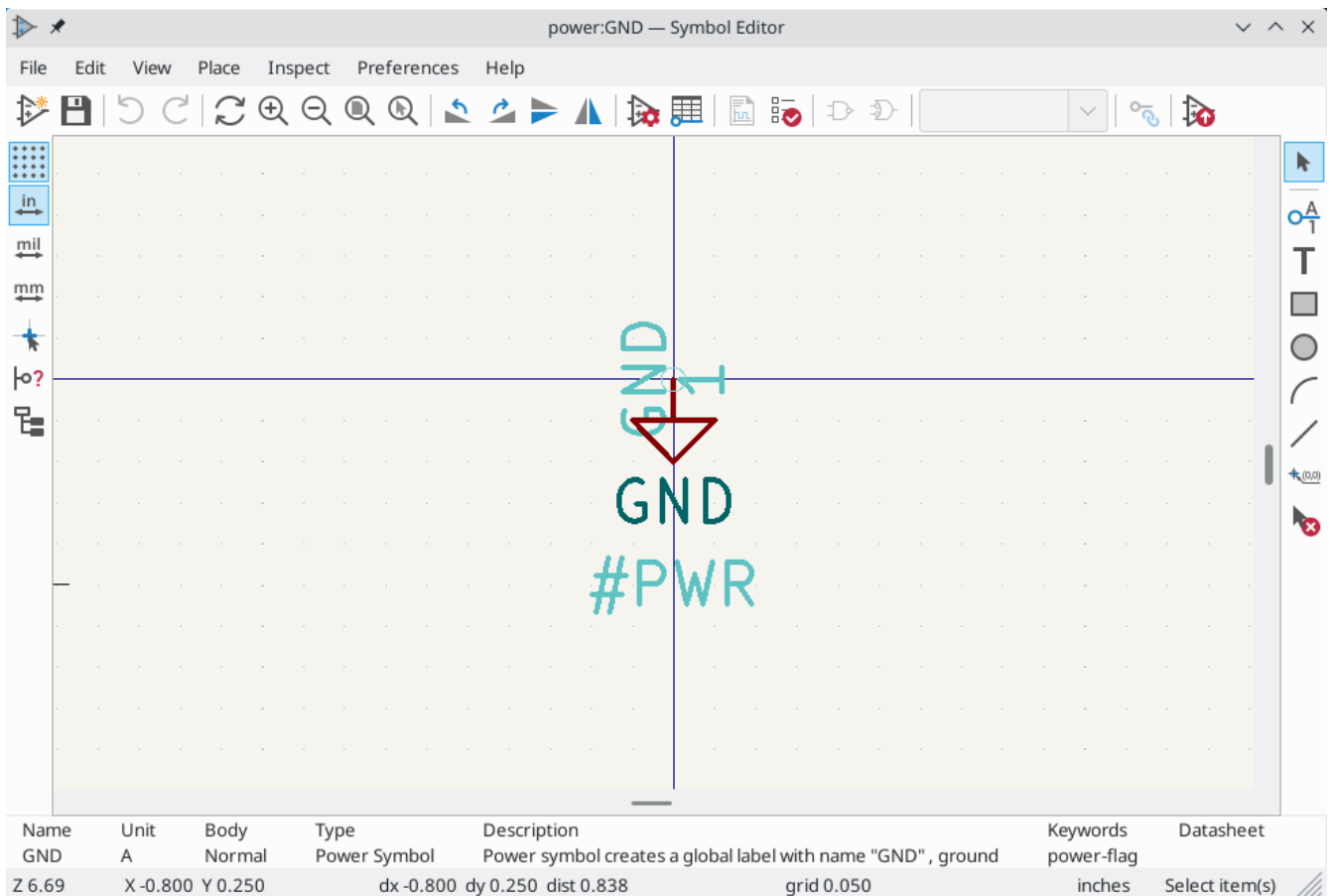
Power ports, or power symbols, are conventionally used to label a wire as part of a power net, like `VCC`, `+5V`, or `GND`. In the schematic below, the `+3.3V` and `GND` symbols are power ports. In addition to acting as a visual indicator that a net is a power rail, a power port will determine the name of the net it is attached to. This is true even if there is another net label attached to the net; the net name determined by the power symbol overrides any other net names.



It may be useful to place power symbols in a dedicated library. KiCad's symbol library places power symbols in the `power` library, and users may create libraries to store their own power symbols. If the "Define as power symbol" box is checked in a symbol's properties, that symbol will appear in the Schematic Editor's "Add Power Port" dialog for convenient access.

Power symbols are handled and created the same way as normal symbols, but there are several additional considerations described below. They consist of a graphical symbol and a pin of the type "Power input" that is marked hidden.

Below is an example of a `GND` power symbol.

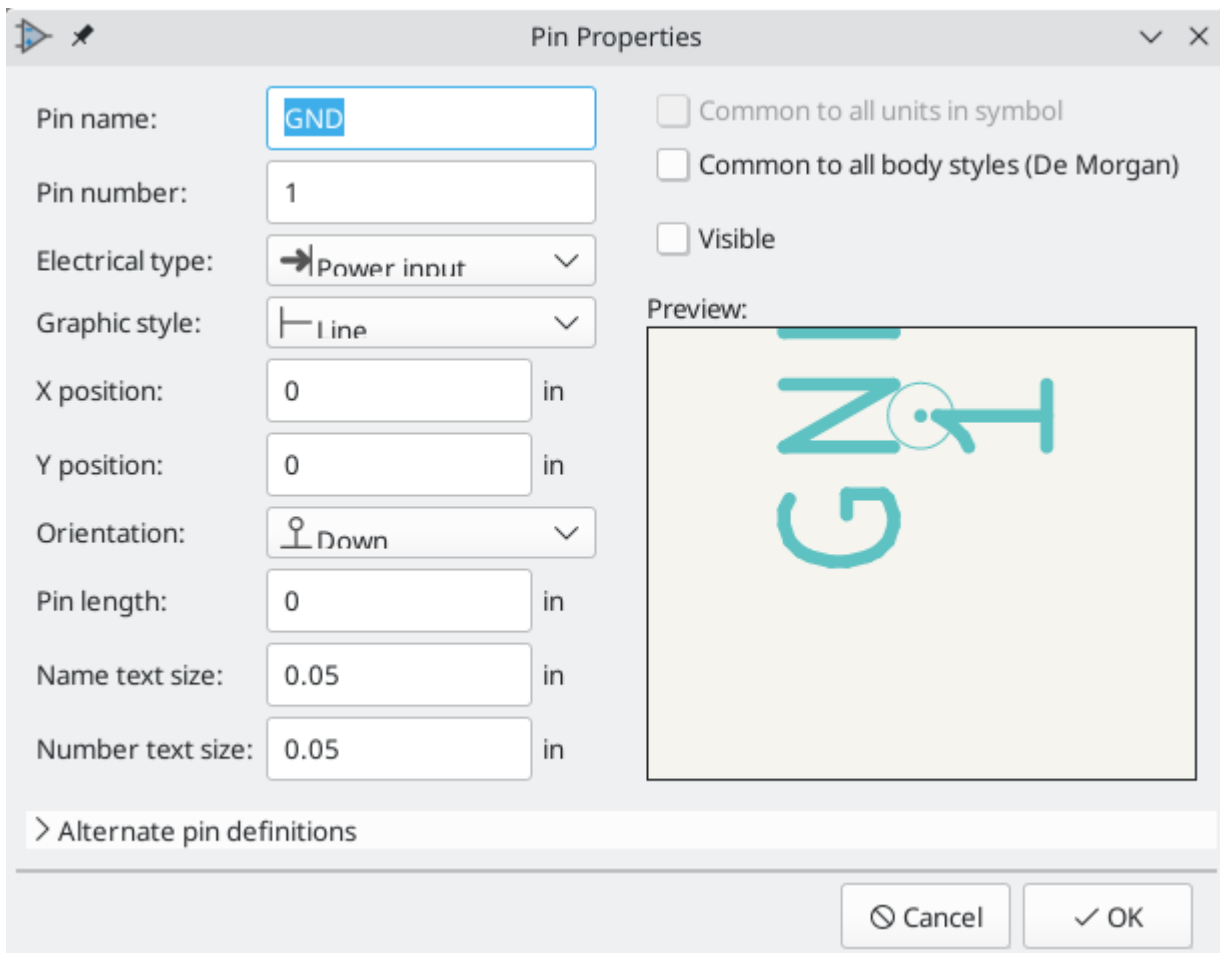


## Creating a Power Port Symbol

Power Port symbols consist of a pin of type "Power input" that is marked invisible. Invisible power input pins have a special property of automatically connecting to a net with the same name as the pin name. A net that is wired to an invisible power input pin will therefore be named after the pin, even if there are other net labels on the net. This connection is global.

### NOTE

If the power symbol has the "Define as power symbol" property checked, the power input pin does not need to be marked invisible. However, the convention is to make these pins invisible anyway.



要创建源符号，请使用以下步骤：

- Add a pin of type "Power input", with "Visible" unchecked, and the pin named according to the desired net. Make the pin number 1, the length 0, and set the graphic style to "Line". The pin name establishes the connection to the net; in this case the pin will automatically connect to the net GND. The pin number, length, and line style do not matter electrically.
- Place the pin on the symbol anchor.
- Use the shape tools to draw the symbol graphics.
- Set the symbol value. The symbol value does not matter electrically, but it is displayed in the schematic. To eliminate confusion, it should match the pin name (which determines the connected net name).
- Check the "Define as power symbol" box in Symbol Properties window. This makes the symbol appear in the "Add Power Port" dialog, makes the Value field read-only in the schematic, prevents the symbol from being assigned a footprint, and excludes the symbol from the board, BOM, and netlists.
- Set the symbol reference and uncheck the "Show" box. The reference text is not important except for the first character, which should be #. For the power port shown above, the reference could be #GND. Symbols with references that begin with # are not added to the PCB, are not included in Bill of Materials exports or netlists, and they cannot be assigned a footprint in the footprint assignment tool. If a power port's reference does not begin with #, the character will be inserted automatically when the annotation or footprint assignment tools are run.

An easier method to create a new power port symbol is to use another symbol as a starting point, [as described earlier](#).




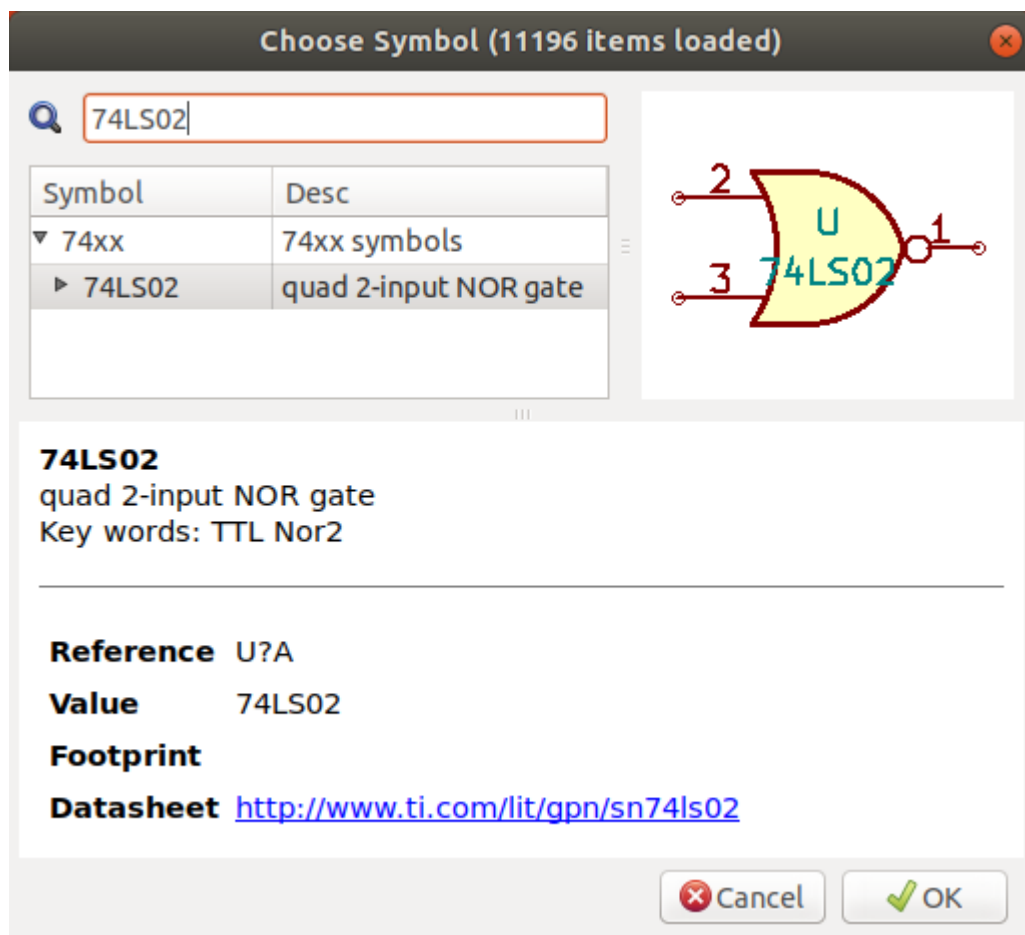
**NOTE**

When modifying an existing power port symbol, make sure to rename the pin name so that the new symbol connects to the appropriate power net.

