

Eeschema 介

The KiCad Team

Table of Contents

Eeschema 介	2
描述	2
技 概述	2
通用 Eeschema 命令	3
鼠 命令	3
	4
格点	7
放	7
示光 坐	7
菜	7
上方工具	8
右 工具	10
左工具	11
出菜 和快速	12
主菜	15
文件菜	15
首 菜	15
帮助菜	22
通用 部工具	23
表格管理	23
搜索工具	23
网表工具	24
批注工具	25
气 工具	26
物料清 工具	29
字段工具	31
用于封装分配的 入工具	32
管理符号	34
符号 表	34
原理 建和	37
介	37
一般考	37
开	37
符号放置和	37
源端口	40
充	46
救 存的符号	48
分 原理	50
介	50
在 次 构中 航	50
本地、分 和全局	51
次 构 建摘要	51

工作表符号	51
接 - 分 引脚	52
接 - 分	53
次 构	55
平面 次 构	55
符号批注工具	58
介	58
一些例子	59
使用 气 行 计	62
介	62
如何使用 ERC	62
ERC 的示例	63
示 断	63
源引脚和 源 志	65
配置	65
ERC 告文件	66
建网 列表	67
概述	67
网表格式	67
网表示例	68
于网表的 明	70
其他格式	72
和打印	74
介	74
常 的打印命令	74
在 Postscript 中 制	74
以 PDF 格式 制	75
在 SVG 中	76
在 DXF 中	76
在 HPGL 中	76
在 上打印	77
符号 器	79
于符号 的一般信息	79
符号 概述	79
符号 器概述	79
与	82
建 符号	84
形元素	87
每个符号多个 位和替代体型 式	89
引脚 建和	91
符号字段	94
源符号	96
LibEdit - 符号	98
概述	98

定位符号 点	98
符号 名	99
符号字段	99
符号文档	100
符号	103
符号 器	104
介	104
-主屏幕	104
符号 器 部工具	105
建自定义网表和 BOM 文件	106
中 网表文件格式	106
新的网表格式	108
XSLT 方法	108
命令行格式：python 脚本的示例	116
中 网表 构	117
有 xsltproc 的更多信息	122
仿真器	126
分配模型	126
Spice 指令	131
仿真	131

参考手册

版

本文档由其 献者 有版 ©2010-2018，如下所列。您可以根据 GNU 通用公共可 (<http://www.gnu.org/licenses/gpl.html>)，版本 3 或更高版本，或知 共享署名可 (<http://creativecommons.org/licenses/by/3.0/>)，版本 3.0 或更高版本的条款分 和/或修改它。

本指南中的所有商 均属于其合法所有者。

献者

Jean-Pierre Charras, Fabrizio Tappero.

翻 人

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

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

反

将任何 告、建 或新版本引 到此：

- 于 KiCad 文档: <https://gitlab.com/kicad/services/kicad-doc/issues>
- 于 KiCad 件: <https://gitlab.com/kicad/code/kicad/issues>
- 于 KiCad 翻： <https://gitlab.com/kicad/code/kicad-i18n/issues>

出版日期和 件版本

2015 年 5 月 30 日出版。

Eeschema 介

描述

Eeschema 是一个原理 计 件，作 KiCad 的一部分分 可在以下操作系 下使用：

- Linux
- Apple OS X
- Windows

无 操作系 如何，所有 Eeschema 文件都可以从一个操作系 100 兼容到另一个操作系

Eeschema 是一个集成的 用程序，其中 控制，布局， 管理和 PCB 计 件的所有功能都在 Eeschema 本身内行。

Eeschema 打算与 PcbNew 合作，后者是 KiCad 的印刷 路 计 件。它 可以 出网表文件，其中列出了其他 件包的所有 气 接。

Eeschema 包含一个符号 器，可以 建和 符号并管理 它 集成了 代原理 捕 件所需的以下附加但必不可少的功能：

- 气 ERC)，用于自 控制 和缺失的 接
- 以多种格式 出 文件（Postscript, PDF, HPGL和SVG）
- 物料清 生成（通 Python 或 XSLT 脚本，允 多灵活的格式）。

技 概述

Eeschema 受可用内存的限制。因此， 元件、元件引脚， 接或板的数量没有 限制。在多 表的情况下，表示是分 的。

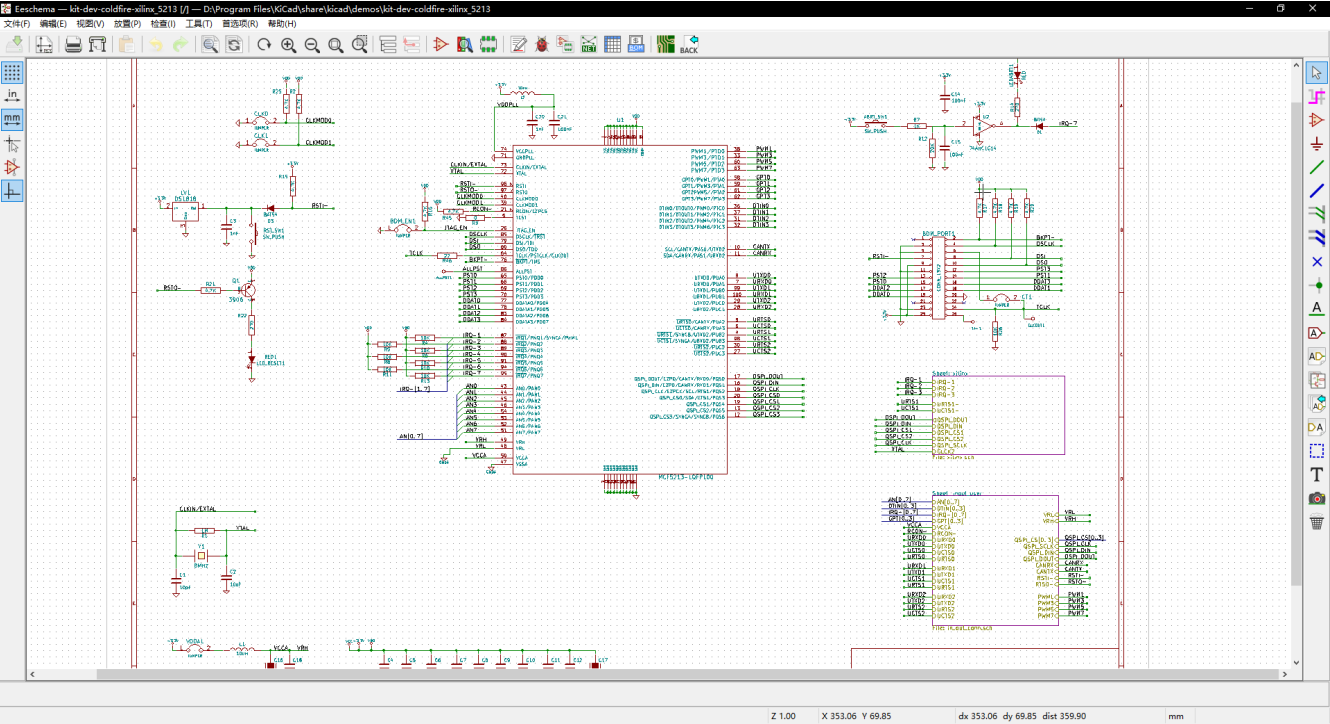
Eeschema可以通 以下几种方式使用多表格 表：

- 的 次 构（每个原理 只使用一次）。
- 的 次 构（一些原理 在多个 例中不止一次使用）。
- 扁平 次 构（原理 未在主 中明确 接）。

通用 Eeschema 命令

命令可以通过以下方式行：

- 菜单（屏幕底部）。
- 屏幕底部的（常用命令）。
- 点屏幕右的（特定命令或“工具”）。
- 屏幕左的（显示）。
- 按下鼠标（重要的充命令）。特别是右击打开光下元素的上下文菜单，网格和元素
- 功能 F1, F2, F3, F4, Insert 和 Space 具体来 Esc 取消正在行的命令。Insert 允制最后建的元素。
- 按通常行工具命令并在当前光位置开始工具操作。有列表，参“帮助 → 列出”菜单或按“Ctrl+F1”



鼠标命令

基本命令

左

- 在状态栏中显示光下的符号或文本的特征。
- 双击如果元素可符号或文本。

右

- 打开出菜

阻止操作

您可以在所有 Eeschema 菜 中移 拖 制和 除所 区域。

通 使用鼠 左 在 目周 制一个框来 区域。

在 期 按住“Shift”，“Ctrl”或“Shift + Ctrl”分 行 制，拖 和 除：

鼠 左	移
Shift + 鼠 左	制
Ctrl + 鼠 左	拖
Ctrl + Shift + 鼠 左	除

拖 或 制 您可以：

- 再次 以放置元素。
- 右 或按 Esc 取消。

如果已启 移 命令， 可以使用右 出菜 另一个命令。



- “Ctrl+F1” 示当前 列表。
- 可以在“原理 器 ” 框的“控件” 卡中重新定义 菜 “首 ” → “常 ”)

是默认的 列表：

帮助（此窗口）	Ctrl+F1
放大	F1
小	F2

放重	F3
放中心	F4
适合屏幕	Home
放到	@
重置本地坐	Space
目	E
除 目	Del
旋 目	R
拖 目	G
撤消	Ctrl+Z
重做	Ctrl+Y
鼠 左	Return
鼠 左 双	End
保存原理	Ctrl+S
加 原理	Ctrl+O
找 目	Ctrl+F
找下一个 目	F5
找下一个DRC	Shift+F5
找和替	Ctrl+Alt+F
重 最后一	Ins
移 → 拖	Tab
制	Ctrl+C
粘	Ctrl+V
切	Ctrl+X
移 原理 目	M
重 的符号或	C
添加符号	A
添加 源	P

像 X	X
像 Y	Y
定向普通符号	N
符号	V
符号参考	U
符号封装	F
使用符号 器	Ctrl+E
开始画	W
开始	B
端 路	K
添加	L
添加分	H
添加全局	Ctrl+L
添加 接点	J
添加无 接 志	Q
添加表	S
添加 入口	Z
添加 入口	/
添加 形折	I
添加 形文字	T
从原理 更新到PCB	F8
自置域	O
留下表	Alt+BkSp
除 点	BkSp
突出 示 接	Ctrl+X

可以使用 器重新定义所有 菜 首 → 常 → 《首 -控件, 控件》)。

可以使用菜 (首 → 入和 出 → 入/ 出) 入/ 出 置。

格点

在 Eeschema 中, 光 始 在网格上移 网格可以自定义 :

- 可以使用 出菜 或使用 “首 / ” 菜 更改大小。
- 可以在 “原理 器 ” 框的 “色” 卡中更改 色 (菜 首 → 常
- 可以使用左 工具 按 切 可 性。

默认网格尺寸 50mil (0.050") 或 1.27mm。

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。

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

放

要更改 放 :

- 右 以打开 出菜 然后 所需的 放。
- 或使用功能 :
 - F1 : 放大
 - F2 小
 - F4 或只需 鼠 中 不移 鼠 光 指 位置居中
- 窗口 放:
 - 鼠 放大/ 小
 - Shift +鼠 向上/向下平移
 - Ctrl +鼠 向左/向右平移

示光 坐

示 位 英寸或毫米。但是, Eeschema 是使用0.001英寸 (mil/thou) 作 其内部 元。

窗口右下角 示以下信息 :

- 放系数
- 光 的 位置
- 光 的相 位置

按空格可以将相 坐 重置 零。 于 量两点之 的距离或 象很有用。

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

菜

部菜 允 打开和保存原理 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

此工具 可以 Eeschema 的主要功能。

如果 Eeschema 以独立模式运行， 是可用的工具集：



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。

	建新原理 （在独立模式下）。
	打开原理 （在独立模式下）。
	保存完整的原理 目。
	尺寸并
	打开打印 框。
	将 制/剪切的 目或 粘 到当前工作表。
	撤消： 原最后一次更改。
	重做： 原最后一次撤消操作。
	示在原理 中搜索符号和文本的 框。
	示用于搜索和替 原理 中文本的 框。
	刷新屏幕；放以适
	放大和 小。
	看和 航 次 构
	保留当前工作表并 入 次 构中。
	用符号 器以 看和修改 和符号。
	符号
	批注符号。
	气 器（ERC），自 气 接。

	用CvPcb 符号分配封装。
	出网表（Pcbnew, SPICE和其他格式）。
	符号字段。
	生成物料清 BOM）。
	用 Pcbnew 行 PCB 布局。
	返回 入封装分配（使用 CvPcb 或 Pcbnew 到“封装”字段中。

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。

	取消激活的命令或工具。
	通 使用不同 色 其 和网 来突出 示网 如果 KiCad 以工程模式运行，那么 于所 网 的 将在 Pcbnew 中也会高亮。
	示符号 器 框以 要放置的新符号。
	示 源符号 器 框以 要放置的 源符号。
	画一根
	画一根
	制 入口点。 些元素 是 形化的，不会 建 接，因此它 不 用于将 接在一起。
	制 到 的入口点。
	放置“无 接”志。 些 志 放在符号引脚上 意味着没有 接。 做是 了通知 气 器特定引脚缺少 接是故意的， 不 告。
	放置一个交叉点。 接两根交叉 或一根 和一个引脚， 当它可能是模糊的（即，如果 端或引脚不是 直接的 接到另一个 端）。
	放置一个本地 本地 接 位于同一 中的物品。 于两个不同工作表之 的 接， 您必 使用全局或 分
	放置一个全局 即使具有相同名称的所有全局 也已 接 位于不同的 上。
	放置分 分 用于 建 子表与包含它的父表之 的 接。
	放置分 子表。您必 指定此子表的文件名。
	从子表 入分 引脚。 命令只能在 行 行 分 子表。它将 建 于分 的分 引脚 放置在目 子表中的
	在子表中放置分 引脚。 命令只能在 行 行 分 子表。它会 建任意分 引脚，即使它 也是如此 目 子表中不存在。
	划一条 些只是 形化的，不能 接任何 西。
	放置文本批注。
	放置位 像。
	除所 元素。

左工具

此工具 管理 示









	切 网格可 性。
	将 位切 英寸。
	将 位切 毫米。
	光 形状（全屏/小）。
	切 “ 藏”引脚的可 性。
	切 自由角度/90 度 和 放置。

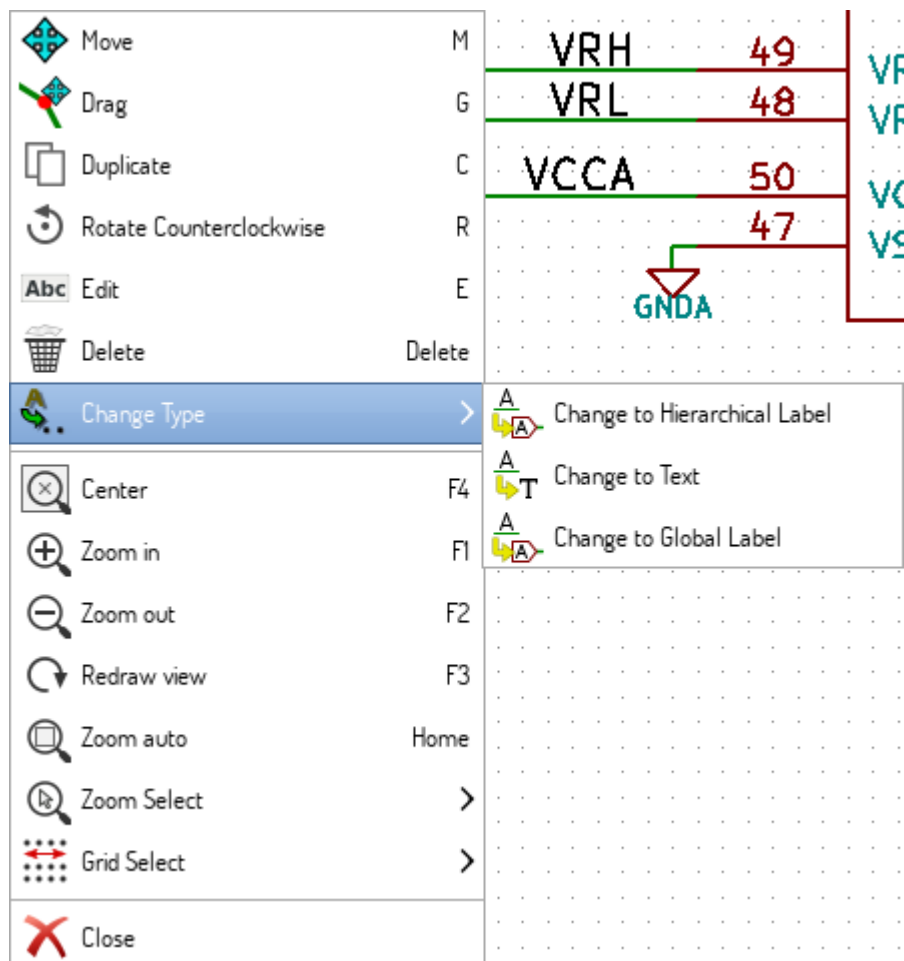
出菜 和快速

右 可打开所 元素的上下文菜 包含：

- 放系数。
- 网格 整。
- 通常 所 元素的参数。

没有 定元素的 出窗口。

	Center	F4
	Zoom in	F1
	Zoom out	F2
	Redraw view	F3
	Zoom auto	Home
	Zoom Select	>
	Grid Select	>
	Close	



符号。

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)

新建	当前原理 并后 一个新原理 在独立模式下)。
打开	加 原理 目 (在独立模式下)。
打开最近	从最近打开的文件列表中打开原理 目 (在独立模式下)。
附加示意	将另一个工作表的内容插入当前工作表。
入非 Kicad 原理 文件	入以其他文件格式保存的原理 目。
保存	保存当前工作表及其所有子工作表。
保存当前工作表	保存当前工作表，而不保存 目中的其他工作表。
将当前工作表另存 ...	以新名称保存当前工作表。
面 置	配置 面尺寸和
打印	打印原理 目 (另 “打印和打印，打印和打印”一章)。
	出 PDF，PostScript，HPGL 或 SVG 格式 (参 “ 和打印， 和打印”一章)。
	止 用程序。

首 菜



管理符号 表	添加/ 除符号
配置路径	置默认搜索路径。
常	首 位，网格大小， 字段名称等）。
置 言	界面 言。
	可 性 置。
入和 出	将首 到/从文件

管理符号 表

符号库

库的范围

全局库工程专用库

文件: C:\Users\taoti\AppData\Roaming\kicad\sym-lib-table

	活动的	别名	库路径	插件类型	
1	<input checked="" type="checkbox"/>	4xxx	\${KICAD_SYMBOL_DIR}/4xxx.lib	Legacy	
2	<input checked="" type="checkbox"/>	4xxx_IEEE	\${KICAD_SYMBOL_DIR}/4xxx_IEEE.lib	Legacy	
3	<input checked="" type="checkbox"/>	74xGxx	\${KICAD_SYMBOL_DIR}/74xGxx.lib	Legacy	
4	<input checked="" type="checkbox"/>	74xx	\${KICAD_SYMBOL_DIR}/74xx.lib	Legacy	
5	<input checked="" type="checkbox"/>	74xx_IEEE	\${KICAD_SYMBOL_DIR}/74xx_IEEE.lib	Legacy	
6	<input checked="" type="checkbox"/>	Amplifier_Audio	\${KICAD_SYMBOL_DIR}/Amplifier_Audio.lib	Legacy	
7	<input checked="" type="checkbox"/>	Amplifier_Buffer	\${KICAD_SYMBOL_DIR}/Amplifier_Buffer.lib	Legacy	
8	<input checked="" type="checkbox"/>	Amplifier_Current	\${KICAD_SYMBOL_DIR}/Amplifier_Current.lib	Legacy	
9	<input checked="" type="checkbox"/>	Amplifier_Difference	\${KICAD_SYMBOL_DIR}/Amplifier_Difference.lib	Legacy	
10	<input checked="" type="checkbox"/>	Amplifier_Operational	\${KICAD_SYMBOL_DIR}/Amplifier_Operational.lib	Legacy	
11	<input checked="" type="checkbox"/>	Amplifier_Instrumentation	\${KICAD_SYMBOL_DIR}/Amplifier_Instrumentation.lib	Legacy	
12	<input checked="" type="checkbox"/>	Amplifier_Video	\${KICAD_SYMBOL_DIR}/Amplifier_Video.lib	Legacy	
13	<input checked="" type="checkbox"/>	Analog	\${KICAD_SYMBOL_DIR}/Analog.lib	Legacy	
14	<input checked="" type="checkbox"/>	Analog_ADC	\${KICAD_SYMBOL_DIR}/Analog_ADC.lib	Legacy	
15	<input checked="" type="checkbox"/>	Analog_DAC	\${KICAD_SYMBOL_DIR}/Analog_DAC.lib	Legacy	
16	<input checked="" type="checkbox"/>	Analog_Switch	\${KICAD_SYMBOL_DIR}/Analog_Switch.lib	Legacy	
17	<input checked="" type="checkbox"/>	Audio	\${KICAD_SYMBOL_DIR}/Audio.lib	Legacy	
18	<input checked="" type="checkbox"/>	Battery_Management	\${KICAD_SYMBOL_DIR}/Battery_Management.lib	Legacy	
19	<input checked="" type="checkbox"/>	Comparator	\${KICAD_SYMBOL_DIR}/Comparator.lib	Legacy	
20	<input checked="" type="checkbox"/>	Connector	\${KICAD_SYMBOL_DIR}/Connector.lib	Legacy	
21	<input checked="" type="checkbox"/>	Connector_Generic	\${KICAD_SYMBOL_DIR}/Connector_Generic.lib	Legacy	

浏览库...

添加库

移除库

上移

下移

替换路径:

	环境变量	路径段
1	KICAD_SYMBOL_DIR	D:\Program Files\KiCad\share\kicad\library
2	KIPRJMOD	D:\Program Files\KiCad\share\kicad\demos\interf_u

确定

取消

Eeschema 使用两个 表来存 可用符号 列表， 些符号 因范 而异：

- 全局

每个 目都可以使用全局 表中列出的 它 保存在您的主目 中的 **sym-lib-table** 中（确切路径取决于操作系； 表 上方的路径）。

- 目 用

目 用 中列出的 可用于当前打开的 目。它 保存在 目目 中的 **sym-lib-table** 文件中（ 表格上方的路径）。

您可以通 表下方的 全局 或 目 用 卡来 看任一列表。

添加一个新

通 ... 按 并 文件或 “附加 ” 并 入 文件的路径来添加 定的 将添加到当前打开的 表（全局/ 目 用） 中。

除

通 一个或多个 并 除 按 来 除

属性

表中的每一行都存 了几个描述 的字段：

活	启用/禁用 减少加 的 集很有用。
昵称	昵称是用于将符号分配 元件的 短唯一 符。符号 由'<Library Nickname> : <Symbol Name>'字符串表示。
路径	路径指向 位置。
插件 型	确定 文件格式。
	存 特定 如果插件使用）。
明	要描述 内容。

常

示

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

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)

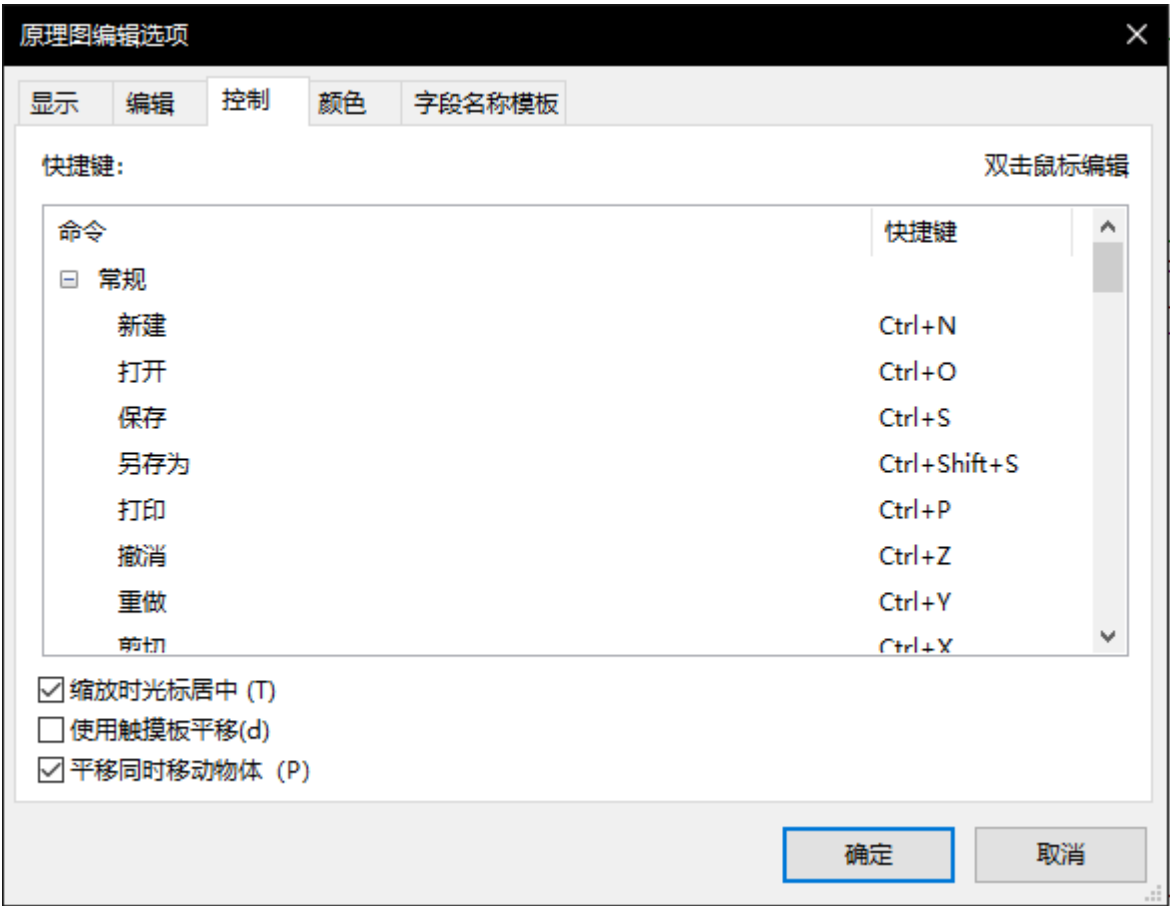
确定

取消

量 位	示和光 坐 位（英寸或毫米）。
重 目的水平 距	元素 制期 X 上的增量（默认 0 （在放置符号， 或 等物 品后， <i>Insert</i> 制）
重 目的垂直 距	在 Y 期 增量 元素 制（默认 0.100英寸或2,54毫米）。
重 的增量	在 制文本 束期 的增量 在一个数字中，例如 成 通常 1 或-1）。
默认文本大小	建新文本 或 使用的文本大小。
自 保存 隔	保存 份之 的 分
自 放置符号字段	如果 中， 符号字段（例如， 和 可能会移 新放置的符号以避免 与其 生冲突 其他 目。
允 字段自 放置更改	展“自 放置符号字段”的 放置 后用符号字段的文本 整一个 新的部分。
始 将自 播放的字段与 50mil 网格	自 展 放置符号字段的 如果 中， 使用 50mil 自 放置字段 网 格， 否 它 是自由放置的。

控制

重新定义 并 置用 界面行



通 双 操作 新的 或右 操作以 示 出菜

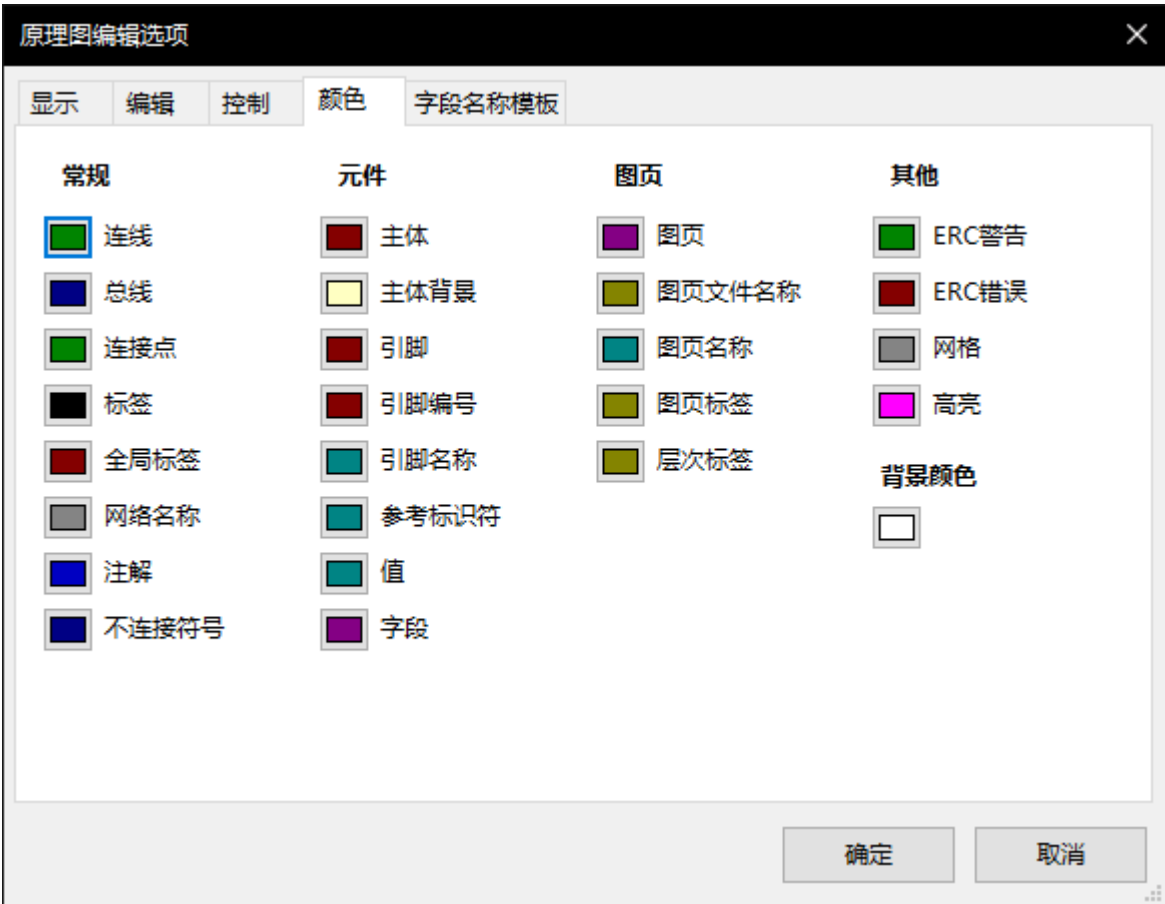
	操作定义新的 与双 相同) 。
撤消更改	原操作的最近 更改。
恢 默认	将操作 置 其默认
撤消所有更改	原操作的所有最近 更改。
全部恢 默认	将所有操作 置 其默认

明:

中心和 形光 焦	如果 中, 指向的位置会 形 放大/ 小 到屏幕中心。
使用触摸板 行平移	启用 使用 或。) 平移 触摸板手 和 放一个需要按住 Ctrl 否 放大/ 小和 Ctrl/Shift 是平移修改器。
移 象 平移	如果 中, 自 平移窗口 如果光 在 或移 程中离开窗口。

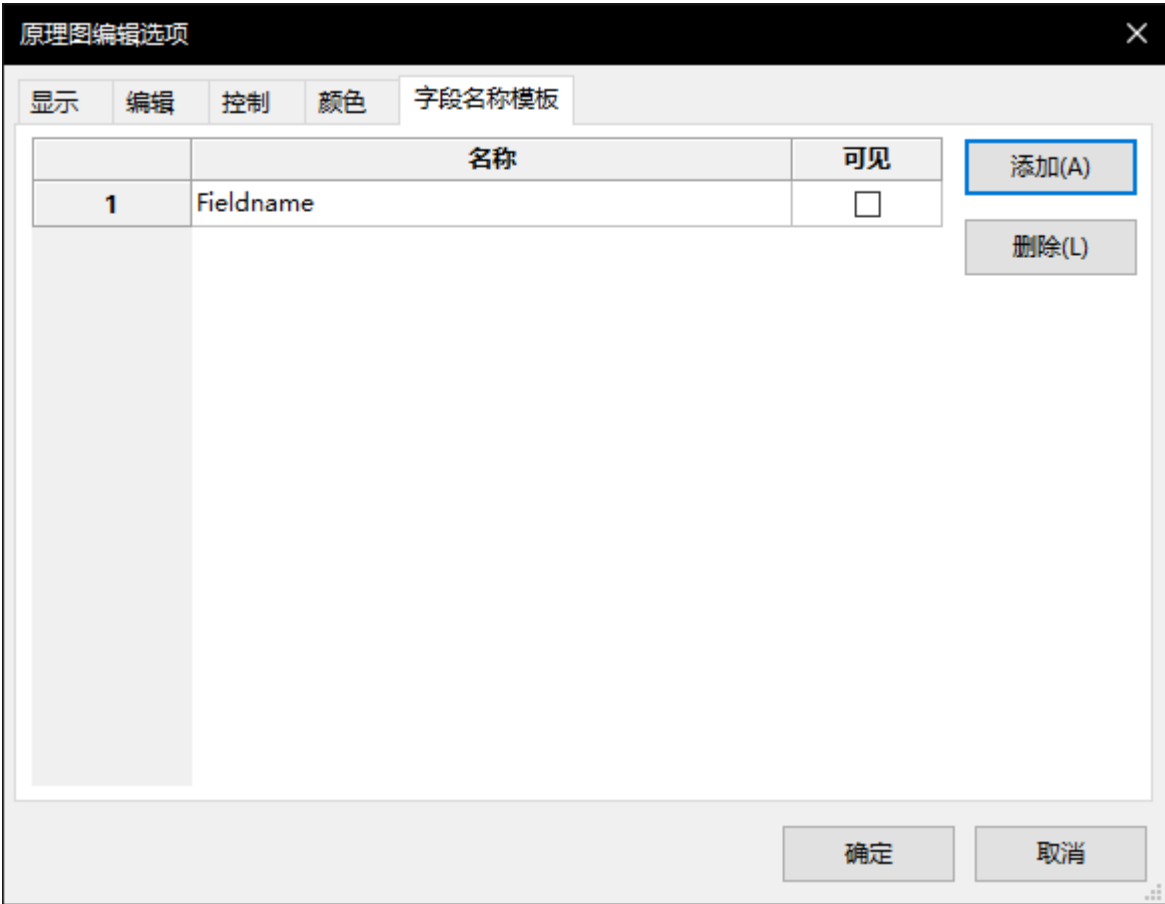
色

各种 形元素的配色方案。 任何 色 本以 特定元素的新 色。



默认字段

定义将在新放置的符号中 示的其他自定义字段和相 的



帮助菜

在 帮助（本文档）， 取有 KiCad 的广泛教程。

在提交 告 使用“制版本信息”来 您的构建和系

通用 部工具

表格管理

“ 置” ) 允 您定义 尺寸和 的内容。

页面设置

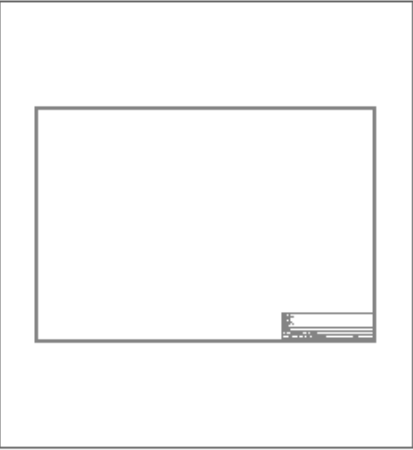
图纸

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

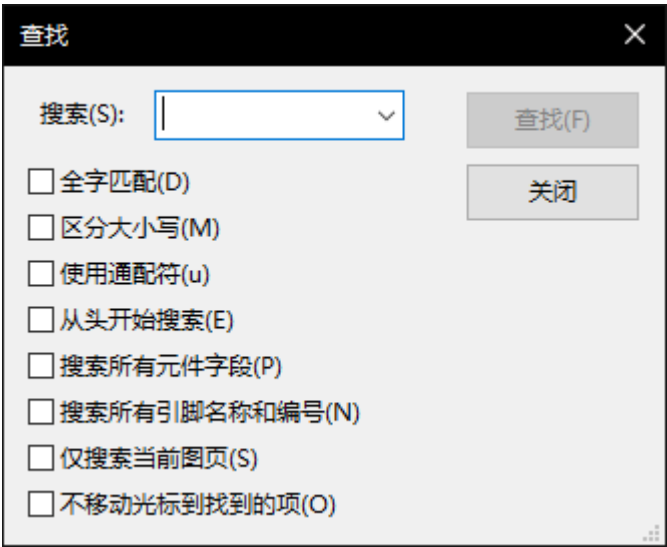
浏览

确定 取消

工作表 号会自 更新。您可以通 按“ 布日期” 按左箭 按 将日期 置 今天, 但不会自 更改。


搜索工具

“ 找” ) 可用于 搜索工具。



您可以在当前工作表或整个 次 构中搜索引用， 或文本字符串。找到后，光 将定位在相 子表中的找到元素上。

网表工具

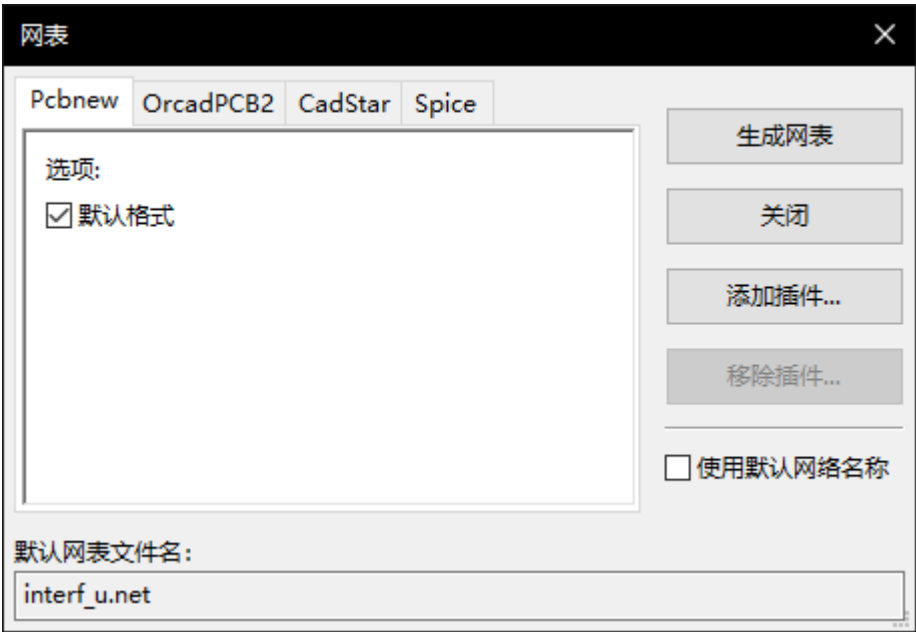
网表 ) 打开网表生成工具。

工具 建一个文件，描述整个 次 构中的所有 接。

在多表 次 构中，任何本地 在其所属的工作表内可 例如：表3的 LABEL1 与表5的 LABEL1 不同（如果没有故意引入 接以 接它 是因 工作表名称路径在内部与本地 相

