



Eeschema

31 октября 2021 г.

Оглавление

1	Общая информация об Eeschema	1
1.1	Описание	1
1.2	Технические характеристики	2
2	Основные команды Eeschema	3
2.1	Команды мыши	4
2.1.1	Основные команды	4
2.1.2	Операции над блоками	4
2.2	Горячие клавиши	5
2.3	Сетка	7
2.4	Изменение масштаба	7
2.5	Отображение координат курсора	8
2.6	Основное меню	8
2.7	Верхняя панель инструментов	8
2.8	Правая панель инструментов	10
2.9	Левая панель инструментов	11
2.10	Контекстное меню и быстрое редактирование	11
3	Основное меню	14
3.1	Меню "Файл"	14
3.2	Меню "Настройки"	15
3.2.1	Менеджер библиотек компонентов	16
3.2.1.1	Добавление новой библиотеки	16
3.2.1.2	Удаление библиотеки	16
3.2.1.3	Настройка библиотек	17

3.2.2	Общие настройки	18
3.2.2.1	Отображение	18
3.2.2.2	Редактирование	19
3.2.2.3	Управление	20
3.2.2.4	Цвета	21
3.2.2.5	Поля по умолчанию	21
3.3	Меню "Справка"	22
4	Верхняя панель инструментов	23
4.1	Настройки страницы	23
4.2	Инструмент поиска	23
4.3	Инструмент формирования списка цепей	24
4.4	Инструмент обозначения элементов	26
4.5	Инструмент проверки электрических правил проектирования	27
4.5.1	Диалоговое окно проверки электрических правил проектирования	28
4.5.2	Настройка электрических правил проектирования	29
4.6	Инструмент формирования перечня элементов	30
4.7	Инструмент редактирования полей	32
4.7.1	Приёмы быстрого заполнения полей	33
4.8	Инструмент импорта значений посадочных мест	34
4.8.1	Доступ	34
5	Менеджер библиотек компонентов	35
5.1	Таблица библиотек компонентов	35
5.1.1	Глобальная таблица библиотек компонентов	36
5.1.2	Таблицы библиотек компонентов проекта	36
5.1.3	Начальная настройка	36
5.1.4	Добавление элементов таблицы	37
5.1.5	Подстановка переменных окружения	37
5.1.6	Варианты использования	38
5.1.7	Загрузка устаревших проектов	38

6	Создание и редактирование схем	40
6.1	Основные положения	41
6.2	Цикл разработки	42
6.3	Размещение и редактирование компонентов	42
6.3.1	Поиск и размещение компонента	42
6.3.2	Символы питания	44
6.3.3	Редактирование размещённых компонентов	44
6.3.3.1	Изменение компонента	44
6.3.3.2	Изменение значения полей	45
6.4	Проводники, шины, метки, символы питания	45
6.4.1	Соединения с помощью проводников и меток	46
6.4.2	Соединения с помощью шин	47
6.4.2.1	Сигналы шины	48
6.4.2.2	Соединение сигналов шины	48
6.4.2.3	Глобальные соединения между шинами	49
6.4.3	Подключение символов питания	49
6.4.4	Флаг «Не подключено»	50
6.5	Чертежные элементы	51
6.5.1	Текстовые комментарии	51
6.5.2	Основная надпись	51
6.6	Восстановление кэшированных компонентов	52
7	Иерархические схемы	54
7.1	Навигация по иерархии	56
7.2	Локальные, иерархические и глобальные метки	56
7.2.1	Свойства	56
7.3	Создание иерархических схем	57
7.4	Символ листа	57
7.5	Иерархические выводы	57
7.6	Иерархические метки	59
7.6.1	Локальные, иерархические, глобальные метки и скрытые выводы питания	60
7.6.1.1	Обычные метки	60
7.6.1.2	Иерархические метки	60

7.6.1.3	Скрытые выводы питания	61
7.6.2	Глобальные метки	61
7.7	Сложные иерархии	61
7.8	Плоские иерархии	62
8	Инструмент обозначения элементов	65
8.1	Примеры работы инструмента	67
8.1.1	Порядок обозначения	67
8.1.2	Выбор варианта обозначений	68
9	Проверка электрических правил проектирования	71
9.1	Как производить проверку правил	72
9.2	Пример выполнения проверки правил	73
9.3	Получение информации об ошибках	73
9.4	Выводы питания и флаги питания	74
9.5	Настройка	75
9.6	Отчёт проверки правил	76
10	Создание списка соединений	77
10.1	Обзор	77
10.2	Форматы списков соединений	77
10.3	Примеры списков соединений	78
10.4	Замечания о списках соединений	81
10.4.1	Предостережение об именовании элементов списка соединений	81
10.4.2	Списки соединений в формате PSPICE	81
10.5	Прочие форматы	83
10.5.1	Подготовка диалогового окна	83
10.5.2	Формат командной строки	84
10.5.3	Конвертер и таблица стилей (плагин)	84
10.5.4	Формат временного файла списка соединений	84