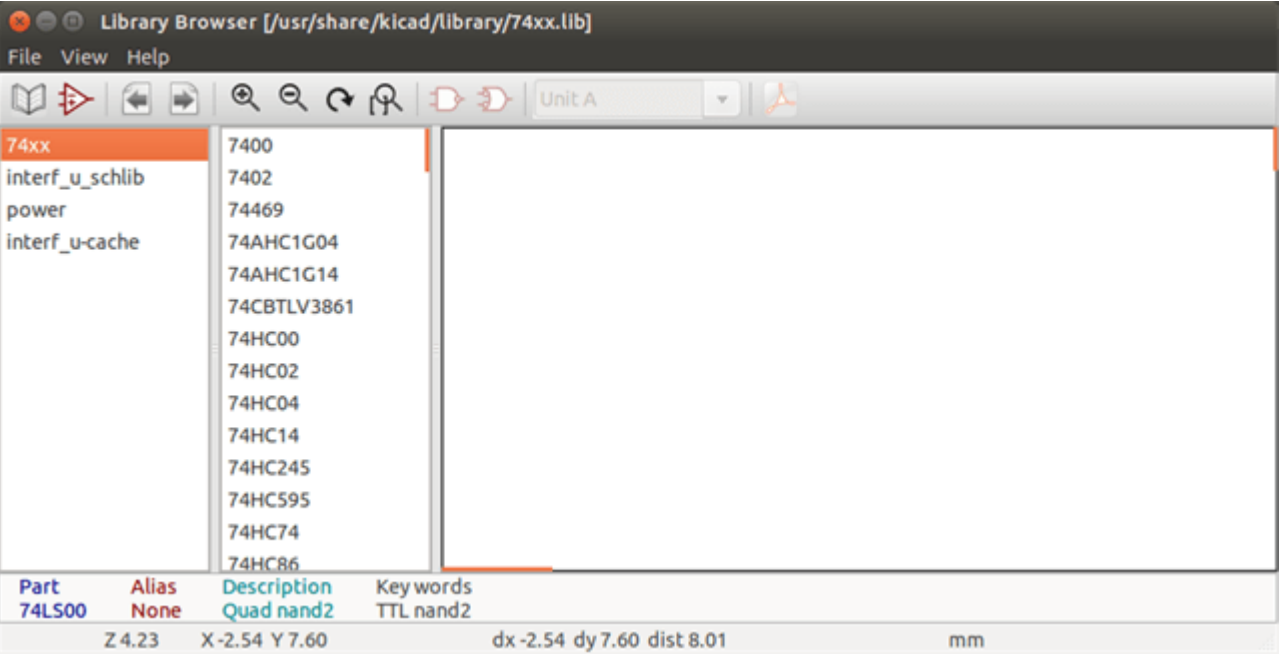
# 符号库器

## 简介

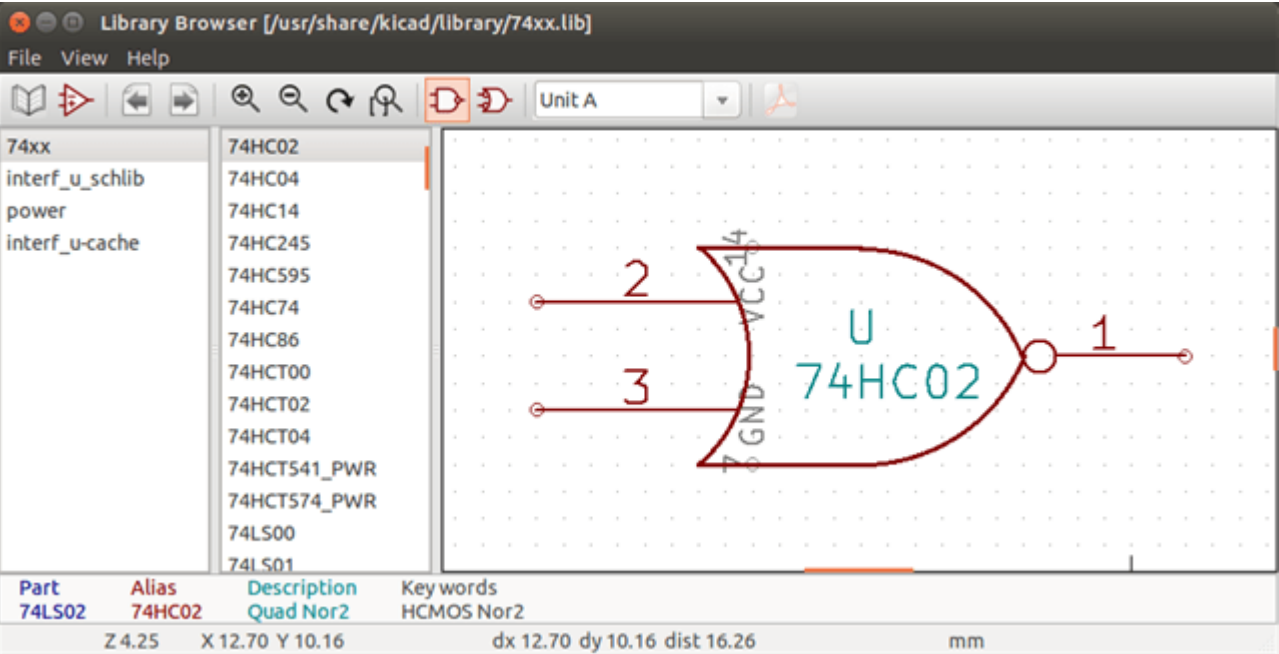
The Symbol Library Browser allows you to quickly examine the content of symbol libraries. The Symbol Library Viewer can be accessed by clicking  icon on the main toolbar, **View** → **Symbol Library Browser**..., or clicking **Select With Browser** in the "Choose Symbol" window.



主屏幕



要查看内容，从左窗格的列表中选择一个。所选中的所有符号都将显示在第二个窗格中。符号名称以查看符号。





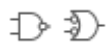
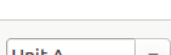




符号器部工具

符号器中的部工具如下所示。



可用的命令是：

	Selection of the symbol which can be also selected in the displayed list.
	Display previous symbol.
	Display next symbol.
	Zoom tools.
	Selection of the representation (normal or alternate) if an alternate representation exists.
	Selection of the unit for symbols that contain multiple units.
	If they exist, display the associated documents.
	Close the browser and place the selected symbol in the schematic.

# 网表列表

## 概述

A netlist is a file which describes electrical connections between symbol pins. These connections are referred to as nets. Netlist files contain:

- A list of symbols and their pins.
- A list of connections (nets) between symbol pins.

Many different netlist formats exist. Sometimes the symbols list and the list of nets are two separate files. This netlist is fundamental in the use of schematic capture software, because the netlist is the link with other electronic CAD software, such as:

- PCB 布局软件。
- 原理图和信号模拟器。
- Programmable logic (FPGA, CPLD, etc.) compilers.

KiCad supports several netlist formats:

- KiCad format, which can be imported by the KiCad PCB Editor. However, the ["Update PCB from Schematic"](#) tool should be used instead of importing a KiCad netlist into the PCB editor.
- OrCAD PCB2 format, for designing PCBs with OrCAD.
- CADSTAR format, for designing PCBs with CADSTAR.
- Spice format, for use with various external circuit simulators.

### NOTE

In KiCad version 5.0 and later, it is not necessary to create a netlist for transferring a design from the schematic editor to the PCB editor. Instead, use the ["Update PCB from Schematic"](#) tool.

## 网表格式

Netlists are exported with the Export Netlist dialog (**File** → **Export** → **Netlist...**).

Several netlist formats are available, and are selectable with the tabs at the top of the window. Some netlist formats have options.

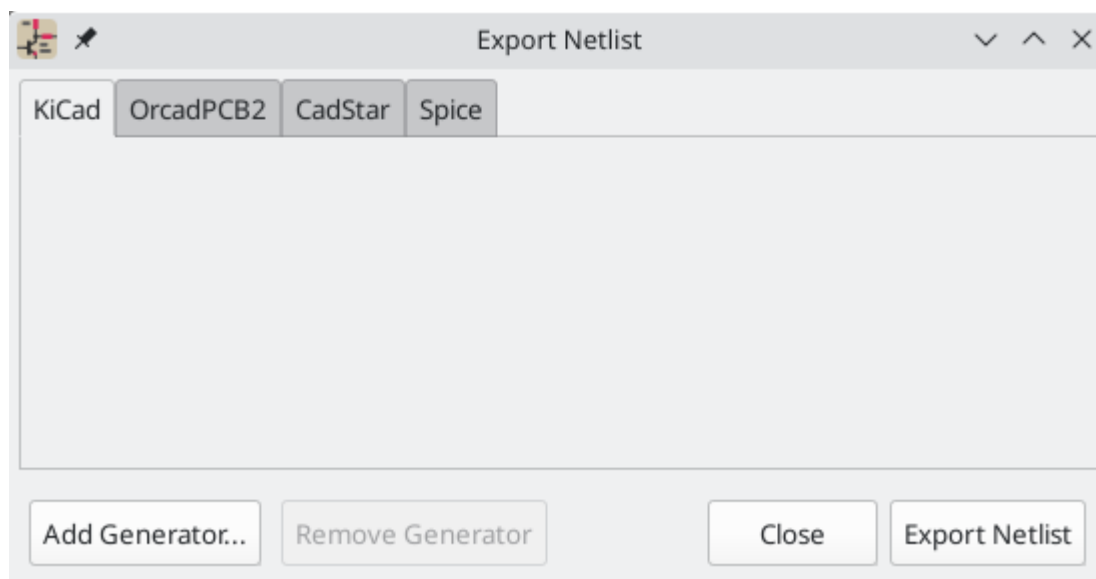
Clicking the **Export Netlist** button prompts for a netlist filename and saves the netlist.

### NOTE

Netlist generation can take up to several minutes for large schematics.

Custom generators can be added by clicking the **Add Generator...** button. Custom generators are external tools that are called by KiCad, for example Python scripts or XSLT stylesheets. For more information on custom netlist generators, see [the section on adding custom netlist generators](#).

## KiCad Netlist Format

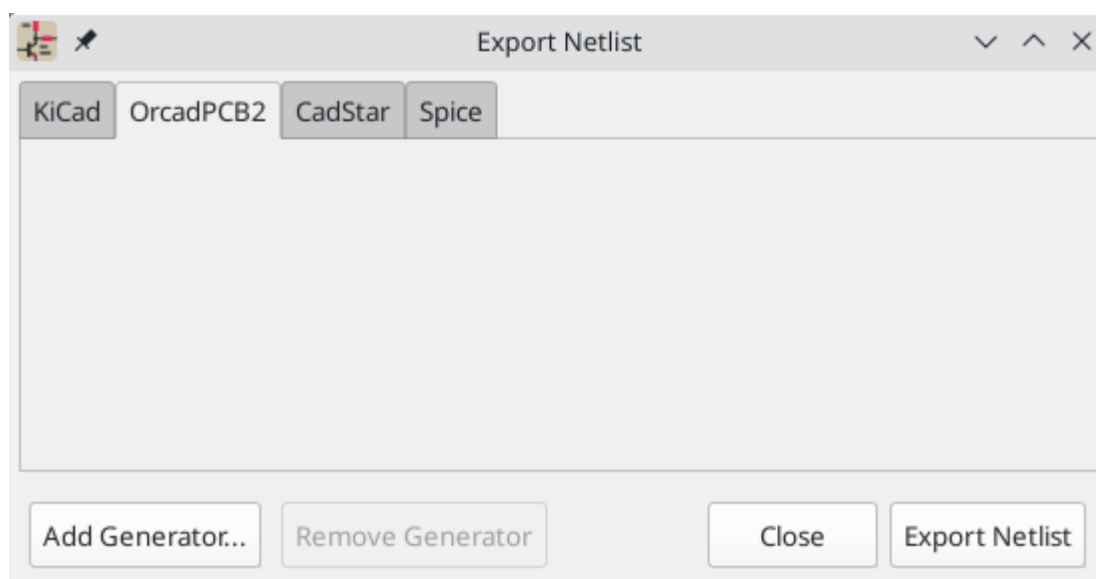


The KiCad netlist exporter does not have any options.