NOTE 即使 Eeschema 中的 没有文本 度限制， 考 到 取生成的网表的其他程序可能存在此 限制。

NOTE 避免 中的空格，因 它 将在生成的文件中 示 独的 它不是 Eeschema 的限制，而是 多网表格式的限制，通常假 没有空格。



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）。

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

批注工具

 启 批注工具。此工具分配 元件的引用。

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

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

☐ 保持对话框打开

☐ 不要求确认

顺序：

☒ X方向排序元件 (X)

☐ Y方向排序元件 (Y)



编号：

☒ 使用该数字之后的编号：

☐ 参考编号X100

☐ 参考编号X1000

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 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。

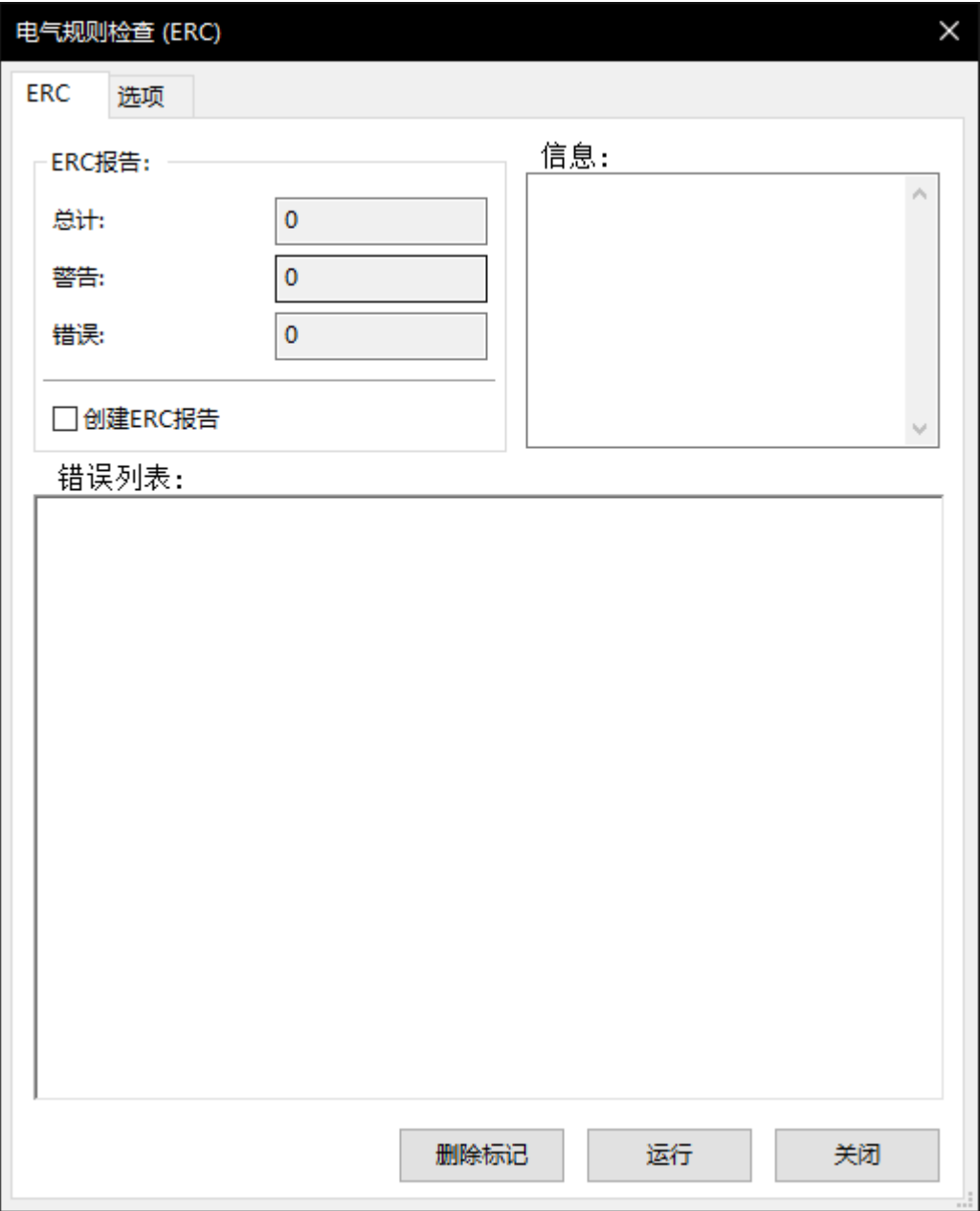
气 工具

 后 子 ERC) 工具。

工具 行 计 能 被 忘的 接和不一致。

运行 ERC 后, Eeschema 会放置 以突出 示 左 后 示 明。 可以生成 告文件。

主要 ERC 框



示在 Electrical Rules Checker 框中：

- 和警告的 数。
- 计数。
- 警告计数。

建 ERC 文件 告	中此 可生成 ERC 告文件。
------------	-----------------


命令：

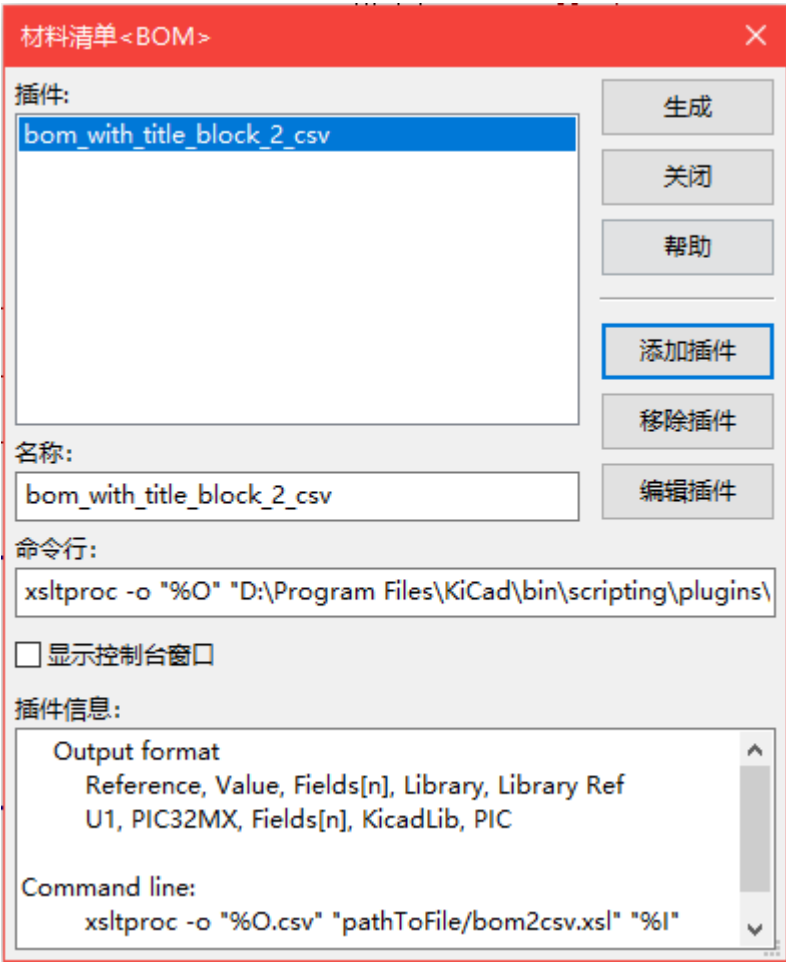
似	告 只有字母大小写（例如 lable/Lable/LaBeL）。网 名称区分大小写，因此 些 被 独的网
独特的全局	告 出 一次的全局 特 网。通常需要至少有两个 接。

命令：

初始化 默认	恢 原始 置。
--------	---------

物料清 工具

 后 物料清 BOM) 生成器。此工具生成一个列出元件和/或分 接（全局 的文件。



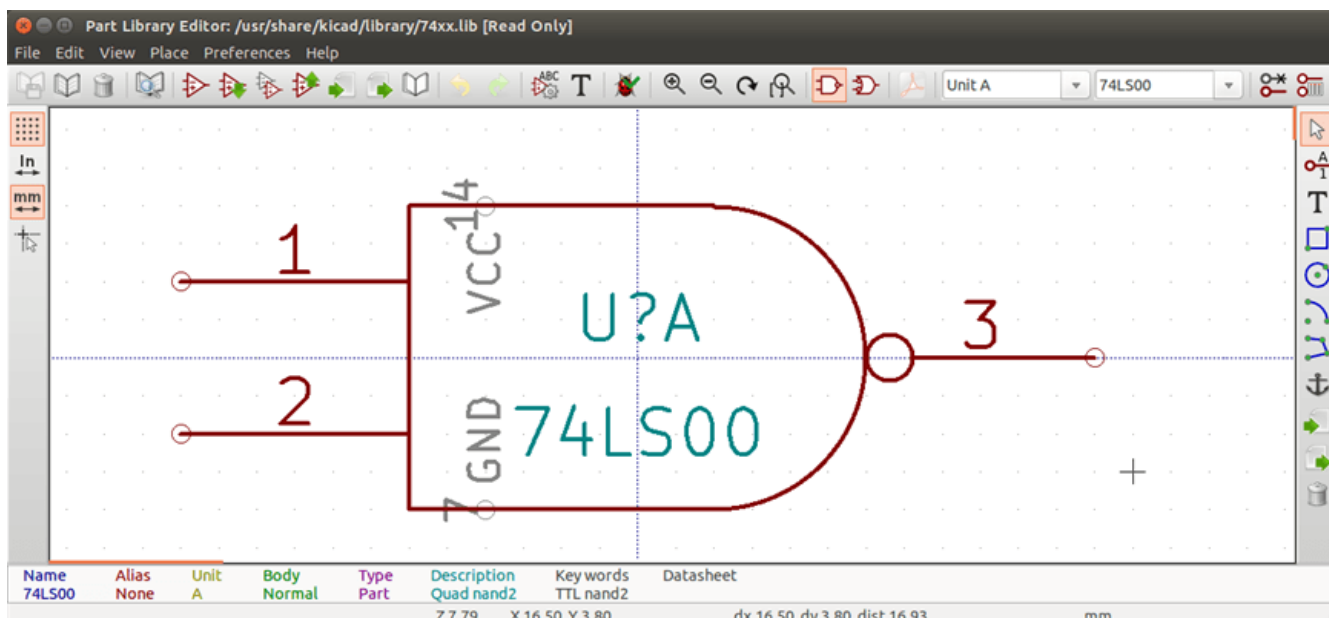
Eeschema 的 BOM 生成器使用外部插件，可以是 XSLT 或 Python 脚本。KiCad 程序文件目 中安装了一些示例。

用于 BOM 的一 有用的元件属性包括：

- -使用的每个部件的唯一名称。
- 封装 - 手 入或反 注（下文）。
- 字段1 - 制造商的名称。
- 字段2 - 制造商的元件号。

例如：

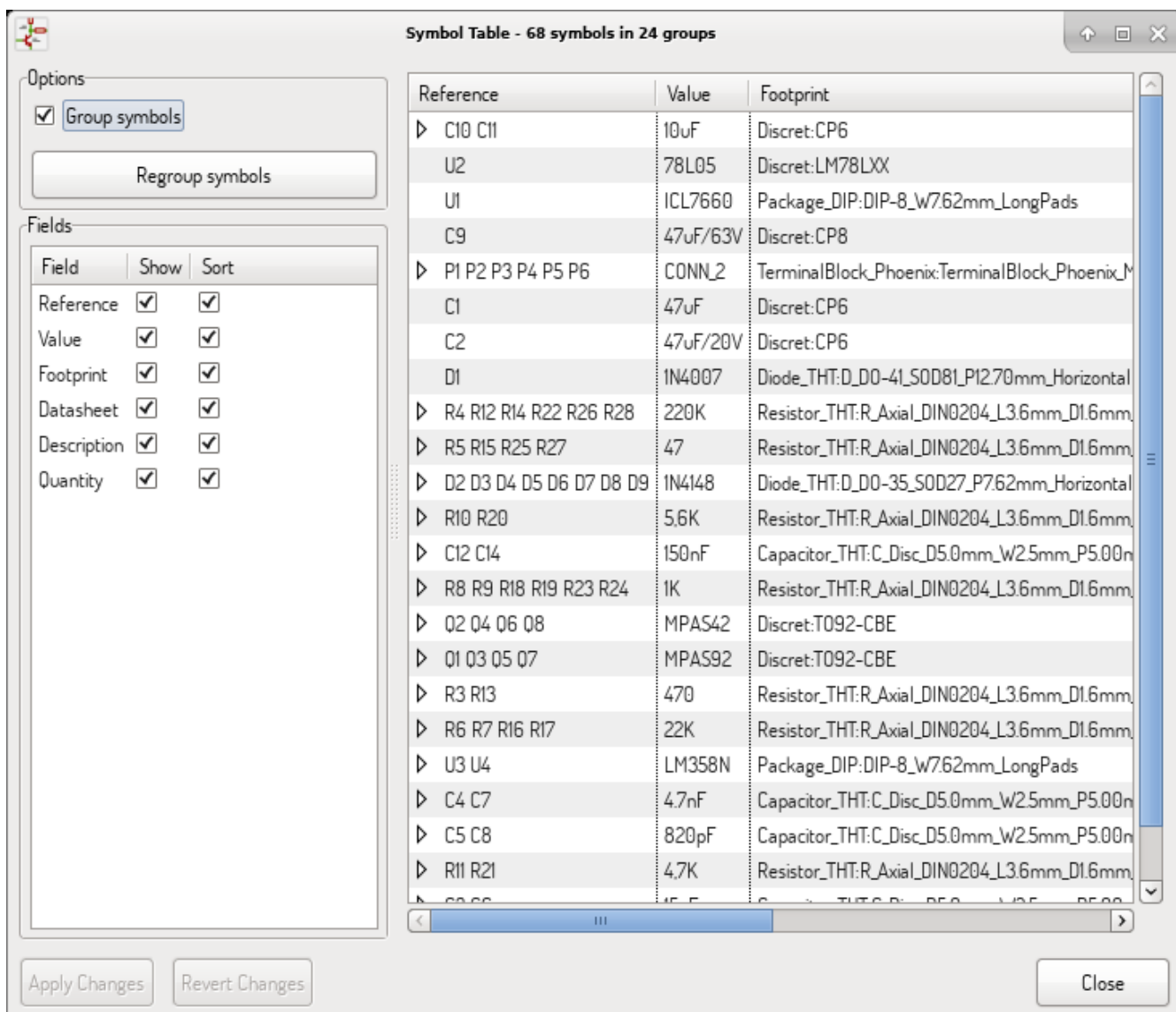
在 **MS Windows** 上, BOM 生成器 框有一个特殊 由 色箭 指示), 用于控制外部插件窗口的可 性。+ 默认情况下, BOM 生成器命令 行控制台窗口 藏, 出重定向到 *Plugin info* 字段。 置此 可 示正在运行的命令的窗口。如果插件提供了 形用 界面, 可能是必要的。



字段工具



打开 子表格以 看和修改所有符号的字段



修改字段 后，您需要通 “用” 按 接受更改，或通 恢 按 撤消更改。

化字段填充的技巧

子表格中有几种特殊的 制/粘 方法。在 入在少数元件中重 的字段 它 可能很有用。

些方法如下所示。

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 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



此工具允 将 PcbNew 中 建的封装更改 入 Eeschema 中的封装字段。

管理符号

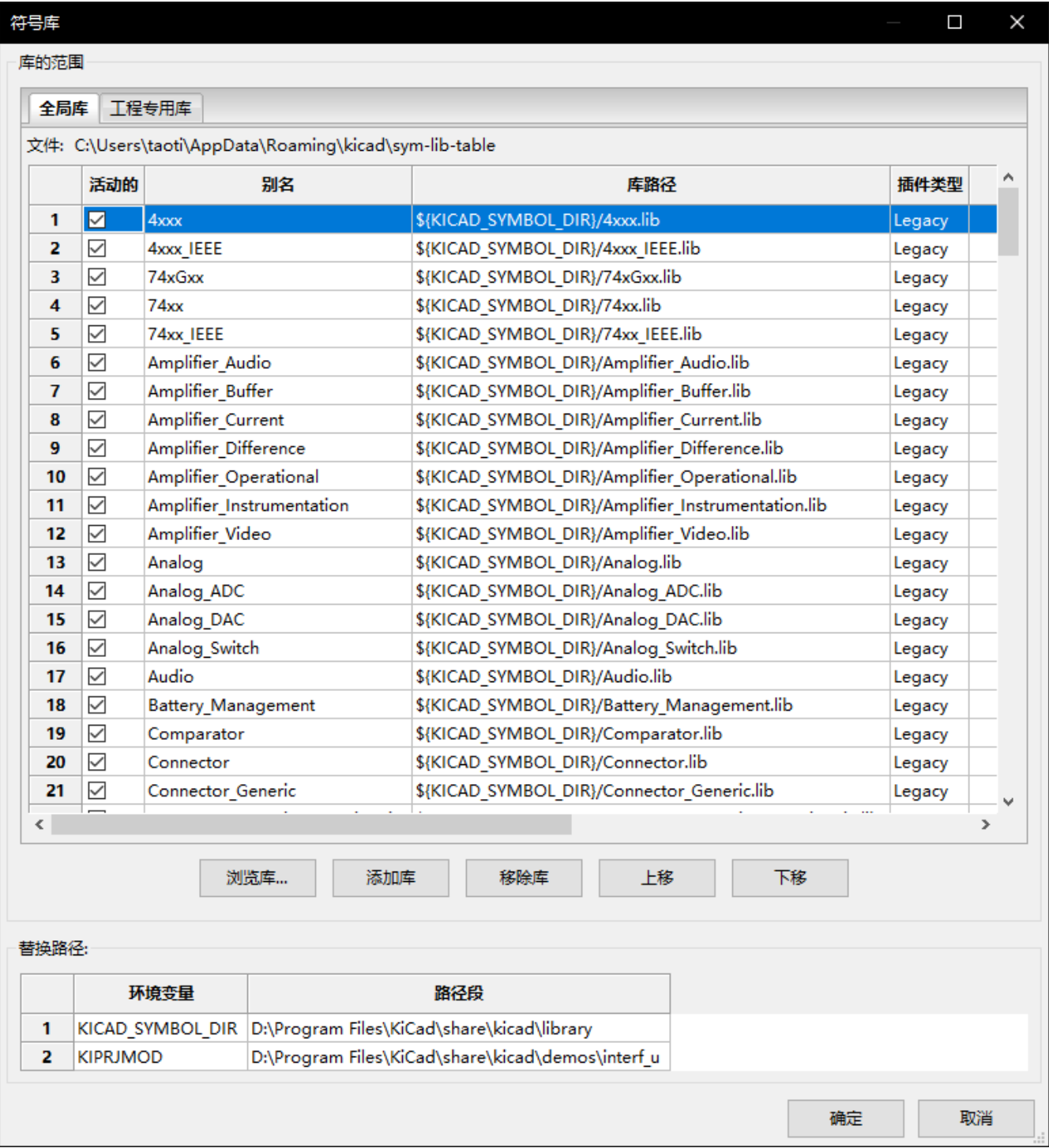
符号 包含 建原理 使用的符号集合。原理 中的每个符号由一个全名唯一 全名由 昵称和符号名称 成。一个例子是“音 AD1853”。

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 目特定符号 表文件构成。

加 符号 Eeschema 使用 昵称（在我 的示例中 “音 ”）来 找符号 表中的 位置。

下 示了符号 表 框，可以通 用 首 菜 中的 管理符号 表条目来打开 框。



全局符号 表

全局符号 表包含始 可用的 列表，与当前加 的 目文件无 表保存在用 主文件 的文件符号列表文件中。此文件的位置取决于所使用的操作系

目特定符号 表

目特定符号 表包含 用于当前加 的 目文件的 列表。 目特定符号 表只能在与 目文件一起加 行 如果未加 目文件或当前 目路径中没有符号 表文件， 会 建一个空表，可以 其 行 然后将其与 目文件一起保存。

初始配置

第一次运行 Eeschema 并且在用 的主文件 中找不到全局符号表文件 **sym-lib-table** Eeschema 将 制存 在系的 KiCad 模板文件 中的默认符号表文件 sym-lib-table 到用 主文件 中的文件 sym-lib-table。如果找不到默认模板 sym-lib-table 文件， 会出 一个 框，提示 入 sym-lib-table 文件的 用位置。如果未找到 sym-lib-table 或解除 框， 将在用 的主文件 中 建空符号 表。如果 生 种情况，用 可以手 制 sym-lib-table 或手 配置表。

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。

每个 条目必 有一个独特的昵称。

不必以任何方式与 文件名或路径相 冒号“:”和“/”字符不能在 昵称中的任何位置使用。每个 条目必 具有有效的路径和/或文件名，具体取决于 的 型。路径可以定义 相 或 境 量替 参 下面的部分)。

必 适当的插件 型才能正确 取 KiCad 目前 支持旧版符号 文件插件。

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 的 昵称。但是，您可以在全局和 目特定的符号 表中包含重 的 昵称。
- 当出 重 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

符号 表的最 大功能之一是 境 量替 允 定义符号 存 在 境 量中的自定义路径。使用 路径中的 法 `$(ENV_VAR_NAME)` 支持 境 量替

默认情况下，在运行 KiCad 定义 两个 境 量：

- **KIPRJMOD** 境 量，始 指向当前打开的 目目 **KIPRJMOD** 无法修改。
- **KICAD_SYMBOL_DIR** 境 量。 指向使用 KiCad 安装的默认符号 的路径。

您可以重写 KICAD_SYMBOL_DIR, 方法是在“首 /配置路径”中自己定义它，路径允 您替 自己的，以取代默认的 KiCad 符号

KIPRJMOD 允 您在没有 目路径的情况下存 必 定义 路径（并不 是已知） 目特定符号 表中的

使用模式

符号 可以全局定义，也可以 定义到当前加 的 目。用 全局表中定义的符号 始 可用，并存 在用 主文件 的 sym-lib-table 文件中。 目特定符号 表 当前打开的 目文件有效。

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。

留 目重新映射

加 在符号 表 之前 建的原理 Eeschema 将 将原理 中的符号 接重新映射到相 的 表符号。 一 程的成功 取决于几个因素：

- 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 行 行所有恢 行 以 建恢 或使 有恢 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 操作以 行恢 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
NOTE	如果已 除原始 并且未 行恢 可以将 存 用作恢 作 最后的手段。将 存 制到新文件名， 并在符号 表 之前使用Eeschema版本将新 文件添加到 列表的 部。

原理 建和

介

原理 可以用 表示, 但是, 如果足 大, 需要多

由几 表示的示意 是分 的, 并且其所有 每个 由其自己的文件表示) 构成Eeschema 目。分 原理 的操作将在《hierarchical-schematics, Hierarchical Schematics》章 中描述。

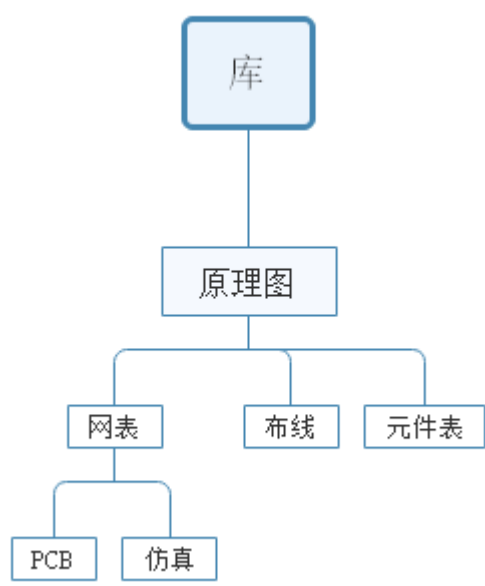
一般考

使用 Eeschema 计的原理 不 是子 的 形表示。它通常是开 的入口点, 允

- 一 ERC 气 以 和 漏。
- 自 生成物料清 BOM) 。
- 用于仿真 件 (如 SPICE) 的 (建 - 定制 - 网表和文件 - 文件, 生成网表) 。
- 建 - 定制 - 网表和网 文件, 生成网表) , 用于 到 PCB 布局。

原理 主要由符号, 接点, 和 源端口 成。 了清晰起 您可以放置 粹的 形元素, 如 条目, 注 和折

开

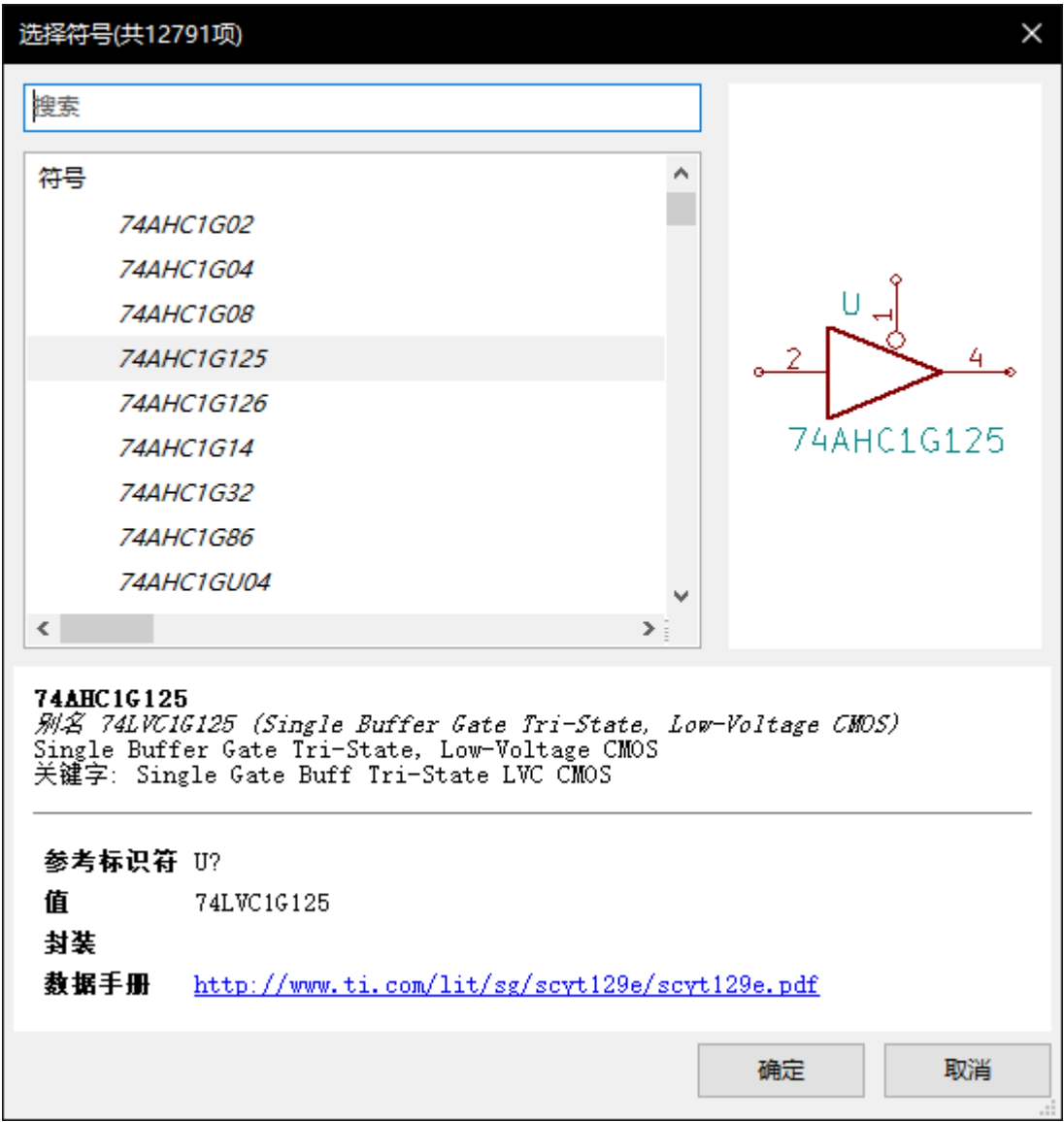


符号从符号 添加到原理 中。在制作原理 之后, 生成一个网表, 稍后用于将 接和封装集 入 PcbNew。

符号放置和

找到并放置一个符号

要将符号加 到原理 中, 可以使用 。使用 框可以 入要加 的符号的名称。



“符号”框将根据您在搜索字段中输入的内容按名称、字母和明符号。只需输入高器即可使用它

- **通配符**：分使用字符“?”和“*”表示“任意字符”和“任意数量的字符”。
- 如果部分的描述或字包含格式“Key: 123”，您可以通过输入相于匹配“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之一：

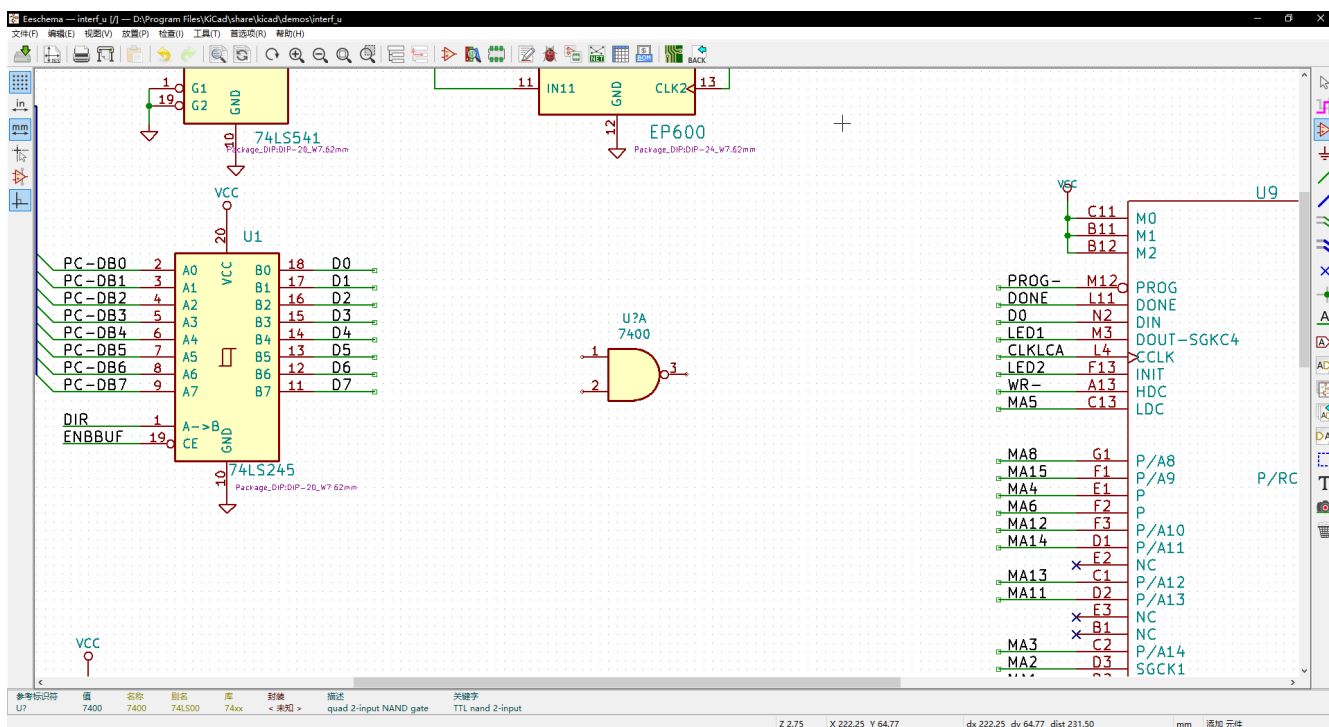
p	n	u	m	k	meg	g	t
10 ⁻¹²	10 ⁻⁹	10 ⁻⁶	10 ⁻³	10 ³	10 ⁶	10 ⁹	10 ¹²

ki	mi	gi	ti
2 ¹⁰	2 ²⁰	2 ³⁰	2 ⁴⁰


- **正则表达式**：如果你熟悉正则表达式，些也可以用。使用的正则表达式味是 wxWidgets 高正表达式 似于 Perl 常表达式。

在将符号放置在原理图之前，您可以使用或右上下文菜单其行旋像和其字段。可以在放置后以相同的方式完成。

是放置期 的符号：



源端口

源端口符号是符号（符号在 源 中分 因此可以使用符号 器放置它 但是，由于 源放置 繁，因此可以使用  工具。 个工具很相似，只是搜索直接在 源 中完成。

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：

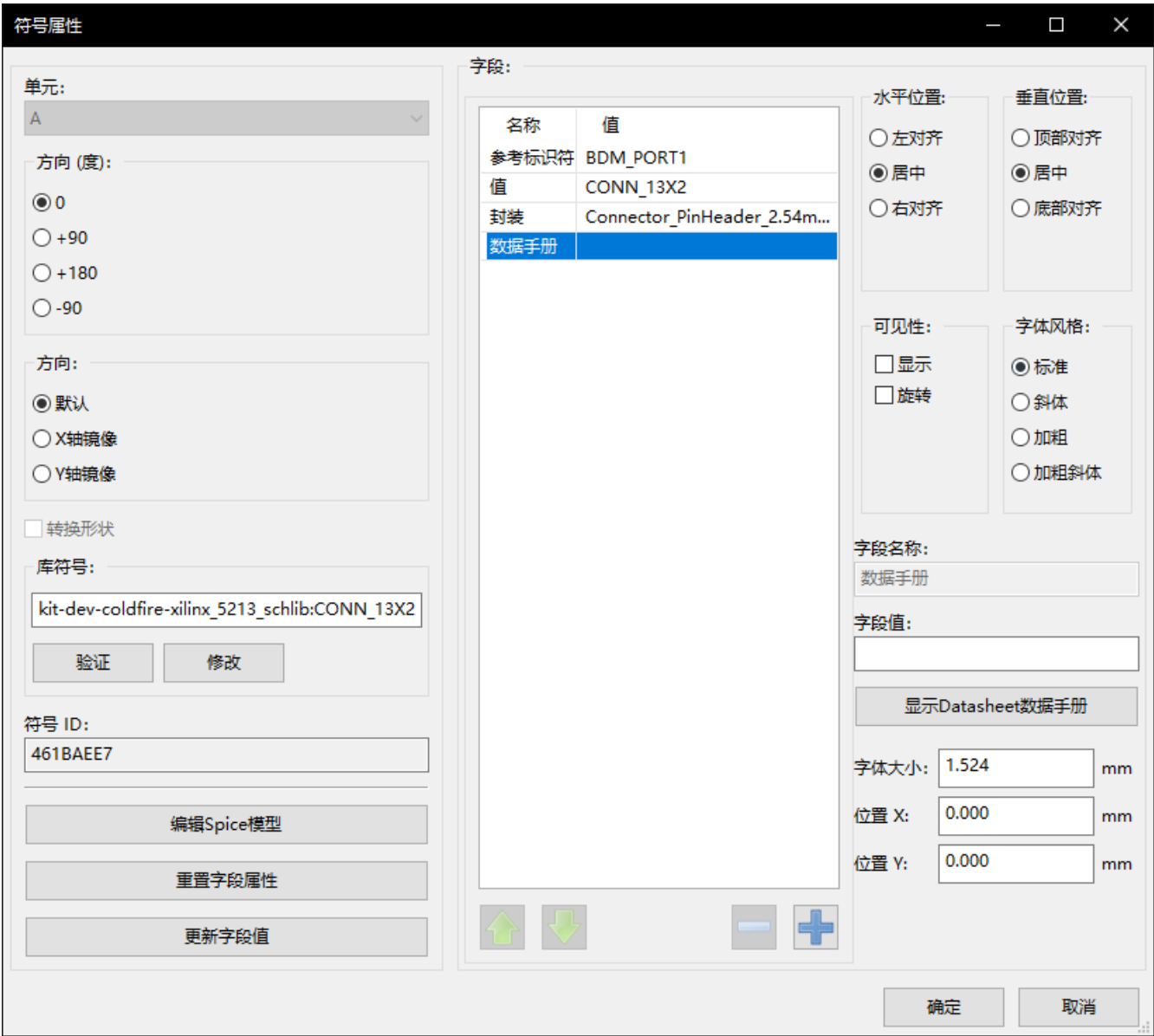
- 双 符号以打开完整的 框。
- 右 以打开上下文菜 并使用以下命令之一：移 方向， 除等。