11 Черчение и печать	85
11.1 Основные команды печати	86
11.2 Чертить в формате Postscript	86
11.3 Чертить в формате PDF	87
11.4 Чертить в формате SVG	88
11.5 Чертить в формате DXF	88
11.6 Чертить в формате HPGL	89
11.6.1 Установка размера листа	89
11.6.2 Настройка смещения	89
11.7 Печать на бумаге	90
12 Просмотр библиотек компонентов	91
12.1 Общая информация о библиотеках компонентов	91
12.2 Обзор библиотеки компонентов	91
12.3 Обзор редактора библиотеки компонентов	92
12.3.1 Основная панель инструментов	92
12.3.2 Панель инструментов чертёжных элементов	94
12.3.3 Панель инструментов параметров	95
12.4 Выбор и управление библиотекой	95
12.4.1 Выбор и сохранение компонента	95
12.4.1.1 Выбор компонента	95
12.4.1.2 Сохранение компонента	96
12.4.1.3 Перемещение компонента в другую библиотеку	96
12.4.1.4 Отмена изменений компонента	97
12.5 Создание компонентов библиотеки	97
12.5.1 Создание нового компонента	97
12.5.2 Создание компонента на основе другого	98
12.5.3 Свойства компонента	99
12.5.4 Компоненты с альтернативным начертанием	100
12.6 Графические элементы	101
12.6.1 Графические элементы в частях и дополнительных начертаниях	101
12.6.2 Графический текст	103
12.7 Компоненты из нескольких частей и с дополнительными начертаниями	103

12.7.1	Пример компонента из нескольких частей и с несколькими начертаниями:	103
12.7.1.1	Элементы графического обозначения	105
12.8	Создание и редактирование выводов	105
12.8.1	Общие сведения о выводах	106
12.8.2	Свойства выводов	106
12.8.3	Графическое представление выводов	107
12.8.4	Электрический тип выводов	107
12.8.5	Глобальные свойства выводов	108
12.8.6	Установка выводов в компонентах с частями и альтернативными начертаниями	108
12.9	Поля компонентов	109
12.9.1	Редактирование полей компонентов	109
12.10	Символы питания	111
13	LibEdit - Редактор компонентов	113
13.1	Обзор	113
13.2	Установка точки привязки	114
13.3	Псевдонимы	115
13.4	Поля компонента	115
13.5	Документация компонента	116
13.5.1	Ключевые слова компонента	117
13.5.2	Документация компонента	117
13.5.3	Связанный файл документации	118
13.5.4	Фильтры посадочных мест для CvPcb	118
13.6	Библиотека символов	119
13.6.1	Экспорт (создание) символа	119
13.6.2	Импорт символа	120
14	Просмотрщик библиотек компонентов	121
14.1	Главное окно	123
14.2	Верхняя панель инструментов просмотрщика компонентов библиотеки	124

15 Создание списков цепей и перечней элементов в различных форматах	126
15.1 Промежуточный файл списка цепей	126
15.1.1 Пример схемы	127
15.1.2 Пример промежуточного файла списка цепей	127
15.2 Преобразование списка цепей в другой формат	131
15.3 Основы XSLT	131
15.3.1 Создание списка цепей в формате Pads-Pcb	131
15.3.2 Создание списка цепей в формате Cadstar	134
15.3.3 Создание списка цепей в формате OrcadPCB2	137
15.3.4 Подключение плагинов в Eeschema	142
15.3.4.1 Настройка диалогового окна	143
15.3.4.2 Параметры плагина	143
15.3.4.3 Создание списка цепей с помощью командной строки	144
15.3.4.4 Формат командной строки: пример для xsltproc	144
15.3.5 Создание перечня элементов	144
15.4 Формат командной строки: пример сценария на языке Python	145
15.5 Структура промежуточного файла списка цепей	145
15.5.1 Основные элементы файла списка цепей	147
15.5.2 Оглавление	147
15.5.3 Компоненты	148
15.5.3.1 Замечание о метке времени компонента	148
15.5.4 Части компонента	148
15.5.5 Библиотеки	149
15.5.6 Цепи	150
15.6 Более подробно о xsltproc	150
15.6.1 Формат командной строки	151
15.6.2 Параметры командной строки	151
15.6.3 Возвращаемые значения xsltproc	153
15.6.4 Больше подробностей о xsltproc	153