### NOTE

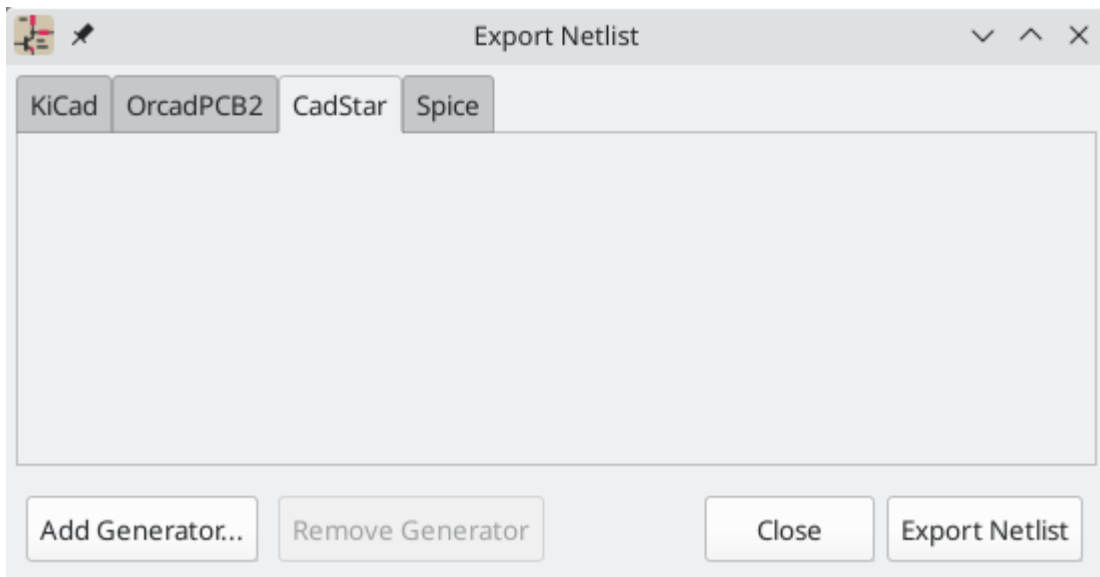
In KiCad version 5.0 and later, it is not necessary to create a netlist for transferring a design from the schematic editor to the PCB editor. Instead, use the ["Update PCB from Schematic"](#) tool.

## OrCAD PCB2 Netlist Format



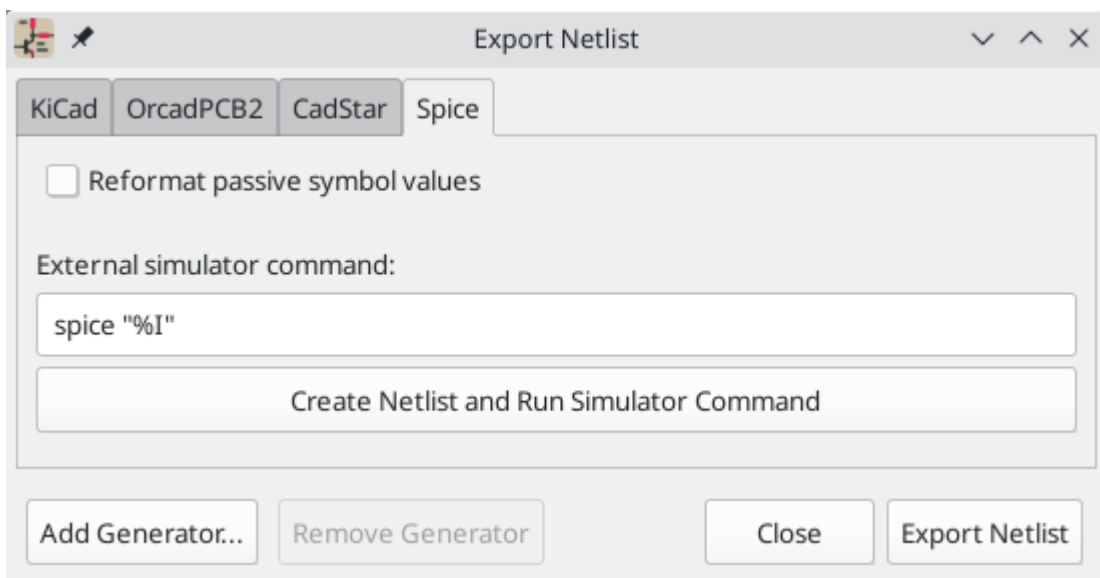
The OrCAD netlist exporter does not have any options.

## CADSTAR Netlist Format



The CADSTAR netlist exporter does not have any options.

## Spice Netlist Format



The Spice netlist format offers several options.

When the **Reformat passive symbol values** box is checked, passive symbol values will be adjusted to be compatible with Spice. Specifically:

- $\mu$  and M as unit prefixes are replaced with u and Meg, respectively
- Units are removed (e.g. 4.7k $\Omega$  is changed to 4.7k)
- Values in RKM format are rewritten to be Spice-compatible (e.g. 4u7 is changed to 4.7u)

The Spice netlist exporter also provides an easy way to simulate the generated netlist with an external simulator. This can be useful for running a simulation without using [KiCad's internal ngspice simulator](#), or for running an ngspice simulation with options that are not supported by KiCad's simulator tool.

Enter the path to the external simulator in the text box, with `%I` representing the generated netlist. Click the **Create Netlist and Run Simulator Command** button to generate the netlist and automatically run the simulator.

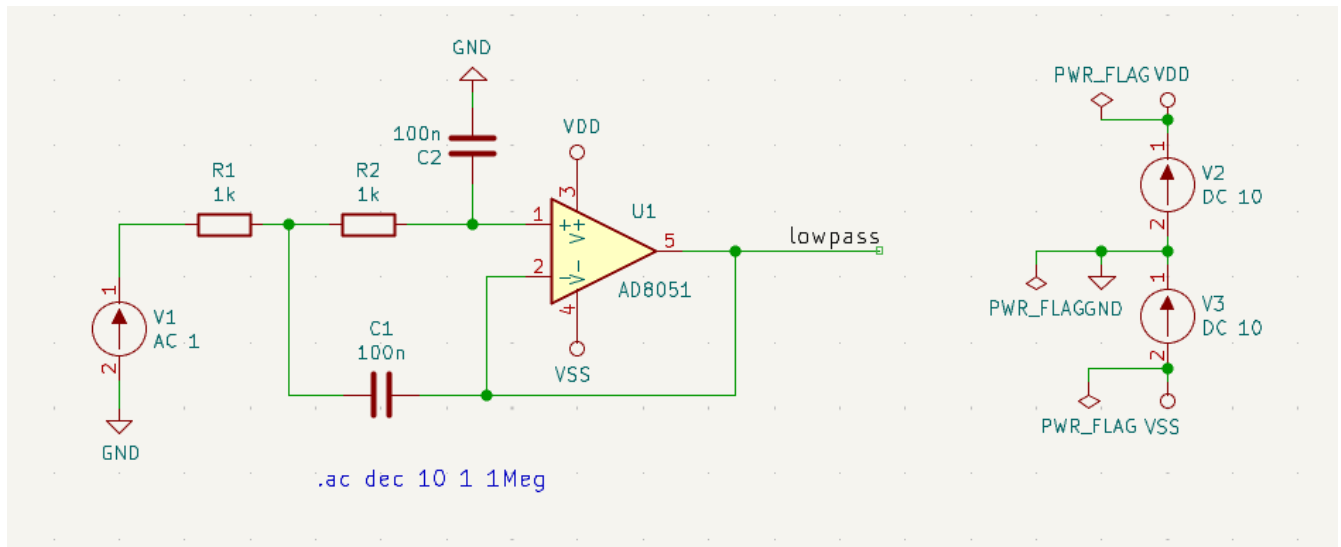
#### NOTE

The default simulator command ( `spice "%I"` ) must be adjusted to point to a simulator installed on your system.

For more information on the contents of Spice netlists, see the [Spice netlist section](#).

## 网表示例

Below is the schematic from the `sallen_key` project included in KiCad's simulation demos.



The KiCad format netlist for this schematic is as follows:



```

(export (version "E")
  (design
    (source "/usr/share/kicad/demos/simulation/sallen_key/sallen_key.kicad_sch")
    (date "Sun 01 May 2022 03:14:05 PM EDT")
    (tool "Eeschema (6.0.4)")
    (sheet (number "1") (name "/") (tstamps "/")
      (title_block
        (title)
        (company)
        (rev)
        (date)
        (source "sallen_key.kicad_sch")
        (comment (number "1") (value ""))
        (comment (number "2") (value ""))
        (comment (number "3") (value ""))
        (comment (number "4") (value ""))
        (comment (number "5") (value ""))
        (comment (number "6") (value ""))
        (comment (number "7") (value ""))
        (comment (number "8") (value ""))
        (comment (number "9") (value ""))))))
    (components
      (comp (ref "C1")
        (value "100n")
        (libsource (lib "sallen_key_schlib") (part "C") (description ""))
        (property (name "Sheetname") (value ""))
        (property (name "Sheetfile") (value "sallen_key.kicad_sch"))
        (sheetpath (names "/") (tstamps "/"))
        (tstamps "00000000-0000-0000-0000-00005789077d"))
      (comp (ref "C2")
        (value "100n")
        (fields
          (field (name "Fieldname") "Value")
          (field (name "SpiceMapping") "1 2")
          (field (name "Spice_Primitive") "C"))
        (libsource (lib "sallen_key_schlib") (part "C") (description ""))
        (property (name "Fieldname") (value "Value"))
        (property (name "Spice_Primitive") (value "C"))
        (property (name "SpiceMapping") (value "1 2"))
        (property (name "Sheetname") (value ""))
        (property (name "Sheetfile") (value "sallen_key.kicad_sch"))
        (sheetpath (names "/") (tstamps "/"))
        (tstamps "00000000-0000-0000-0000-00005789085b"))
      (comp (ref "R1")
        (value "1k")
        (fields
          (field (name "Fieldname") "Value")
          (field (name "SpiceMapping") "1 2")
          (field (name "Spice_Primitive") "R"))
        (libsource (lib "sallen_key_schlib") (part "R") (description ""))
        (property (name "Fieldname") (value "Value"))
        (property (name "SpiceMapping") (value "1 2"))
        (property (name "Spice_Primitive") (value "R"))
        (property (name "Sheetname") (value ""))
        (property (name "Sheetfile") (value "sallen_key.kicad_sch"))
        (sheetpath (names "/") (tstamps "/"))
        (tstamps "00000000-0000-0000-0000-0000578906ff"))
      (comp (ref "R2")
        (value "1k")
        (fields
          (field (name "Fieldname") "Value")
          (field (name "SpiceMapping") "1 2"))

```

In Spice format, the netlist is as follows:

```
.title KiCad schematic
.include "ad8051.lib"
XU1 Net-_C2-Pad1_ /lowpass VDD VSS /lowpass AD8051
C2 Net-_C2-Pad1_ GND 100n
C1 /lowpass Net-_C1-Pad2_ 100n
R2 Net-_C2-Pad1_ Net-_C1-Pad2_ 1k
R1 Net-_C1-Pad2_ Net-_R1-Pad2_ 1k
V1 Net-_R1-Pad2_ GND AC 1
V2 VDD GND DC 10
V3 GND VSS DC 10
.ac dec 10 1 1Meg
.end
```

## 关于网表的说明

### 网表名称注意事项

Many software tools that use netlists do not accept spaces in component names, pins, nets, or other fields. Avoid using spaces in pins, labels, names, and value fields of components to ensure maximum compatibility.

In the same way, special characters other than letters and numbers can cause problems. Note that this limitation is not related to KiCad, but to the netlist formats that can then become untranslatable by other software that reads those netlist files.

## Spice netlists

Spice simulators expect simulation commands ( `.PROBE` , `.AC` , `.TRAN` , etc.) to be included in the netlist.

Any text line included in the schematic diagram starting with a period ( `.` ) will be included in the netlist. If a text object contains multiple lines, only the lines beginning with a period will be included.

`.include` directives for including model library files are automatically added to the netlist based on the Spice model settings for the symbols in the schematic.