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 性：

- 双 文本字段 行修改。
- 右 以打开上下文菜 并使用以下命令之一：移 旋 除等。

要 得更多 或者要 建字段， 双 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。

重置默认 将符号置回原始方向，并重置每个字段的字体大小和位置。但是，文本字段不会被修改，因为它们可能会破坏原理图。

源端口

简介

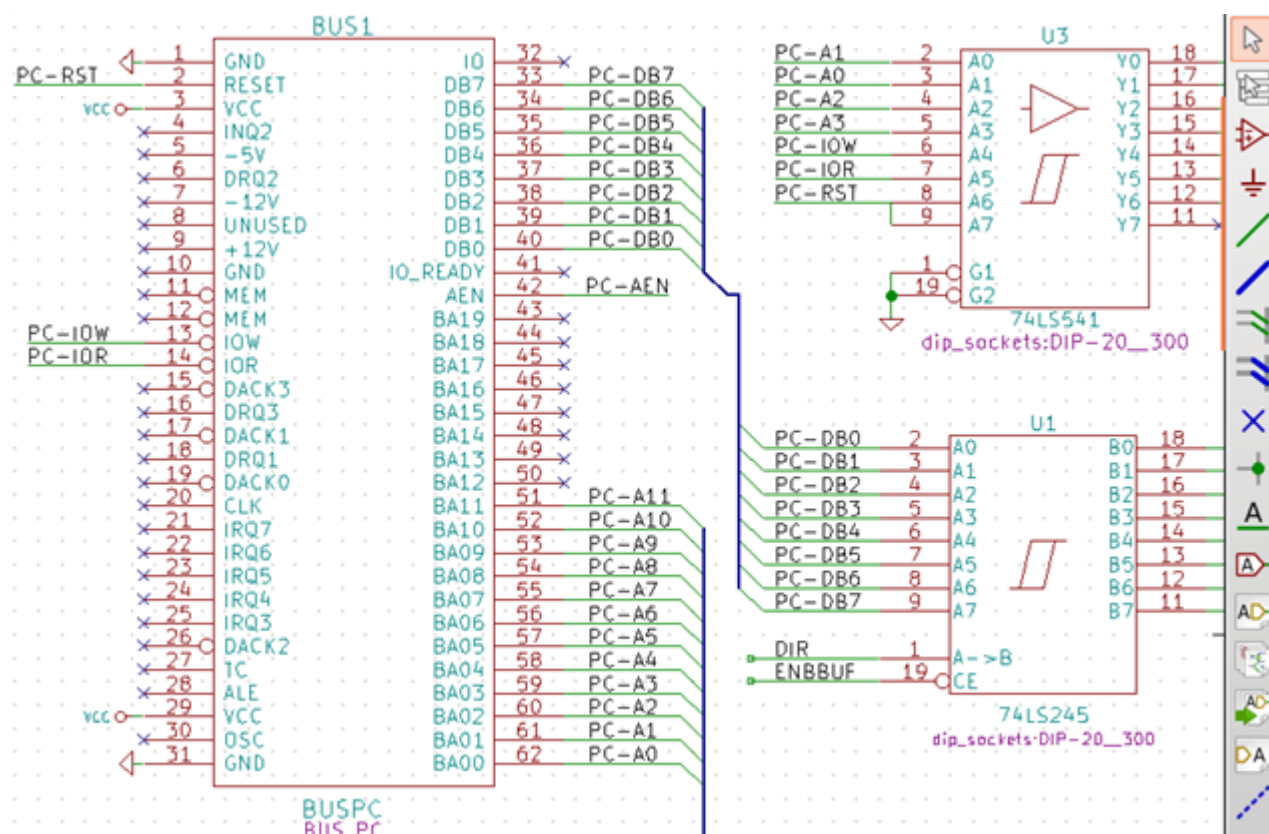
所有某些元素也可以与垂直右工具上的工具一起放置。

某些元素是：

- 符号之的大多数接口。
- 以图形方式加入。
- **折** 用于图形演示。
- **接点**：用于在交叉或分支处建立接口。
- **入口**：指示和分支的接口。

接（

在下面的原理 中， 多引脚 接到



号

是一种在原理 中 相 信号 行分 的方法，以 化 的 计。可以使用 工具将 制成 并使用与信号 相同的命名。KiCad 6.0 及更高版本中有两种 型的 矢量 和

一个 向量 是以公共前 开 并以数字 尾的信号集合。向量 命名 ‘<PREFIX> [M..N]’，其中‘PREFIX’是任何有效的信号名称，‘M’是第一个后 号，‘N’是最后一个后 号。例如， ‘DATA [0..7]’包含信号‘DATA0’，‘DATA1’，依此推，直到‘DATA7’。指定‘M’和‘N’的 序无 要，但两者都必 是非 的。

一个 是一个或多个信号和/或矢量 的集合。 可用于将相 信号捆 在一起，即使它 具有不同的名称。 使用特殊 法：

‘<OPTIONAL_NAME>{SIGNAL1 SIGNAL2 SIGNAL3}’

的成 列在由空格字符分隔的花括号（‘{}’）内。 的可 名称位于左大括号之前。如果 未命名， PCB 上生成的网 将只是 内的信号名称。如果 具有名称， 生成的网 将具有名称作 前 其中句点（‘.’）将前 与信号名称分开。

例如， ‘{SCL SDA}’有两个信号成 在网表中 些信号将是‘SCL’和‘SDA’。 “USB1 {DP DM}”将生成名 “USB1.DP”和“USB1.DM”的网 于在几个 似 路上重 使用 大 的 计，使用 种技 可以 省

可以包含矢量 例如， ‘MEMORY {A [7..0] D [7..0] OE WE}’包含矢量 和普通信号，并将 生 如“MEMORY.A7”和“MEMORY.OE”之 的网 在 PCB 上的。

可以与信号 相同的方式 制和 接，包括使用 接点在交叉 之 建立 接。与信号一 不能有多 名称 - 如果两个 冲突的 接到同一 会 生 ERC

成 之 的 接

成 之 的 接 接在 的相同成 之 的 接必 通 接。无法将引脚直接 接到 ；Eeschema 将忽略 种 型的接。

在上面的示例中， 接是通 放置在 接到引脚的 上的 行的。到 的 入口（45度 段） 是 形化的，并不是形成 接所必需的。

上，使用重 命令（*Insert* 如果元件引脚按 增 序排列， 可以通 以下方式快速建立 接（ 上在存 器，微 理器等元件上的常 情况）：

- 放置第一个 例如 PCA0)
- 尽可能多地使用重 命令来放置成 Eeschema 将自 建垂直 的下一个 PCA1, PCA2), 理 上是在其他引脚的位置上。
- 在第一个 下画 然后使用重 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 命令）。

NOTE

在“首 / ”菜 中，您可以 置重 参数：

- 垂直步
- 水平步
- 增量（因此可以 增 2,3 或 减）。

正在展开

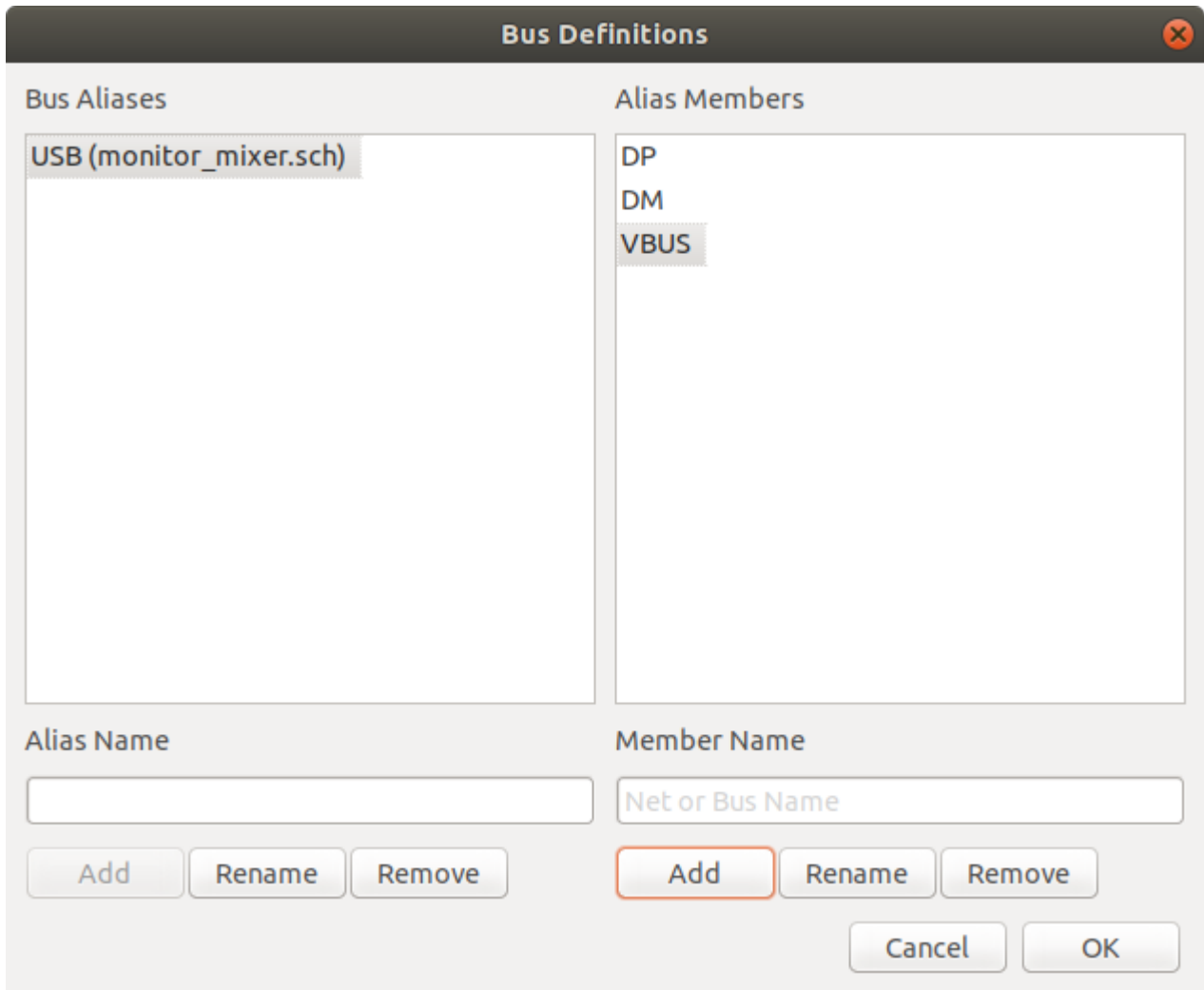
展开工具允 您快速分离来自 的信号。要展开信号，右 象（ 等）并 “展开 ”。或者，当光 位于 象上使用“展开 ” 默认：“D”）。菜 允 您 要展开的 成

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。

要 建 名， 在“工具”菜 中打开“ 定义” 框。



名可以被命名 任何有效的信号名称。使用 框，您可以向 名添加信号或矢量 作 一种快捷方式，您可以 入 或粘 由空格分隔的信号和/或 列表，并将它 全部添加到 名定义中。在 个例子中，我 定义了一个名 ‘USB’的 名，其成 “DP”，“DM”和“VBUS”。

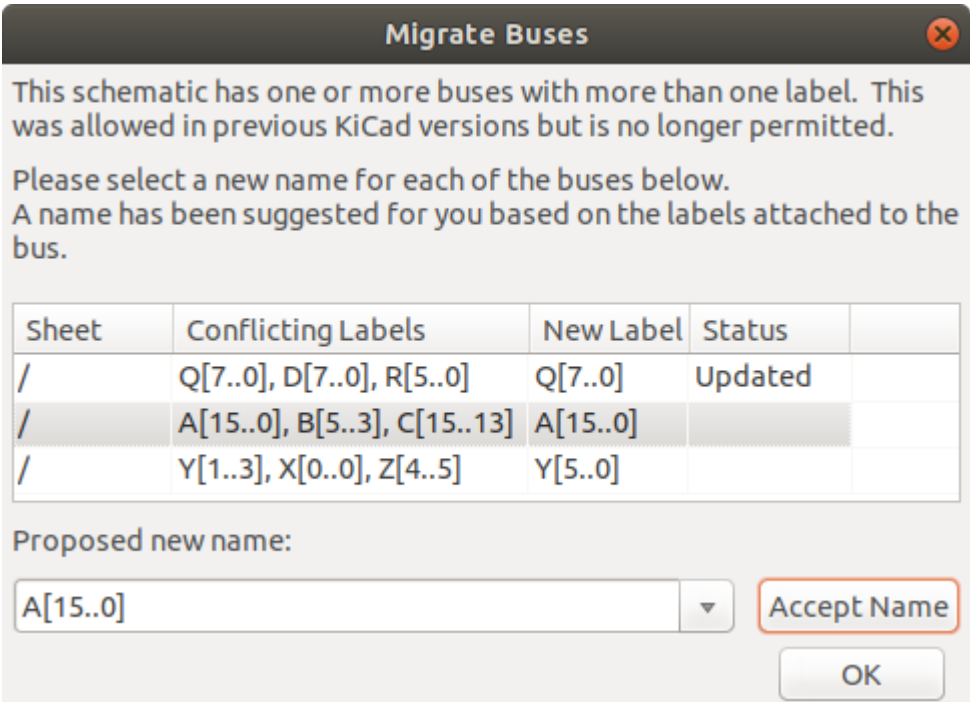
定义 名后，可以通 将 名放在 的大括号内来用于 ‘{USB}’。与 “{DP DM VBUS}”具有相同的效果。您 可以 添加前 名称，例如“USB1 {USB}” 会 生如上所述的“USB1.DP”等网 于 的 使用 名可以使原理 上的 更短。住， 名只是一个快捷方式， 名的名称不包含在网表中。

名保存在原理 文件中。在 定的原理 工作表中 建的任何 名都可用于同一 次 构 计中的任何其他原理 工作 表。

有多个 的

KiCad 5.0 及更早版本允 将具有不同 的 接在一起，并且在网 列表期 将加入 些 的成 此行 已在 KiCad 6.0 中 除，因 它与 不兼容，并且 致令人困惑的网表，因 不容易 定信号将接收的名称。

如果您在 代版本的 KiCad 中打开使用此功能的 计，您将看到“迁移 ” 框， 框将指 您更新原理 以便在任何 定的 路上只存在一个



于具有多个 的每 您必 要保留的 下拉名称框允 您在 计中存在的 之 行 或者您可以通 手 将其 入新名称字段来 其他名称。

源端口 接


当符号的 源引脚可 它 必 接，就像任何其他信号一

和触 器等符号可能有不可 的 源引脚。必 小心 些因

- 由于它 不可 你无法 接
- 你不知道他 的名字。

此外，将它 可 并将它 像其他引脚一 接将是一个坏主意，因 原理 将 得不可 并且不符合通常的 例。

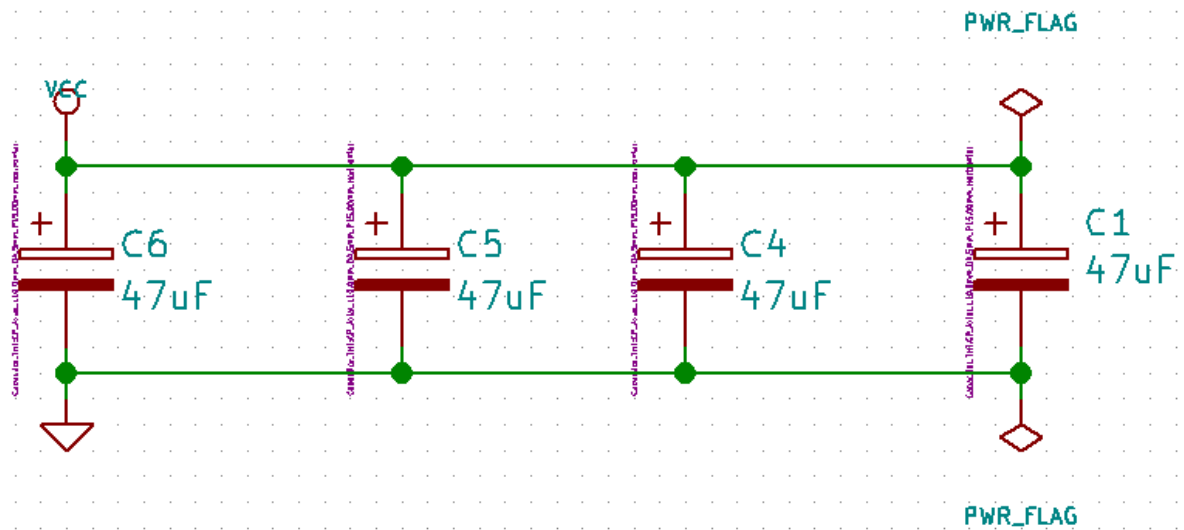
NOTE

如果要 制 示 些不可 的 源引脚，必 在主菜 的首 / 框中 中 示不可 的 源引脚 或者  左 工具 上的。

Eeschema 自 将同名的 形 源引脚 接到 名称的 源网。可能需要 接不同名称的 源网（例如，TTL 元件中的 *GND* 和 MOS 元件中的 *VSS*）；此使用 源端口。

建 不要使用 行 源 接。它 只有一个_本地_ 接范 不会 接不可 的 源引脚。

下 示了 源端口 接的示例。




在示例中，地（GND）接到源端口 VSS，源端口 VCC 接到 VDD。

可以看到两个 PWR_FLAG 符号。它表明两个源端口 VCC 和 GND 确 接到 源。如果没有 两个 志，ERC 工具将 断出：警告：源端口未通 。

所有 些符号都可以在“源”符号 中找到。



“无接”志

些符号 于避免意外的 ERC 警告非常有用。 气 确保没有 接意外地保持未 接状

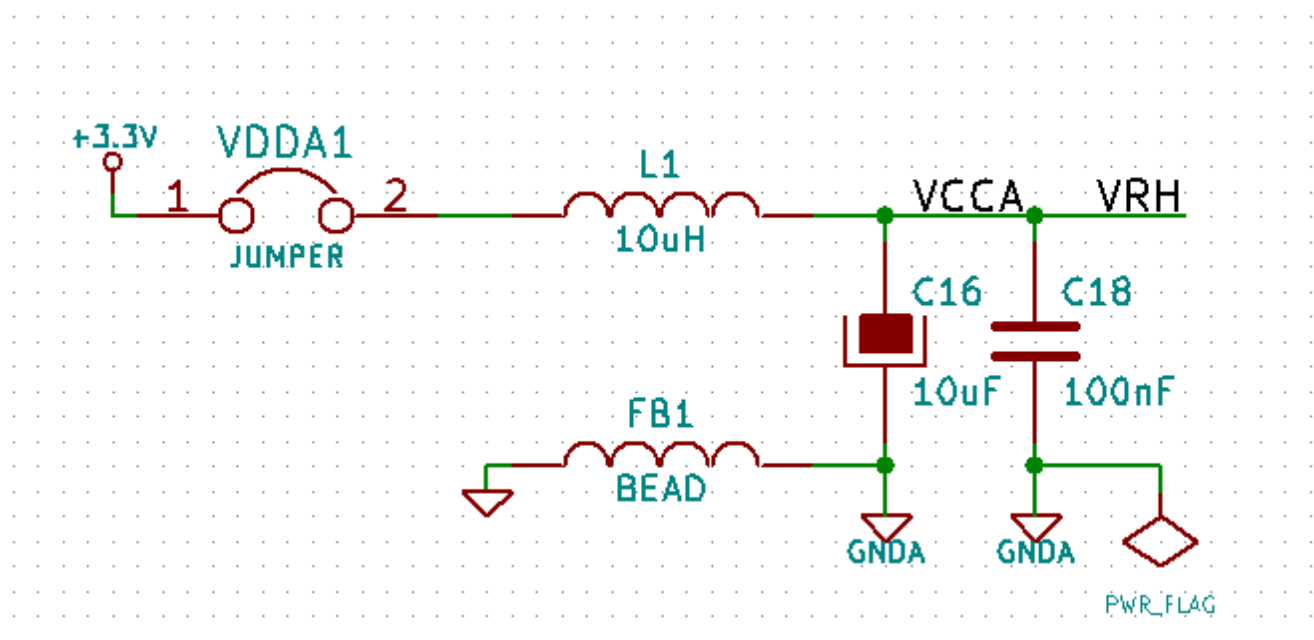
如果引脚必 保持未 接状 必 在 些引脚上放置“无接”志（工具 ）。 些符号 生成的网表没有任何影响。

充

文本注

放置批注(例如文本字段和框架)可能很有用(有助于理解原理)。文本字段(工具 )和多段（工具 )用于此用途，而 和 是 接元素。

在 里，您可以找到 有文本注 的框架示例。



表格

使用工具 像



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

47

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

工作表 号 (工作表X/Y) 会自 更新。

救 存的符号

默认情况下, Eeschema 根据 置的路径和从 目 加 符号 序。在加 非常旧的 目 可能会 致 如果 已更改或 已被 除, 或者 自使用后已不存在。在 目中, 目中的 目将自 替 新版本。新版本可能没有正确 或者可能方 向不同, 致出 破 的原理

保存 目 将包含具有当前 符号内容的 存 以及原理 允 在没有完整 的情况下分 目。如果加 其 存和系 中 存在符号的 目, Eeschema 将 描 以 找冲突。找到的任何冲突都将在以下 框中列出:

This project uses symbols that no longer match the ones in the system libraries. Using this tool, you can rescue these cached symbols into a new library.

Choose "Rescue" for any parts you would like to save from this project's cache, or press "Cancel" to allow the symbols to be updated to the new versions.

All rescued components will be renamed with a new suffix of "-RESCUE-kicad_test" to avoid naming conflicts.

Symbols with cache/library conflicts:

scue symbol	Symbol name
<input checked="" type="checkbox"/>	DIODE

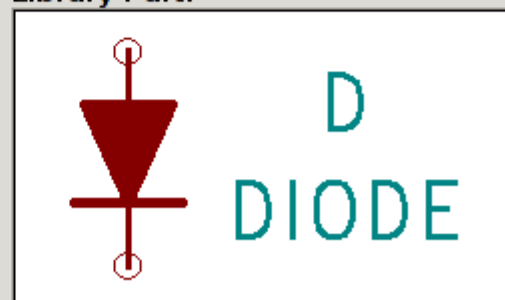
Instances of this symbol:

Reference	Value
D1	DIODE
D2	DIODE
D3	DIODE


Cached Part:



Library Part:



Never Show Again

 Cancel

 OK

您可以在此示例中看到 目最初使用的是 极朝上的二极管，但 在 中包含 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 中，并重命名所有符号以避免命名冲突。

如果按 取消 不会 行任何恢 因此 Eeschema 默认会加 所有新元件。如果此 保存原理 将覆盖 存并且旧符号将无法恢 如果已保存原理 仍可以通 “工具” 菜 中的“恢 存元件”再次 用恢 框，再次运行恢 功能。

如果您不想看到此 框，可以按 从不再 示。默认 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

于大于几 的 目，分 表示通常是一个很好的解决方案。如果要管理此 目， 需要：

- 使用大 会 致打印和 理
- 使用多个工作表， 将引 您 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 独的表格通常会提高其可 性。

从根表中，您必 能 找到所有子表。借助可通 部工具 的  的集成 次 构 航器，Eeschema 可以 松 分 原理 管理。

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

Eeschema 目前不会 理第二种情况。

次 构可以是：

- 定： 定的工作表只使用一次
- 倍数： 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。

Eeschema 可以 理所有 些 次 构。

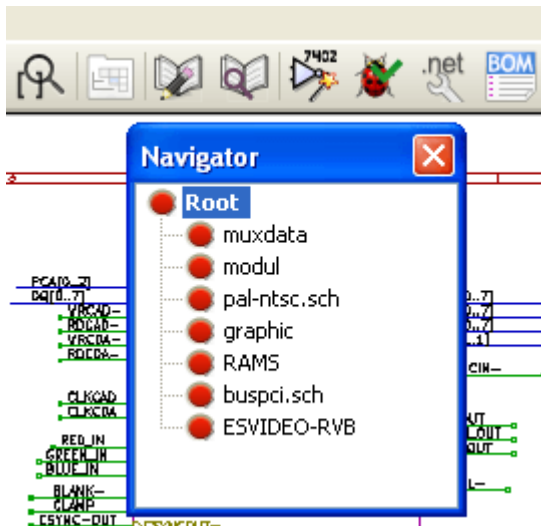
建分 原理 很容易，整个 次 构从根原理 开始 理，就像您只有一个原理 一

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

通 使用可通 部工具 上的按  的 航工具来 子表之 的 航。




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 中 “离开工作表” 或按 “Alt + Backspace”。

本地、分 和全局

属性

本地 工具  在工作表内 接信号。分 工具  在工作表内 接信号，并 接到父工作表中的分 引脚。

全局 工具  接所有 次 构中的信号。源引脚（型 源 入和 源 出）不可 就像全局 一 因 它 在所有 次 构中被 接在它 之

NOTE 在 次 构（ 或 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）。

些 似于通常的符号引脚。 工具 .

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

接-分 引脚

您将在此 建 建的符号的 接点（次 构引脚）。

些 接点 似于普通符号引脚，但是只需一个 接点就可以 接一个完整的

有两种方法可以做到 一点：

- 在 制子表之前放置不同的引脚（手 放置）。
- 在 制子表和全局 半自 放置）后放置不同的引脚。

第二种解决方案是非常 的。

手 放置：

- 工具 。
- 要放置引脚的 次 构符号。

有 建名 “接”的分 引脚的示例， 参 下文：

Sheet Pin Properties

Name:

Text height: millimeters

Text width: millimeters

Connection type:

您可以在 建期 或之后定义引脚的名称，大小和方向，方法是右 引脚并在 出菜 中 引脚。

在工作表内部，必 使用与“分 引脚”相同的名称 “分 ”。注意正确匹配 些名称必 手 完成， 就是下面第二种方法首 的原因。


自 放置：

工具 。

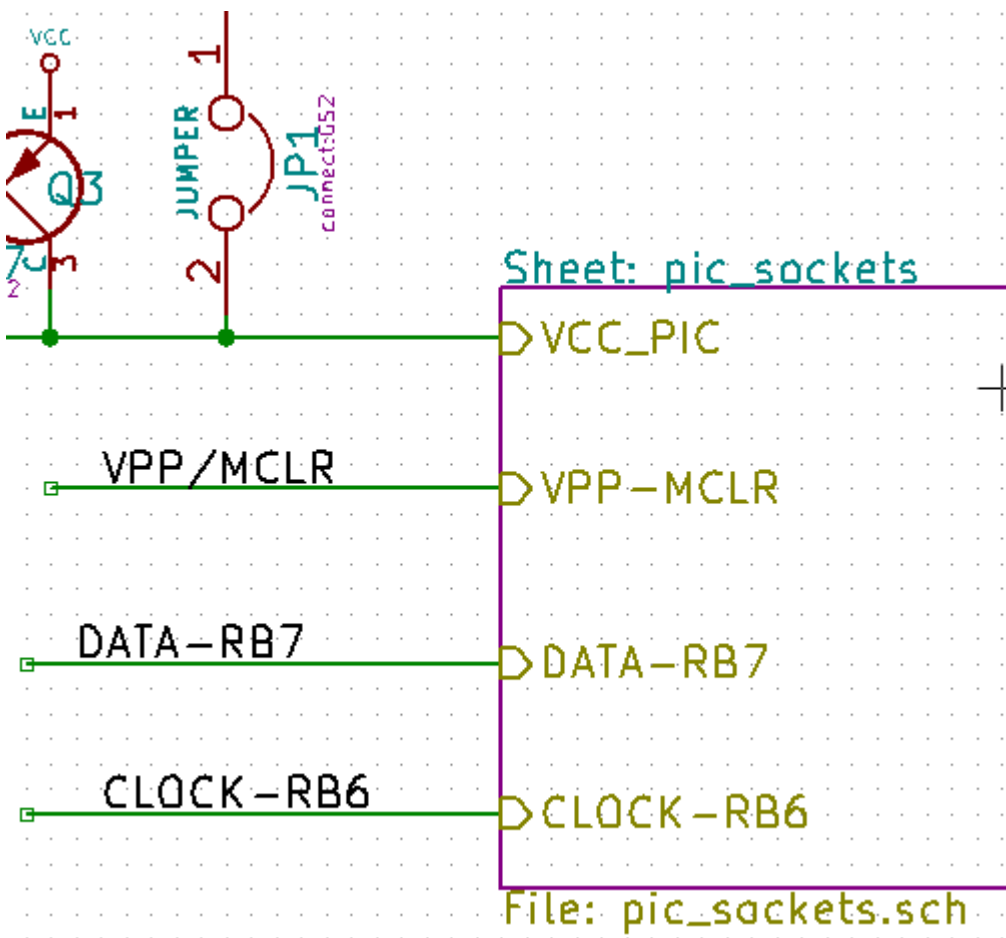
- 要从中 入与相 原理 中放置的全局 的引脚的 次 构符号。如果存在新的全局 出 分 引脚, 即不 于 已 放置的引脚。
- 要放置此引脚的位置。

因此可以快速且无 地放置所有必需的引脚。他 的方面符合相 的全局

接 - 分

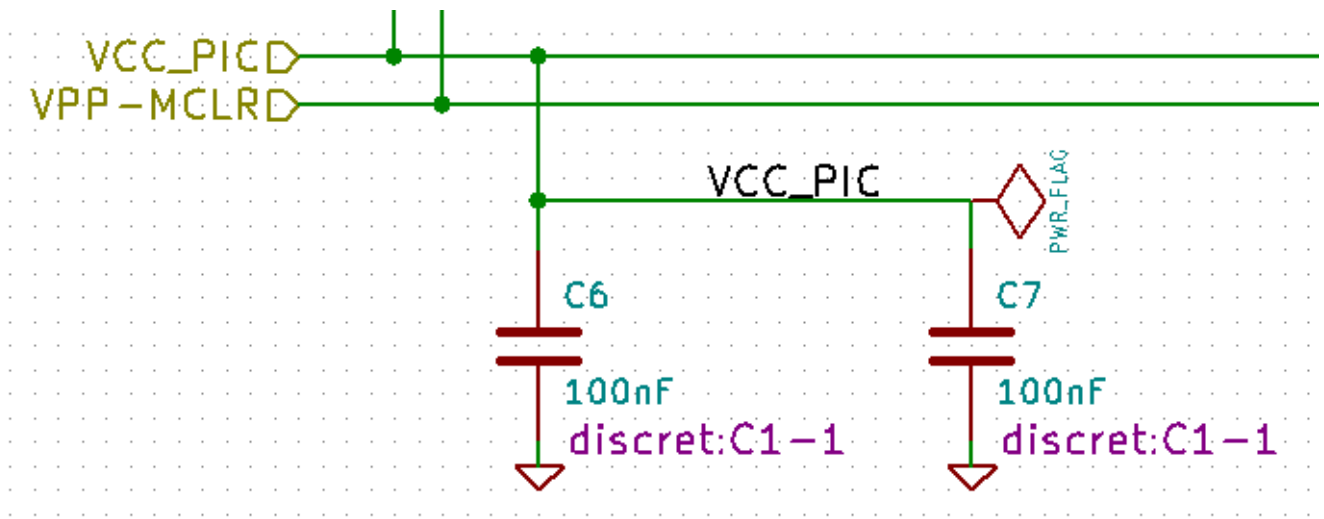
建的工作表符号的每个引脚必 于子工作表中名 分 的 分 与 似, 但它 提供子表和根表之 的 接。两个互 pin 和 HLabel) 的 形表示是 似的。使用工具 像 建分 .