16	Симулятор	154
16.1	Назначение моделей	154
16.1.1	Пассивный	155
16.1.2	Модель	157
16.1.3	Источник	158
16.2	Директивы Spice	159
16.3	Симуляция	160
16.3.1	Меню	161
16.3.1.1	Меню "Файл"	161
16.3.1.2	Симуляция	161
16.3.1.3	Меню "Вид"	161
16.3.2	Панель инструментов	161
16.3.3	Панель графиков	162
16.3.4	Сообщения консоли	162
16.3.5	Перечень сигналов	162
16.3.6	Перечень курсоров	162
16.3.7	Панель подстройки	163
16.3.8	Подстройка	163
16.3.9	Пробник	163
16.3.10	Параметры симуляции	164

Руководство пользователя

Авторские права

Авторские права © 2010-2018 на данный документ принадлежит его разработчикам (соавторам), перечисленным ниже. Документ можно распространять и/или изменять в соответствии с правилами лицензии GNU General Public License (<http://www.gnu.org/licenses/gpl.html>), версии 3 или более поздней, или лицензии типа Creative Commons Attribution License (<http://creativecommons.org/licenses/by/3.0/>), версии 3.0 или более поздней.

Все торговые знаки этого руководства принадлежат его владельцам.

Соавторы

Jean-Pierre Charras, Fabrizio Tappero.

Перевод

Барановский Константин <baranovskiykonstantin@gmail.com>, 2016-2019

Обратная связь

Оставить свои комментарии или замечания можно на следующих ресурсах:

- О документации KiCad: <https://gitlab.com/kicad/services/kicad-doc/issues>
- О программном обеспечении KiCad: <https://gitlab.com/kicad/code/kicad/issues>
- О переводе программного обеспечения KiCad: <https://gitlab.com/kicad/code/kicad-i18n/issues>

Дата публикации

30 мая 2015 года

Глава 1

Общая информация об Eeschema

1.1 Описание

Eeschema - это приложение для создания электрических схем, которое является частью САПР KiCad, и доступно для пользователей следующих операционных систем:

- Linux
- Apple macOS
- Windows

Не зависимо от типа операционной системы все файлы Eeschema полностью совместимы.

Eeschema - это прикладное программное обеспечение в котором содержится весь необходимый функционал для черчения схем, настройки, управления библиотеками и генерации необходимых данных для дальнейшей передачи в приложения проектирования печатных плат.

Eeschema предназначена для работы совместно с трассировщиком печатных плат из комплекта приложений САПР KiCad - PcbNew. Но, несмотря на это, есть возможность генерации списка соединений (Netlist) других форматов, для совместной работы с другими САПР.

Eeschema включает в себя редактор библиотеки компонентов, который позволяет создавать, редактировать компоненты и управлять библиотеками. Также имеются дополнительные, но очень важные, инструменты, необходимые при проектировании современных электрических схем:

- Проверка электрических правил проектирования (ERC) для автоматического обнаружения неверных или пропущенных соединений.
 - Экспорт изображения схемы в файлы различных форматов (Postscript, PDF, HPGL и SVG).
 - Генератор перечня элементов (BOM) различных форматов с помощью внешних инструментов.
-

1.2 Технические характеристики

Eeschema ограничена лишь доступным объемом памяти. В ней нет абсолютно никаких ограничений ни на количество компонентов, ни на количество выводов, ни на количество листов. В случае многолистных схем, они представляются в виде иерархии.

В Eeschema многолистные схемы могут быть следующих типов:

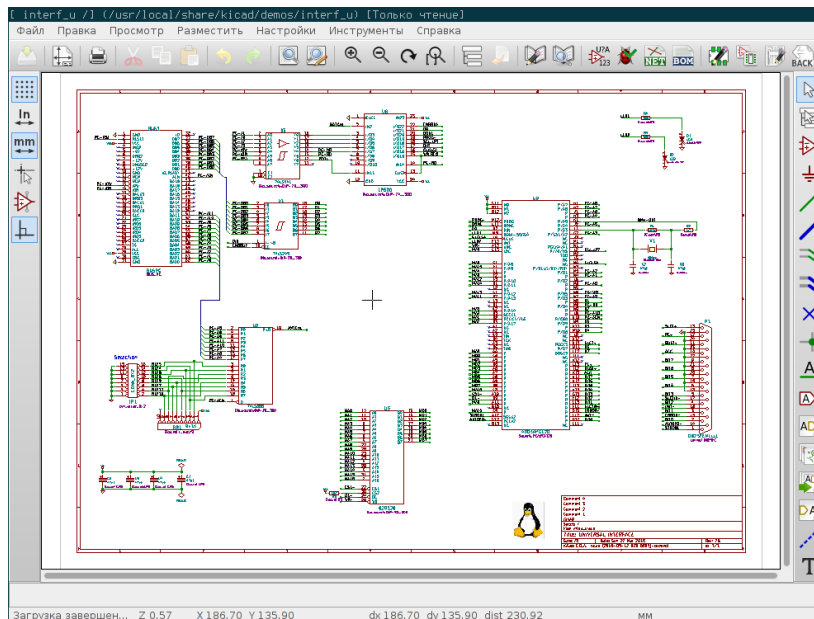
- Простые иерархии (каждый лист используется лишь один раз).
 - Сложные иерархии (некоторые листы используются два и более раз).
 - Плоские иерархии (иерархические листы не связаны явным образом на корневом листе схемы).
-