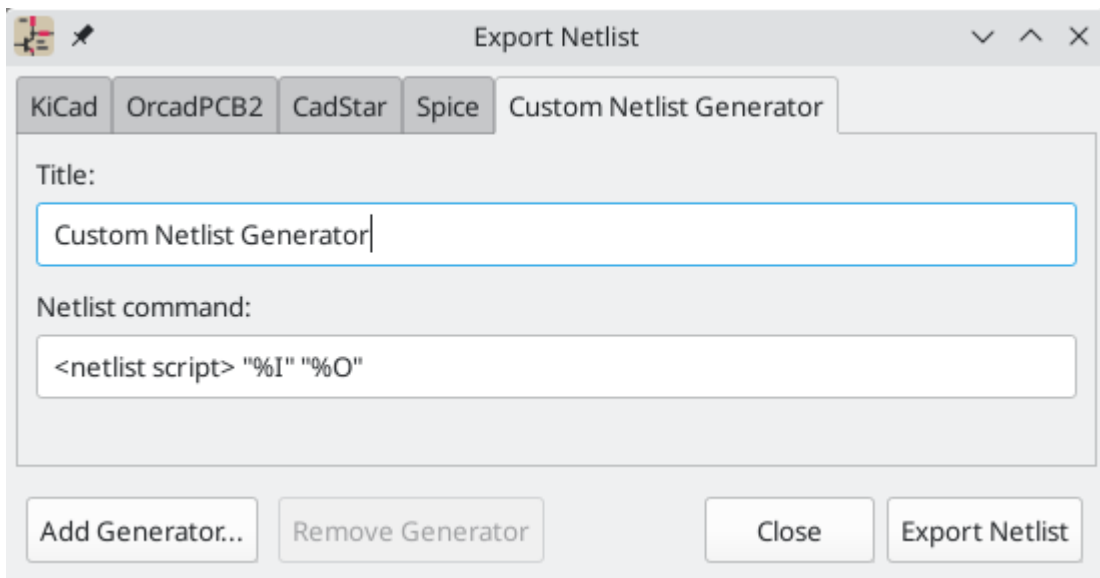
## 其他格式

KiCad supports custom netlist generators for exporting netlists in other formats. Some examples of netlist generators are given in the [custom netlist generators section](#).

A netlist generator is a script or program that converts the intermediate netlist file created by KiCad into the desired netlist format. The intermediate netlist file contains all of the netlist information required to create an arbitrary netlist for the schematic. Python and XSLT are commonly used tools to create custom netlist generators.

## Adding new netlist generators

New netlist generators are added by clicking the **Add Generator...** button.



New generators require a name and a command. The name is shown in the tab label, and the command is run whenever the **Export Netlist** button is clicked.

When the netlist is generated, KiCad creates an intermediate XML file which contains all of the netlist information from the schematic. The generator command is then run in order to transform the intermediate netlist into the desired netlist format.

The netlist command must be set up properly so that the netlist generator script takes the intermediate netlist file as input and outputs the desired netlist file. The `%I` argument represents the input intermediate netlist filename and the `%O` argument represents the output netlist filename. The exact netlist command will depend on the generator script used.

## 命令行格式

Consider the following example which uses `xsltproc` to generate a netlist in PADS ASC format. `xsltproc` converts the intermediate netlist using the `netlist_form_pads-pcb.asc.xsl` stylesheet to define the output format:

```
xsltproc -o %O.net /usr/share/kicad/plugins/netlist_form_pads-pcb.asc.xsl %I
```

The purpose of each part of the command is as follows:

<code>xsltproc</code>	A tool to convert an XML file (the intermediate netlist) according to an XSLT stylesheet.
<code>-o %O.net</code>	Output filename. %O is replaced with the name of the intermediate netlist file, which is <code>&lt;schematic name&gt;.xml</code> . Therefore in this example the complete output filename is <code>&lt;schematic name&gt;.xml.net</code> . An arbitrary output filename can be specified if desired with <code>-o &lt;filename&gt;</code> .
<code>/usr/share/kicad/plugins/netlist_form_pads-pcb.asc.xsl</code>	XSLT stylesheet which determines how the output is formatted. This particular stylesheet is included with KiCad, but custom stylesheets can also be created.
<code>%I</code>	Input (intermediate netlist) filename. %I is replaced with the name of the intermediate netlist file, which is <code>&lt;schematic name&gt;.xml</code> .

For netlist generators that do not use `xsltproc`, the generator command will differ.

## 中□网表文件格式

See the [custom netlist generators section](#) for more information about netlist generators, a description of the intermediate netlist format, and some examples of netlist generators.

# 创建自定义网表和 BOM 文件

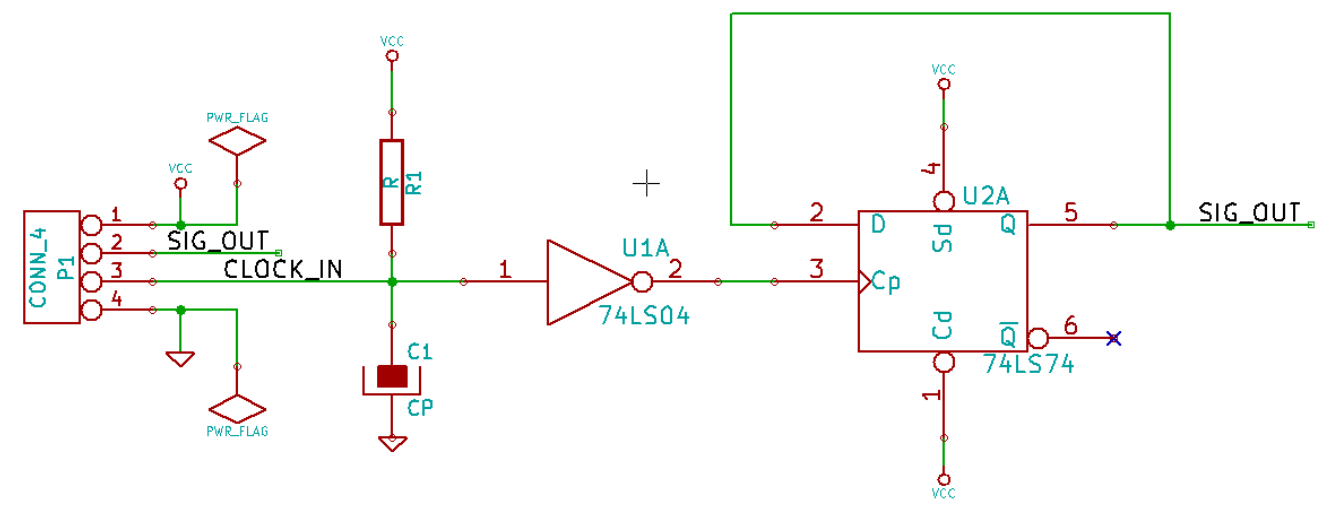
## 中间网表文件格式

BOM files and netlist files can be converted from an Intermediate netlist file created by KiCad.

此文件使用 XML 语法，称为中间网表。中间网表包含有关您的电路板的大量数据，因此，它可以与后处理一起用于创建 BOM 或其他报告。

根据输出（BOM 或网表），将在后处理中使用完整的中间网表文件的不同子集。

## 原理图示例



## 中间网表文件示例

上述电路的中间网表 (使用 XML 语法) 如下所示。

```

<?xml version="1.0" encoding="utf-8"?>
<export version="D">
  <design>
    <source>F:\kicad_aux\netlist_test\netlist_test.sch</source>
    <date>29/08/2010 20:35:21</date>
    <tool>eeschema (2010-08-28 BZR 2458)-unstable</tool>
  </design>
  <components>
    <comp ref="P1">
      <value>CONN_4</value>
      <libsource lib="conn" part="CONN_4"/>
      <sheetpath names="/" tstamps="/" />
      <tstamp>4C6E2141</tstamp>
    </comp>
    <comp ref="U2">
      <value>74LS74</value>
      <libsource lib="74xx" part="74LS74"/>
      <sheetpath names="/" tstamps="/" />
      <tstamp>4C6E20BA</tstamp>
    </comp>
    <comp ref="U1">
      <value>74LS04</value>
      <libsource lib="74xx" part="74LS04"/>
      <sheetpath names="/" tstamps="/" />
      <tstamp>4C6E20A6</tstamp>
    </comp>
    <comp ref="C1">
      <value>CP</value>
      <libsource lib="device" part="CP"/>
      <sheetpath names="/" tstamps="/" />
      <tstamp>4C6E2094</tstamp>
    </comp>
    <comp ref="R1">
      <value>R</value>
      <libsource lib="device" part="R"/>
      <sheetpath names="/" tstamps="/" />
      <tstamp>4C6E208A</tstamp>
    </comp>
  </components>
  <libparts>
    <libpart lib="device" part="C">
      <description>Condensateur non polarise</description>
      <footprints>
        <fp>SM*</fp>
        <fp>C?</fp>
        <fp>C1-1</fp>
      </footprints>
      <fields>
        <field name="Reference">C</field>
        <field name="Value">C</field>
      </fields>
      <pins>
        <pin num="1" name="~" type="passive"/>
        <pin num="2" name="~" type="passive"/>
      </pins>
    </libpart>
    <libpart lib="device" part="R">
      <description>Resistance</description>
      <footprints>
        <fp>R?</fp>
        <fp>SM0603</fp>
        <fp>SM0805</fp>
      </footprints>
    </libpart>
  </libparts>
</export>

```

## 新的网表格式

通过后将处理器用于中间网表文件，您可以生成外部网表文件以及 BOM 文件。由于此是文本到文本因此可以使用 Python, XSLT 或任何其他能将 XML 作为输入的工具来写此处理器。

XSLT itself is an XML language very suitable for XML transformations. There is a free program called *xsltproc* that you can download and install. The *xsltproc* program can be used to read the Intermediate XML netlist input file, apply a style-sheet to transform the input, and save the results in an output file. Use of *xsltproc* requires a style-sheet file using XSLT conventions. The full conversion process is handled by KiCad, after it is configured once to run *xsltproc* in a specific way.

## XSLT 方法

描述 XSL (XSLT) 的文档可在此得：

<http://www.w3.org/TR/xslt>

## 创建 Pads-Pcb 网表文件

“pads-pcb” 的格式由两部分成。

- 封装列表。
- 网表: 按网表引用行分

下面是式表，它将中间网表文件 pad-pcb 网表格式：

```

<?xml version="1.0" encoding="ISO-8859-1"?>
<!--XSL style sheet to Eeschema Generic Netlist Format to PADS netlist format
Copyright (C) 2010, SoftPLC Corporation.
GPL v2.

如何使用:
https://lists.launchpad.net/kicad-developers/msg05157.html
-->

<!DOCTYPE xsl:stylesheet [
  <!ENTITY nl "&#xd;&#xa;"> <!--new line CR, LF -->
]>

<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:output method="text" omit-xml-declaration="yes" indent="no"/>

<xsl:template match="/export">
  <xsl:text>*PADS-PCB*&nl;*PART*&nl;</xsl:text>
  <xsl:apply-templates select="components/comp"/>
  <xsl:text>&nl;*NET*&nl;</xsl:text>
  <xsl:apply-templates select="nets/net"/>
  <xsl:text>*END*&nl;</xsl:text>
</xsl:template>

<!-- for each component -->
<xsl:template match="comp">
  <xsl:text> </xsl:text>
  <xsl:value-of select="@ref"/>
  <xsl:text> </xsl:text>
  <xsl:choose>
    <xsl:when test = "footprint != '' ">
      <xsl:apply-templates select="footprint"/>
    </xsl:when>
    <xsl:otherwise>
      <xsl:text>unknown</xsl:text>
    </xsl:otherwise>
  </xsl:choose>
  <xsl:text>&nl;</xsl:text>
</xsl:template>

<!-- for each net -->
<xsl:template match="net">
  <!-- nets are output only if there is more than one pin in net -->
  <xsl:if test="count(node)>1">
    <xsl:text>*SIGNAL* </xsl:text>
    <xsl:choose>
      <xsl:when test = "@name != '' ">
        <xsl:value-of select="@name"/>
      </xsl:when>
      <xsl:otherwise>
        <xsl:text>N-</xsl:text>
        <xsl:value-of select="@code"/>
      </xsl:otherwise>
    </xsl:choose>
    <xsl:text>&nl;</xsl:text>
    <xsl:apply-templates select="node"/>
  </xsl:if>
</xsl:template>

<!-- for each node -->
<xsl:template match="node">
  <xsl:text> </xsl:text>

```



□是运行 xsltproc 后的 pads-pcb □出文件：

```
*PADS-PCB*
*PART*
P1 unknown
U2 unknown
U1 unknown
C1 unknown
R1 unknown
*NET*
*SIGNAL* GND
U1.7
C1.2
U2.7
P1.4
*SIGNAL* VCC
R1.1
U1.14
U2.4
U2.1
U2.14
P1.1
*SIGNAL* N-4
U1.2
U2.3
*SIGNAL* /SIG_OUT
P1.2
U2.5
U2.2
*SIGNAL* /CLOCK_IN
R1.2
C1.1
U1.1
P1.3

*END*
```

□行此□□的命令是：

```
kicad\\bin\\xsltproc.exe -o test.net kicad\\bin\\plugins\\netlist_form_pads-pcb.xml
test.tmp
```