参 下面的根表示例：



注意引脚 VCC_PIC 接到 接器 JP1。

以下是子表中的相 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 提供 接的各种方法的一些注 而不是有 接。

的

具有局部 接能力，即限于放置它 的示意 是因

- 每 都有一个 号。
- 此工作表 号与 相

因此，如果在 n°3 中放置 “TOTO” 上真正的 是 “TOTO_3”。如果您 在工作表 n°1 (根表) 中放置了 “TOTO” 上放置一个名 “TOTO_1” 的 与 “TOTO_3” 不同。即使只有一 也是如此。

分

于 而言， 于分 也是如此。

因此，在同一 中，分 “TOTO” 被认 接到本地 “TOTO”，但没有 接到另一 中称 “TOTO” 的分 或 分 被认 接到放置在父表中的分 符号中的相 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 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 接在它 所放置的工作表内。

意味着如果将 VCC 放在子表中，它将不会 接到 VCC 引脚，因 上是 VCC_n，其中 n 是表 号。

如果您希望此 VCC 真正 接到整个原理 的 VCC 必 通 VCC 源符号将其明确 接到不可 的 源引脚。

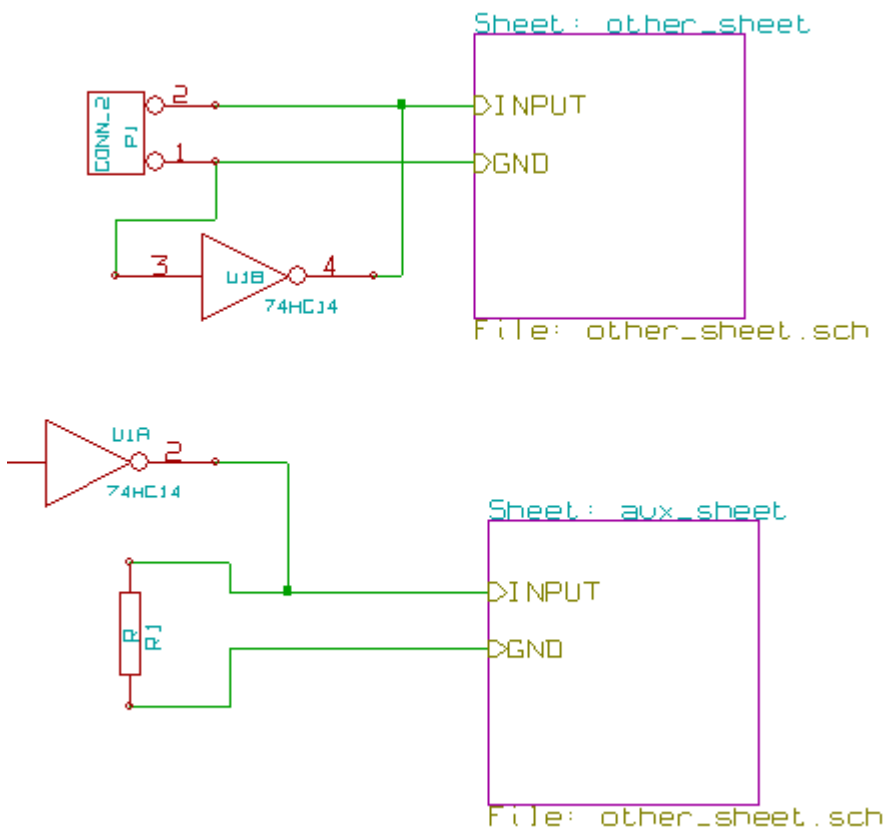
全局

具有相同名称的全局 跨整个 次 构 接。

(像vcc 的 力是全局

次 构

是一个例子。相同的原理 使用两次（两个 例）。 两个工作表共享相同的原理 因 两个工作表的文件名相同（*other_sheet.sch*）。工作表名称必 是唯一的。



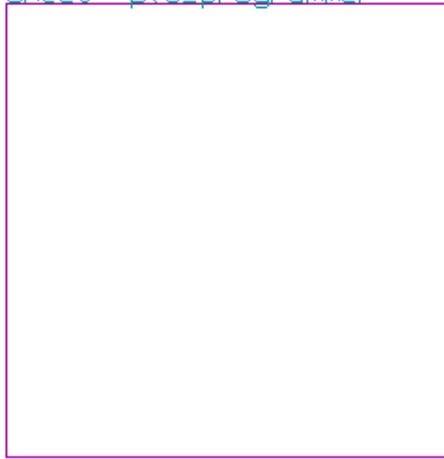
平面 次 构

如果遵守以下 可以使用多个工作表 建 目，而无需在 些工作表（平面 次 构）之 建 接：

- 建包含其他工作表的根工作表， 些工作表充当其他工作表之 的 接。
- 不需要明确的 接。
- 在所有工作表中使用全局 而不是分

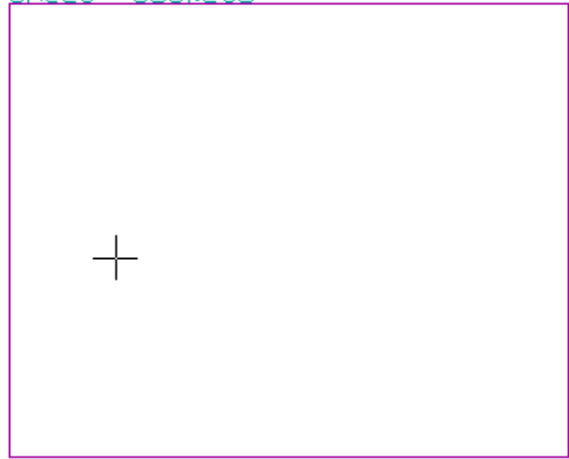
以下是根表的示例。

Sheet: pic_programmer



File: pic_programmer.sch

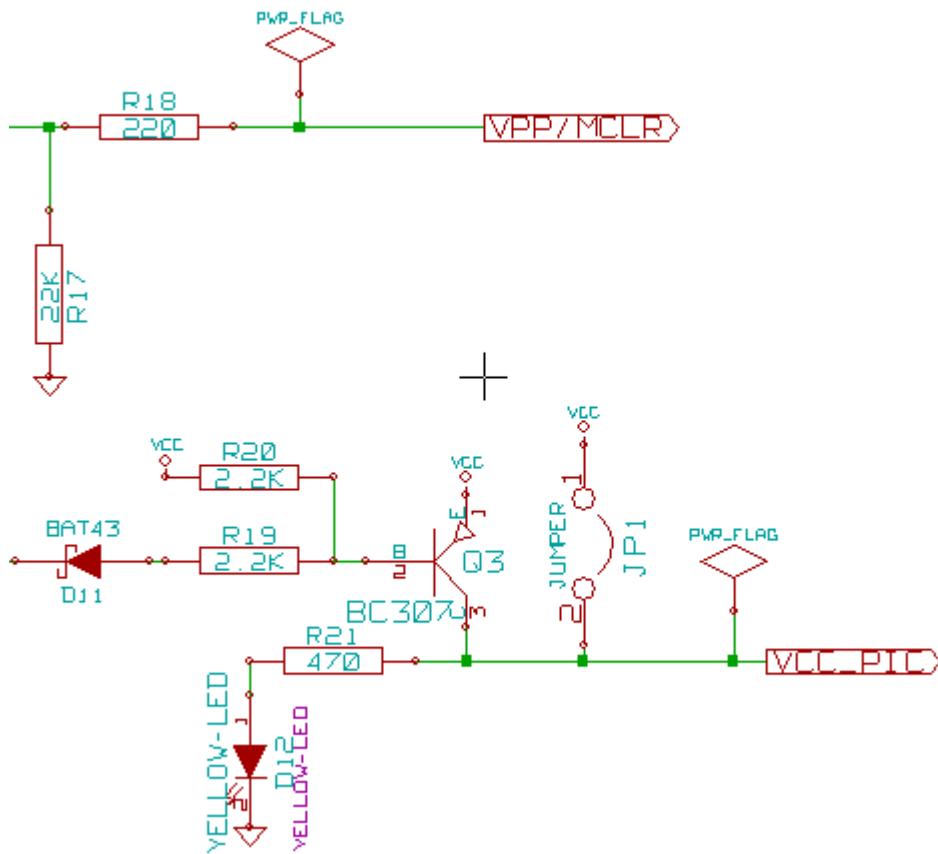
Sheet: sockets



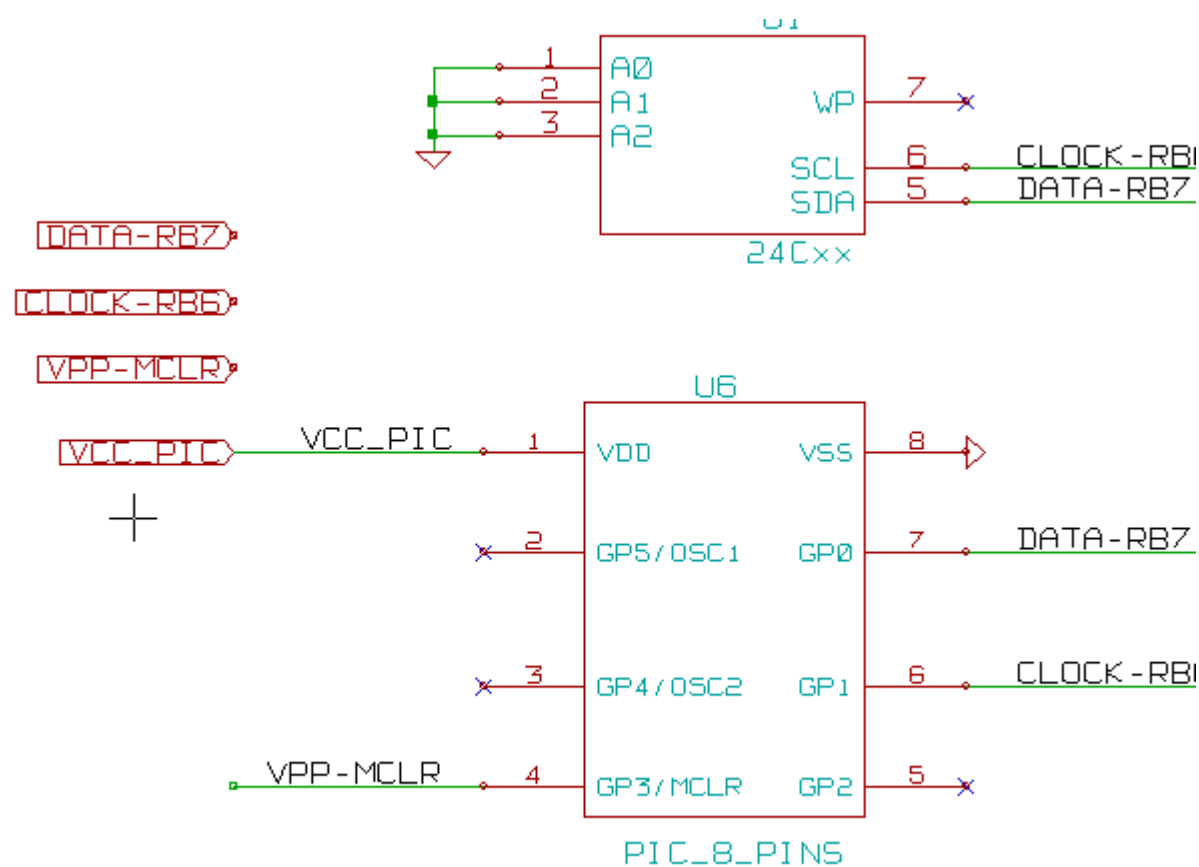
File: pic_sockets.sch

是两 由全局 接。

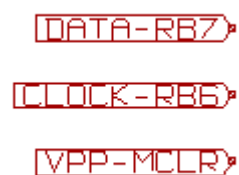
是pic_programmer.sch。



是pic_sockets.sch。



看全局



符号批注工具

介

批注工具允 您自 原理 中的符号指定一个指示符。具有多个 位的符号的批注将分配一个唯一的后 ,以最大限度地减少 些符号的数量。批注批注工具可通 像 :image:images/icons/annotate.png [icons_annotate_png]。在 里,你可以找到它的主窗口。

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 有批注)
- 批注所有符号,但不要交 任何以前批注的多 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 尾的指示符。
- 批注整个 次 构 (使用整个原理)。
- 批注当前工作表 (使用当前)。

“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 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。

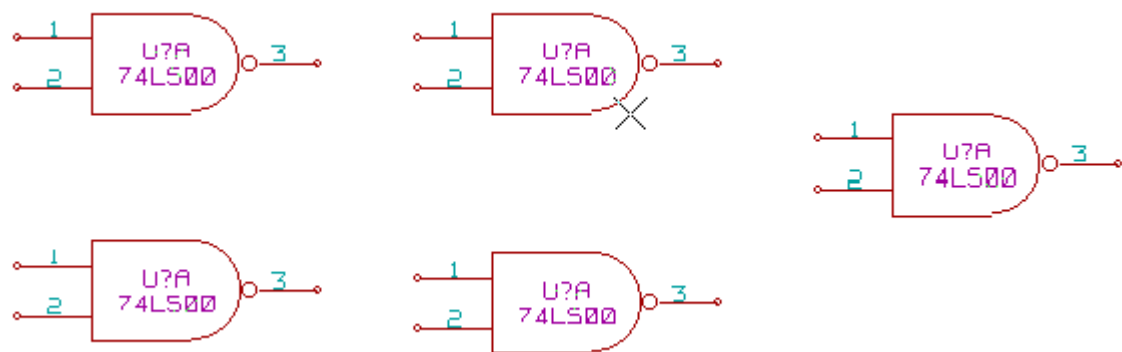
“批注 ” 出了用于计算参考的方法:

- 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

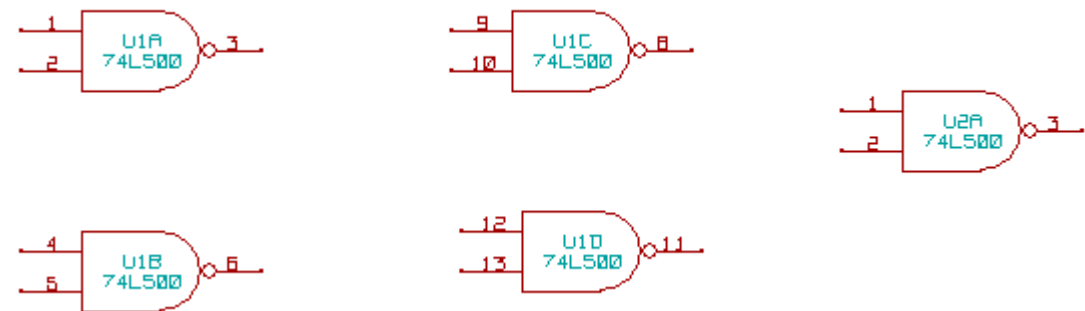
批注 序

此示例 示放置了 5 个元素, 但未批注。

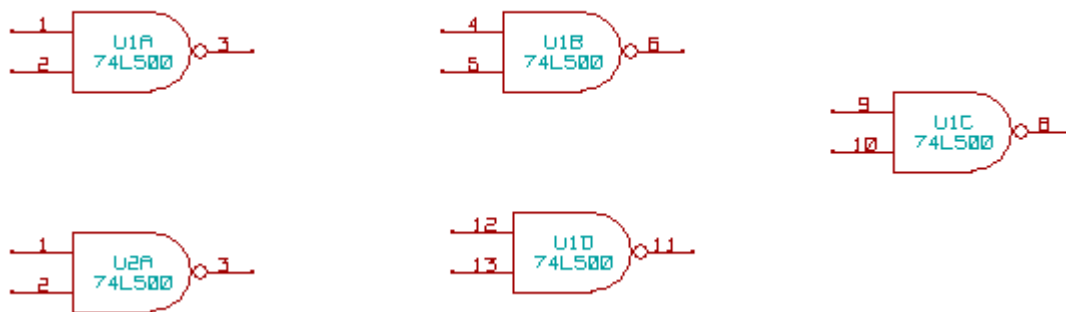


行批注工具后, 得以下 果。

按 X 位置排序。



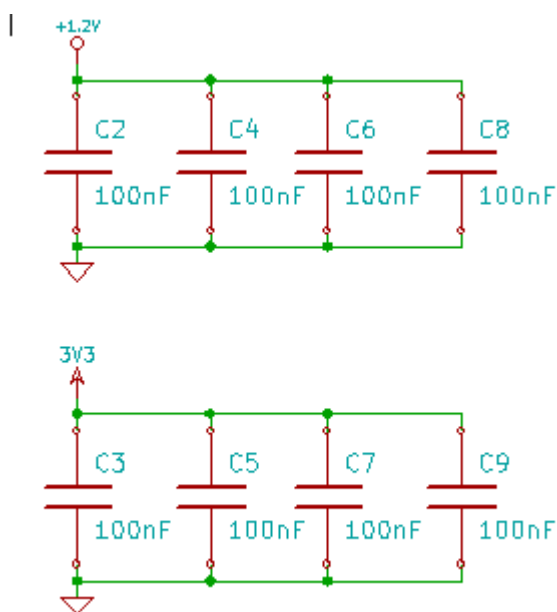
按 Y 位置排序。



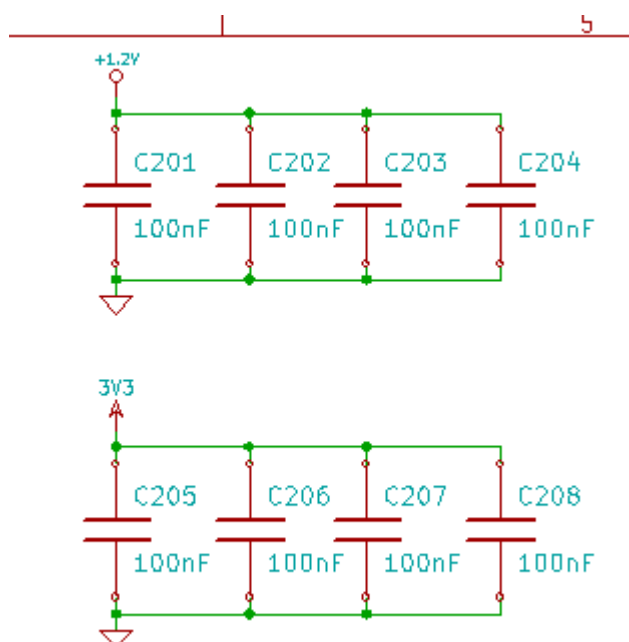
您可以看到四个 74LS00 分布在 U1 包中，第五个 74LS00 已分配 下一个 U2。

批注

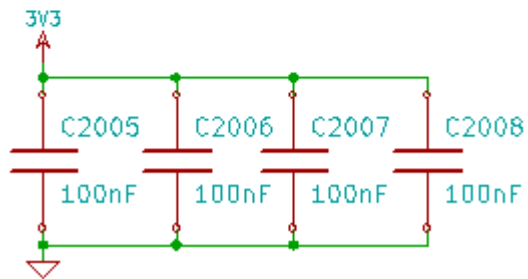
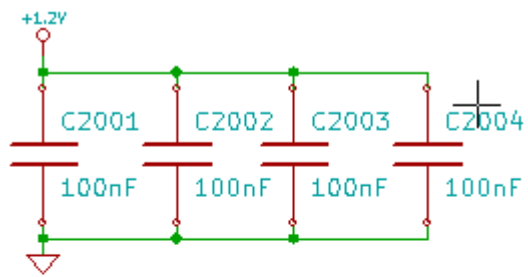
是表 2 中的批注，其中 使用原理 中的第一个空 号。



开始到工作表 号 *100并使用第一个空 号 出以下 果。



开始到工作表 号 *1000并使用第一个空 号 出以下 果。

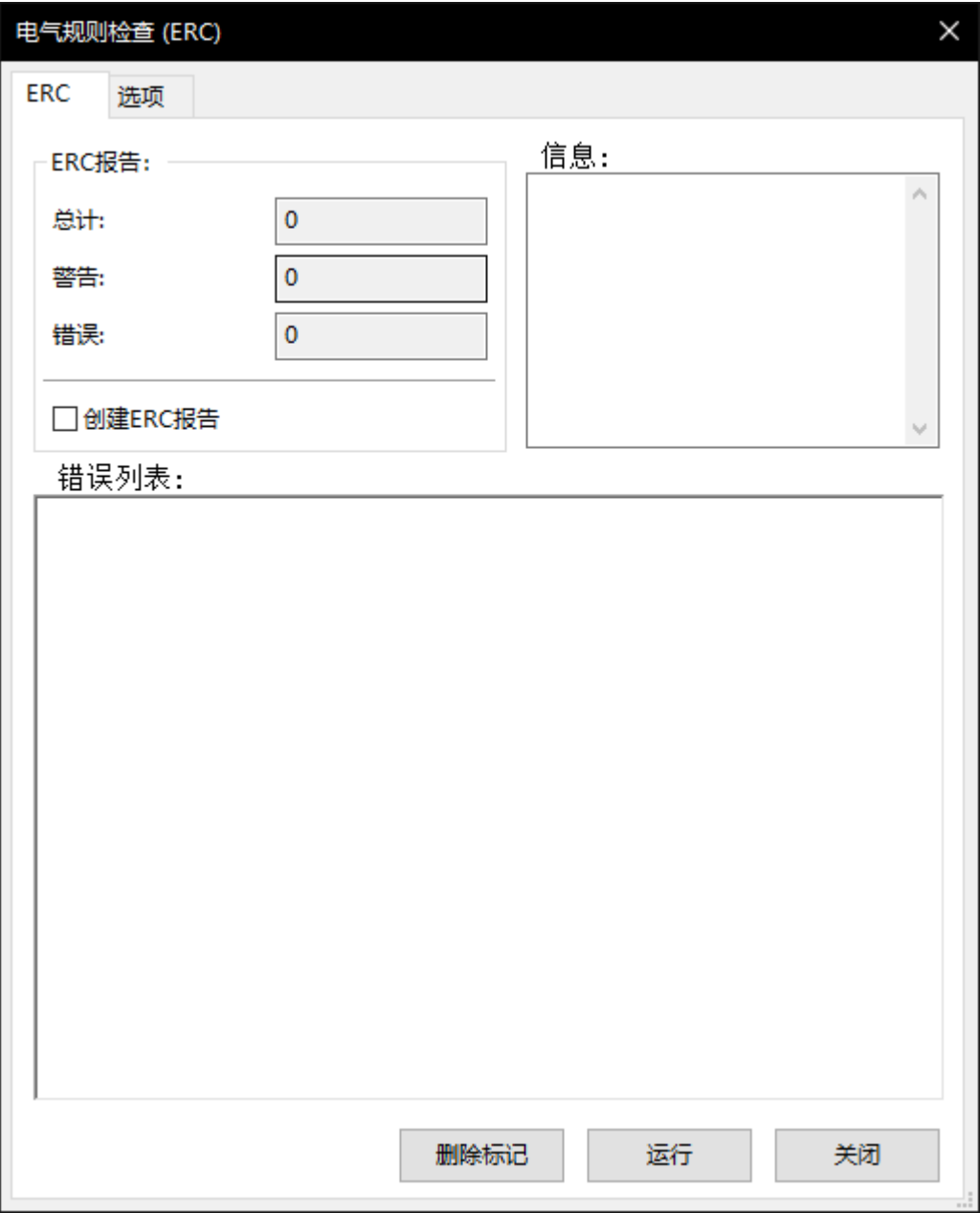


使用 气 行 计

介

气 (ERC) 工具会自 原理 ERC 工作表中的任何 例如未 接的引脚, 未 接的分 符号, 短路 出等。当然, 自 不是 可靠的, 并且可以 所有 计 的 件 不是100% 完成。 的 非常有用, 因 它允 您 多疏忽和小

上, 必 所有 到的 然后在正常 行之前 行 正。ERC 的 量与在符号 建期 声明 引脚属性 所采取的 慎直接相 ERC 出 告 或 警告。



如何使用 ERC

像可以后 ERC : images/icons/erc.png[ERC icon] 。

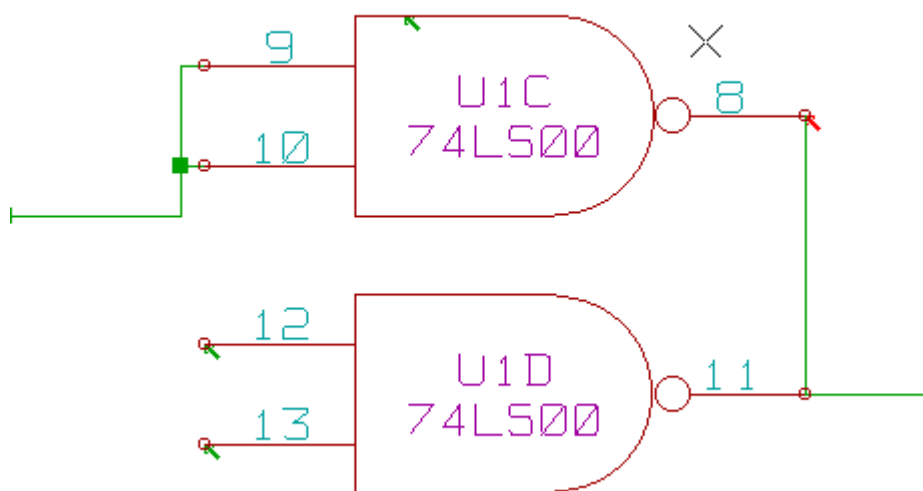
在原理 元素上 置警告, 以引 ERC (引脚或)。

NOTE

- 在此 框窗口中, 消息 您可以跳 到原理 中的相
- 在原理 中, 右 以 相 的 断消息。

您 可以从 框中 除

ERC 的示例

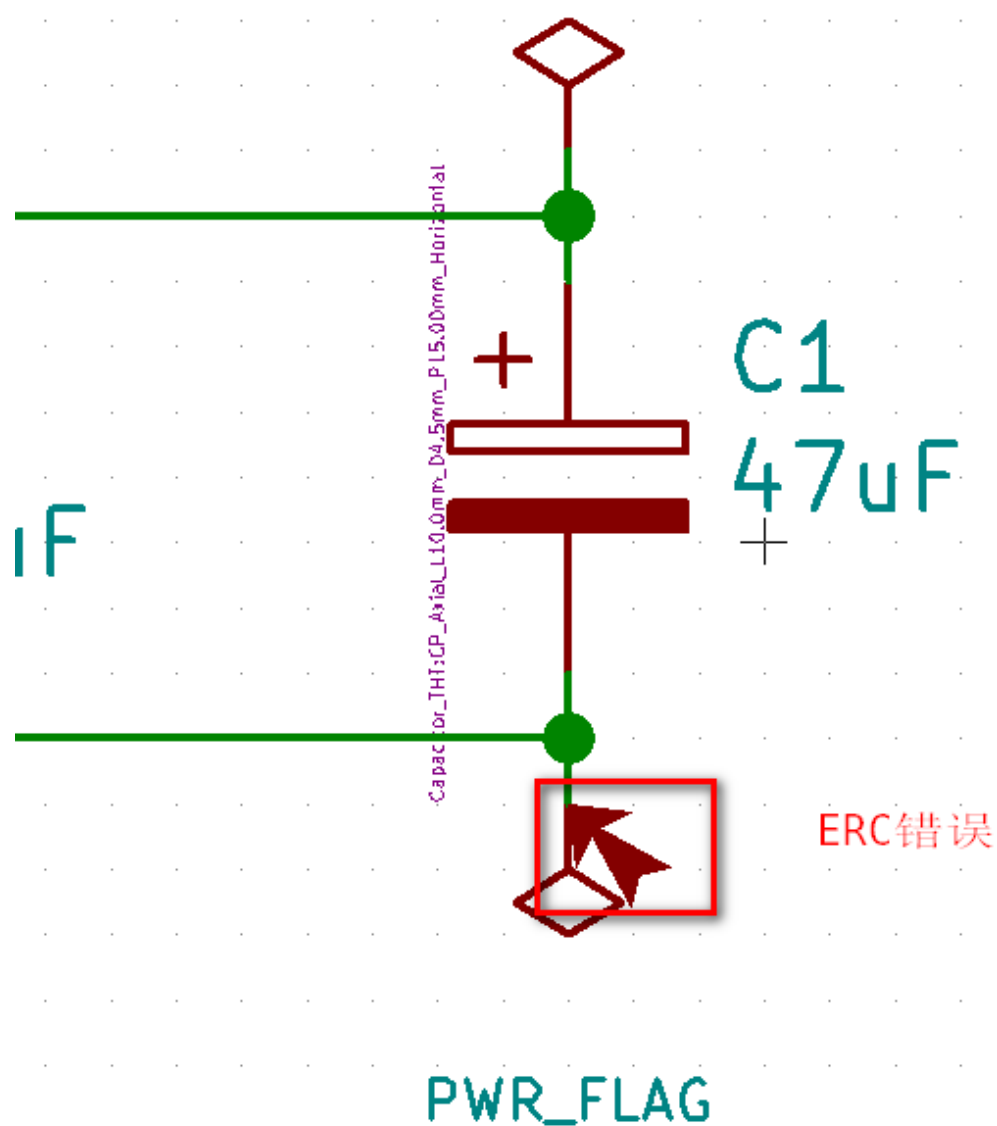


在 里, 您可以看到四个 :

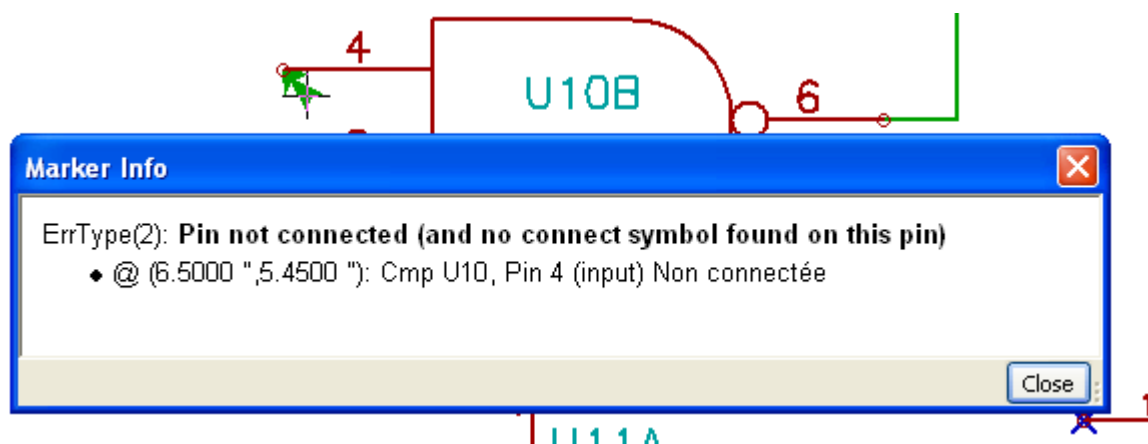
- 两个 出 地 接在一起 (色箭
- 两个 入未 接 (色箭
- 藏 源端口出 缺少 源 志 (部 色箭

示 断

通 右 出菜 允 您 ERC 断窗口。



当 信息 您可以 得 的描述。

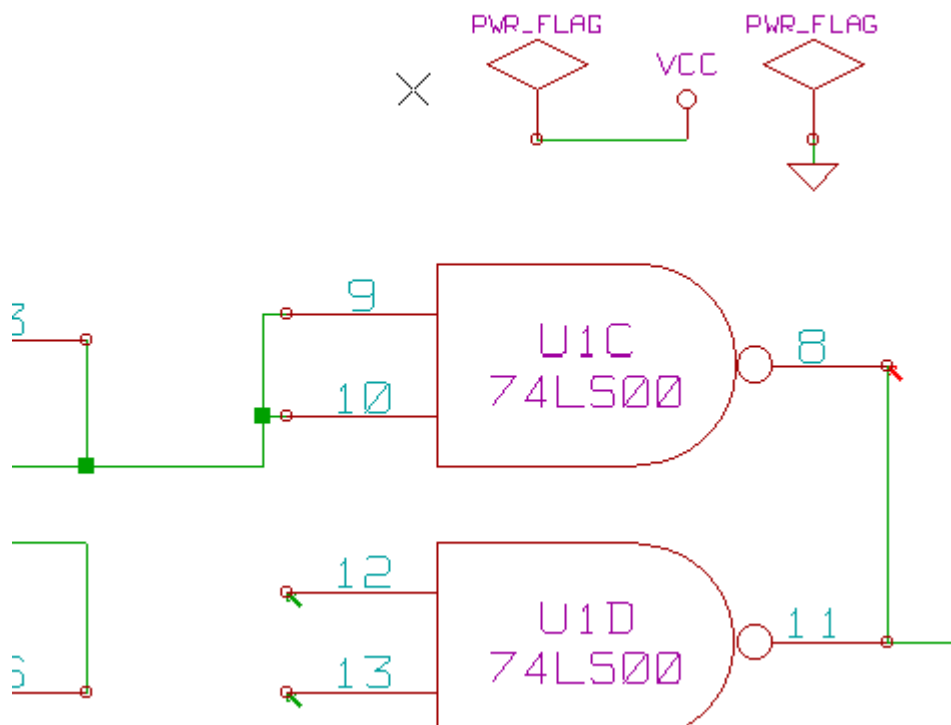


源引脚和 源 志

通常在 源引脚上出 或警告，即使一切看起来都很正常。 上面的例子。之所以会 生 种情况，是因 在大多数计中， 源是由不是 源的 接器提供的（如 器 出，它被声明 源 出）。

因此，ERC 不会 到任何 源 出引脚来控制 并声明它 不是由 源 的。

要避免此警告，您必 在此 源端口上放置“PWR_FLAG”。看一下下面的例子：

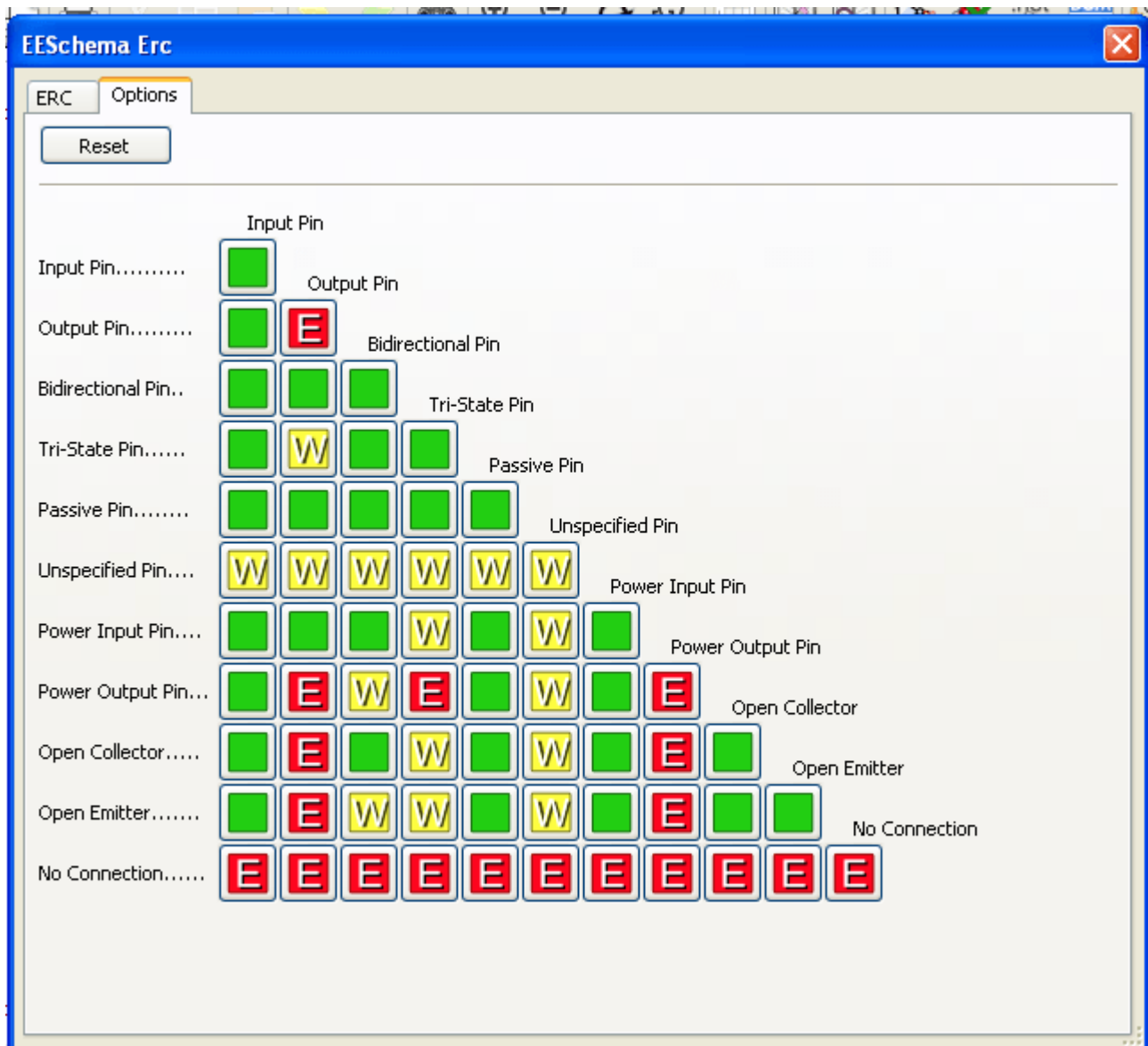


然后 将消失。

大多数情况下，PWR_FLAG 必 接到 GND，因 器的 出声明 断 但接地引脚永 不会断 正常属性是 源 入），因此，在没有 源 志的情况下，接地不会与 源相