Глава 2

Основные команды Eeschema

Команду можно вызвать из:

- основного меню (в верхней части окна);
 - верхней панели инструментов (основные команды);
 - правой панели инструментов (вспомогательные команды или "инструменты");
 - левой панели инструментов (параметры отображения);
 - кнопок мыши (важные дополнительные команды). В частности, правая кнопка мыши открывает контекстное меню для элемента под курсором (масштаб, сетка и команды редактирования элементов);
 - функциональных кнопок (F1, F2, F3, F4, Insert и Пробел). В особенности: Escape - позволяет прервать выполнение текущей команды. Insert - создать копию последнего добавленного компонента.
 - горячих клавиш, которые, обычно, выбирают один из инструментов и задействуют его в месте текущего положения курсора. Для просмотра перечня горячих клавиш, откройте меню "Справка→Список горячих клавиш" или нажмите клавишу *Ctrl+F1*.
-



2.1 Команды мыши

2.1.1 Основные команды

Левая кнопка

- Одинарный щелчок: отобразить в строке состояния характеристики компонента или текста под курсором.
- Двойной щелчок: редактировать (если редактирование возможно) компонент или текст.

Правая кнопка

- Открывает контекстное меню.

2.1.2 Операции над блоками

В Eeschema можно перемещать, перетаскивать, копировать или удалять выделенные области с помощью контекстного меню.

Нужные области выделяются путем перетаскивания курсора мыши с нажатой левой кнопкой.

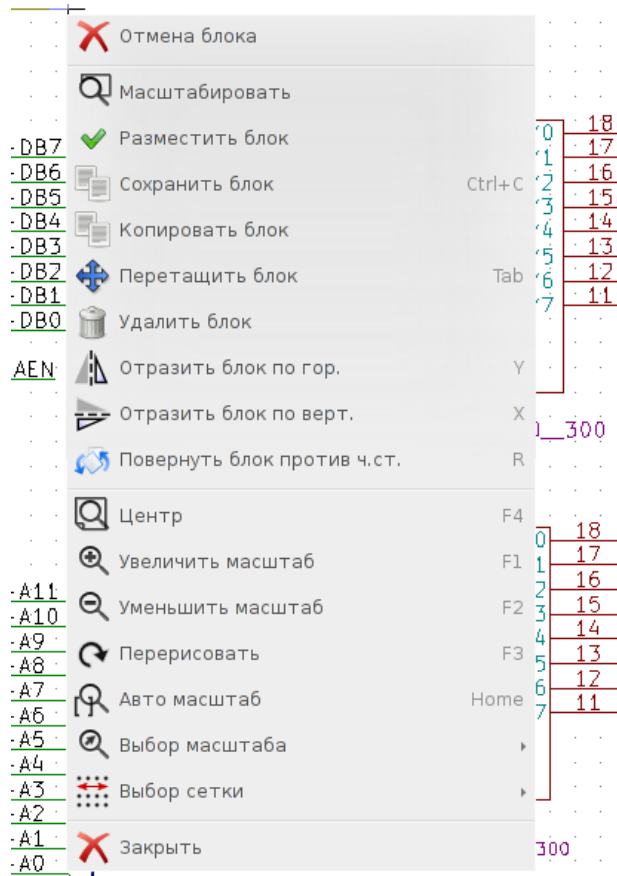
Нажимая клавишу «Shift», «Ctrl» или «Shift + Ctrl» в процессе выделения можно сразу копировать, перетаскивать или удалять, соответственно.

Левая кнопка мыши	Переместить выделенную область.
Shift + левая кнопка мыши	Копировать выделенную область.
Ctrl + левая кнопка мыши	Перетащить выделенную область.
Ctrl + Shift + левая кнопка мыши	Удалить выделенную область.

При перетаскивании или копировании можно:

- нажать левую кнопку мыши, чтобы расположить выделенный блок;
- нажать правую кнопку мыши или клавишу Esc, чтобы отменить текущее действие.

При перемещении выделенного блока можно выбрать другую операцию из контекстного меню (правая кнопка мыши).