## □建一个 Cadstar 网表文件

Cadstar 格式由两个部分□成。

- 封装列表。
- 网表: 按网□□□□引用□行分□□

以下是□行此特定□□的□式表文件：

```

<?xml version="1.0" encoding="ISO-8859-1"?>
<!--XSL style sheet to Eeschema Generic Netlist Format to CADSTAR netlist format
      Copyright (C) 2010, Jean-Pierre Charras.
      Copyright (C) 2010, SoftPLC Corporation.
      GPL v2.

<!DOCTYPE xsl:stylesheet [
      <!ENTITY nl "&#xd;&#xa;"> <!--new line CR, LF -->
]>

<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:output method="text" omit-xml-declaration="yes" indent="no"/>

<!-- Netlist header -->
<xsl:template match="/export">
      <xsl:text>.HEA&nl;</xsl:text>
      <xsl:apply-templates select="design/date"/> <!-- Generate line .TIM <time> -->
      <xsl:apply-templates select="design/tool"/> <!-- Generate line .APP <eeschema version>
-->
      <xsl:apply-templates select="components/comp"/> <!-- Generate list of components -->
      <xsl:text>&nl;&nl;</xsl:text>
      <xsl:apply-templates select="nets/net"/> <!-- Generate list of nets and
connections -->
      <xsl:text>&nl;.END&nl;</xsl:text>
</xsl:template>

      <!-- Generate line .TIM 20/08/2010 10:45:33 -->
<xsl:template match="tool">
      <xsl:text>.APP "</xsl:text>
      <xsl:apply-templates/>
      <xsl:text>"&nl;</xsl:text>
</xsl:template>

      <!-- Generate line .APP "eeschema (2010-08-17 BZR 2450)-unstable" -->
<xsl:template match="date">
      <xsl:text>.TIM </xsl:text>
      <xsl:apply-templates/>
      <xsl:text>&nl;</xsl:text>
</xsl:template>

<!-- for each component -->
<xsl:template match="comp">
      <xsl:text>.ADD_COM </xsl:text>
      <xsl:value-of select="@ref"/>
      <xsl:text> </xsl:text>
      <xsl:choose>
        <xsl:when test = "value != '' ">
          <xsl:text>"</xsl:text> <xsl:apply-templates select="value"/> <xsl:text>"
</xsl:text>
        </xsl:when>
        <xsl:otherwise>
          <xsl:text>"</xsl:text>
        </xsl:otherwise>
      </xsl:choose>
      <xsl:text>&nl;</xsl:text>
</xsl:template>

<!-- for each net -->
<xsl:template match="net">
      <!-- nets are output only if there is more than one pin in net -->
      <xsl:if test="count(node)>1">
        <xsl:variable name="netname">

```

是 Cadstar 出文件。

```
.HEA
.TIM 21/08/2010 08:12:08
.APP "eeschema (2010-08-09 BZR 2439)-unstable"
.ADD_COM P1 "CONN_4"
.ADD_COM U2 "74LS74"
.ADD_COM U1 "74LS04"
.ADD_COM C1 "CP"
.ADD_COM R1 "R"

.ADD_TER U1.7 "GND"
.TER      C1.2
          U2.7
          P1.4
.ADD_TER R1.1 "VCC"
.TER      U1.14
          U2.4
          U2.1
          U2.14
          P1.1
.ADD_TER U1.2 "N-4"
.TER      U2.3
.ADD_TER P1.2 "/SIG_OUT"
.TER      U2.5
          U2.2
.ADD_TER R1.2 "/CLOCK_IN"
.TER      C1.1
          U1.1
          P1.3

.END
```

## 创建 OrcadPCB2 网表文件

此格式只有一个部分是封装列表。每个封装包括其参考网的列表。

以下是此特定表的式表：

```

<?xml version="1.0" encoding="ISO-8859-1"?>
<!--XSL style sheet to Eeschema Generic Netlist Format to CADSTAR netlist format
Copyright (C) 2010, SoftPLC Corporation.
GPL v2.

如何使用:
https://lists.launchpad.net/kicad-developers/msg05157.html
-->

<!DOCTYPE xsl:stylesheet [
  <!ENTITY nl "&#xd;&#xa;"> <!--new line CR, LF -->
]>

<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:output method="text" omit-xml-declaration="yes" indent="no"/>

<!--
  Netlist header
  Creates the entire netlist
  (can be seen as equivalent to main function in C
-->
<xsl:template match="/export">
  <xsl:text>( { Eeschema Netlist Version 1.1  </xsl:text>
  <!-- Generate line .TIM <time> -->
<xsl:apply-templates select="design/date"/>
<!-- Generate line eeschema version ... -->
<xsl:apply-templates select="design/tool"/>
<xsl:text>}&nl;</xsl:text>

<!-- Generate the list of components -->
<xsl:apply-templates select="components/comp"/> <!-- Generate list of components -->

<!-- end of file -->
<xsl:text>)&nl;*&nl;</xsl:text>
</xsl:template>

<!--
  Generate id in header like "eeschema (2010-08-17 BZR 2450)-unstable"
-->
<xsl:template match="tool">
  <xsl:apply-templates/>
</xsl:template>

<!--
  Generate date in header like "20/08/2010 10:45:33"
-->
<xsl:template match="date">
  <xsl:apply-templates/>
  <xsl:text>&nl;</xsl:text>
</xsl:template>

<!--
  This template read each component
  (path = /export/components/comp)
  creates lines:
  ( 3EBF7DBD $noname U1 74LS125
    ... pin list ...
  )
  and calls "create_pin_list" template to build the pin list
-->
<xsl:template match="comp">
  <xsl:text> ( </xsl:text>

```

是 OrcadPCB2 出文件。

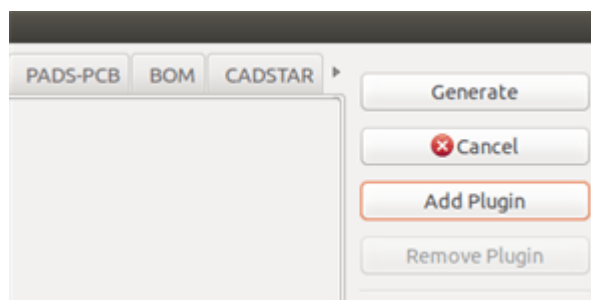
```
( { Eeschema Netlist Version 1.1 29/08/2010 21:07:51
eeschema (2010-08-28 BZR 2458)-unstable}
( 4C6E2141 $noname P1 CONN_4
( 1 VCC )
( 2 /SIG_OUT )
( 3 /CLOCK_IN )
( 4 GND )
)
( 4C6E20BA $noname U2 74LS74
( 1 VCC )
( 2 /SIG_OUT )
( 3 N-04 )
( 4 VCC )
( 5 /SIG_OUT )
( 6 ? )
( 7 GND )
( 14 VCC )
)
( 4C6E20A6 $noname U1 74LS04
( 1 /CLOCK_IN )
( 2 N-04 )
( 7 GND )
( 14 VCC )
)
( 4C6E2094 $noname C1 CP
( 1 /CLOCK_IN )
( 2 GND )
)
( 4C6E208A $noname R1 R
( 1 VCC )
( 2 /CLOCK_IN )
)
)
*
```

## Netlist plugins interface

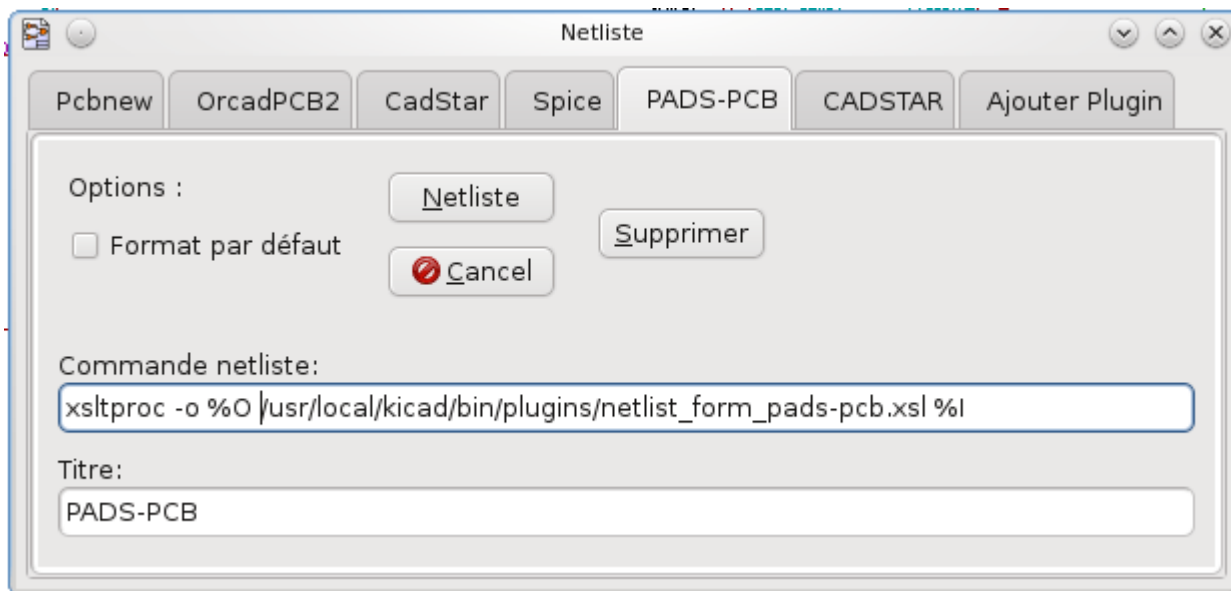
Intermediate Netlist converters can be automatically launched within the Schematic Editor.

### 初始化窗口

可以通过 添加插件 按钮添加新的网表插件用界面卡。



以下是 PadsPcb 卡的配置数据：



## 插件配置参数

The netlist plug-in configuration dialog requires the following information:

- 例如，网表格式的名称。
- 用于后处理器的命令行。

网表按钮后，将产生以下情况：

1. KiCad creates an intermediate netlist file \*.xml, for instance test.xml.
2. KiCad runs the plug-in by reading test.xml and creates test.net.

## 使用命令行生成网表列表文件

假如我使用程序 `xsltproc.exe` 将工作表格式用于中间文件，使用以下命令行 `xsltproc.exe`：

```
xsltproc.exe -o <output filename> < style-sheet filename> <input XML file to convert>
```

在 Windows 下的 KiCad 中，命令行如下：

```
f:/kicad/bin/xsltproc.exe -o "%O" f:/kicad/bin/plugins/netlist_form_pads-pcb.xml "%I"
```

在 Linux 下，命令如下：

```
xsltproc -o "%O" /usr/local/kicad/bin/plugins/netlist_form_pads-pcb.xml "%I"
```

Where `netlist_form_pads-pcb.xml` is the style-sheet that you are applying. Do not forget the double quotes around the file names, this allows them to have spaces after the substitution by KiCad.

命令行格式接受文件名的参数：

支持的格式参数是。

- %B = 基本文件名和所输出文件的路径，减去路径和扩展名。
- %I = 完整的文件名和输入文件的路径（中间网表文件）。
- %O = 完整的文件名和用输出的输出文件的路径。

%I 将被[]的中[]文件名替[]

%O 将替[]出文件名。

## 命令行格式：xsltproc 的示例

xsltproc 的命令行格式如下：

```
<path of xsltproc> xsltproc <xsltproc parameters>
```

在 Windows 下：

```
f:/kicad/bin/xsltproc.exe -o "%O" f:/kicad/bin/plugins/netlist_form_pads-pcb.xml "%I"
```

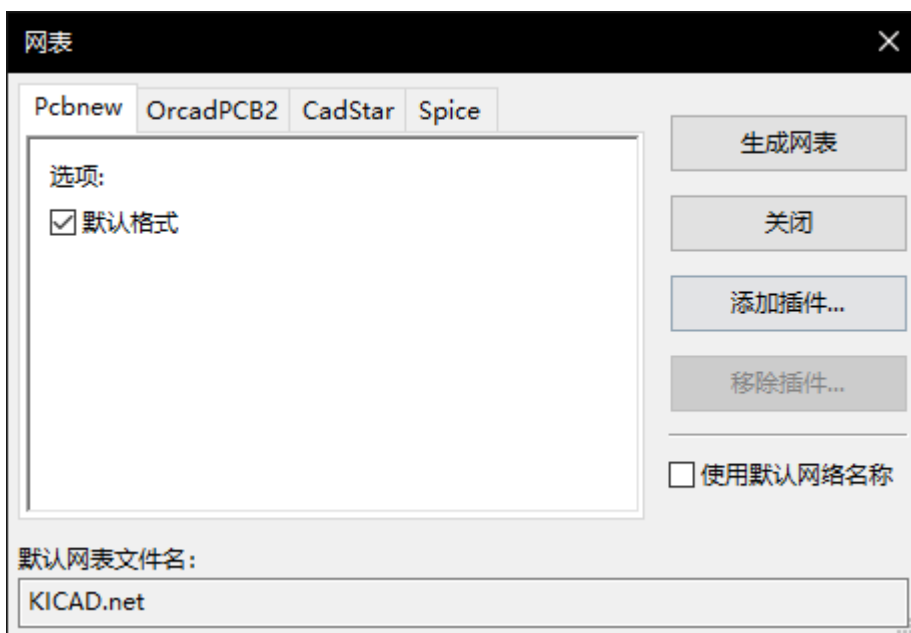
在 Linux 下：

```
xsltproc -o "%O" /usr/local/kicad/bin/plugins/netlist_form_pads-pcb.xml "%I"
```