配置

面板允 您配置 接 以定义 和警告 的 气条件。



所需的 元格方 可以更改 使其循 正常，警告，

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 展名 .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
```

建网 列表

概述

网表是描述符号之 的 接的文件。 些 接称 网 在网表文件中，您可以找到：

- 符号列表
- 符号之 的 接(网) 列表。


存在 多不同的网表格式。有 符号列表和网 列表是两个 独的文件。 网表是使用原理 捕 件的基 因 网表是与其他 子CAD 件的 接，例如：

- PCB 布局 件。
- 原理 和 信号模 器。
- CPLD（和其他可 程 IC 器。

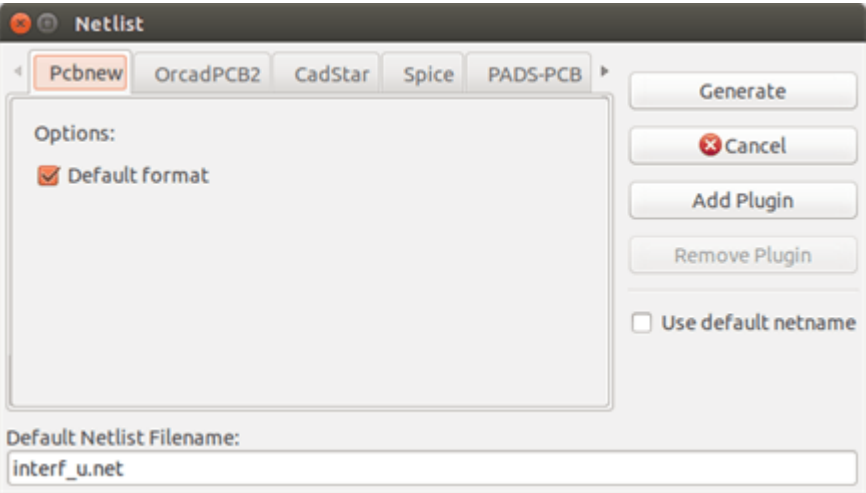
Eeschema 支持几种网 列表格式。

- PCBNEW 格式（印刷 路）。
- ORCAD PCB2 格式（印刷 路）。
- CADSTAR 格式（印刷 路）。
- Spice 格式，适用于各种模 器（其他模 器也使用 Spice 格式）。

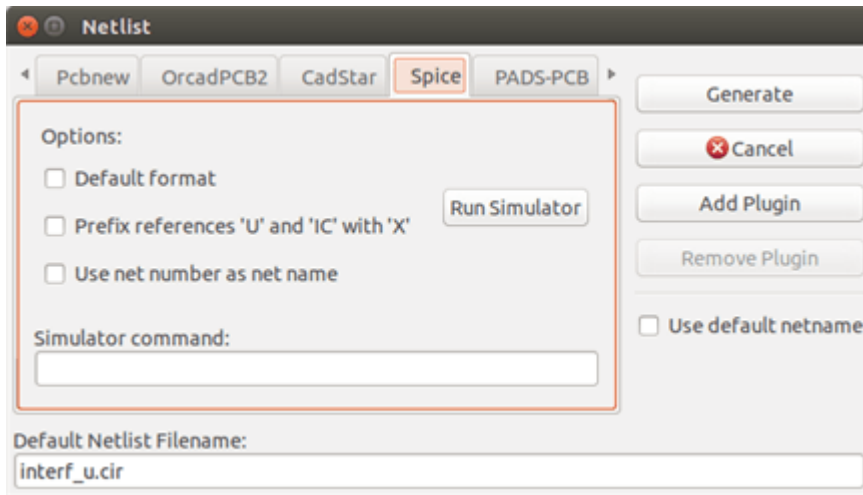
网表格式

工具  以打开网表 建 框。

的 Pcbnew



的 Spice

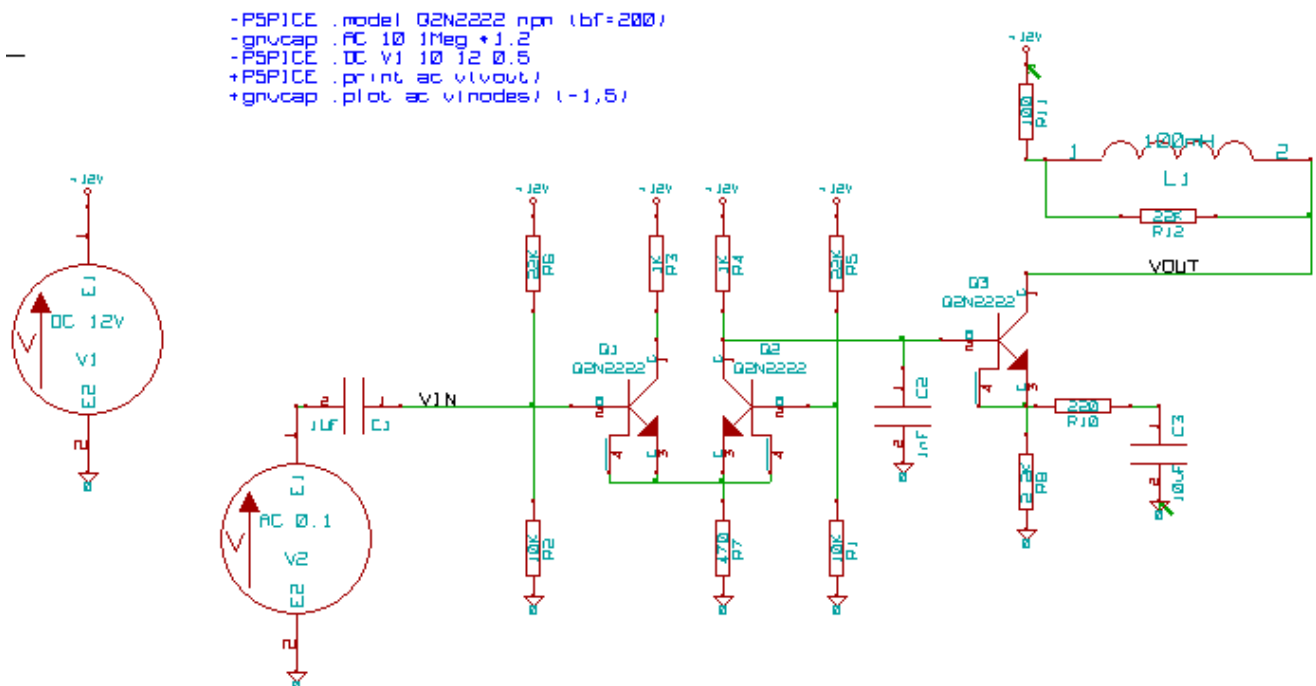


使用不同的 卡，您可以 所需的格式。在 Spice 格式中，您可以使用网 名称生成网表， 使得 SPICE 文件更易于 或者使用旧版 Spice 使用的网 号。通 “网表”按 将要求您提供网表文件名。

NOTE 于大型原理，网表生成最多可能需要几分

网表示例

您可以在下面看到使用 PSPICE 的原理 计：



PCBNEW 网表文件示例：


```

# Eeschema Netlist Version 1.0 genereee le 21/1/1997-16:51:15
(
(32E35B76 $noname C2 1NF {Lib=C}
(1 0)
(2 VOUT_1)
)
(32CFC454 $noname V2 AC_0.1 {Lib=VSOURCE}
(1 N-000003)
(2 0)
)
(32CFC413 $noname C1 1UF {Lib=C}
(1 INPUT_1)
(2 N-000003)
)
(32CFC337 $noname V1 DC_12V {Lib=VSOURCE}
(1 +12V)
(2 0)
)
(32CFC293 $noname R2 10K {Lib=R}
(1 INPUT_1)
(2 0)
)
(32CFC288 $noname R6 22K {Lib=R}
(1 +12V)
(2 INPUT_1)
)
(32CFC27F $noname R5 22K {Lib=R}
(1 +12V)
(2 N-000008)
)
(32CFC277 $noname R1 10K {Lib=R}
(1 N-000008)
(2 0)
)
(32CFC25A $noname R7 470 {Lib=R}
(1 EMET_1)
(2 0)
)
(32CFC254 $noname R4 1K {Lib=R}
(1 +12V)
(2 VOUT_1)
)
(32CFC24C $noname R3 1K {Lib=R}
(1 +12V)
(2 N-000006)
)
(32CFC230 $noname Q2 Q2N2222 {Lib=NPN}
(1 VOUT_1)
(2 N-000008)
(3 EMET_1)
)
(32CFC227 $noname Q1 Q2N2222 {Lib=NPN}
(1 N-000006)
(2 INPUT_1)
(3 EMET_1)
)
)
# End

```

在 PSPICE 格式中, 网表如下 :

```
* Eeschema 网表 1.1 版(Spice 格式) 建日期 : 18/6/2008-08:38:03

.model Q2N2222 npn (bf=200)
.AC 10 1Meg \*1.2
.DC V1 10 12 0.5

R12 /VOUT N-000003 22K
R11 +12V N-000003 100
L1 N-000003 /VOUT 100mH
R10 N-000005 N-000004 220
C3 N-000005 0 10uF
C2 N-000009 0 1nF
R8 N-000004 0 2.2K
Q3 /VOUT N-000009 N-000004 N-000004 Q2N2222
V2 N-000008 0 AC 0.1
C1 /VIN N-000008 1uF
V1 +12V 0 DC 12V
R2 /VIN 0 10K
R6 +12V /VIN 22K
R5 +12V N-000012 22K
R1 N-000012 0 10K
R7 N-000007 0 470
R4 +12V N-000009 1K
R3 +12V N-000010 1K
Q2 N-000009 N-000012 N-000007 N-000007 Q2N2222
Q1 N-000010 /VIN N-000007 N-000007 Q2N2222

.print ac v(vout)
.plot ac v(nodes) (-1,5)

.end
```

于网表的 明

网表名称注意事

多使用网表的 件工具不接受元件名称、引脚、网 或其他信息中的空格。避免在 中使用空格, 或元件或其引脚的名称和 字段, 以确保最大程度的兼容性。

同 字母和数字以外的特殊字符也可能 致 注意, 此限制与 Eeschema 无 而是与网表格式无 后者可以 不可翻 使用网表文件的 件。

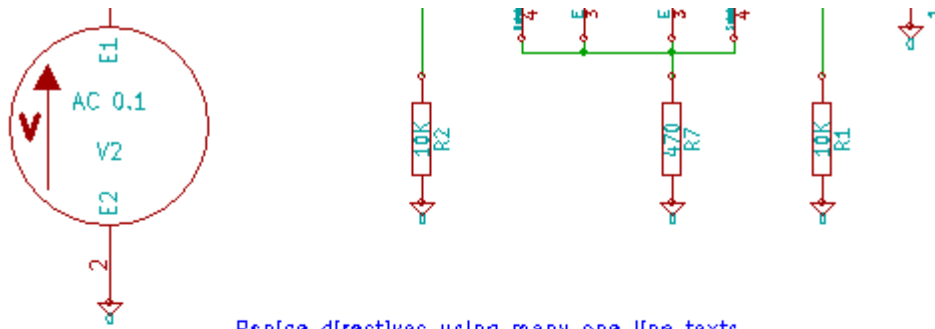
PSPICE 网表

于 Pspice 模 器, 您必 在网表本身 (.PROBE, .AC 等) 中包含一些命令行。

从 字 **-pspice** 或 **-gnuap** 开始的示意 中包含的任何文本行都将在网表的 部插入 (不 字)。

以 字 **+pspice** 或 **+gnuap** 开 的示意 中包含的任何文本行都将在网表的末尾插入 (不 字)。

下面是一个使用 多 行文本和一个多行文本的示例:



Pspice directives using many one line texts

```
-PSPICE .model Q2N2222 npn (bf=200)
-gnucap .AC dec 10 1Meg *1.2
-PSPICE .DC V1 10 12 0.5
+PSPICE .print ac v(vout)
+gnucap .plot ac v(nodes) (-1.5)
```

Pspice directives using one multiline text:

```
+PSPICE .model NPN NPN
.model PNP PNP
.lib C:\Program Files\LTC\LTspiceIV\lib\cmp\standard.bjt
.backanno
```

例如，如果您 入以下文本（不要使用

```
-PSPICE .PROBE
```

一行 .PROBE 将被插入网表中。

在前面的例子中，在网表的开 插入了三行，用 种技 插入了两行。

如果您使用的是多行文本， 只需要 **+pspice** 或 **+gnucap** 字一次：

```
+PSPICE .model NPN NPN
.model PNP PNP
.lib C:\Program Files\LTC\LTspiceIV\lib\cmp\standard.bjt
.backanno
```

建四行:

```
.model NPN NPN
.model PNP PNP
.lib C:\Program Files\LTC\LTspiceIV\lib\cmp\standard.bjt
.backanno
```

另 注意，Pspice 的 GND 网 必 命名 0（零）。

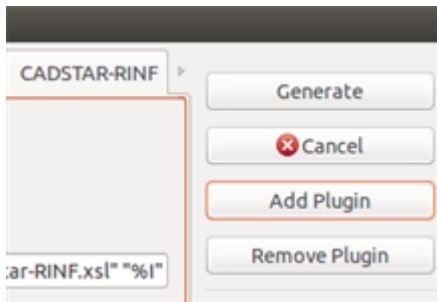
其他格式

于其他网表格式，您可以以插件的形式添加网表器。 些器由 Eeschema 自启 第14章 出了 器的一些解和示例。

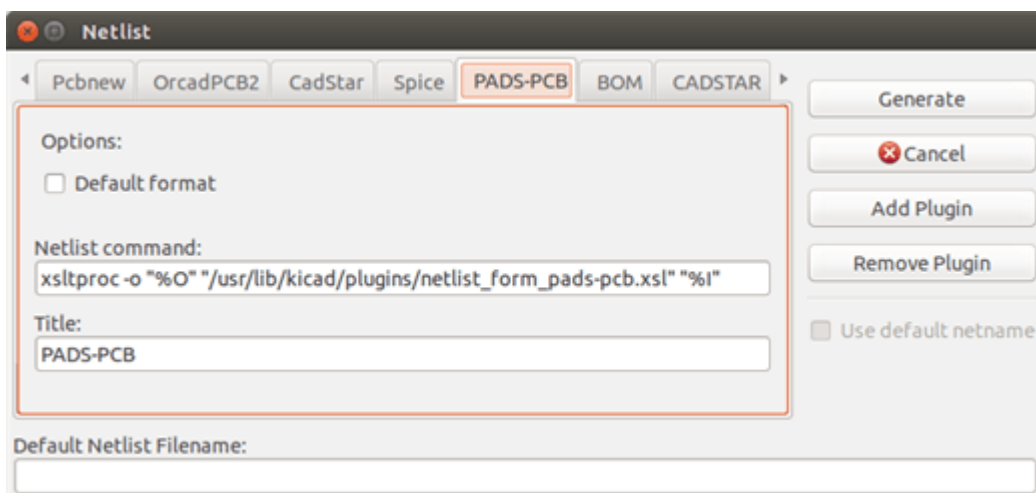
器是一个文本文件（xsl 格式），但可以使用其他言，如 Python。使用 xsl 格式 工具（xsltproc.exe 或 xsltproc 取 Eeschema 建的中 文件和 器文件以 建 出文件。在 种情况下， 器文件（工作表 式）非常小并且非常容易 写。

在 框窗口中

您可以通 添加插件按 添加新的网表插件。



是插件 PadsPcb 置窗口：



置将需要：

- 例如，网表格式的名称）。
- 要启 的插件。

生成网表

1. Eeschema 建一个中 文件 *.tmp，例如 test.tmp。
2. Eeschema 运行插件， 取 test.tmp 并 建 test.net。

命令行格式

下面是一个示例，使用 xsltproc.exe 作 .xsl 文件的工具，使用文件 netlist_form_pads-pcb.xml 作 表 式：

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

附：

f:/kicad/bin/xsltproc.exe	A tool to read and convert xsl file
-o %O.net	Output file: %O will define the output file.
f:/kicad/bin/plugins/netlist_form_pads-pcb.xsl	File name converter (a sheet style, xsl format).
%I	Will be replaced by the intermediate file created by Eeschema (*.tmp).

于名 test.sch 的原理 的命令行是：