2.2 Горячие клавиши

- Чтобы просмотреть текущий список горячих клавиш нужно нажать "Ctrl+F1" на клавиатуре.
- Горячие клавиши можно отредактировать на вкладке Управление в диалоговом окне Параметров редактора схем (меню Настройки → Общие настройки).

Ниже показан список горячих клавиш со значениями по умолчанию.

Помощь (данное окно)	Ctrl+F1
Увеличить	F1
Уменьшить	F2
Перерисовать	F3
По центру	F4
Вписать в экран	Home

Увеличить выбранное	@
Сбросить локальные координаты	Space
Редактировать элемент	E
Удалить элемент	Del
Повернуть элемент	R
Перетащить элемент	G
Отмена	Ctrl+Z
Повтор	Ctrl+Y
Щелчок левой кнопки мыши	Return
Двойной щелчок левой кнопки мыши	End
Сохранить схему	Ctrl+S
Загрузить схему	Ctrl+L
Искать элемент	Ctrl+F
Искать следующий элемент	F5
Искать следующий DRC маркер	Shift+F5
Найти и заменить	Ctrl+Alt+F
Повторить последнее действие	Ins
Переместить блок → Перетащить блок	Tab
Копировать блок	Ctrl+C
Вставить блок	Ctrl+V
Вырезать блок	Ctrl+X
Переместить элемент схемы	M
Дубликат компонента или метки	C
Добавить компонент	A
Добавить питание	P
Отразить по верт.	X
Отразить по гориз.	Y
Ориентировать компонент нормально	N
Редактировать значение компонента	V
Редактировать обозначение компонента	U
Редактировать посад.место компонента	F
Редактировать в редакторе компонентов	Ctrl+E
Начать проводник	W
Начать шину	B
Завершить линию, проводник, шину	K
Добавить метку	L
Добавить иерархическую метку	H
Добавить глобальную метку	Ctrl+L
Добавить соединение	J
Добавить флаг "Не подключено"	Q
Добавить лист	S
Добавить ввод проводника	Z
Добавить ввод шины	/

Добавить графическую линию	I
Добавить графический текст	T
Обновить плату со схемы	F8
Авторазмestить поля	O
Покинуть лист	Alt+BkSp
Удалить узел	BkSp
Подсветить соединение	Ctrl+X

Любую горячую клавишу можно изменить используя редактор горячих клавиш (меню Настройки→Общие настройки→[Управление](#)

Существует возможность импорта/экспорта установленных горячих клавиш через меню Настройки→Импорт/Экспорт горячих клавиш.

2.3 Сетка

В Eeschema курсор всегда перемещается по сетке. Шаг сетки можно менять:

- Изменить шаг сетки можно через контекстное меню или основное меню "Настройки"→"Параметры".
- Цвет можно изменить на вкладке Цвета в диалоговом окне Параметров редактора схем (меню Настройки → Общие настройки).
- Видимость сетки можно переключить используя кнопку на левой панели инструментов.

По умолчанию шаг сетки равен 50 мил (0.050") или 1,27 мм.

Это предпочтительный размер сетки для расположения компонентов и проводников на схеме, а также для размещения выводов при создании компонентов в Редакторе библиотек компонентов.

Допускается работа и с сетками меньшего размера от 25 мил до 10 мил. Это приемлемо при построении символа компонента или расположении текста и комментариев, но нежелательно при размещении выводов или проводников.

2.4 Изменение масштаба

Изменить масштаб можно с помощью:















- Правой кнопки мыши: откроется контекстное меню из которого можно выбрать нужный масштаб.
- Функциональных клавиш:
 - F1: Увеличить масштаб
 - F2: Уменьшить масштаб
 - F4 или щелчок средней кнопкой мыши (без перемещения курсора): Отцентрировать изображение схемы по позиции курсора
- Колеса мыши:







	Создать новую схему (только в автономном режиме).
	Открыть схему (только в автономном режиме).
	Сохранить все (включая иерархические) листы.
	Настроить параметры листа и заполнить основную надпись.
	Открыть диалог печати.
	Вставить скопированный/вырезанный элемент или блок в текущий лист.
	Отменить последнее изменение.
	Повторить предыдущее изменение.
	Открыть диалог для поиска компонентов и текста в схеме.
	Открыть диалог для поиска и замены текста в схеме.
	Обновить изображение схемы. Вместить схему в окно программы.
	Масштаб (увеличить, уменьшить).
	Навигация по иерархическим листам схемы.
	Покинуть текущий лист и перейти на уровень выше.
	Открыть редактор компонентов для просмотра и редактирования библиотек и компонентов.
	Просмотреть библиотеки компонентов.
	Обозначить компоненты схемы.
	Выполнить проверку электрических правил проектирования (ERC).
	Запустить CvPcb для назначения посадочных мест компонентам схемы.
	Сформировать список цепей (для Pcbnew, SPICE и других).
	Редактировать поля компонента.
	Сформировать перечень элементов (BOM).
	Запустить Pcbnew для проектирования печатной платы.
	Загрузить значения посадочных мест компонентов (выбранных в CvPcb) в соответствующие поля "Посадочное место".