上面的示例假[] xsltproc 安装在 Windows 下的 PC 上，所有文件都位于 kicad/bin 中。

## 物料清[]BOM) 生成

由于中[]网表文件包含有关已使用元件的所有信息，因此可以从中提取 BOM。 以下是用于[]建自定义物料清[] (BOM) 文件的插件[]置窗口（在 Linux 上）：



[]式表 bom2csv.xml 的路径取决于系[] 目前用于 BOM 生成的最佳 XSLT []式表称[] bom2csv.xml。 您可以根据自己的需要自由修改它，如果您开[]了一些非常有用的[]西，[]让它成[] KiCad []目的一部分。

## 命令行格式：python 脚本的示例

python 的命令行格式如下：

```
python <脚本文件名> <[]入文件名> <[]出文件名>
```

在 Windows 下：

```
python *.exe f:/kicad/python/my_python_script.py "%I" "%O"
```

在 Linux 下：

```
python /usr/local/kicad/python/my_python_script.py "%I" "%O"
```

假如你的 PC 上安装了 python。

## **中网表构**

此示例提供了网表文件格式的概念。



```

<?xml version="1.0" encoding="utf-8"?>
<export version="D">
  <design>
    <source>F:\kicad_aux\netlist_test\netlist_test.sch</source>
    <date>29/08/2010 21:07:51</date>
    <tool>eeschema (2010-08-28 BZR 2458)-unstable</tool>
  </design>
  <components>
    <comp ref="P1">
      <value>CONN_4</value>
      <libsource lib="conn" part="CONN_4"/>
      <sheetpath names="/" tstamps="/" />
      <tstamp>4C6E2141</tstamp>
    </comp>
    <comp ref="U2">
      <value>74LS74</value>
      <libsource lib="74xx" part="74LS74"/>
      <sheetpath names="/" tstamps="/" />
      <tstamp>4C6E20BA</tstamp>
    </comp>
    <comp ref="U1">
      <value>74LS04</value>
      <libsource lib="74xx" part="74LS04"/>
      <sheetpath names="/" tstamps="/" />
      <tstamp>4C6E20A6</tstamp>
    </comp>
    <comp ref="C1">
      <value>CP</value>
      <libsource lib="device" part="CP"/>
      <sheetpath names="/" tstamps="/" />
      <tstamp>4C6E2094</tstamp>
    <comp ref="R1">
      <value>R</value>
      <libsource lib="device" part="R"/>
      <sheetpath names="/" tstamps="/" />
      <tstamp>4C6E208A</tstamp>
    </comp>
  </components>
  <libparts/>
  <libraries/>
  <nets>
    <net code="1" name="GND">
      <node ref="U1" pin="7"/>
      <node ref="C1" pin="2"/>
      <node ref="U2" pin="7"/>
      <node ref="P1" pin="4"/>
    </net>
    <net code="2" name="VCC">
      <node ref="R1" pin="1"/>
      <node ref="U1" pin="14"/>
      <node ref="U2" pin="4"/>
      <node ref="U2" pin="1"/>
      <node ref="U2" pin="14"/>
      <node ref="P1" pin="1"/>
    </net>
    <net code="3" name="">
      <node ref="U2" pin="6"/>
    </net>
    <net code="4" name="">
      <node ref="U1" pin="2"/>
      <node ref="U2" pin="3"/>
    </net>
  </nets>

```

## 一般网表文件结构

中网表占五个部分。

- “” 部分。
- “元件” 部分。
- “元件” 部分。
- “” 部分。
- “网” 部分。

文件内容具有分隔符 <export>

```
<export version="D">
...
</export>
```

### “” 部分

具有分隔符 <design>

```
<design>
<source>F:\kicad_aux\netlist_test\netlist_test.sch</source>
<date>21/08/2010 08:12:08</date>
<tool>eeschema (2010-08-09 BZR 2439)-unstable</tool>
</design>
```

此部分可被批注部分。

### “元件” 部分

元件部分具有分隔符 <components>

```
<components>
<comp ref="P1">
<value>CONN_4</value>
<libsource lib="conn" part="CONN_4"/>
<sheetpath names="/" tstamps="/">
<tstamp>4C6E2141</tstamp>
</comp>
</components>
```

本包含原理中的元件列表。每个元件都是描述的：

```
<comp ref="P1">
<value>CONN_4</value>
<libsource lib="conn" part="CONN_4"/>
<sheetpath names="/" tstamps="/">
<tstamp>4C6E2141</tstamp>
</comp>
```

libsource	找到此元件的库的名称。
part	此库中的元件名称。
sheetpath	本次结构中工作表的路径：工作表 在完整的原理图本次结构中。
tstamps (time stamps)	原理图文件的戳。
tstamp (time stamp)	元件的戳。

### 关于元件的戳的注意事项

要网表中的元件，从而板上，戳参考每个元件都是唯一的。然而，KiCad 提供了一种助方法来元件，元件是路板上相的占位面 允重新批注原理图中的元件，并且不会失元件与其占用空之的接。

戳是原理图中每个元件或工作表的唯一符。但是, 在复的本次结构中, 同一工作表多次使用, 因此此工作表包含具有相同戳的元件。

复本次结构中的定工作表具有唯一符：其 sheetpath。定元件（在复本次结构内）具有唯一符：sheetpath + 其 tstamp

### “部件” 部分

部件部分具有分隔符 <libparts>, 并且此部分的内容在原理图中定义。部件部分包含

- 允的封装名称(名称使用通配符)以 <fp> 分隔符。
- 分隔符中定义的字段 <fields>。
- 引脚列表分隔 <pins>。

```
<libparts>
<libpart lib="device" part="CP">
  <description>Condensateur polarise</description>
  <footprints>
    <fp>CP*</fp>
    <fp>SM*</fp>
  </footprints>
  <fields>
    <field name="Reference">C</field>
    <field name="Valeur">CP</field>
  </fields>
  <pins>
    <pin num="1" name="1" type="passive"/>
    <pin num="2" name="2" type="passive"/>
  </pins>
</libpart>
</libparts>
```

␣似 <pin num="1" type="passive"/> 的␣路也␣出了␣气引脚␣型。可能的␣气引脚␣型有

Input	␣入引脚
Output	␣出引脚
Bidirectional	␣入或␣出
Tri-state	␣␣␣入/␣出
Passive	无源元件的␣束
Unspecified	未知␣气␣型
Power input	␣元件␣源␣入引脚
Power output	␣源␣出引脚作␣␣␣器␣出
Open collector	模␣比␣器中常␣的开路集␣极␣出
Open emitter	有␣在␣␣中找到开放␣射器␣出。
Not connected	必␣在原理␣中保持未␣接状␣

“␣” 部分

␣部分具有分隔符<libraries>。本␣包含␣目中使用的原理␣␣列表。

```
<libraries>
  <library logical="device">
    <uri>F:\kicad\share\library\device.lib</uri>
  </library>
  <library logical="conn">
    <uri>F:\kicad\share\library\conn.lib</uri>
  </library>
</libraries>
```

## “网” 部分

“网” 部分具有分隔符 <nets>。本部分包含原理图的“连接”。

```
<nets>
  <net code="1" name="GND">
    <node ref="U1" pin="7"/>
    <node ref="C1" pin="2"/>
    <node ref="U2" pin="7"/>
    <node ref="P1" pin="4"/>
  </net>
  <net code="2" name="VCC">
    <node ref="R1" pin="1"/>
    <node ref="U1" pin="14"/>
    <node ref="U2" pin="4"/>
    <node ref="U2" pin="1"/>
    <node ref="U2" pin="14"/>
    <node ref="P1" pin="1"/>
  </net>
</nets>
```

本部分列出了原理图中的所有网表

可能的网表包含以下内容。

```
<net code="1" name="GND">
  <node ref="U1" pin="7"/>
  <node ref="C1" pin="2"/>
  <node ref="U2" pin="7"/>
  <node ref="P1" pin="4"/>
</net>
```

net code	是此网表的内部标识符
name	是此网表的名称
node	指出一个连接到网表的引脚引用

## 有关 xsltproc 的更多信息

参见页面：<http://xmlsoft.org/XSLT/xsltproc.html>

## 简介

xsltproc 是一个命令行工具，用于将 XSLT 样式表用于 XML 文档。虽然它是作 GNOME 项目的一部分开发的，但可以独立于 GNOME 桌面运行。

从命令行用 xsltproc，其中包含要使用的样式表的名称，后跟要用样式表的文件的名称。如果提供的文件名是 -，它将使用标准输入。

如果样式表包含在具有样式表处理指令的 XML 文档中，不需要在命令行中命名样式表。xsltproc 将自动包含的样式表并使用它。默认情况下，输出到 *stdout*。您可以使用 -o 指定要输出的文件。

## 简介

```
xsltproc [[-V] | [-v] | [-o *file* ] | [--timing] | [--repeat] |
[--debug] | [--novalid] | [--noout] | [--maxdepth *val* ] | [--html] |
[--param *name* *value* ] | [--stringparam *name* *value* ] | [--nonet] |
[--path *paths* ] | [--load-trace] | [--catalogs] | [--xinclude] |
[--profile] | [--dumpextensions] | [--nowrite] | [--nomkdir] |
[--writesubtree] | [--noddattr]] [ *stylesheet* ] [ *file1* ] [ *file2* ]
[ *...* ]
```

## 命令行选项

*-V o --version*

显示使用的 libxml 和 libxslt 的版本。

*-v o --verbose*

输出 xsltproc 在处理样式表和文档时采取的每个步骤

*-o o --output file*

直接输出到名为 *file* 的文件。对于多个输出，也称 *chunking*，-o *directory/* 将输出文件定向到指定的目录，目录必须已存在。

*--timing*

显示用于解析样式表，解析文档和用样式表并保存结果的总时间以毫秒显示。

*--repeat*

运行 20 次。用于定时

*--debug*

输出后文档的 XML 以行

*--novalid*

跳过加载文档的 DTD。

*--noout*

不输出结果。

*--maxdepth value*

在 libxslt 断定它于无限循环之前，整模板堆的最大深度。默认 500。

*--html*

输入文档是 HTML 文件。

*--param name value*

将名称 *name* 和 *value* 的参数以键值对形式表。您可以指定多个名称/值对，最多 32。如果 *value* 是字符串而不是点号，则改用 *--stringparam*。

*--stringparam name value*

名称 *name* 和 *value* 的参数，其中 *value* 是字符串而不是点号。（注意：字符串必须是 utf-8。）

*--nonet*

不要使用互联网来检索 DTD 实体或文档。

*--path paths*

使用 *paths* 指定的文件系统路径的列表（由空格或列分隔）来加载 DTD 实体或文档。

*--load-trace*

向 stderr 显示加载的所有文档。

*--catalogs*

使用 SGML\_CATALOG\_FILES 中指定的 SGML 目录来解析外部实体的位置。默认情况下，xsltproc 查找 XML\_CATALOG\_FILES 中指定的目录。如果未指定，则使用 /etc/xml/catalog。

*--xinclude*

使用 Xinclude 规范处理输入文档。有关该方面的更多信息，请参阅 Xinclude 规范：  
<http://www.w3.org/TR/xinclude/> [<http://www.w3.org/TR/xinclude/>]

*--profile --norman*

输出分析信息，详细说明每个部分所花费的时间在优化表达式性能时很有用。

*--dumpextensions*

将所有已注册扩展名的列表输出到 stdout。

*--nowrite*

拒绝写入任何文件或源。

*--nomkdir*

拒绝建立目录。

*--writesubtree path*

□允□在 *path* 子□内写入文件。

`--nodtdattr`

不要从文档的 DTD □用默认属性。

## Xsltproc 返回□

xsltproc 返回一个状□号，在脚本中□用它□非常有用。

0: 正常

1: 无参数

2: 参数太多

3: 未知□□

4 : 无法解析□式表

5: □式表中的□□

6 : 其中一个文件出□

7: 不支持的 xsl: □出方法

8 : 字符串参数包含引号和双引号

9: 内部□理□□

10 : 通□□止消息停止□理

11 : 无法将□果写入□出文件

## 有关 xsltproc 的更多信息

libxml 网□□<http://www.xmlsoft.org/>[\[http://www.xmlsoft.org/\]](http://www.xmlsoft.org/)

W3C XSLT □面 : <http://www.w3.org/TR/xslt>[\[http://www.w3.org/TR/xslt\]](http://www.w3.org/TR/xslt)



# 仿真器

KiCad provides an embedded electrical circuit simulator using [ngspice](#) as the simulation engine.