```
f:/kicad/bin/xsltproc.exe -o test.net f:/kicad/bin/plugins/netlist_form_pads-pcb.xsl test.tmp.
```

器和工作表 式 (插件)

是一个非常 的 件，因 它的目的只是将 入文本文件（中 文本文件） 另一个文本文件。此外，从中 文本文件中，您可以 建 BOM 清

使用 xsltproc 作 器工具 生成工作表 式。

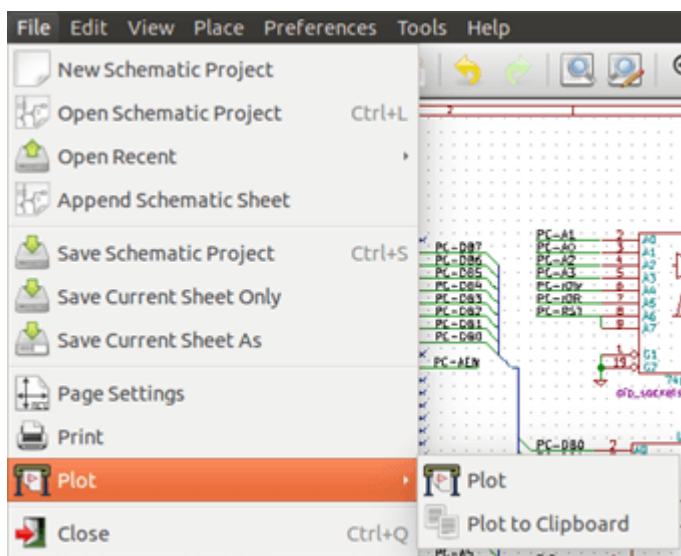
中 网表文件格式

有 xslproc 的更多 明，中 文件格式的 明以及 器的工作表 式的一些示例， 参 第14章。

和打印

介

您可以通过 文件菜单 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

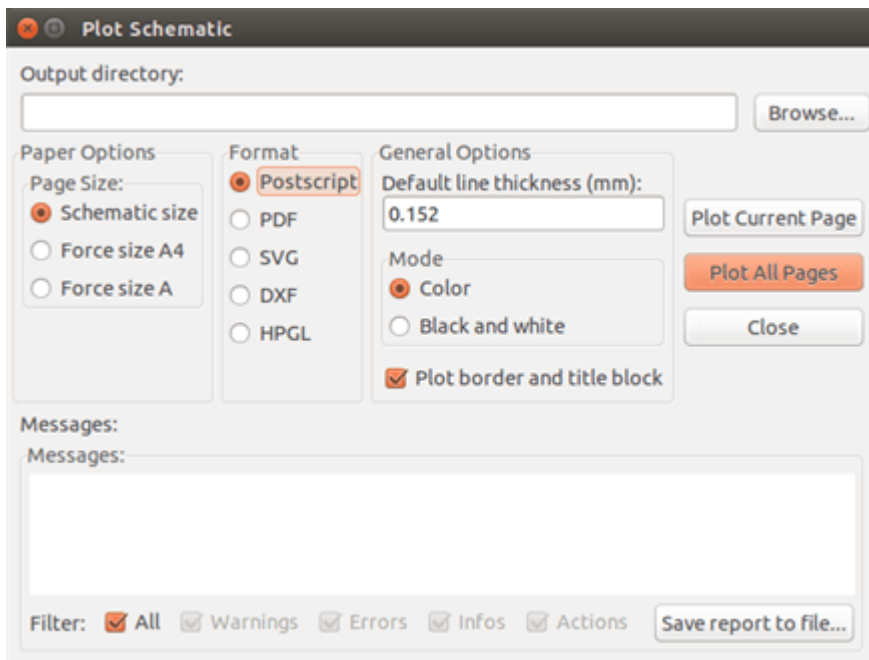
打印当前工作表的一个文件。

制所有 面

允 您 制整个 次 构 (每个工作表生成一个打印文件)。

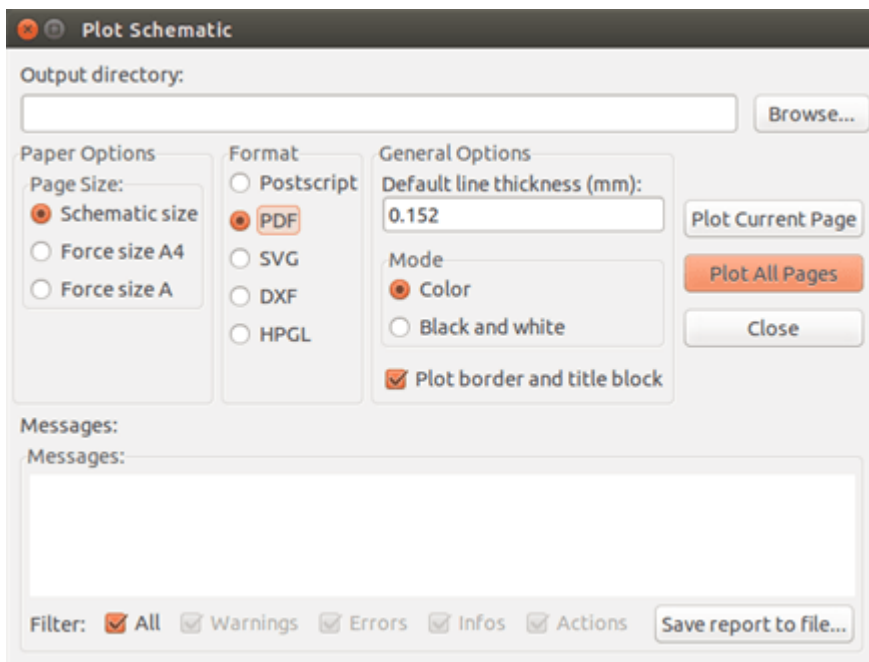
在 Postscript 中 制

此命令允 您 建 PostScript 文件。



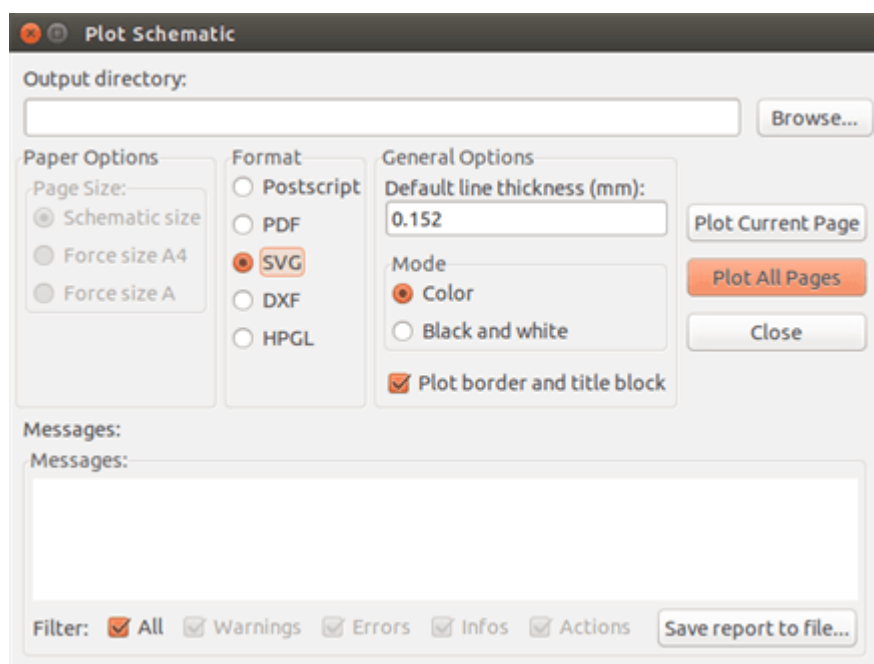
文件名是 展名 .ps 的工作表名称。您可以禁用”制 框和 “ 如果要 建用于封装的 postscript 文件（格式 .eps 通常用于在文字 理 件中插入 表， 非常有用。消息窗口 示 建的文件名。

以PDF 格式 制



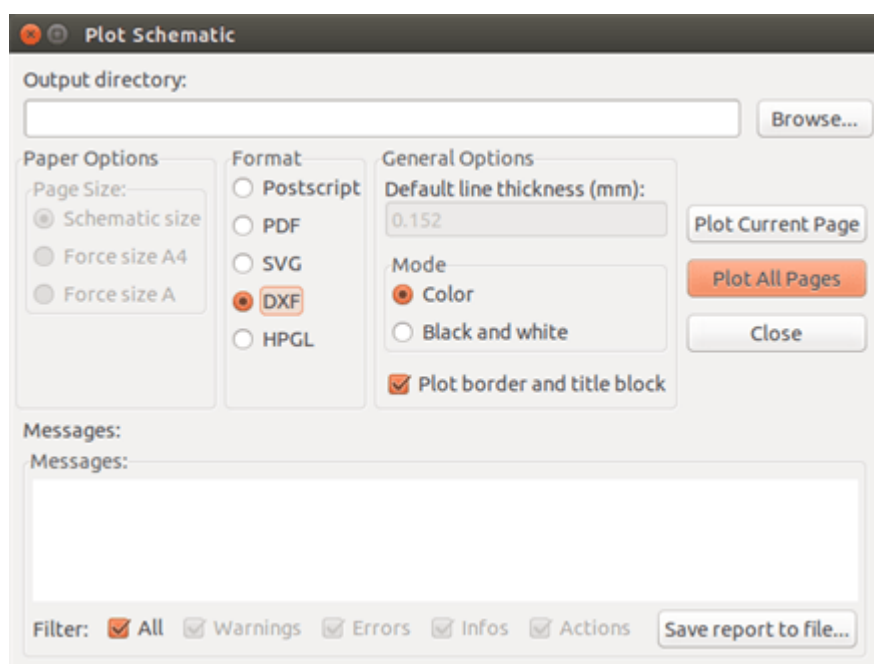
允 您使用 PDF 格式 建打印文件。文件名是 展名 .pdf 的工作表名称。

在 SVG 中



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

在 DXF 中



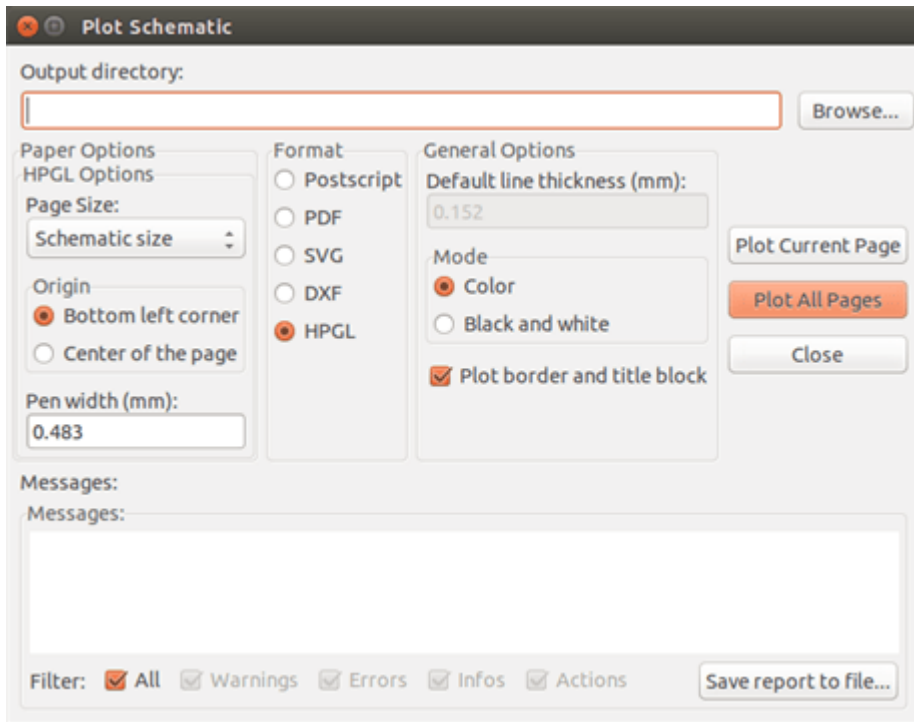
允 您使用 DXF 格式 建打印文件。文件名是 展名 .dxf 的工作表名称。

在 HPGL 中

此命令允 您 建 HPGL 文件。在 种格式中，您可以定义：

- 面大小。
- 原点。
- 笔 mm) 。

置 框窗口如下所示：



出文件名将是工作表名称加上 展名 .plt。

尺寸

通常 尺寸。在 种情况下，将使用 菜 中定义的 尺寸，并且所 的比例将 1.如果 了不同的 尺寸（A4 A0, 或 A E 自 整比例以填充 面。

偏移 整

于所有 准尺寸，您可以 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 能引入偏移以便正确


一般来

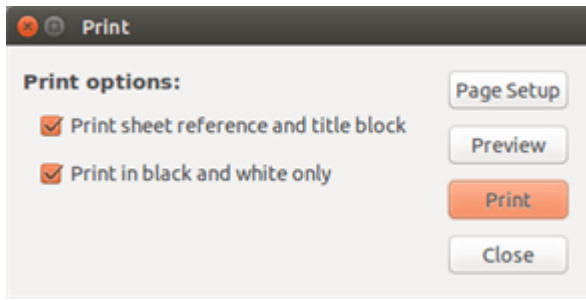
- 于原点位于 中心的 偏移量必 并 置 尺寸的一半。
- 于原点位于 左下角的 偏移量必 置 0

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

此命令可通  得，允 您可 化并生成 准打印机的 计文件。



“打印表格参考和” 可启用或禁用 参考和

“黑白打印” 置 色打印。如果使用黑白激光打印机，通常需要此 因 色打印成半色 通常不太可

符号 器

关于符号 器的一般信息

符号是一个示意 元素，包含 形表示， 气 接和定义符号的字段。原理 中使用的符号存 在符号 中。Eeschema 提供了一个符号 工具，允 您在 之 建 添加， 除或 符号，将符号 出到文件以及从文件 入符号。 工具提供了一种管理符号 文件的方法。

符号 器概述

符号 由一个或多个符号 成。通常，符号按功能， 型和/或制造商 行 分

符号由以下部分 成：

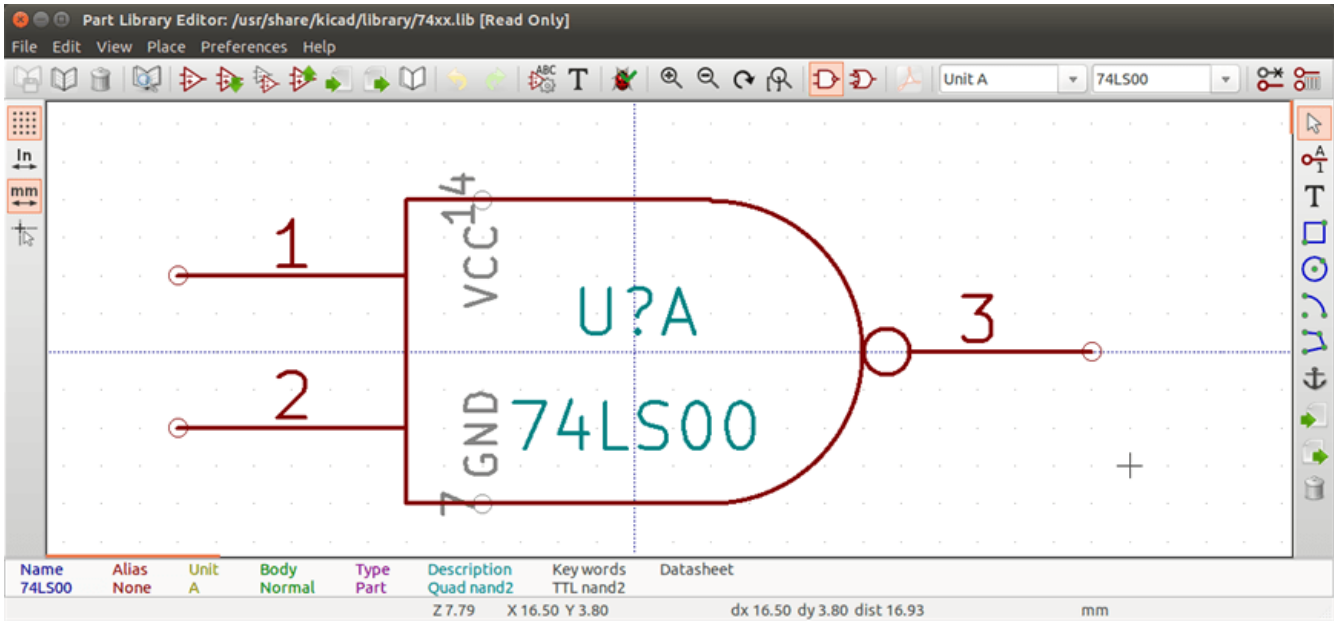
- 提供符号定义的 形 弧，文本等）。
- 具有 形属性（ 反 低 平有效等）和 气 ERC）工具使用的 气属性（ 入， 出，双向等）的引脚。
- 如参考， PCB 计的相 封装名称等字段等。
- 名用于将常用符号（如7400）与其所有衍生物（如 74LS00,74HC00 和 7437）相 所有 些 名共享相同的 符号。

正确的符号 计需要：

- 定义符号是否由一个或多个 元 成。
- 定义符号是否具有替代的体型，也称 De Morgan 表示。
- 使用 条，矩形， 形，多 形和文本 计其符号表示。
- 通 仔 定义每个引脚的 形元素，名称， 号和 气属性（ 入， 出，三 源端口等）来添加引脚。
- 如果其他符号具有相同的 计， 添加 名，如果已从其他符号 建符号， 将其固定或 除。
- 添加可 字段，例如 PCB 计 件使用的封装名称和/或定义其可 性。
- 通 添加描述字符串和数据表 接等来 符号。
- 将其保存在所需的 中。

符号 器概述

符号 器主窗口如下所示。它由三个工具 成，可快速 常用功能和符号 看/ 区域。并非所有命令都可在工具 上使用，但可以使用菜



主工具

通常位于主窗口 部的主工具 包括 管理工具，撤消/重做命令， 放命令和符号属性 框。



	保存当前 定的 如果没有， 按 将被禁用 当前已 中或当前未 更改 已 制作完成。
	要 的
	从当前 定的 或任何 中 除符号 如果当前没有 由 自定义。
	打开符号 器以 和符号
	建一个新符号。
	从当前 定的 中加 符号以 行
	从当前加 的符号 建新符号。
	将当前符号更改保存在内存中。 文件不是 改
	从文件中 入一个符号。
	将当前符号 出到文件。
	建包含当前符号的新 文件。注意：新的 不会自 添加到 目中。
	撤消上次
	重做最后一次撤消。
	当前符号属性。

	当前符号的字段。
	当前符号是否存在 计
	放大。
	小。
	刷新 示。
	放以适合 示中的符号。
	正常的体型。如果当前， 按 被禁用 符号没有替代的体型 格。
	用体型。如果当前， 按 被禁用 符号没有替代的体型 格。
	示相 文档。如果没有， 按 将被禁用 文档是 当前符号定义的。
	要 示的 位。如果， 将禁用下拉控件 当前符号不是从多个 元派生的。
	名。如果是， 下拉控件将被禁用 当前符号没有任何 名。
	引脚 引脚形状和位置的独立 具有多个 位和替代符号的符号。
	示引脚表。




元素工具

垂直工具 通常位于主窗口的右 允 您放置 计符号所需的所有元素。下表定义了每个工具 按


	工具。使用 工具右 可打开上下文菜单于光标下的对象。使用 工具 鼠标左 在消息中显示光标下对象的属性主窗口底部的面板。双左 工具将打开对象下的属性框光
	引脚工具。 鼠标左 以添加新引脚。
	形文本工具。 鼠标左 以添加新的 形文本
	矩形工具。 鼠标左 以开始 制a的第一个角 形矩形。再次 鼠标左 以放置 角 矩形。
	形工具。 鼠标左 以开始 制新的 形 圈 中心。再次 鼠标左 以定义 的半径。
	弧工具。 鼠标左 以开始从中 制新的 形弧 中央。再次 鼠标左 以定义第一个 弧 点。左 点 再次定义第二个弧 点。
	多 形工具。 鼠标左 以开始 制新的 形多 形 在当前的符号。左 每个加法多 形 双左 以完成多 形。
	点工具。 鼠标左 以 置符号的 点位置。
	从文件 入符号。
	将当前符号 出到文件。
	除工具。 鼠标左 以从当前符号中 除 象。

工具



垂直工具 通常位于主窗口的左 允 您 置一些 器 下表定义了每个工具 按

	打开和 网格可 性。
	将 位 置 英寸。
	将 位 置 毫米。
	打开和 全屏光

与

可以通  当前 它会 示所有可用 并允 您 一个 加 或保存符号 它将被放入此 中。符号的 名称是其 value 字段的内容。


NOTE

- 您必 将 加 到 Eeschema 中才能 其内容。
- 通 主工具 上的  , 可以在修改后保存当前 的内容。
- 通 像可以从任何 中 除符号 :  。


并保存符号

当您 符号 您 上并不是在其 中的符号上工作，而是在计算机内存中的符号上 制它。任何 操作都可以 松撤消。可以从本地 或 有符号加 符号。



符号

主工具 上的  将 示可从当前所 中 和加 的可用符号列表。

NOTE


如果通 名 了符号， 加 的符号的名称将 示在窗口 上，而不是 示在 定的 名上。符号 名列表始 随每个符号一起加 并且可以 行 您可以通过 从 像中 当前符号的 名来 建新符号： 。 名列表中的第一 是符号的根名称。

NOTE

或者，  可以加 先前由 像保存的符号： 。

保存符号

修改后，符号可以保存在当前 中，新 中，也可以 出到 份文件中。

要将修改后的符号保存在当前 中， 。 注意，更新命令 将符号更改保存在本地内存中。 您可以在恢之前做出决定。


要将符号更改永久保存到 文件， ，它将覆盖 有 文件并更改符号。

如果要 建包含当前符号的新 。系 将要求您 入新的 名称。

NOTE






新 不会自 添加到当前 目中。

您必 使用《manage-sym-lib-table, Symbol Library Table 框》将您希望在原理 中使用的任何新 添加到 Eeschema 中的 目 列表中。

 以 建 包含当前符号的文件。 文件将是一个 准 文件，它只包含一个符号。此文件可用于将符号 入另一个 上，建新的 命令和 出 命令基本相同。

将符号 移到另一个

您可以使用以下命令很容易地将符号从源 制到目 中：

- 像 源 .
- 通 像加 要 的符号： 。符号将 示在 区域中。
- 像 目 .
- 像将当前符号保存到本地内存中的新 .
- 像将符号保存在当前本地 文件中： .


弃符号 化

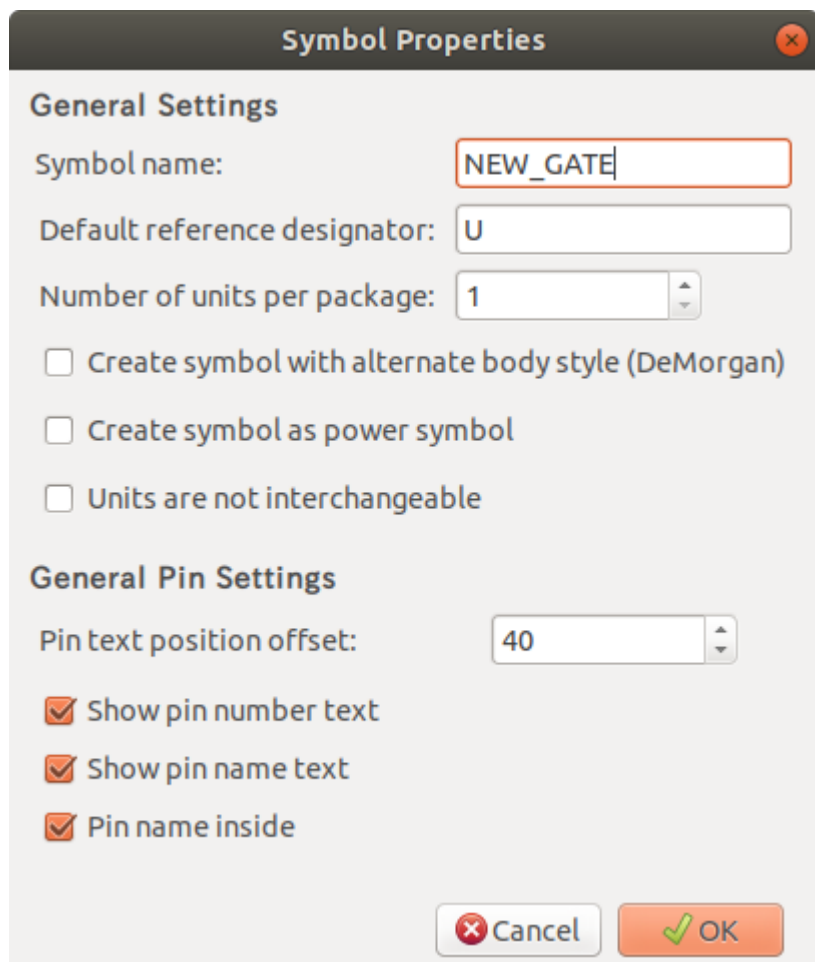
理符号 的符号只是其 中 符号的工作副本。 意味着只要你没有保存它，你可以重新加 它以 弃所做的所有更改。如果您已在本地内存中更新它并且尚未将其保存到 文件， 可以随 退出并重新启 Eeschema 将撤消所有更

改。

建 符号

建一个新符号

像可以 建新符号： 。系 将要求您 入符号名称（此名称用作原理 器中 字段的默认 参考 号（U, IC, R ...），每个包装的 位数（例如一个 7400 由每个包装4个 元 成）并且如果需要替代的体型（有 称 DeMorgan）。如果参考指示符字段 空， 默认 U。稍后可以更改 些属性，但最好在 建符号 正确 置它

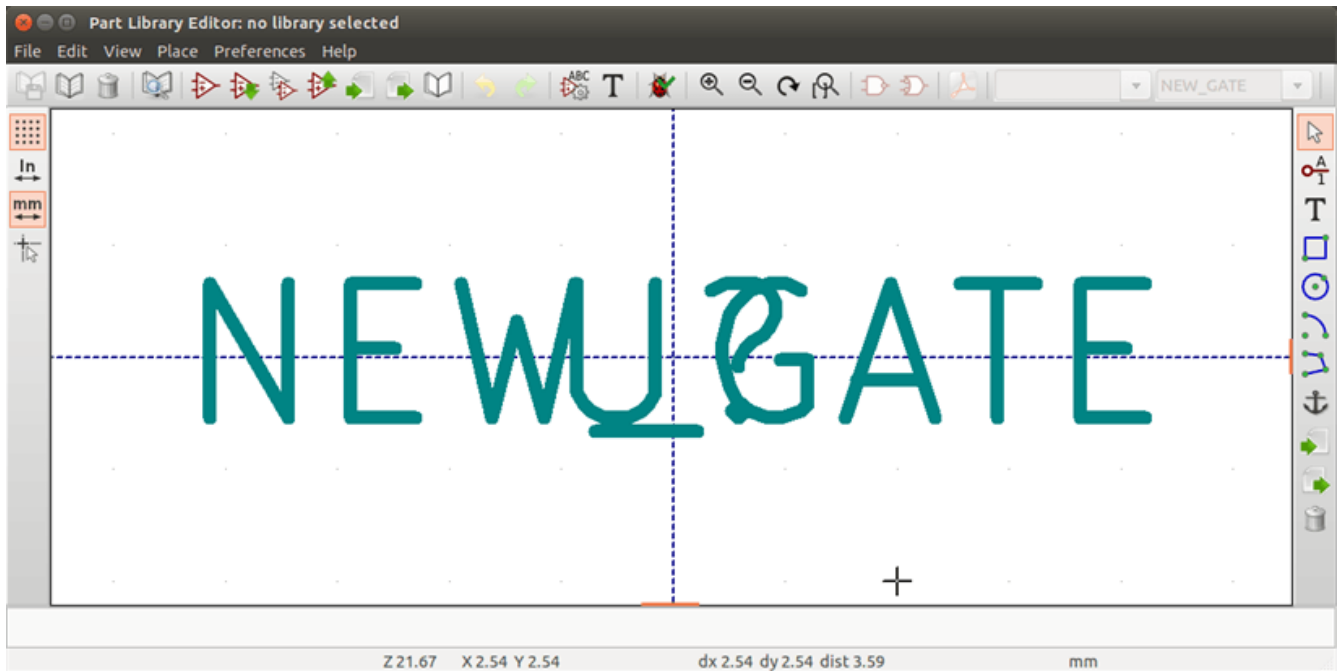


The image shows a 'Symbol Properties' dialog box with the following settings:

- General Settings**
 - Symbol name: NEW_GATE
 - Default reference designator: U
 - Number of units per package: 1
 - ☐ Create symbol with alternate body style (DeMorgan)
 - ☐ Create symbol as power symbol
 - ☐ Units are not interchangeable
- General Pin Settings**
 - Pin text position offset: 40
 - ☒ Show pin number text
 - ☒ Show pin name text
 - ☒ Pin name inside






Buttons: Cancel, OK

将使用上面的属性 建一个新符号，它将出 在 器中，如下所示。



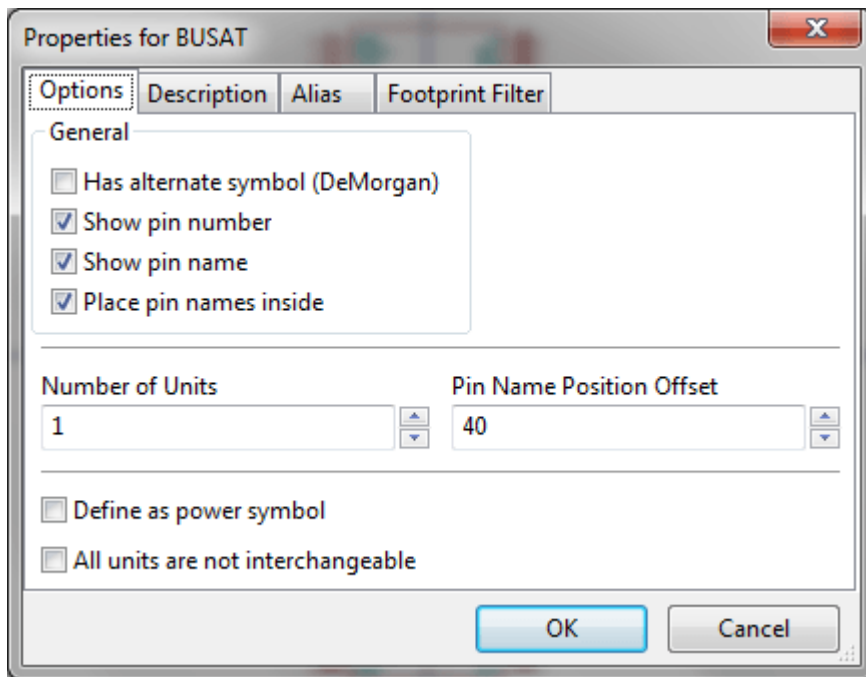
从另一个符号 建符号

通常，您要制作的符号 似于符号 中已有的符号。在 种情况下，很容易加 和修改 有符号。

- 加 将用作起点的符号。
-  或通 右 字段并 文本来修改其名称。如果您 制当前符号，系 将提示您 入新的符号名称。
- 如果模型符号具有 名，系 将提示您从新符号中 除与当前 冲突的 名。如果答案 否， 将中止新的符号 建。符号 不能包含任何重 的名称或 名。
- 根据需要 新符号。
- 像更新当前 中的新符号：  或 像保存到新  或者 如果要将此新符号保存在其他 有 中， 像其他  并保存新符号。
- 像将当前 文件保存到磁 。

符号属性

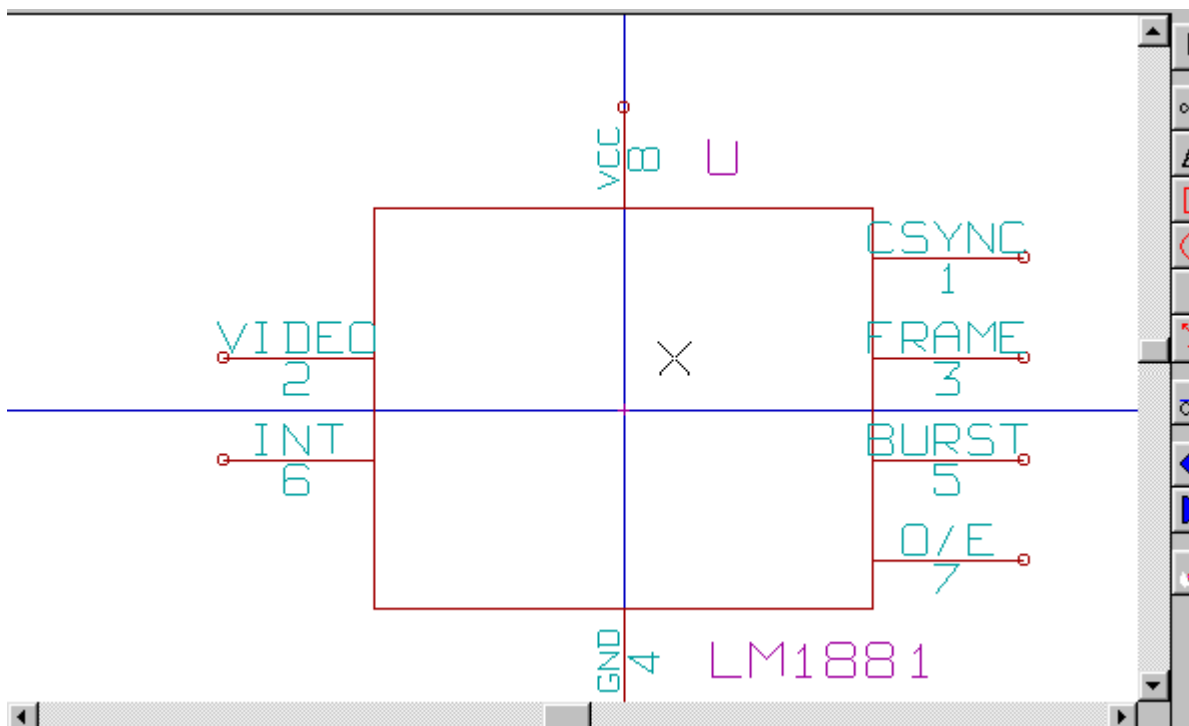
在符号 建期 仔 置符号属性，或者从 制的符号 承符号属性。要更改符号属性，  以 示下面的 框。




正确 置每个封装的 元数和 用符号表示（如果已启用）非常重要，因 在 或 建引脚 每个 元的相 引脚都会受到影响。如果在 建和 引脚后更改每个封装的 元数， 需要 外的工作来添加新的 元引脚和符号。然而，可以随 修改 些属性。


形 示引脚 号和 示引脚名称定义了引脚 号和引脚名称文本的可 性。如果 中相 的 此文本将可 将引脚名称放在内部 定义了相 于引脚体的引脚名称位置。如果 中 文本将 示在符号 廓内。在 种情况下，引脚名称位置偏移属性定义文本从引脚的主体端移开。30 到 40（1/1000英寸）的 是合理的。

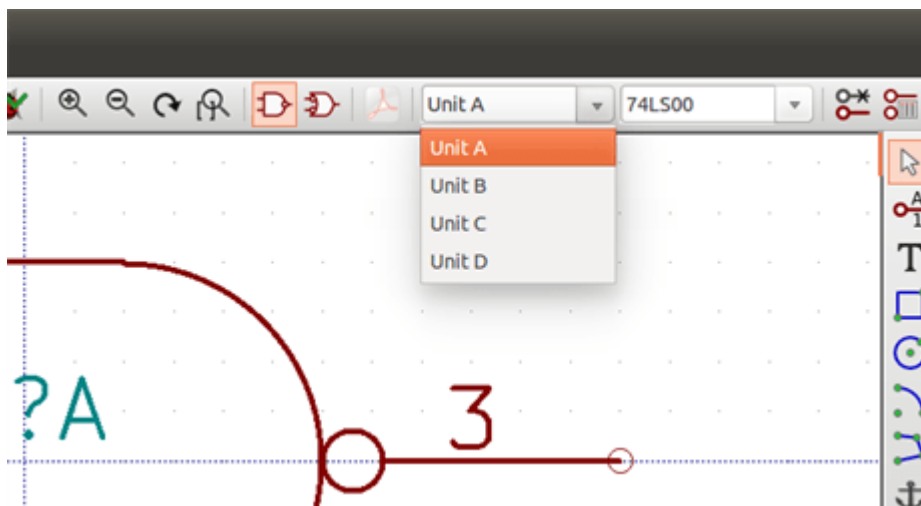
下面的示例 示了未 中“放置引脚名称” 的符号。注意名称和引脚号的位置。



有替代符号表示的符号

如果符号具有多个符号再存 必 一个表示来 它 要 常 表示， .

要用表示，。使用下面 示的 要的 位。



形元素

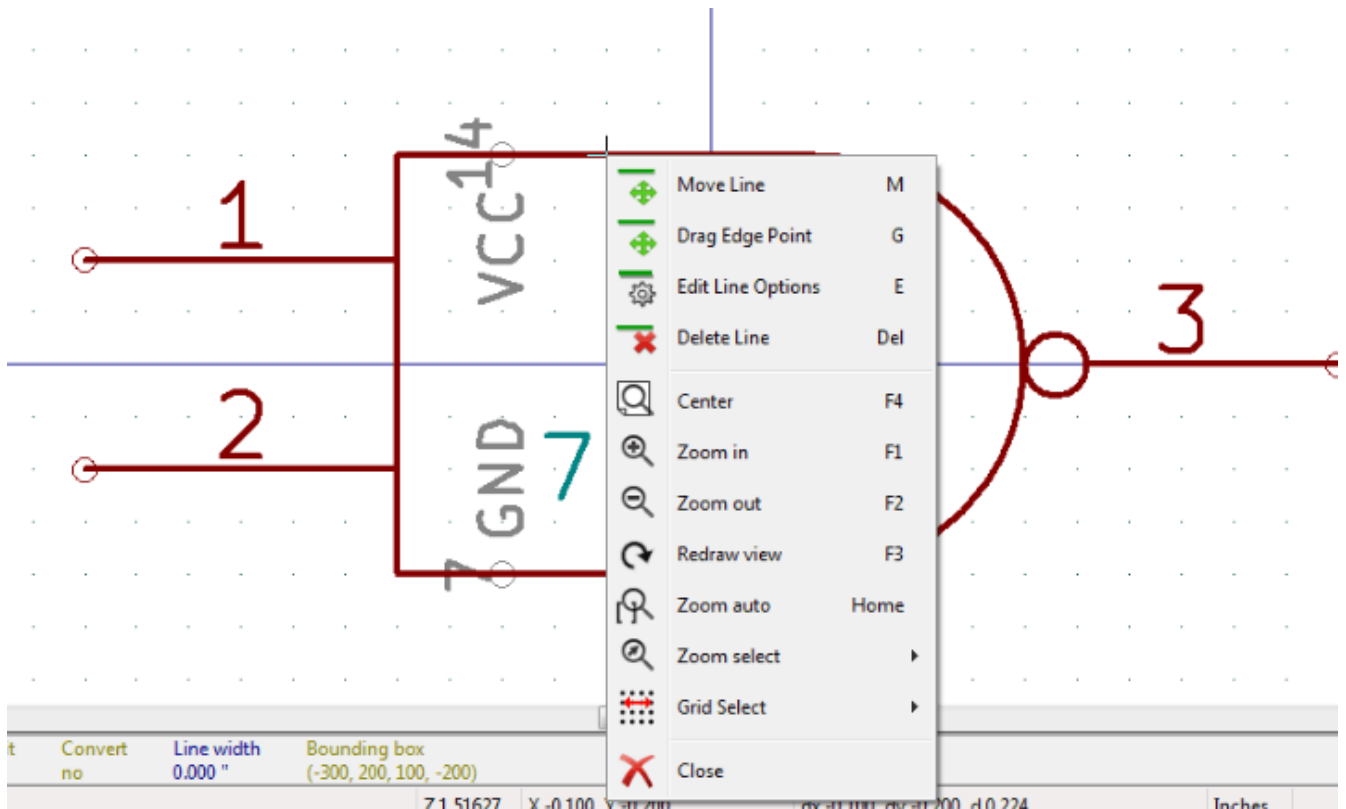
形元素 建符号的表示，不包含 气 接信息。使用以下工具可以 计它

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

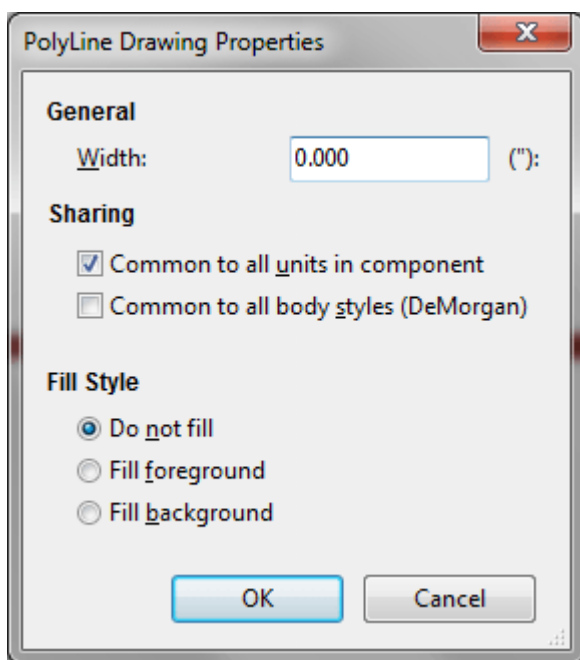
主窗口右 的垂直工具 允 您放置 计符号表示所需的所有 形元素。

形元素成 格

每个 形元素（ 弧， 等）可以被定义 于所有 元和/或主体 式是共同的或者 于 定 元和/或主体 式是特定的。右 元素可以快速 元素 以 示所 元素的上下文菜 下面是 元素的上下文菜



您可以通过双击元素以修改其属性。下面是多边形元素的属性对话框。



多边形元素的属性是：

- 用于定义当前多边形位中元素线条的宽度。
- 符号中所有位的共同置定义是否每个包含多个位的符号制多边形元素，或者是否当前位制多边形元素。
- “所有主体式的共同（DeMorgan）”置定义是否具有替代主体式的符号中的每个符号表示制多边形元素，或者是否当前主体式制多边形元素。
- 填充式置确定多边形元素定义的符号是否要制未填充，背景填充或前景填充。

形文本元素

一个 **T** 允 建 形文本。即使 像符号， 形文本也始 可 注意， 形文本 不是字段。

每个符号多个 位和替代体型式

符号可以具有两个符号表示（ 准符号和 用符号，通常称 *DeMorgan* ）和/或每个包具有多于一个 元（例如 某些符号每个封装可以有多个 元，每个 元具有不同的符号和引脚配置。

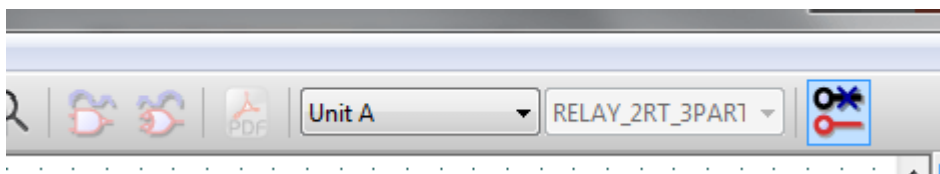
例如，考 具有两个开 的 器，其可以被 计 具有三个不同 元的符号： 圈，开 1和开 2. 计每个封装具有多个 元和/或交替的主体 式的符号是非常灵活的。引脚或体型符号 于所有 元可以是共同的或者 于 定 元是特定的，或者它 于两个符号表示可以是共同的，因此特定于 定符号表示。

默认情况下，引脚特定于每个 元的每个符号表示，因 引脚 号特定于 元，并且形状取决于符号表示。当引脚 于每个 元或每个符号表示是公共的 您需要 所有 元和所有符号表示 建一个引脚（ 通常是 源引脚的情况）。 也是身体 式 形形状和文本的情况，每个 元可能是共同的（但通常特定于每个符号表示）。

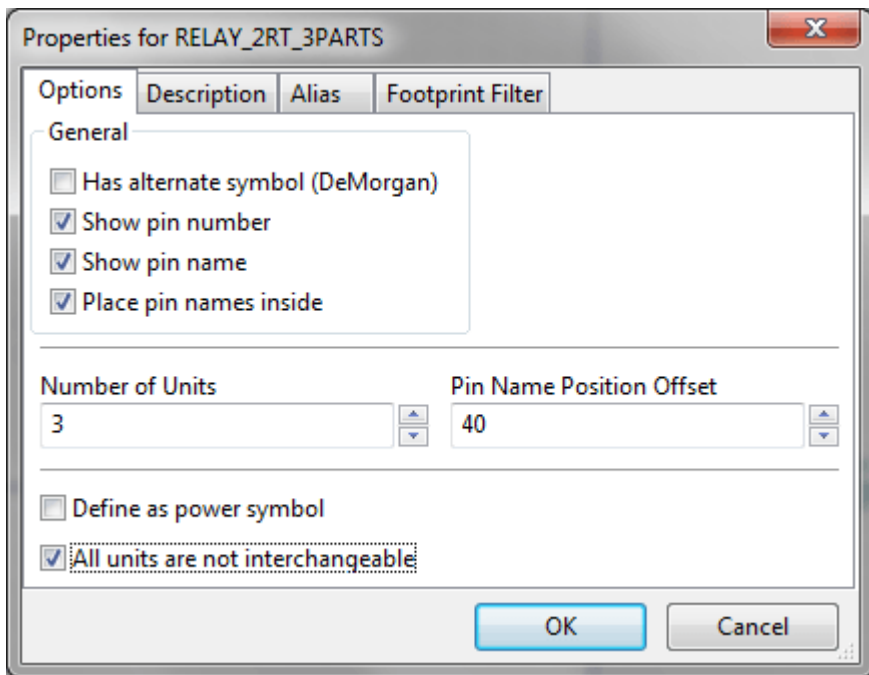
具有不同符号的多个 元的符号示例：

是一个 器的例子，每个包装有三个 元，开 1 开 2和 圈：

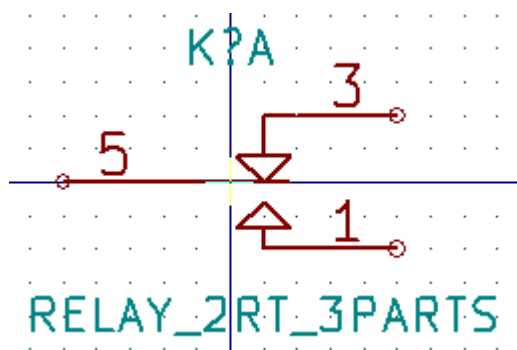
引脚未 接。可以 每个 元添加或 引脚，而无需与其他 元的引脚耦合。



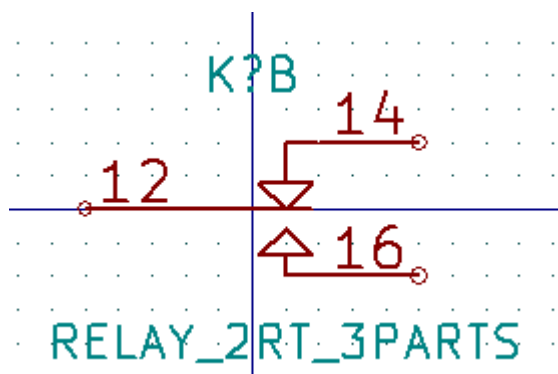
必 所有不可互 的 元。



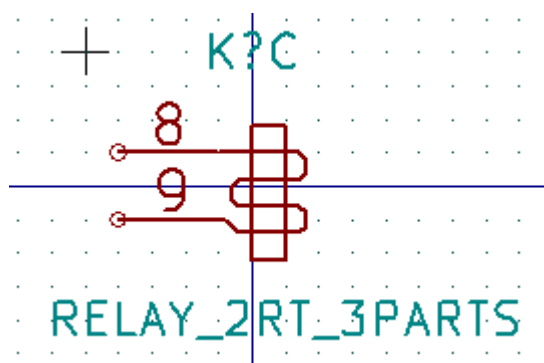
元1



元2



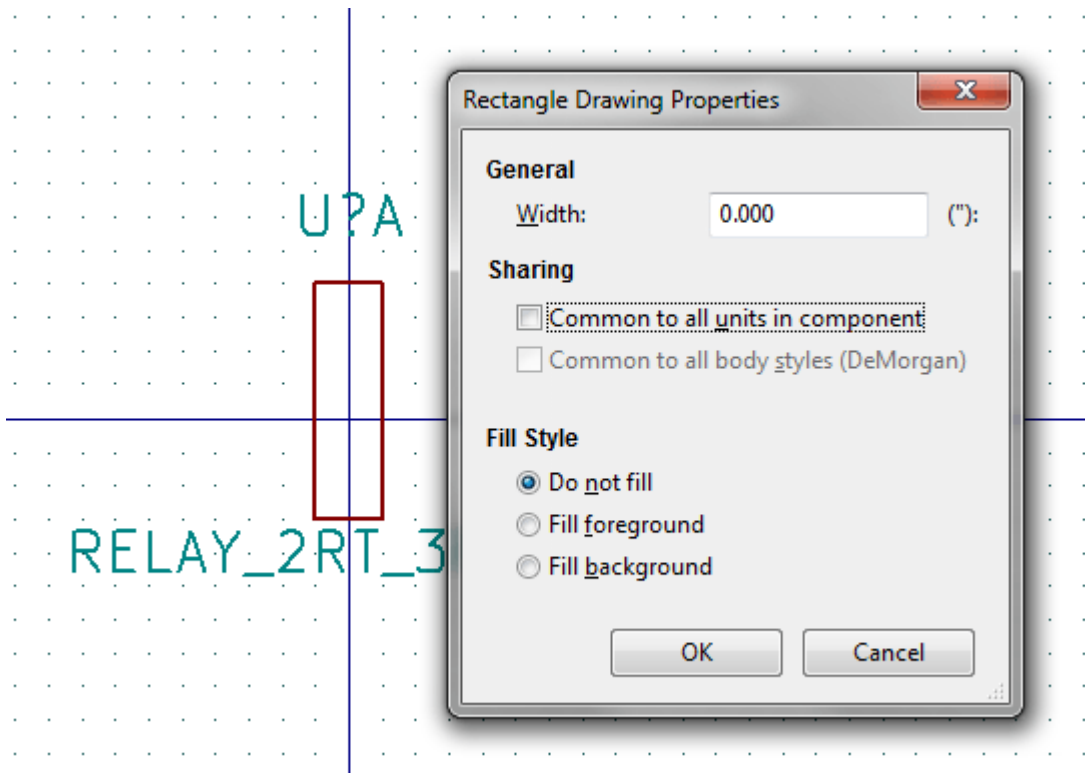
元3




它没有相同的符号和引脚布局，因此不能与 元1和2互

形符号元素

下面 示的是 形主体元素的属性。从上面的 器示例中，三个 元具有不同的符号表示。因此，每个 元都是 独 建的，并且 形主体元素必 禁用“符号中所有 元的通用”。



引脚 建和

您可以  来 建和插入引脚。通 双 引脚或右 引脚以打开引脚上下文菜 可以 所有引脚属性。必 仔 建 引脚, 因 任何 都会 PCB 计 生影响。可以 除和/或移 已放置的任何引脚。