2.8 Правая панель инструментов

Эта панель содержит инструменты для:







- Размещения компонентов, проводников, шин, соединений, меток, текста и т.п.
- Создания иерархических листов и их выводов

	Прервать выполнение текущей команды или инструмента.
	Подсветить цепь, выделяя проводники и метки особым цветом. Если KiCad работает в режиме проекта, то в Pcbnew будут подсвечены медные проводники, относящиеся к выделенной цепи.
	Открыть диалог выбора компонента для добавления в схему.
	Открыть диалог выбора символа питания для добавления в схему.
	Чертить проводник.
	Чертить шину.
	Разместить соединение проводника с шиной. На самом деле этот элемент носит лишь графический характер и не создает никаких соединений, поэтому с его помощью нельзя соединять проводники между собой.
	Разместить соединение шины с шиной.
	Разместить флаг "Не подключено". Этот флаг устанавливается на выводах, которые умышленно оставлены без соединений. Делается это для избежания предупреждений при выполнении проверки электрических правил проектирования.
	Разместить соединение. Используется для соединения пересекающихся проводников или проводника с выводом в местах где это соединение не очевидно (например, когда один конец проводника или вывода упирается в другой проводник, а не соединяется с его концом).
	Разместить локальную метку (имя цепи). Два проводника с одинаковыми метками и расположенные на одном листе электрически соединены между собой. Для соединения проводников при помощи меток на разных листах нужно использовать глобальные или иерархические метки.
	Разместить глобальную метку. Все проводники (включая иерархические листы) с одинаковыми глобальными метками электрически соединены между собой.
	Разместить иерархическую метку. Они позволяют соединять проводники между иерархическим листом и, содержащим его, родительским листом.
	Разместить иерархический лист. При создании нужно указать имя файла для этого листа.

	Импорт иерархических меток из иерархического листа. Данный инструмент применим только к иерархическим листам. Он создает выводы на символе иерархического листа для каждой иерархической метки, содержащейся внутри этого листа.
	Разместить вывод на символе иерархического листа. Данный инструмент применим только к иерархическим листам. Он создаёт вывод с указанным именем на символе иерархического листа.
	Чертить линию. Добавляет графическую линию, которая не вносит никаких электрических соединений.
	Разместить текстовый комментарий.
	Разместить графическое изображение.
	Удалить выбранный элемент.

2.9 Левая панель инструментов

Эта панель позволяет настроить параметры отображения:

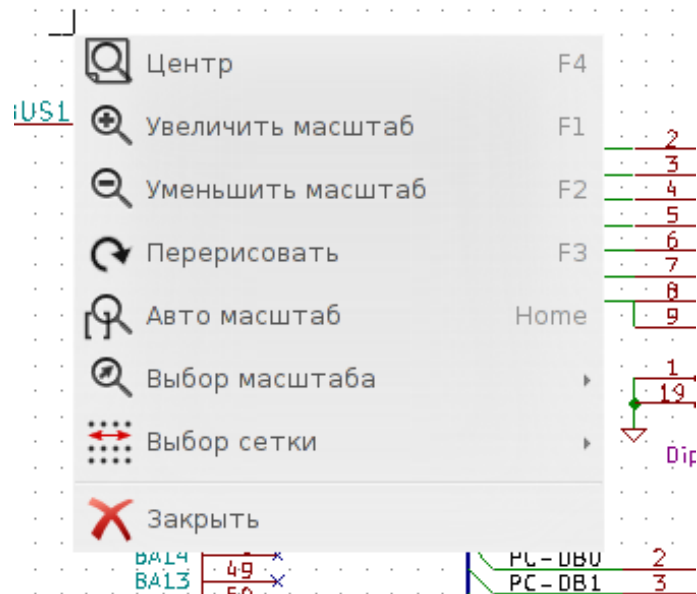
	Toggle grid visibility.
	Switch units to inches.
	Switch units to millimeters.
	Choose the cursor shape (full screen/small).
	Toggle visibility of "invisible" pins.
	Toggle free angle/90 degrees wires and buses placement.