使用模拟器您可能需要官方的 *pspice* 很有用。它包含用于模拟的公共符号，如电压源或晶体管，其引脚号与 ngspice 点序号范相匹配。

有一些演示目录来明确模拟器的功能。您将在 *demos/simulation* 目录中找到它们。

## 分配模型

在启动模拟之前，元件需要分配 Spice 模型。

即使元件由多个元件组成，每个元件也只能分配一个模型。在这种情况下，第一个元件具有指定的模型。

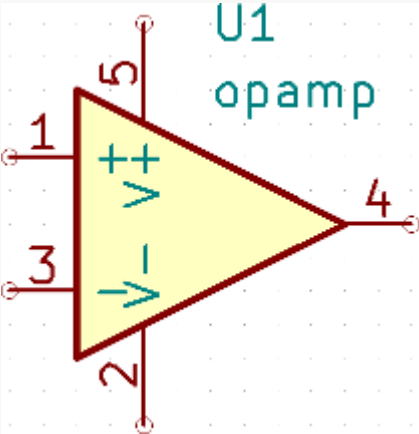
“无源模型”参考匹配 Spice 表示法中的器件类型的无源元件（ $R^*$  表示电阻器， $C^*$  表示电容器， $L^*$  表示电感器）将公式分配模型并使用 `model` 字段确定他的属性。

### NOTE

请记住，在 Spice 表示法中，“M”代表 milli，“Meg”代表 mega。如果您更喜欢使用“M”来表示超前您可以在（模型位置，模型位置框）中请求这样做。

Spice 模型信息作为文本存储在符号字段中，因此您可以在符号器件或原理图器件中定义它。打开符号属性框，然后单击 *Spice 模型* 按钮以打开 Spice 模型器件框。

Spice 模型器件框有三个用于不同模型类型的卡。所有模型类型共有两个子卡。

禁用模型的符号	模型中元件将从模型中排除。
模型点序列	<p>允许模型将符号引脚覆盖模型点映射。要定义不同的映射，模型按模型期的顺序指定引脚号。</p> <p>例子：'+</p> <p>“* 接：”+ “* 1: 非反相输入”</p> <p>“* 2: 反相输入”</p> <p>“* 3: 正电源”</p> <p>“* 4: 负电源”</p> <p>“* 5: 输出”</p> <p>“子电路 tl071 1 2 3 4 5”</p>  <p>要将符号引脚与上面所示的 Spice 模型点相匹配，需要使用具有模型的模型点序列“1 3 5 2 4”。它是与 Spice 模型点顺序的引脚号列表。</p>

无源

无源模型卡允许模型将无源器件模型（电阻，电容或电感）分配给元件。它是一个很少使用的模型，因为它通常被元件的模型分配了模型无源模型，模型形，除非元件引用与模型模型不匹配。

NOTE

明确定义的被模型模型先于模型分配的模型。这意味着一旦分配了被模型模型，在模型期不会考虑参考和模型字段。当指定的模型与原理图上所示的模型不匹配可能会导致混乱的情况。

Spice Model Editor

Passive

Model

Source

Type:Resistor

Passive type

Value:1k

Spice value in simulation

In Spice values,the decimal separator is the point.  
Values can use Spice unit symbols.

Spice unit symbols in values (case insensitive):

f	femto	1e-15
p	pico	1e-12
n	nano	1e-9
u	micro	1e-6
m	milli	1e-3
k	kilo	1e3
meg	mega	1e6
g	giga	1e9
t	tera	1e12

☐ Disable symbol for simulation

☐ Alternate node sequence:

Cancel

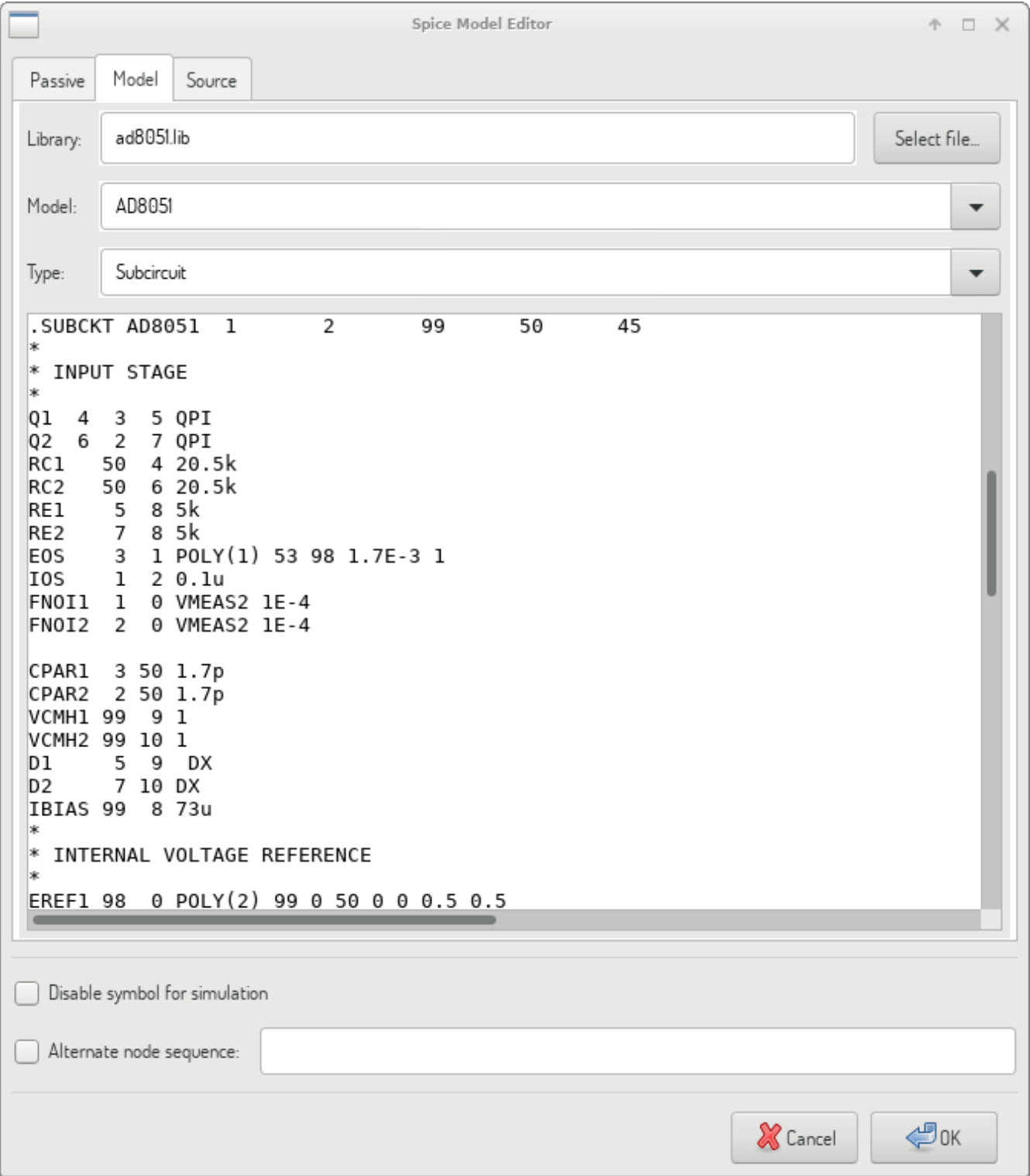
OK

□型	□□器件□型 (□阻, □容或□感)。
□	定义器件属性 (□阻, □容或□感)。□ 可以使用常□的 Spice □元前□□如文本□入字段下方列出的) 和 □□使用点作□小数点分隔符。□注意, Spice 不正确 解□在□中交□的前□□例如 1k5)。

模型

Model □□卡用于分配外部□文件中定义的半□体或复□模型。Spice 模型□通常由□□制造商提供。

主文本小部件显示所的文件内容。将模型描述放在文件中是常见的做法，包括点顺序。

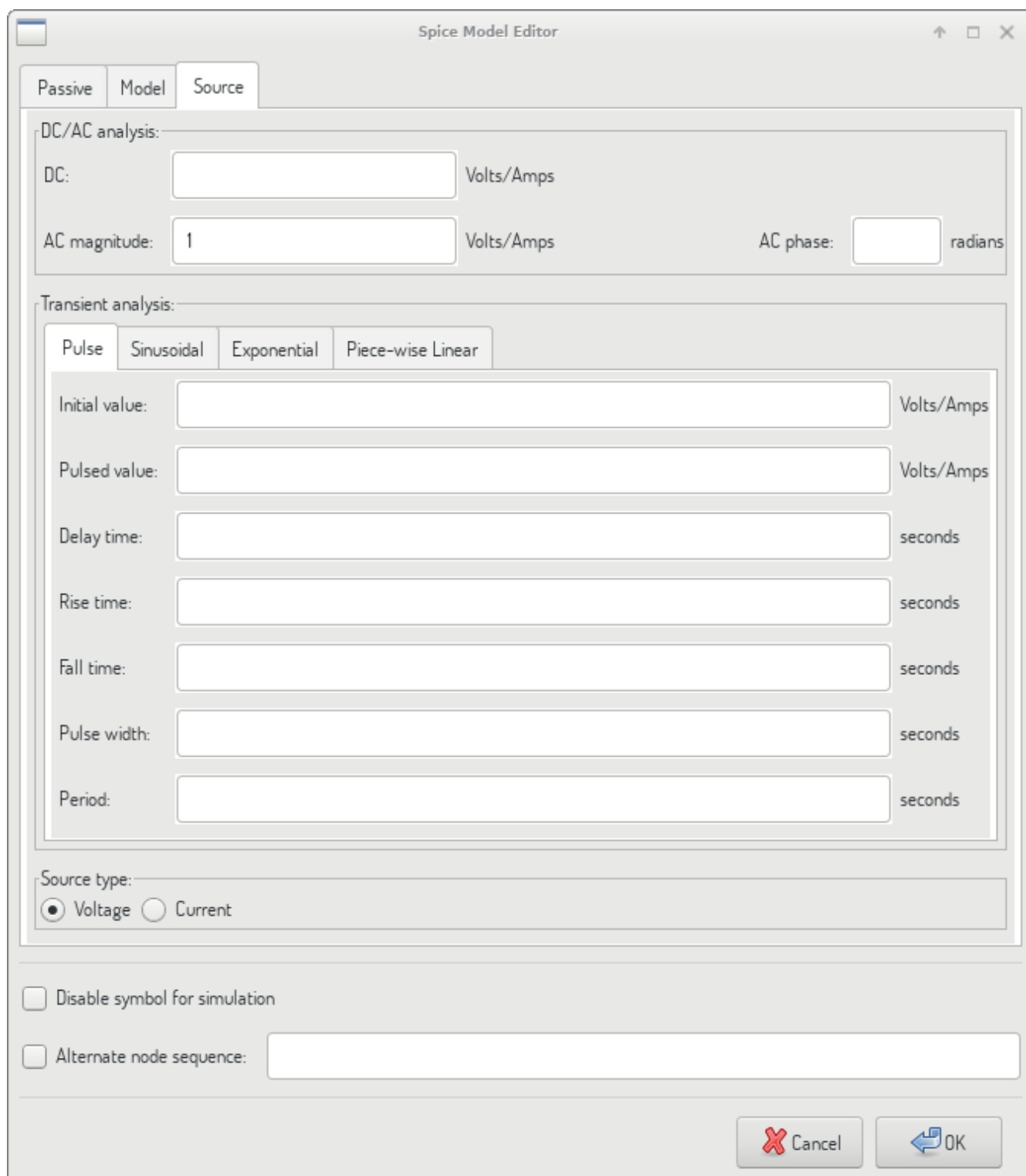


文件	Spice 文件的路径。文件将由模拟器使用，因为它使用 <code>.include</code> 指令添加的。
型号	所型号。文件后，列表将填充可用模型可供
型	型号型（子路，BJT，MOSFET 或二极管）。通常是定的模型自

## 源

Source 卡用于分配源或信号源模型。有两个部分：“DC/AC 分析”和“瞬态分析”。每个都定义了相应模型类型的源参数。

Source type 适用于所有模型。



The image shows the 'Spice Model Editor' dialog box with the 'Source' tab selected. The dialog is divided into two main sections: 'DC/AC analysis' and 'Transient analysis'.