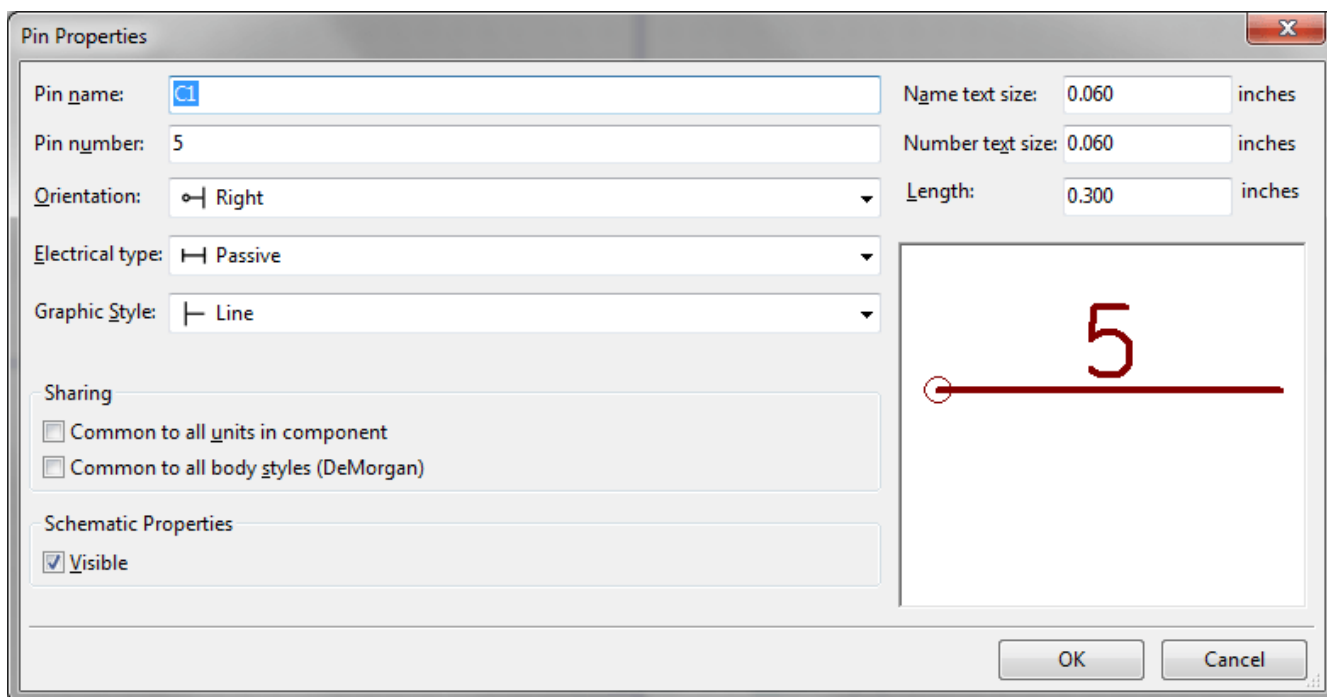
引脚概述

引脚由其 形表示, 名称和 “ 号” 定义。引脚的 “数字” 由一 4个字母和/或 号定义。要使 气 ERC) 工具有 用, 必 正确定义引脚的 “ 气” 型 (入, 出, 三)。如果未正确定义此 型, 原理 ERC 果可能无效。

重要笔

- 不要在引脚名称和数字中使用空格。
- 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.
- 如果引脚名称减少 个符号, 引脚被 未命名。
- 以 “#” 开 的引脚名称保留用于 源端口符号。
- 引脚 “ 号” 由1到4个字母和/或数字 成。1,2, ... 9999是有效数字。A1, B3, Anod, Gnd, Wire 等也有效。
- 符号中不能存在重 的引脚 “ 号”。

引脚属性



The image shows a 'Pin Properties' dialog box with the following fields and options:

- Pin name:** C1
- Pin number:** 5
- Orientation:** Right (indicated by a right-pointing arrow icon)
- Electrical type:** Passive (indicated by a horizontal line with a circle at the end icon)
- Graphic style:** Line (indicated by a horizontal line icon)
- Name text size:** 0.060 inches
- Number text size:** 0.060 inches
- Length:** 0.300 inches
- Sharing:**
 - ☐ Common to all units in component
 - ☐ Common to all body styles (DeMorgan)
- Schematic Properties:**
 - ☒ Visible

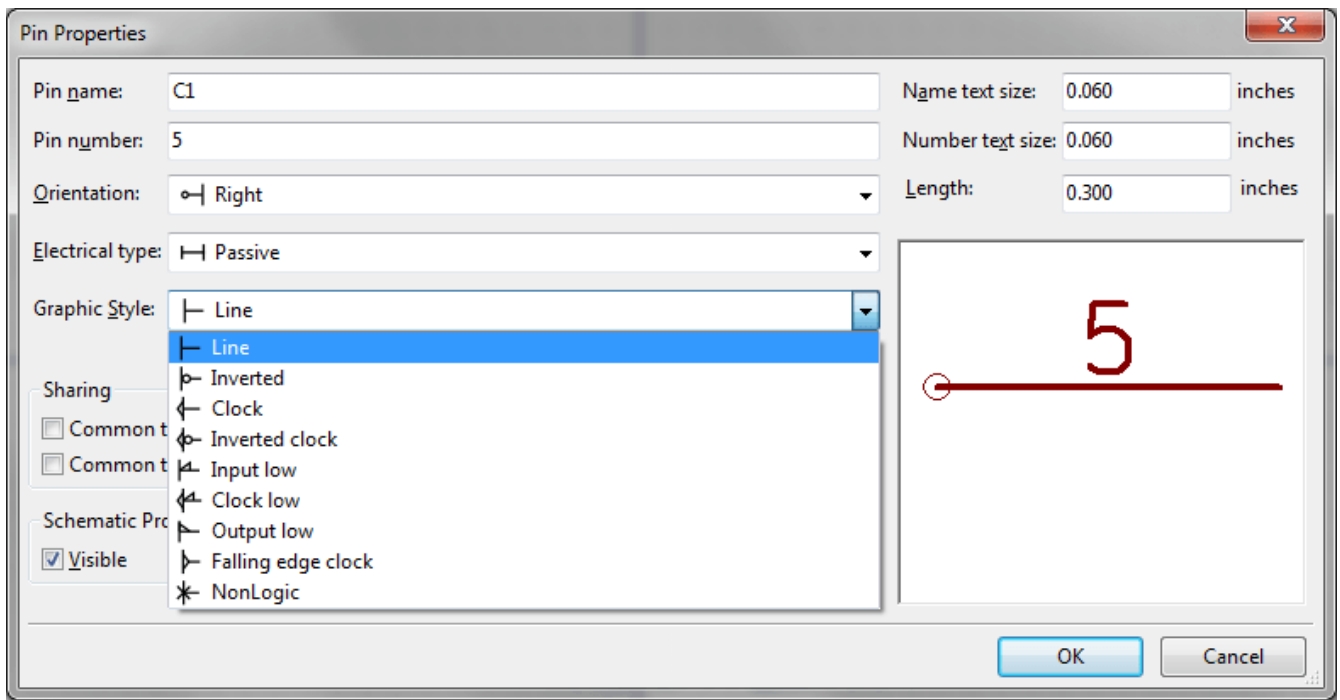
On the right side of the dialog, there is a preview window showing a red horizontal line with a circle at the left end and the number '5' centered above it.

引脚属性 框允 您 引脚的所有特性。 建引脚或双 有引脚 会自 出此 框。此 框允 您修改：

- 名称和名称的文字大小。
- 数字和数字的文字大小。
- 度。
- 气和 形 型。
- 位和替代代表成 格。
- 可 性。

引脚 形 式

下 示了不同的 形引脚 式。 形 式的 引脚的 气 型没有任何影响。



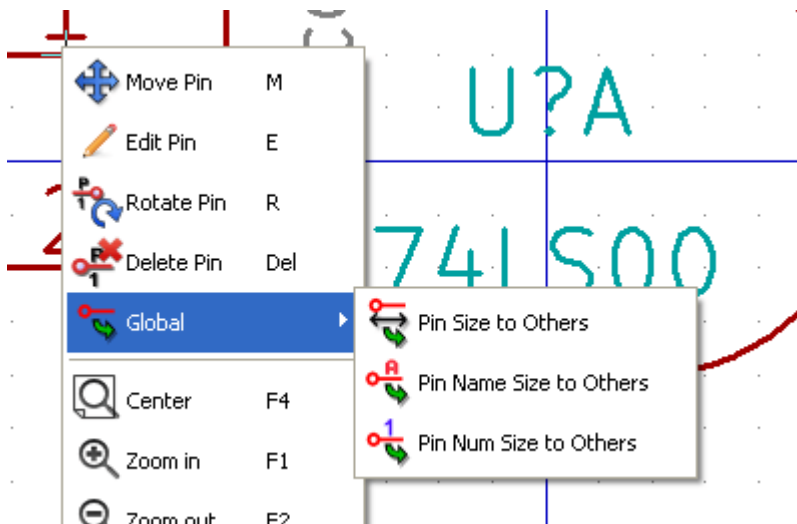
引脚气型

正确的气型于原理 ERC 工具很重要。定义的气型是：

- 双向指示 入和 出之 可交 的双向引脚（例如微 理器数据
- 三 是通常的3 出。
- 无源用于无源符号引脚， 阻， 接器等。
- 当 ERC 无 要 可以使用未指定的。
- 源 入用于符号的 源引脚。 源引脚自 接到具有相同名称的其他 源 入引脚。
- 功率 出用于 器 出。
- 开路 射极和开路集 极 型可用于如此定义的 出。
- 当符号具有没有内部 接的引脚 使用未 接。


引脚全局属性

您可以使用引脚上下文菜 的全局命令条目修改所有引脚的名称和/或 号的 度或文本大小。 要修改的参数，然后 入新 然后将 用于所有当前符号的引脚。



多个 元和 用符号表示定义引脚


在 建和 引脚 具有多个 元和/或 形表示的符号尤其成 大多数引脚特定于每个 元（因 它 的引脚 号特定于每个 元）和每个符号表示（因 它 的形式和位置特定于每个符号表示）。 于每个封装具有多个 元的符号和替代符号表示，引脚的 建和 可能是有 的。符号 器允 同 建引脚。默认情况下， 多个 位符号的所有 位以及具有替代符号表示的符号的两个表示都 行 引脚所做的更改。

唯一的例外是引脚的 形 型和名称。建立此依 系以便在大多数情况下更容易 建和 引脚。可以通 切 主工具 上的  来禁用此依 系。 将允 您完全独立地 每个 元和表示 建引脚。

符号可以具有两个符号表示（表示“Demorgan”），并且可以由多个 元 成，如具有 的符号的情况。 于某些符号，您可能需要几个不同的 形元素和引脚。与“前一部分具有多个不同符号的符号的示例”，中所示的 器 本 似， 器可以由三个不同的 元表示： 圈，开 触点1，并切 开 触点2。

具有多个 元的符号的管理和具有替代符号表示的符号是灵活的。引脚可以是通用的或特定于不同的 元。引脚也可以是符号表示或每个符号表示特有的。

默认情况下，引脚特定于每个 元的每个表示，因 它 的数量因每个 元而不同，并且它 的 计 于每个符号表示是不同的。当一个引脚 所有 元都是通用的 它只需要 制一次，例如在 源引脚的情况下。

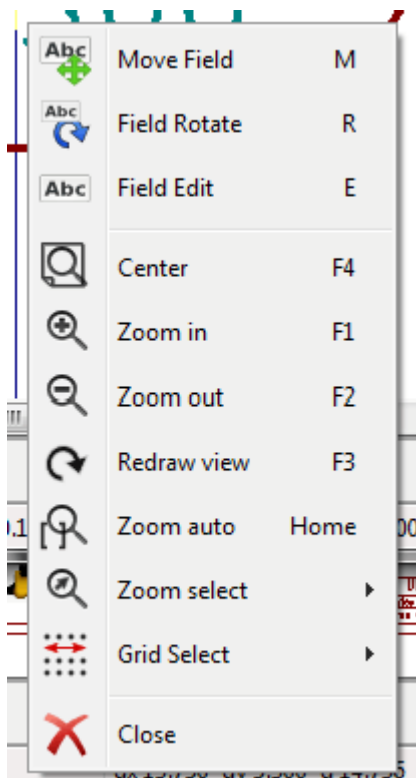
一个例子是 出引脚 7400 四路双 入与非 由于有四个 元和两个符号表示，因此在符号定义中定义了八个 独的 出引脚。 建新的 7400 符号 正常符号表示的 元A将 示在 器中。要在 用符号表示中 引脚 式，必 首先通 工具 上的  按 启用它。要 每个 元的引脚 号，使用以下 像 相 的 元： 下拉控件。

符号字段

所有 符号都定义了四个默认字段。无 何 建或 制符号，都会 建参考指示符， 占用空 分配和文档文件 接字段。 需要参考指示符和 字段。 于 有字段，可以通 右 引脚来使用上下文菜 命令。 中定义的符号通常使用四个默认字段定义。如供 商，部件号， 位成本等附加字段可以添加到 符号中，但通常 是在原理 器中完成的，因此附加字段可以 用于原理 中的所有符号。

符号字段

要 有符号字段， 右 字段文本以 示下面 示的字段上下文菜



要 未定义的字段，添加新字段或 除主工具 上的可 字段 **T** ，以打开下面 示的字段属性 框。

Field Properties

Name	Va...
Reference	U
Value	74LS00
Footprint	
Datasheet	

Add Field
Delete Field
Move Up

Horiz. Justify

☐ Left
☒ Center
☐ Right

Vert. Justify

☐ Bottom
☒ Center
☐ Top

Visibility

☒ Show
☐ Rotate

Style:

☒ Normal
☐ Italic
☐ Bold
☐ Bold Italic

Field Name

Reference

Field Value

U

Show in Browser

Size

0.060

in

PosX

0.000

in

PosY

-0.050

in

OK

Cancel

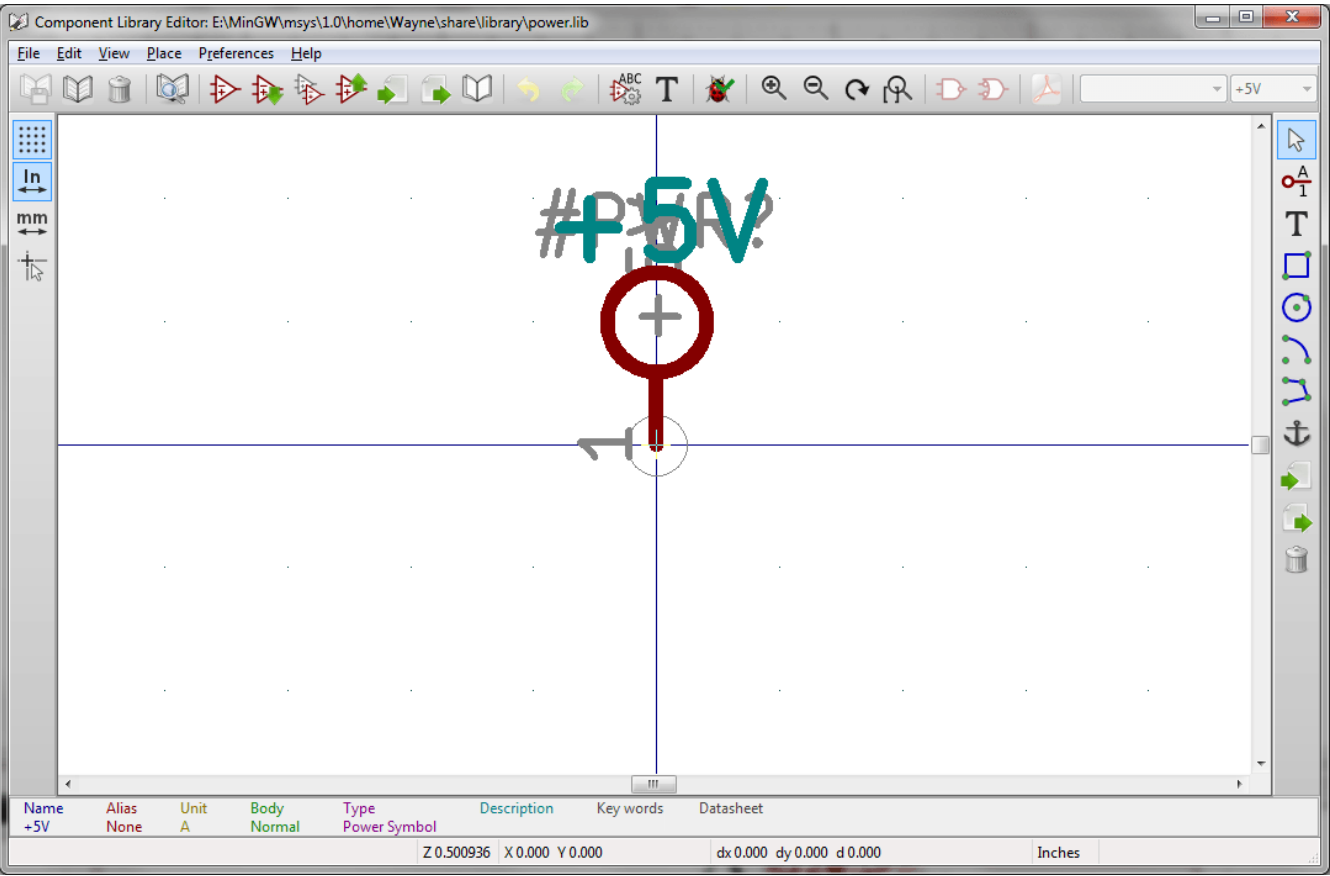
字段是与符号 的文本部分。不要将它 与属于此符号 形表示的文本混淆。

重要笔

- 修改 字段有效地使用当前符号作 新符号的起点 建新符号。将新符号保存到当前 定的 将包含在 字段中。
- 上面的字段 框必 用于 空的字段或启用了不可 属性。
- 封装定义 使用 LIBNAME:FPNAME 格式的 封装，其中 LIBNAME 是封装 表中定义的封装 的名称（ 参 Pcbnew 参考手册中的 封装 表部分），FPNAME 是 LIBNAM 中的封装名称。

源符号

功率符号的 建方式与普通符号相同。将它 放在 用 如 power.lib) 中可能很有用。 源符号由 形符号和 源 藏 型的引脚 成。原理 捕 件可以像 理任何其他符号一 理 源端口符号。一些 防措施至 重要。以下是 源 +5V 符号的示例。



要 建 源符号，使用以下步：

- 添加名 +5V 的“源 入”型的引脚（重要的是因 此名称将建立与网 +5V 的 接），引脚 号 1 不重要的数字）， 度 0 以及“路”“形 格”。
- 如 所示，将一个小 圈和一个从引脚到 圈的部分放置。
- 符号的 点在引脚上。
- 符号 “+5V”。
- 符号引用是“#+5V”。除了必 “#”的第一个字符表示符号是 符号外，参考文本并不重要。按照 例，引用字段以“#”开 的每个符号都不会出 在符号列表或网表中，并且引用被声明 不可

建新的 源端口符号的更 方法是使用另一个符号作 模型：

- 加 有的 源符号。
- 使用新 源符号的名称 引脚名称。
- 如果要 示 源端口，将 字段 与引脚相同的名称。
- 保存新符号。

LibEdit - 符号

概述

符号由以下元素 成

- 形表示（几何形状，文本）。
- 引脚。
- 后置 理器使用的字段或 文本：网表，符号列表。

要初始化两个字段：引用和 与符号 的 计的名称以及 的足迹的名称，其他字段是空 字段，它 通常可以保持空，并且可以在原理 捕 期 填充。

但是，管理与任何符号相 的文档有助于 的研究，使用 and 相 文档包括

- 一行注
- 一系列 字，如 TTL CMOS NAND2，由空格分隔。
- 附加文件名（例如 用程序注 或 pdf 文件）。

附加文件的默认目

kicad/share/library/doc

如果没有找到：

kicad/library/doc

在 linux 下：

/usr/local/kicad/share/library/doc

/usr/share/kicad/library/doc

/usr/local/share/kicad/library/doc

字允 您根据各种 准有 地搜索符号。注 和 字 示在各种菜 中，特 是从 中 符号

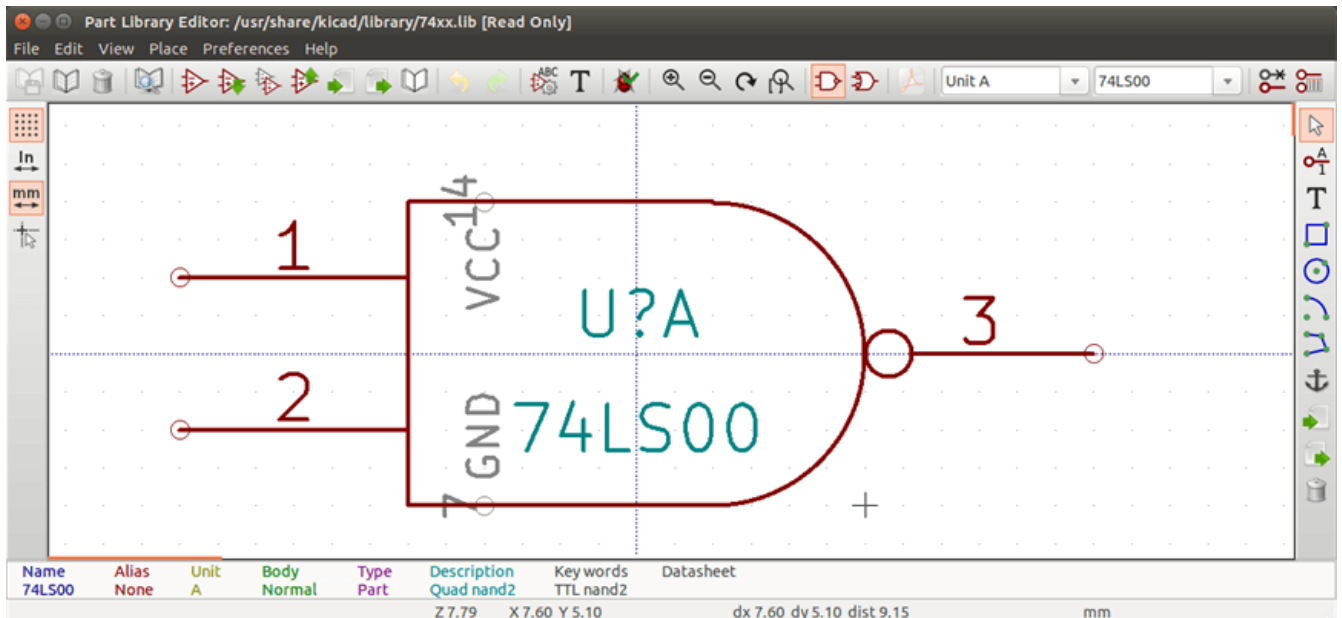
符号也有一个 点。相 于 点 行旋 或 像，并且在放置期 点用作参考位置。因此准确定位 是有用的。


符号可以具有 名，即等效名称。 允 您 着减少需要 建的符号数量（例如，74LS00 可以具有 名，例如 74000,74HC00,74HCT00）。

最后，符号分布在 中（按主 或制造商分 以便于管理。

定位符号 点

点位于坐 0,0 并由屏幕上 示的 色 示。



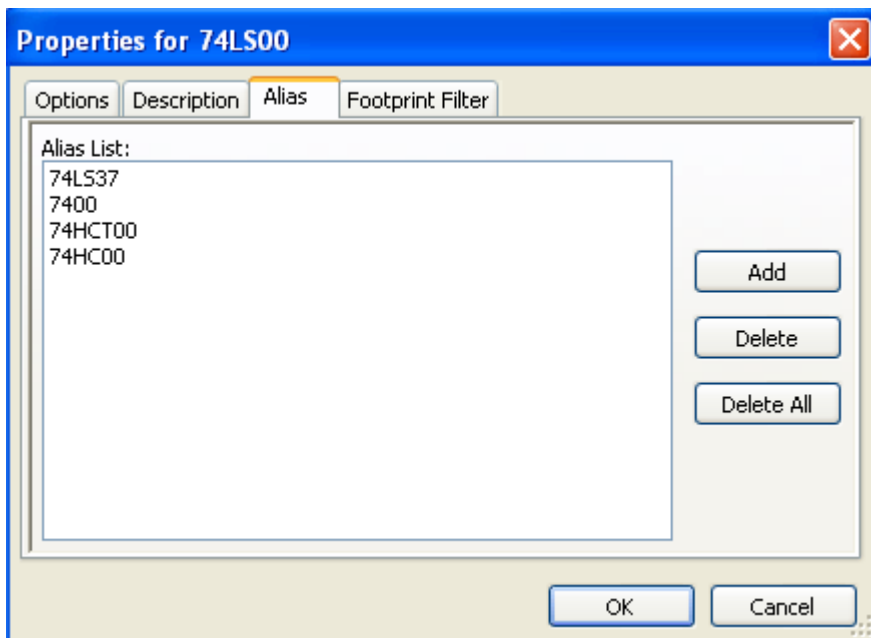
可以通过  并 新的所需 点位置来重新定位 点。 将自 重新定位在新 点上。

符号 名

名是与 中相同符号 的另一个名称。具有 似引脚和表示的符号然后可以 由一个符号表示，具有若干 名（例如，具有 名 74LS00,74HC00,74LS37 的 7400）。

使用 名可以快速构建完整的 此外， 些 更加 凑，可以 松加 KiCad。

要修改 名列表，必 通  主 窗口，然后 名文件



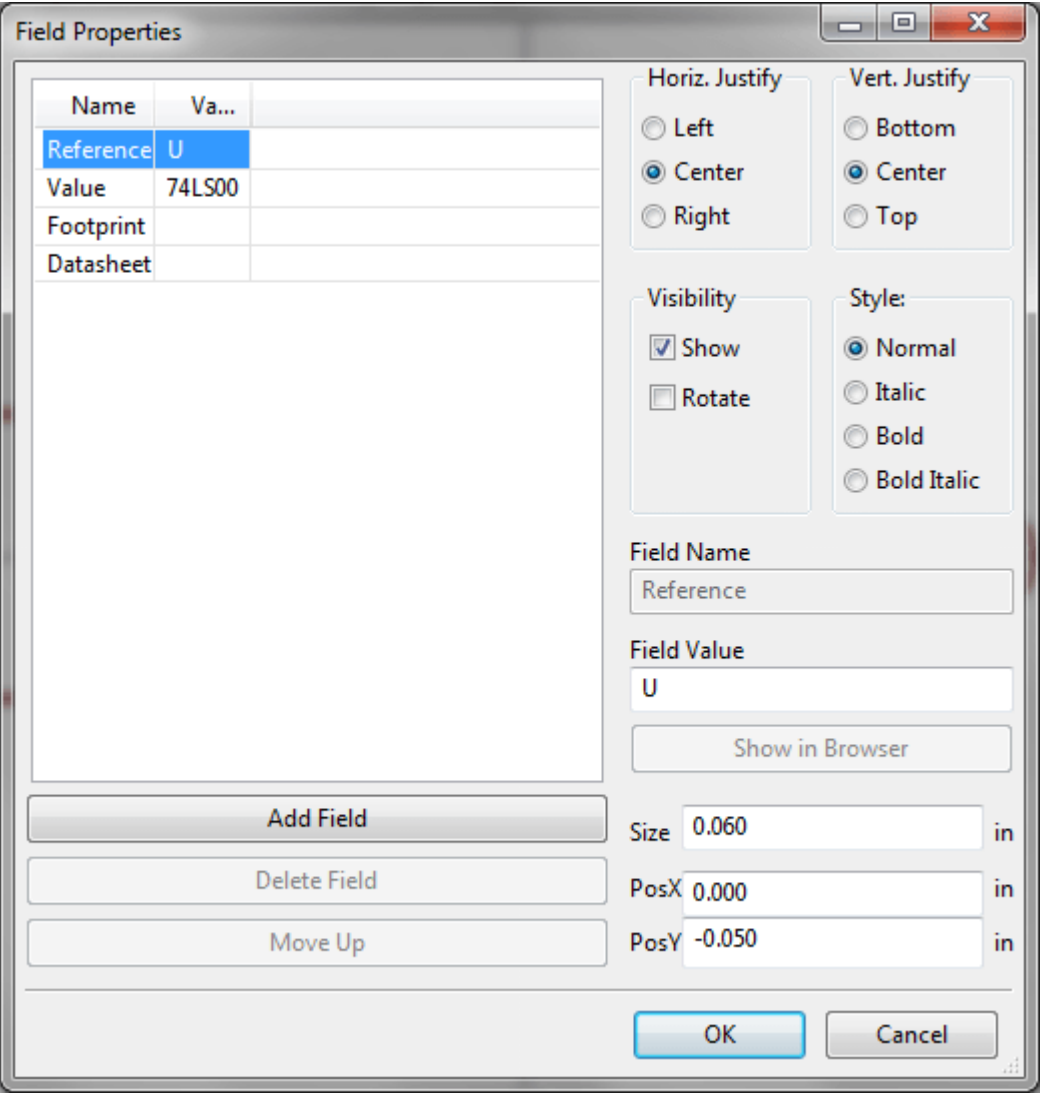
因此，您可以添加或 除所需的 名。由于 了当前的 名，因此 然无法将其 除。

要 除所有 名，首先要 根符号。 主工具 窗口中 名列表中的第一个符号。

符号字段

字段 器通 用： **T**。


有四个特殊字段（符号附加的文本）和可配置的用 字段

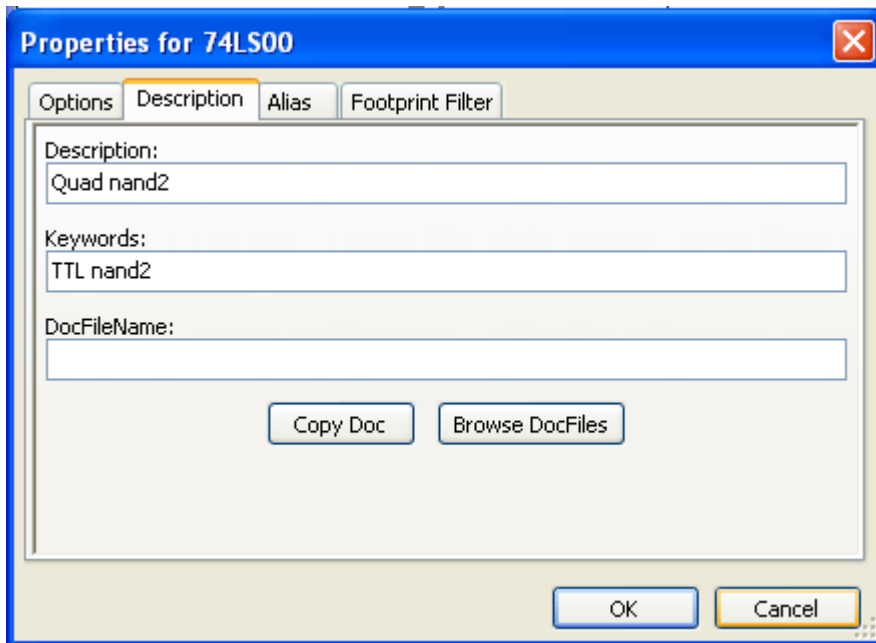


特殊字段

- 参考。
- 它是 中的符号名称和原理 中的默认 字段。
- 封装。它是用于 路板的封装名称。使用 CvPcb 置封装列表 不是很有用，但如果不使用 CvPcb 必 使用。
- 表。它是一个保留字段,在 写 不使用。

符号文档

要 文档信息，需要通  用符号的主 窗口，并 文档文件



必须正确的名称或根符号，因此文档是名称之唯一不同的特征。“制文档”按钮允许您将文档信息从根符号复制到当前的名称。

符号 字

允许您根据特定的标准（功能，技术系列等）以不同的方式搜索符号

Eeschema 研究工具不区分大小写。中使用的最新是

- 用于系列的 CMOS TTL
- AND2 NOR3 XOR2 INV ...用于 AND2 = 2 输入 AND NOR3 = 3 输入 NOR
- JKFF DFF ... 用于 JK 或 D 触发器。
- ADC, DAC, MUX...
- 具有开路集电极输出的 OpenCol。因此，如果在原理图中，您搜索符号：通 字 NAND2 OpenCol Eeschema 将显示具有两个字的符号列表。

符号文档 (Doc)

注 行（和 字）示在各种菜单中，尤其是在的 示符号列表和 ViewLib 菜单中 符号

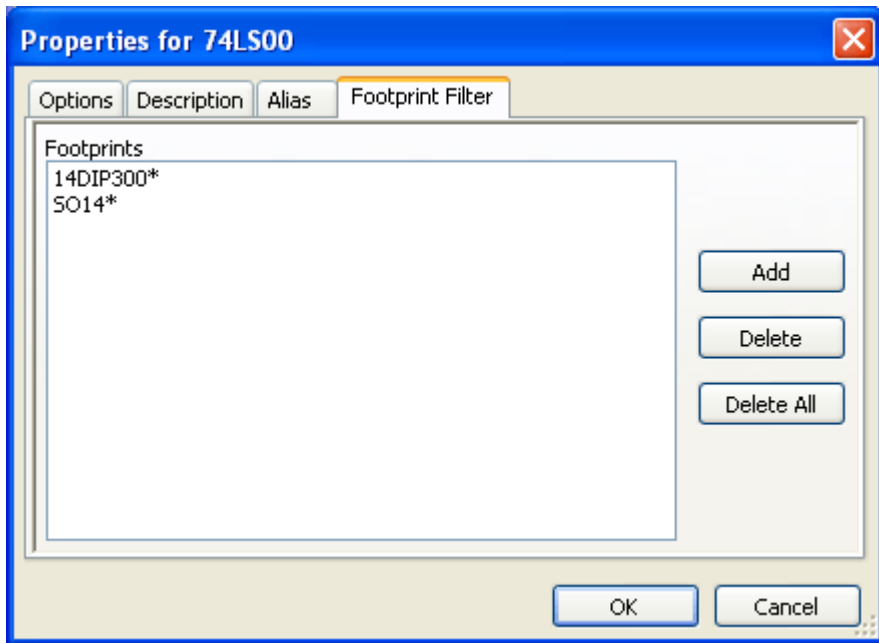
如果一个 Doc. 文件存在，也可以在原理图中，通过右 符号 示的 出菜单中 它。

相关文档文件 (DocFileName)

表示可用的附件（文档，用原理 pdf 文件，原理 等）。

CvPcb 的封装

您可以输入符号允许的覆盖区列表。此列表充当 CvPcb 用于 示允许的覆盖区的 器。无效列表不会 任何内容。



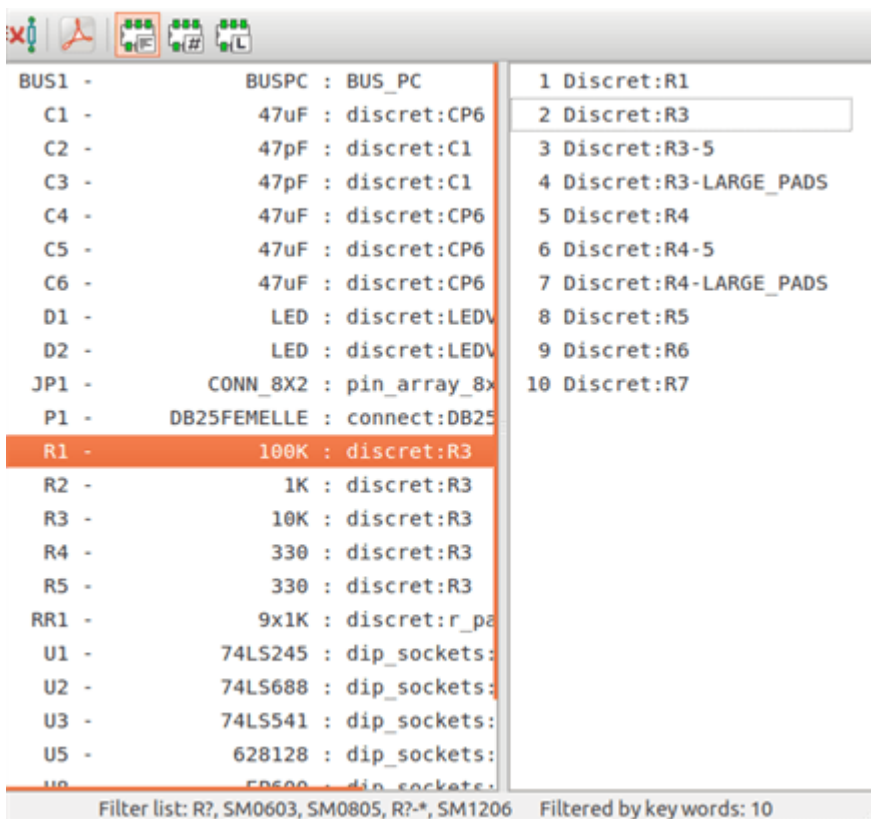
允 使用通配符。

SO14* 允 CvPcb 示名称以 SO14 开 的所有封装。

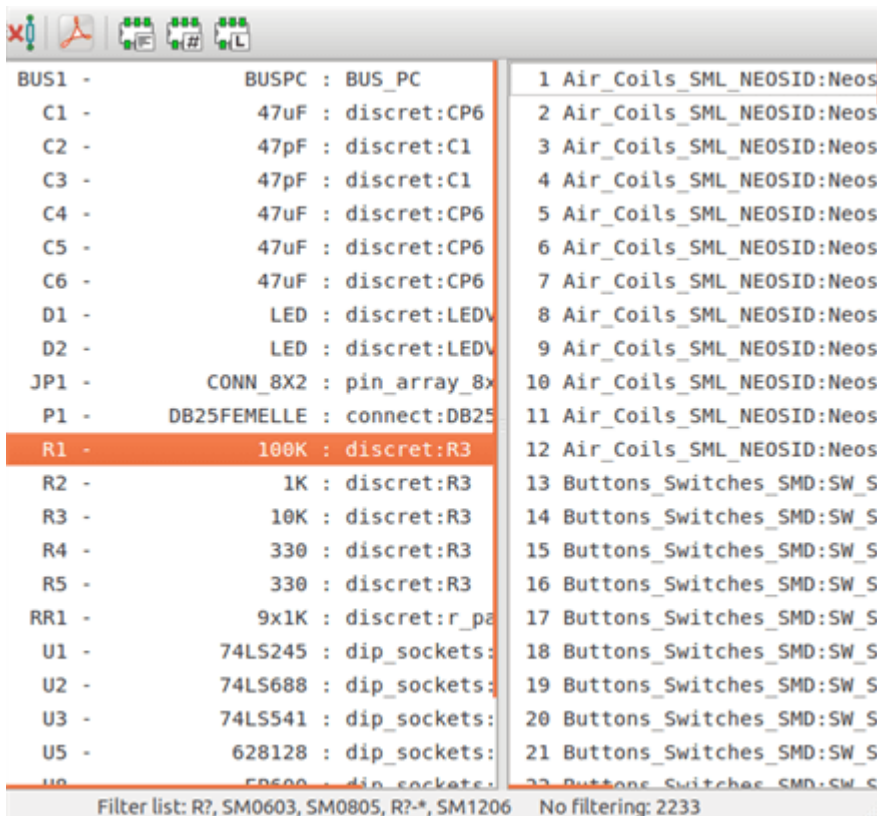
于 阻器, R 示所有 有以 R 开 的2个字母名称的封装。

以下是 本 : 有和没有

有



没有




符号


您可以 松 包含常用符号的 形符号 文件。 可以用于 建符号（三角形，与，或，异或 等的形状）以用于保存和随后的重 使用。

些文件默认存 在 目 中，并具有“.sym” 展名。 些符号不像普通符号那 收集在 中，因 它 通常不是那么多。

出或 建符号


可以使用按 像 出符号： 。您通常只能 建一个 形， 除所有引脚（如果存在）也是一个好主意。

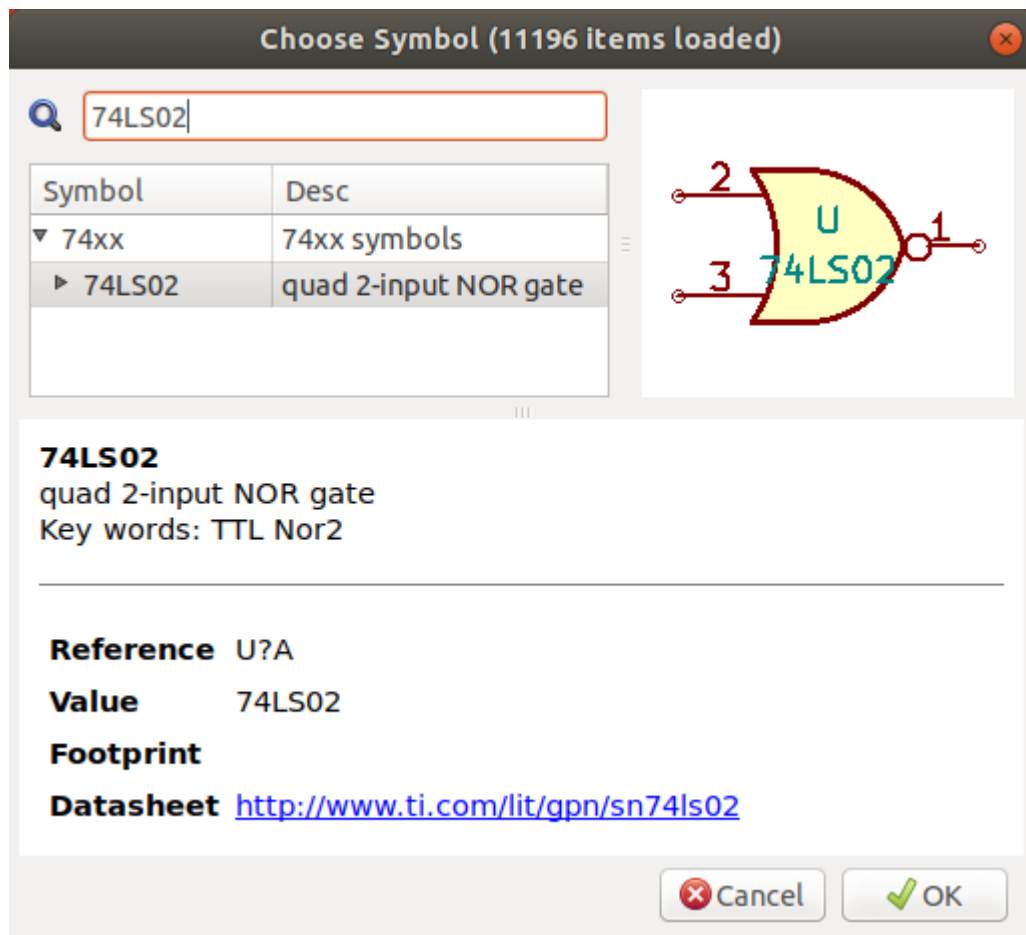
入符号

入允 您将 形添加到正在 的符号中。使用按 像 入符号： 。 入的 形在 有 形中 建 添加。

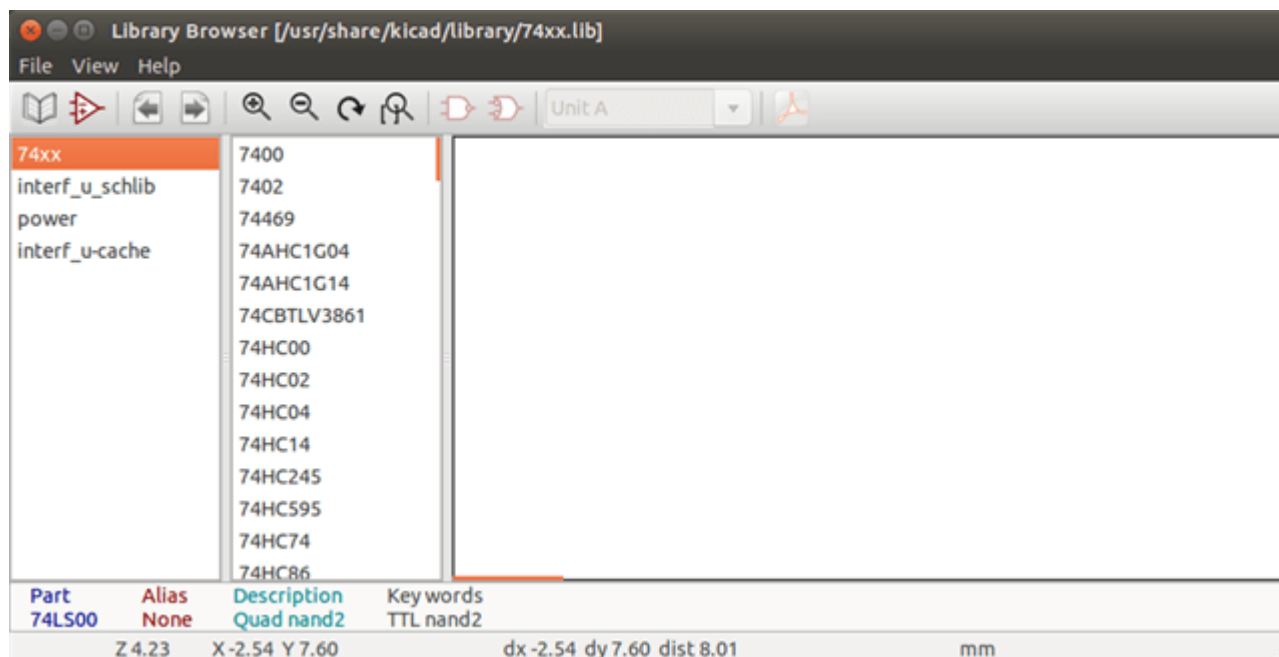
符号器

简介

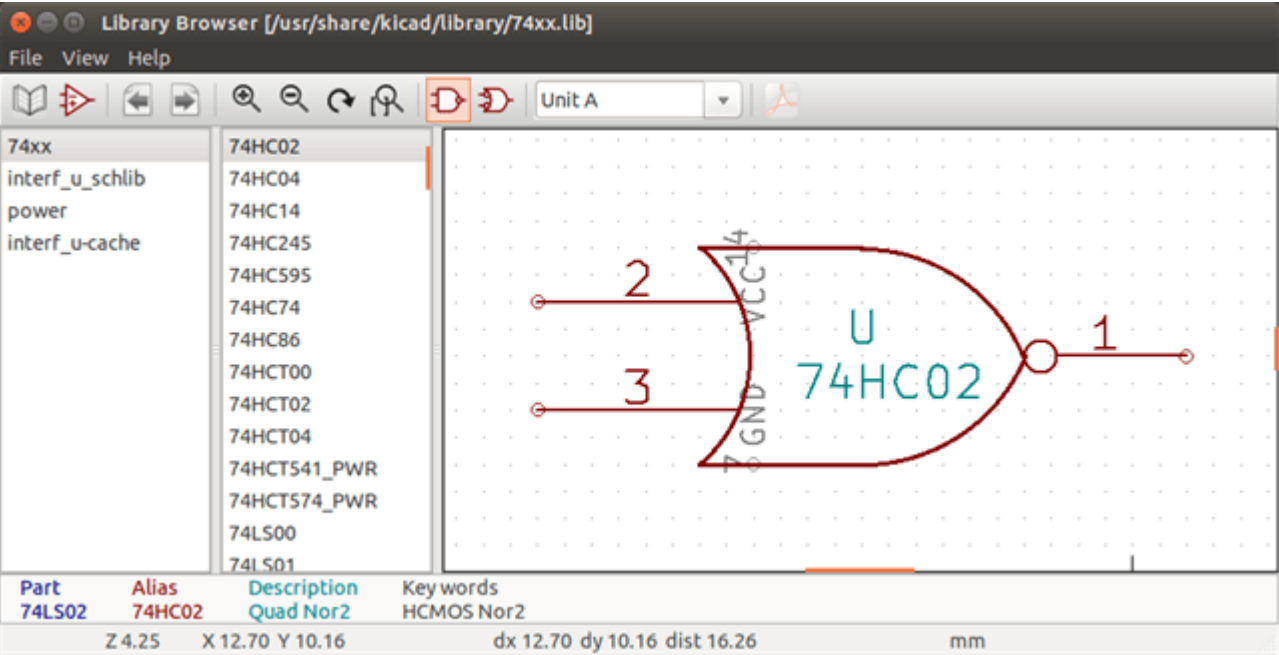
符号器允许您快速查看符号的内容。您可以通过主工具上的  在菜单中查看器条目或双符号上的符号像来查看符号查看器窗口。



-主屏幕



要 的内容， 从左 窗格的列表中 一个 所 中的所有符号都将 示在第二个窗格中。 符号名称以 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：

	所需的 也可以在中 示列表。
	可在 示中 的符号 名
	示上一个符号。
	示下一个符号。
	放工具。
	如果存在， 表示（正常或
	包含多个 位的符号的 位。
	如果存在， 示 的文档。 在被叫 存在 来自 Eeschema 的地方符号 框。
	器并将所 符号放在 Eeschema 中。 只有在从 Eeschema 用 器 才会 示此 双 在元件 器中的符号上）。

建自定义网表和BOM文件

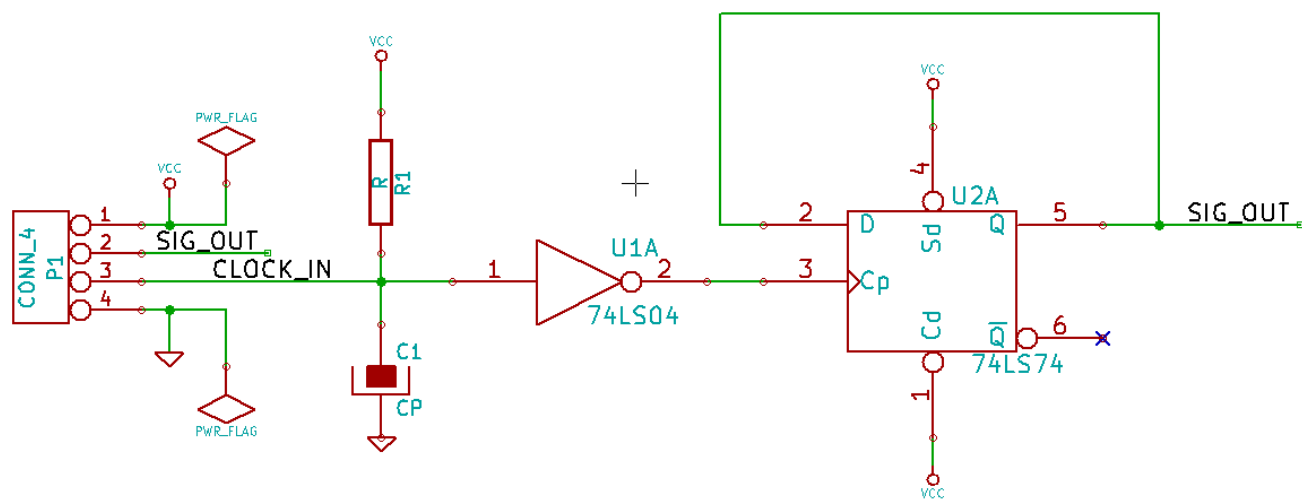
中 网表文件格式

BOM 文件和网表文件可以从 Eeschema 建的中 网表文件

此文件使用 XML 法，称 中 网表。中 网表包含有 您的 路板的大量数据，因此，它可以与后 理一起用于 建 BOM 或其他 告。

根据 出（BOM 或网表），将在后 理中使用完整的中 网表文件的不同子集。

原理 本



中 网表文件示例

上述 路的相 中 网表 (使用 XML 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

通常将后处理器用于中网表文件，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 本身是一种非常适合 XML 的 XML 语言。有一个名为 *xsltproc* 的免费程序，您可以下载并安装。*xsltproc* 程序可用于取中 XML 网表输入文件，用样式表来输入，并将结果保存在输出文件中。使用 *xsltproc* 需要使用 XSLT 定义的样式表文件。完成程序由 Eeschema 处理，在配置一次后以特定方式运行 *xsltproc*。

XSLT 方法

描述 XSL (XSLT) 的文档可在此获得：

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

建立 Pads-Pcb 网表文件

“pads-pcb”的格式由两部分组成。

- 封装列表。
- 网表: 按网络引用行分

接下面是样式表，它将中网表文件 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 的命令行是：