2.10 Контекстное меню и быстрое редактирование

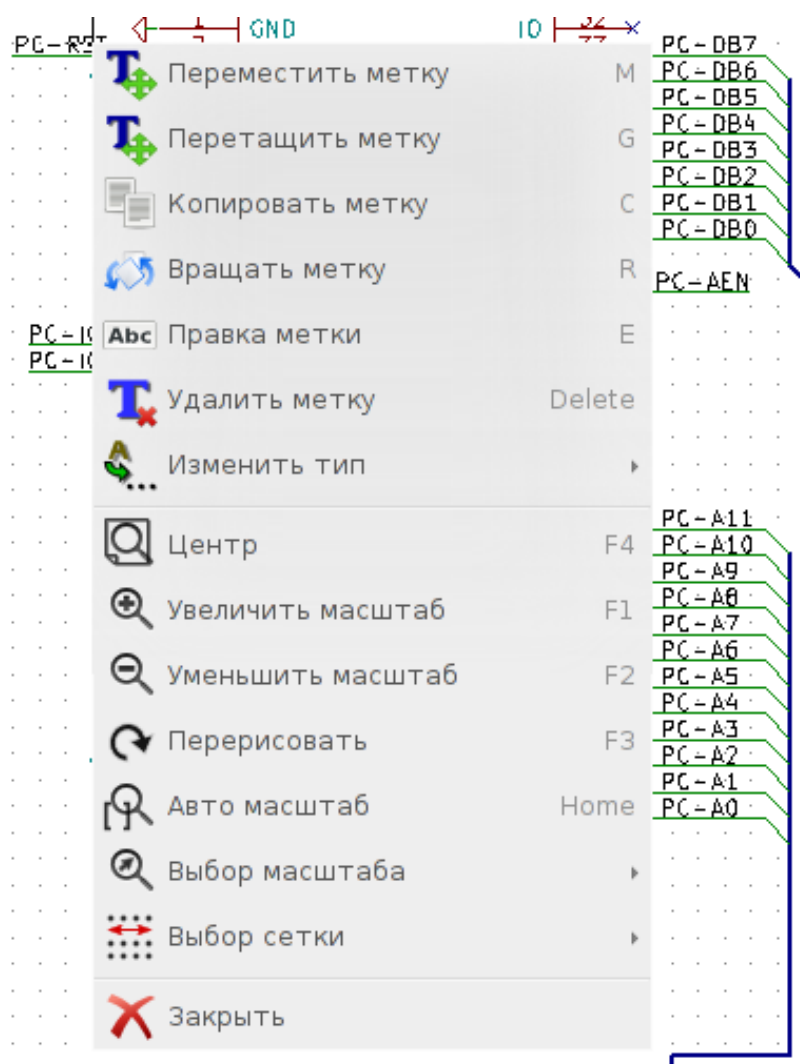
Щелчок правой кнопки мыши открывает контекстное меню, которое позволит:

- Изменить масштаб.
- Настроить сетку.
- Редактировать основные параметры выделенного элемента.

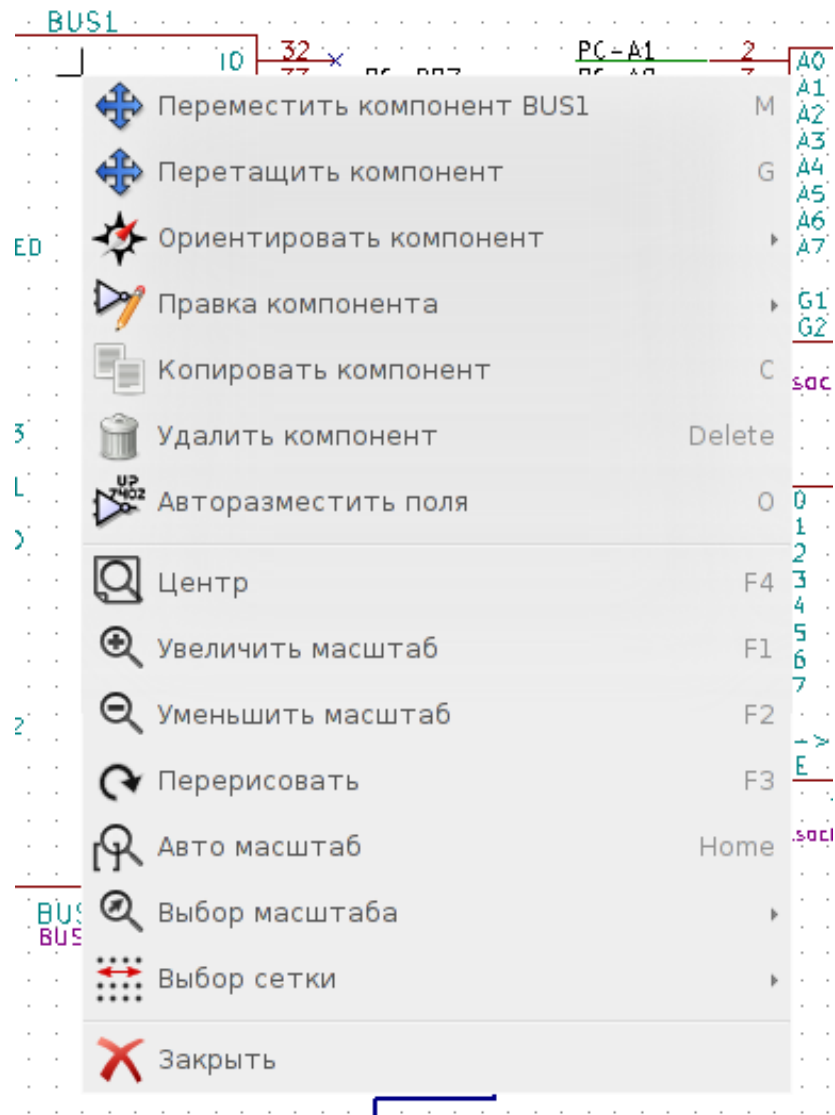
Обычное контекстное меню (без выделенных элементов).