**DC/AC analysis:**

- DC: [ ] Volts/Amps
- AC magnitude: 1 [ ] Volts/Amps
- AC phase: [ ] radians

**Transient analysis:**

Sub-tabs: Pulse (selected), Sinusoidal, Exponential, Piece-wise Linear

- Initial value: [ ] Volts/Amps
- Pulsed value: [ ] Volts/Amps
- Delay time: [ ] seconds
- Rise time: [ ] seconds
- Fall time: [ ] seconds
- Pulse width: [ ] seconds
- Period: [ ] seconds

**Source type:**

- ☒ Voltage ☐ Current

**Options:**

- ☐ Disable symbol for simulation
- ☐ Alternate node sequence: [ ]

**Buttons:** Cancel, OK

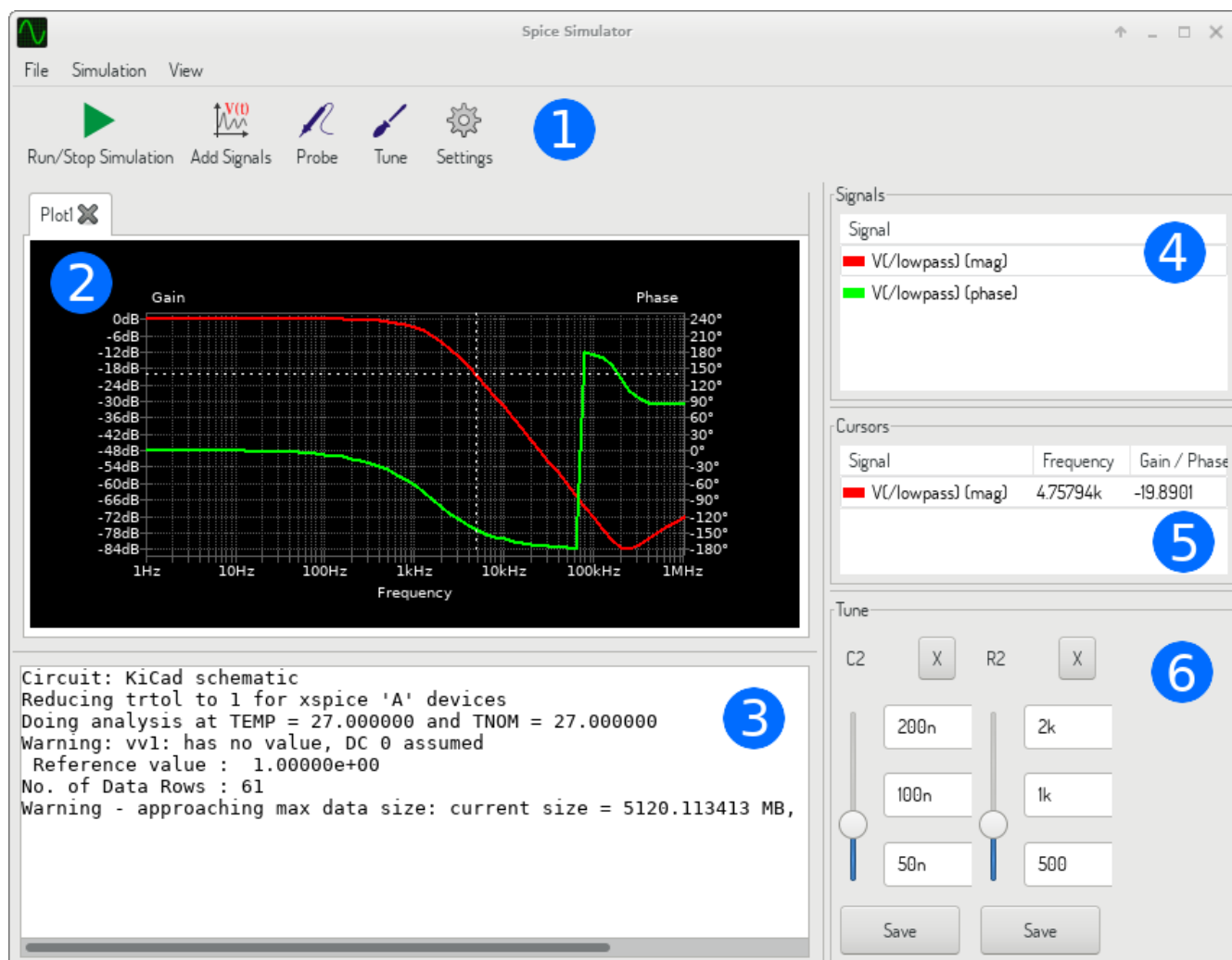
有关源的更多信息，请参考 [ngspice文档](#)，第4章（源和流源）。

## Spice 指令

可以通过将 Spice 指令放在原理图工作表的文本字段中来添加它。此方法便于定义默认模型。此功能仅限于以点开  
的 Spice 指令（例如“.tran 10n 1m”），无法使用文本字段放置其他元件。

## 仿真

要启动模型在原理图器窗口中，菜单 **工具** → **仿真** 打开 Spice 仿真框。



该框分为几个部分：

- 《模型-工具》工具
- 《模型面板, 信号面板》
- 《模型输出控制台, 信号输出控制台》
- 《模型信号列表, 信号列表》
- 《模型游标列表, 游标列表》
- 《模型面板, 信号面板》

## 菜单

## 文件

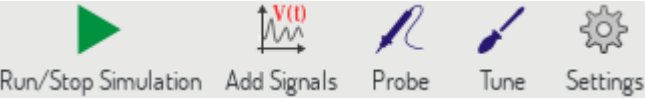
新制	在面板中新建一个新卡。
打开工作簿	打开制信号列表。
保存工作簿	保存制信号列表。
另存像	将活出 .png 文件。
另存 .csv 文件	将活原始数据点出到 .csv 文件。
退出模	关框。

仿真

运行模	使用当前置行模
添加信号.....	打开一个框以要制的信号。
原理探	后原理“模探工具，探”工具。
整元件	后“模工具，”工具。
示 SPICE 网表...	打开一个框，示生成的网表 模路。
置...	打开“模置，模置框”。

小	小活
适合屏幕	整放置以示所有
示网格	切网格可性。
示度	切表例可性。

工具



部工具提供最常行的操作的

运行/停止模	后或停止模
添加信号	打开一个框以要制的信号。
探	后原理“模探工具，探”工具。
	后”模工具，“工具。
置	打开“模置，模置框”。

## 面板

将模拟结果可视化可以在单独的卡片中打开多个，但只有在并行模拟才会更新活动。这样就可以比较不同运行的模拟结果。

可以使用“模拟菜单”菜单切换网格和示例可配置性来自定义。当示例可以拖动它来改变其位置。

面板交互：

- 鼠标放大/缩小
- 右键打开上下文菜单以调整
- 绘制矩形以放大所区域
- 拖光以更改其坐

## 出控制台

出控制台示来自模拟器的消息。建控制台出以确认没有或警告。

## 信号列表

示活中示的信号列表。

信号列表交互：

- 右键打开上下文菜单以隐藏信号或切光
- 双以隐藏信号

## 游列表

示游列表及其坐。每个信号可以示一个光。使用“模拟信号列表，信号”列表置游可性。

## 面板

示使用“模拟工具，”工具取的元件。面板允许快速修改元件并察它模拟结果的影响 - 每次更改元件都会重新运行模拟并更新形。

于每个元件，有一些控件关

- 部文本字段置最大元件
- 中文本字段置的元件
- 底部文本字段置最小元件
- 滑允用以平滑的方式修改元件
- Save 按钮将原理上的元件修改使用滑的元件
- X 按钮从面板中除元件并恢复其原始

三个文本字段 Spice 元前

## 工具

器工具允许要整的元件。



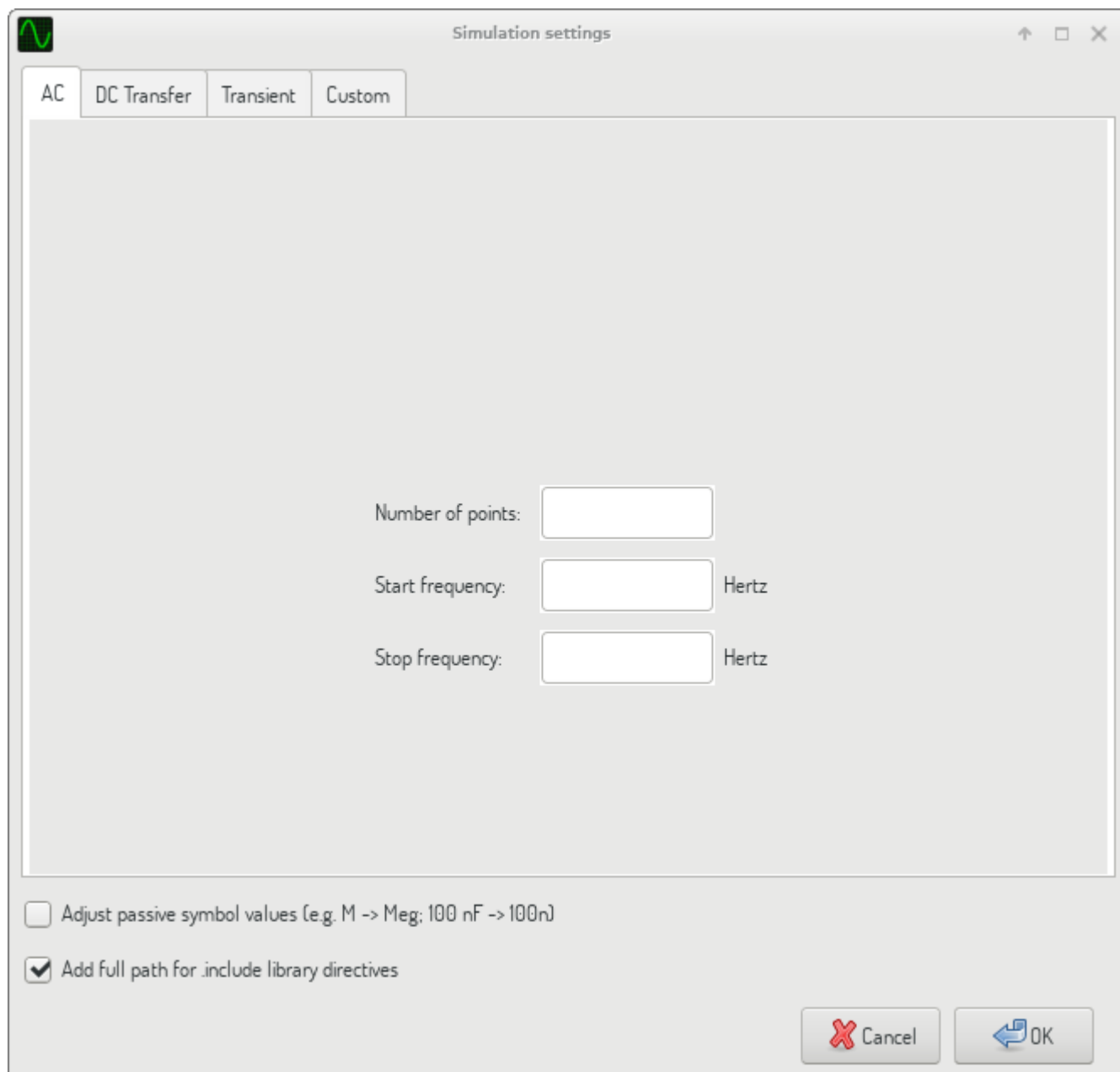
要调整元件，在工具栏中激活原理图器中的一个元件。所选元件将出现在“模型工具，”面板中。只能调整被选元件。

## 探针工具

探针工具提供了一种用户友好的方式来用于信号的。

要向添加信号，在工具栏中激活原理图器中的相应

## 仿真设置

The image shows a "Simulation settings" dialog box. At the top, there is a green waveform icon and the title "Simulation settings". Below the title bar are four tabs: "AC", "DC Transfer", "Transient", and "Custom". The "AC" tab is currently selected. The main area of the dialog is a large, empty rectangular box. Below this box, there are three input fields with labels: "Number of points:" followed by a text box, "Start frequency:" followed by a text box and the unit "Hertz", and "Stop frequency:" followed by a text box and the unit "Hertz". At the bottom of the dialog, there are two checkboxes: "Adjust passive symbol values (e.g. M -> Meg; 100 nF -> 100n)" which is unchecked, and "Add full path for .include library directives" which is checked. In the bottom right corner, there are two buttons: "Cancel" with a red X icon and "OK" with a blue arrow icon.

模型设置框允许用设置模型型和参数。有四个卡：

- 交流
- 直流
- 短的
- 自定义

前三个卡提供可以指定模型参数的表 最后一个卡允许输入自定义 Spice 指令以配置模型 有关仿真模型和参数的更多信息，参 [ngspice文档](#)，第1.2章。

配置模型的另一种方法是在原理图上的文本字段中输入“模型指令，Spice 指令”。与模型相关的任何文本字段指令都会被框中指令的覆盖。意味着一旦开始使用模型框，框将覆盖原理图指令，直到重新打开模型器。

所有模型共有两个

被符号	替被符号以常元件符号表示 Spice 表示法。
.include 指令添加完整路径	Prepend Spice 模型文件名完整路径。通常，ngspice 需要完整路径才能文件。