```
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
```

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 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。

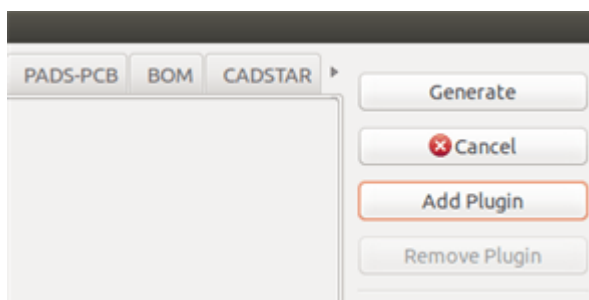
```
( { Eeschema Netlist Version 1.1 29/08/2010 21:07:51
eeschema (2010-08-28 BZR 2458)-unstable}
( 4C6E2141 $noname P1 CONN_4
( 1 VCC )
( 2 /SIG_OUT )
( 3 /CLOCK_IN )
( 4 GND )
)
( 4C6E20BA $noname U2 74LS74
( 1 VCC )
( 2 /SIG_OUT )
( 3 N-04 )
( 4 VCC )
( 5 /SIG_OUT )
( 6 ? )
( 7 GND )
( 14 VCC )
)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

Eeschema 插件界面

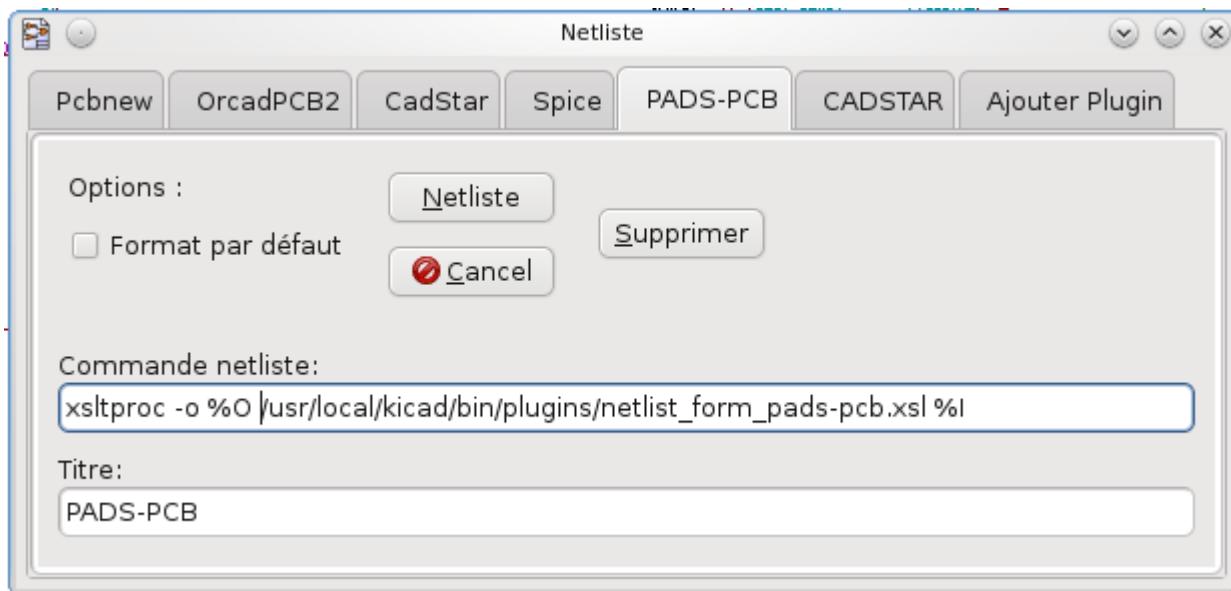
中 网表 器可以在 Eeschema 中自 启

初始化 窗口

可以通 添加插件 按 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

Eeschema 插件配置 框需要以下信息：

- 例如，网表格式的名称。
- 用于启 器的命令行。

网表按 后，将 生以下情况：

1. Eeschema 建了一个中 网表文件 *.xml，例如 test.xml。
2. Eeschema 通 取 test.xml 来运行插件，并 建 test.net。

使用命令行生成网 列表文件

假 我 使用程序 `xsltproc.exe` 将工作表 式 用于中 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"`

`netlist_form_pads-pcb.xml` 是您要 用的 式表。 不要忘 文件名周 的双引号， 允 它 在 Eeschema 替 后有空格。

命令行格式接受文件名的参数：

支持的格式 置参数是。

- %B = 基本文件名和所 出文件的路径， 减去路径和 展名。
- %I = 完整的文件名和 入文件的路径（中 网 文件）。
- %O = 完整的文件名和用 的 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"
```

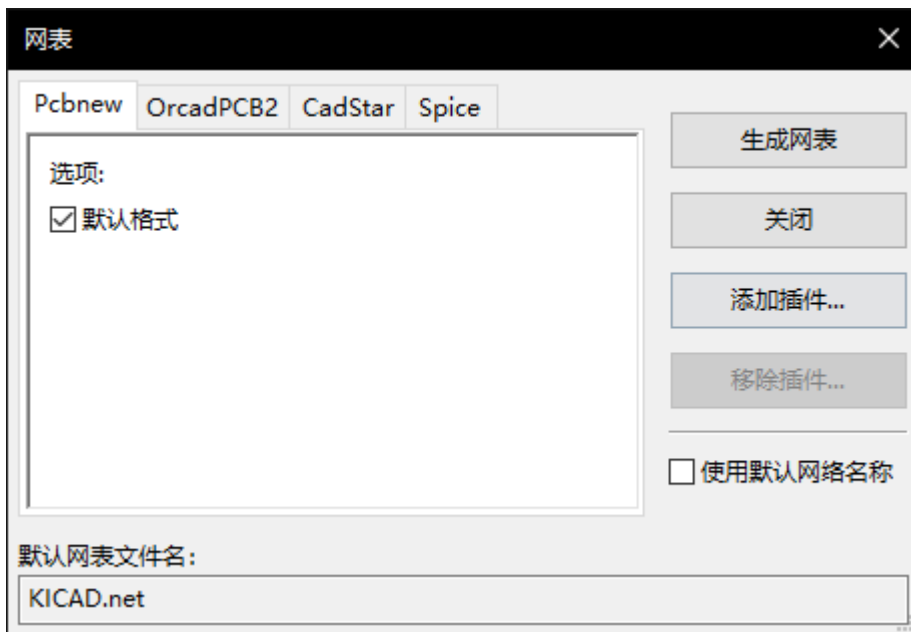
在 Linux 下：

```
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 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 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 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 的名称。
part	此 中的 件名称。
sheetpath	次 构中工作表的路径： 工作表 在完整的原理 次 构中。
tstamps (time stamps)	原理 文件的 戳。
tstamp (time stamp)	元件的 戳。

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 目中的元件，并且不会 失元件与其占用空 之 的 接。

戳是原理 目中每个元件或工作表的唯一 符。但是,在 的 次 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。

次 构中的 定工作表具有唯一 符：其 sheetpath。 定元件（在 次 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 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 包含原理 的 “ 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 列出了原理 中的所有网

可能的网 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 的内部 符
name	是此网 的名称
node	出一个 接到 网 的引脚引用

有 xsltproc 的更多信息

参 面 : <http://xmlsoft.org/XSLT/xsltproc.html>

介

xsltproc 是一个命令行工具，用于将 XSLT 式表 用于 XML 文档。 然它是作 GNOME 目的一部分开 的，但它 可以独立于 GNOME 桌面运行。

从命令行 用 xsltproc，其中包含要使用的 式表的名称，后跟要 用 式表的文件的名称。如果提供的文件名是 -， 它将使用 准 入。

如果 式表包含在 有 式表 理指令的 XML 文档中， 不需要在命令行中命名 式表。xsltproc 将自 包含的 式表 并使用它。默认情况下， 出 *stdout*。您可以使用 -o 指定要 出的文件。

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] | [--nodtdattr]] [ *stylesheet* ] [ *file1* ] [ *file2* ]
[ *....* ]
```

命令行

-V o --version

示使用的 libxml 和 libxslt 的版本。

-v o --verbose

出 xsltproc 在 理 式表和文档 采取的每个步

-o o --output file

直接 出到名 *file* 的文件。 于多个 出，也称 *chunking*，-o directory/ 将 出文件定向到指定的目 目 必 已存 在。

--timing

示用于解析 式表，解析文档和 用 式表并保存 果的 以毫秒 示。

--repeat

运行 20次。用于定

--debug

出 后文档的 XML 以 行

--novalid

跳 加 文档的 DTD。

--noout

不 出 果。

--maxdepth value

在 libxslt 断定它 于无限循 之前 整模板堆 的最大深度。默认 500

--html

入文档是 HTML 文件。

--param name value

将名称 *name* 和 *value* 的参数 式表。您可以 多个名称/ 最多 32。如果 的是字符串而不是 点 符， 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 符。（注意：字符串必 是utf-8。）

--nonet

不要使用互 网来 取 DTD 体或文档。

--path paths

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

--load-trace

向 stderr 示 理期 加 的所有文档。

--catalogs

使用 SGML_CATALOG_FILES 中指定的 SGML 目 来解析外部 体的位置。默认情况下， xsltproc 找 XML_CATALOG_FILES 中指定的目 如果未指定， 使用 /etc/xml/catalog。

--xinclude

使用 Xinclude 范 理 入文档。有 方面的更多 信息， 参 Xinclude 范：
<http://www.w3.org/TR/xinclude/>[<http://www.w3.org/TR/xinclude/>]

--profile --norman

出分析信息， 明 式表的每个部分所花 的 在 化 式表性能 很有用。

--dumpextensions

将所有已注册 展名的列表 到 stdout。

--nowrite

拒 写入任何文件或 源。

--nomkdir

拒 建目

--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 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

Eeschema 使用 [ngspice](#) 作为模型引擎提供嵌入式电路仿真器。

使用模型器 您可能会 官方的 *pspice* 很有用。它包含用于模型的公共符号，如 / 流源或晶体管，其引脚号与 ngspice 点序号范相匹配。

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

分配模型

在启模之前，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。在某种情况下，第一个元件具有指定的模型。

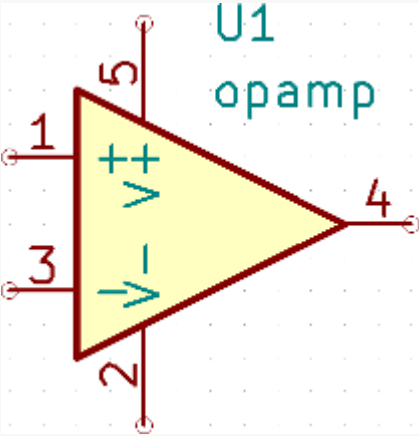
“无源模型”参考匹配 Spice 表示法中的器件型的无源元件（ R^* 表示电阻器， C^* 表示电容器， L^* 表示电感器）将公式分配模型并使用 `value` 字段确定他的属性。

NOTE

注意，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（模型器，模型器框）中请求做。

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或原理器中定义它。打开符号属性框，然后 *Spice 模型* 按钮以打开 Spice 模型器框。

Spice 模型器框有三个用于不同模型类型的卡。所有模型型共有两个

禁用模型的符号	中 元件将从模 中排除。
用 点序列	<p>允 用 将符号引脚覆盖 模型 点映射。要定义不同的映射， 按模型 期的 序指定引脚 号。</p> <p>'例子：'+</p> <p>“* 接：”+ “* 1: 非反相 入”</p> <p>“* 2: 反相 入”</p> <p>“* 3: 正 源”</p> <p>“* 4: 源”</p> <p>“* 5: 出”</p> <p>“.子 路 tl071 1 2 3 4 5”</p>  <p>要将符号引脚与上面 示的 Spice 模型 点相匹配，需要使用具有 的 用 点序列 “1 3 5 2 4”。它是与 Spice 模型 点 序 的引脚 号列表。</p>

无源

无源 卡允 用 将无源器件模型（阻， 容或 感）分配 元件。 是一个很少使用的 因 通常被 元件的模型分配了 模 无源模型， 形， 除非元件引用与 型不匹配。

NOTE

明确定义的被 模型 先于 式分配的模型。 意味着一旦分配了被 模型，在模 期 不会考 参考和 字段。当指定的模型 与原理 上 示的模型 不匹配 可能会 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

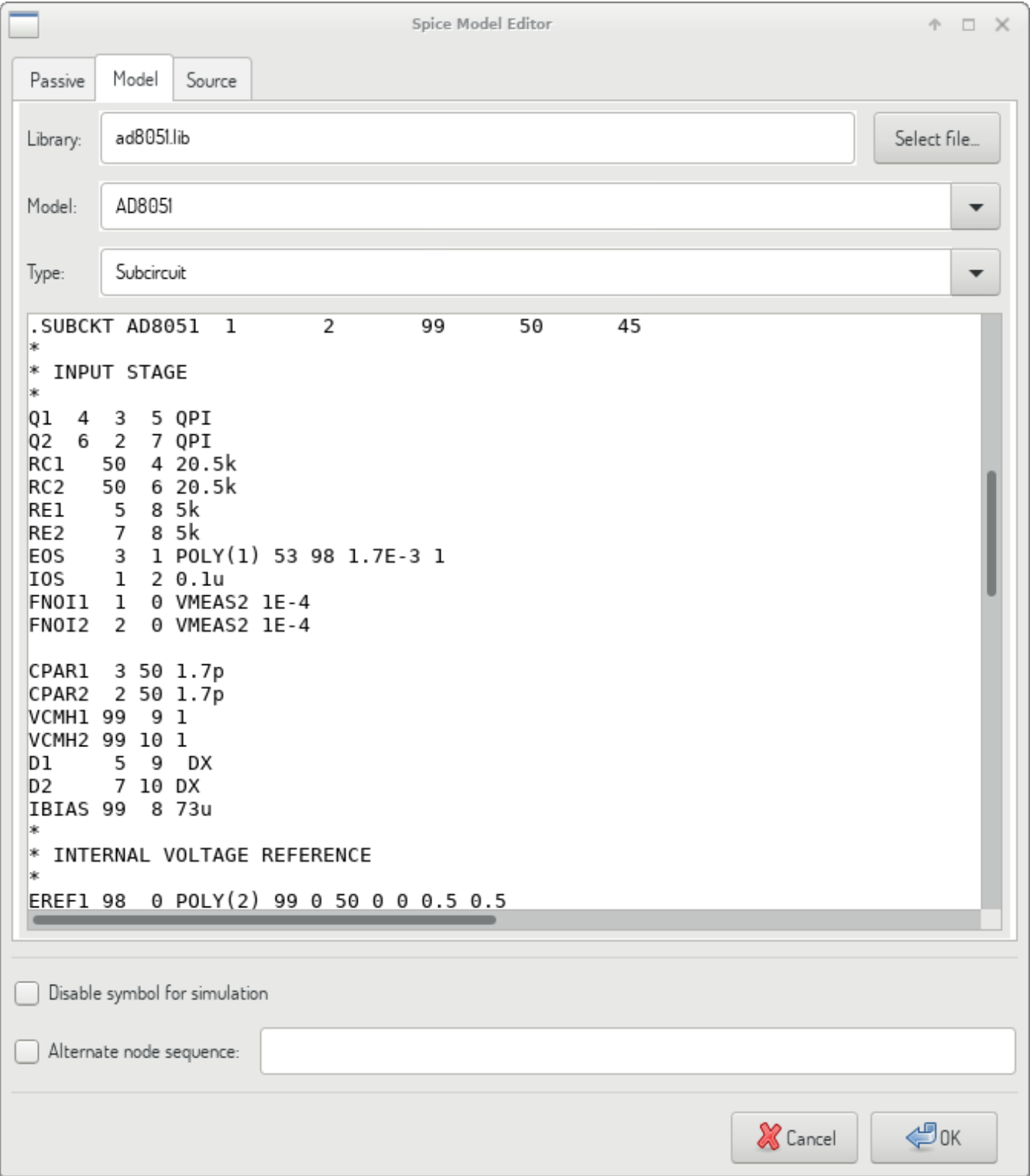
OK

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 模型。 Spice 模型 通常由 制造商提供。

主文本小部件 示所 的文件内容。将模型描述放在 文件中是常 的做法，包括 点 序。



文件	Spice 文件的路径。 文件将由模 器使用， 因 它是使用 .include 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“ 分析”。每个都定义了相 模 型的源参数。

Source type 适用于所有模 型。

Spice Model Editor

PassiveModelSource

DC/AC analysis:

DC:Volts/Amps

AC magnitude:

1

Volts/Amps

AC phase:radians

Transient analysis:

PulseSinusoidalExponentialPiece-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

☐ Disable symbol for simulation

☐ Alternate node sequence:

Cancel

OK

有 源的更多 信息， 参 [ngspice文档](#)， 第4章（ 和 流源）。

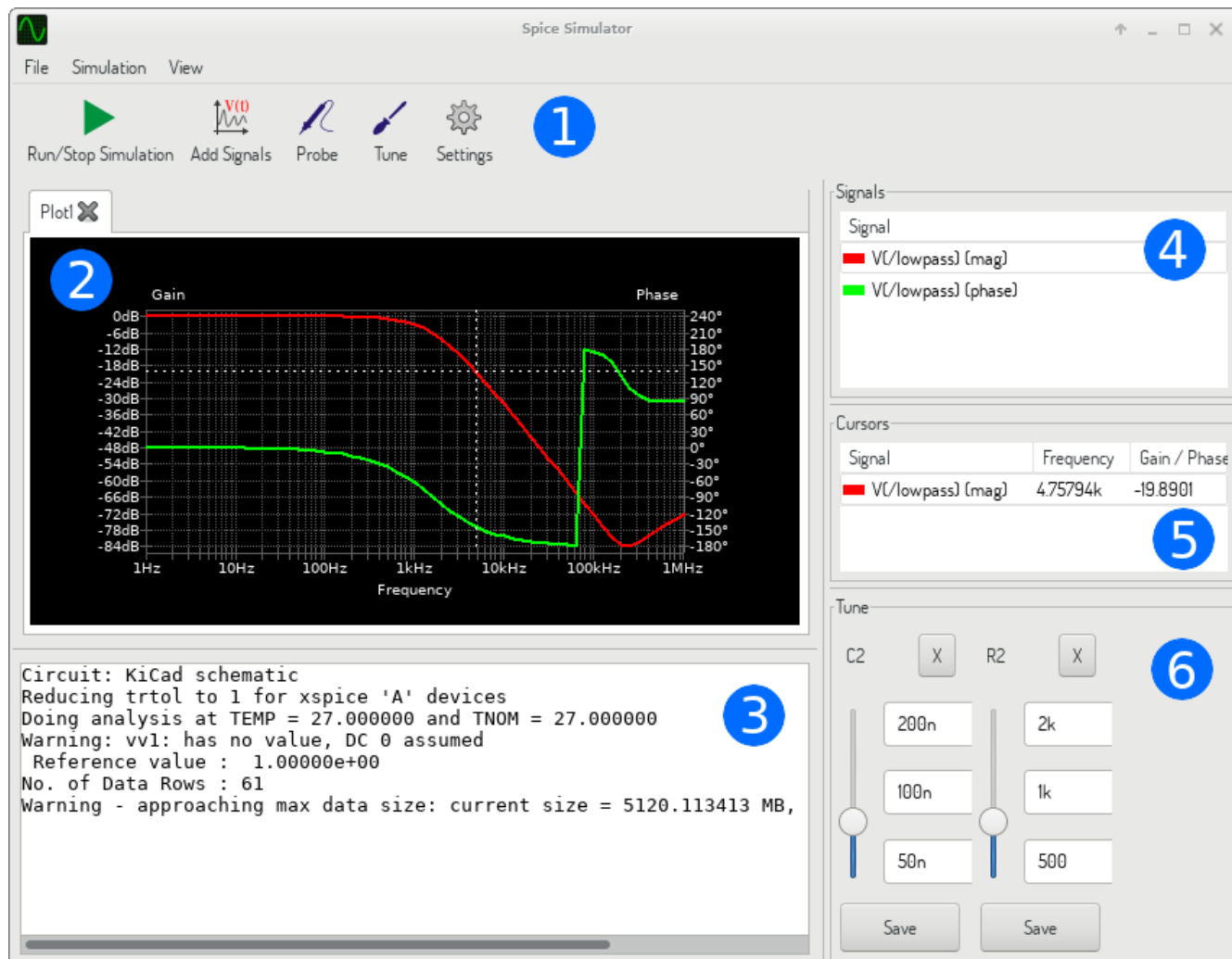
130

Spice 指令

可以通过将 Spice 指令放在原理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，在原理器窗口中，菜单 **工具** → **仿真** 打开 Spice 仿真框。



框分几个部分：

- 《模型 - 工具 工具
- 《模型 面板, 面板》
- 《模型 出控制台, 出控制台》
- 《模型 信号列表, 信号列表》
- 《模型 游 列表, 游 列表》
- 《模型 面板, 面板》

菜单

文件

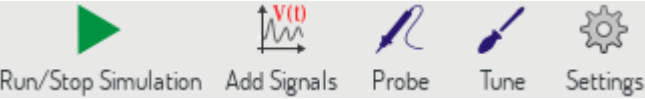
新制	在 面板中 建一个新 卡。
打开工作簿	打开 制信号列表。
保存工作簿	保存 制信号列表。
另存 像	将活 出 .png 文件。
另存 .csv 文件	将活 原始数据点 出到 .csv 文件。
退出模	框。

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	启 原理 “模 探 工具, 探 ” 工具。
整元件	启 “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	启 或停止模
添加信号	打开一个 框以 要 制的信号。
探	启 原理 “模 探 工具, 探 ” 工具。
	启 ”模 工具, “ 工具。
置	打开 “模 置, 模 置 框”。

面板

将模型可视化可以在单独的卡中打开多个，但只有在行模式下才会更新活动，这样就可以比较不同运行的模型。

可以使用“模型”菜单、切片网格和示例库来自定义。当示例可以拖拽它来改变其位置。

面板交互：

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

出控制台

出控制台显示来自模型器的消息。建立控制台输出以确认没有错误或警告。

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换光
- 双击以隐藏信号

游列表

示游列表及其坐。每个信号可以示一个光。使用“模型信号列表，信号”列表置游可能性。

面板

示使用“模型”工具，“工具”取的元件。面板允许快速修改元件并观察它模型果的影响 - 每次更改元件都会重新运行模型并更新形。

于每个元件，有一些控件

- 顶部文本字段置最大元件
- 中间文本字段置的元件
- 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。

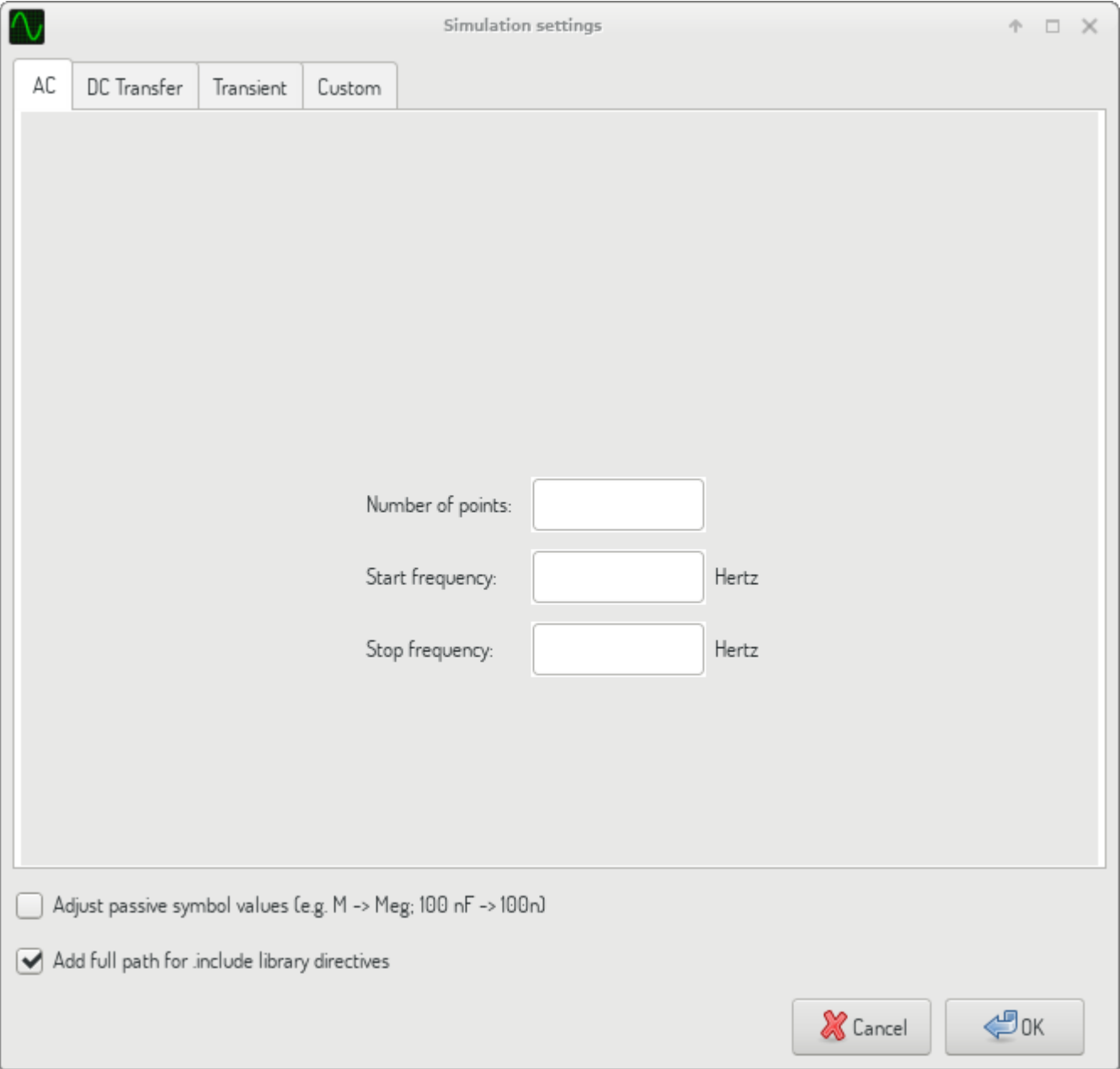
要 要 整的元件， 在工具 于活 状 原理 器中的一个元件。所 元件将出 在“模 工具， ”面板中。只 能 整被 元件。

探 工具

探 工具提供了一种用 友好的方式来 用于 的信号。

要向 添加信号， 在工具 于活 状 原理 器中的相

仿真 置



模 置 框允 用 置模 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 参数的表 最后一个 卡允 用 入自定义 Spice 指令以 置模 有 仿真 型和参数的更多信息， 参 [ngspice文档](#)，第1.2章。

配置模 的另一种方法是在原理 上的文本字段中 入“模 指令，Spice 指令”。与模 型相 的任何文本字段指令都会被 框中 的 置覆盖。 意味着一旦开始使用模 框， 框将覆盖原理 指令，直到重新打开模 器。

所有模 型共有两个

整被 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。