Контекстное меню для метки.



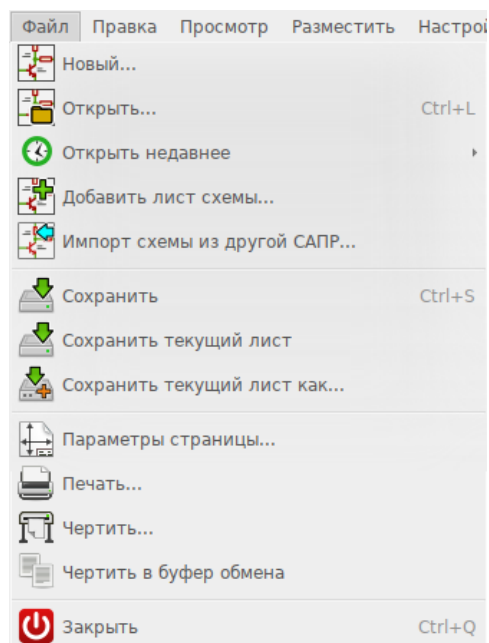
Контекстное меню редактирования компонента.



Глава 3

Основное меню

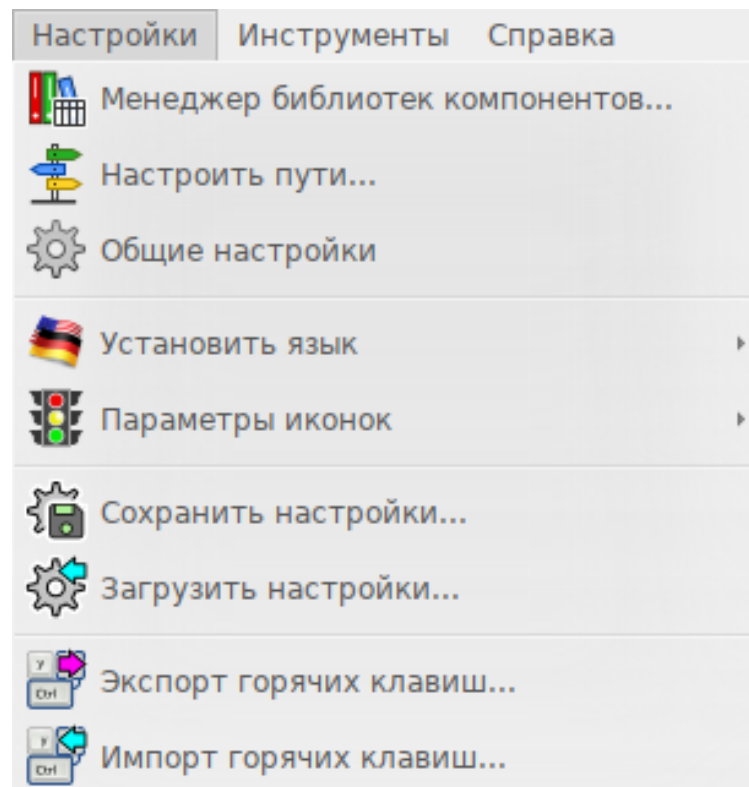
3.1 Меню "Файл"



Новый	Закреть текущую схему и создать новую (только в автономном режиме).
Открыть	Загрузить проект схемы (только в автономном режиме).
Открыть недавнее	Открыть проект схемы из списка недавно используемых файлов (только в автономном режиме).
Импортировать содержимое листа схемы	Вставить содержимое другого листа в текущую схему.
Импорт схемы из другой САПР	Импорт проекта схемы, сохранённого в другом формате файла.
Сохранить	Сохранить все листы схемы, включая иерархические.
Сохранить текущий лист схемы	Сохранить только текущий лист схемы, иерархические листы не сохраняются.

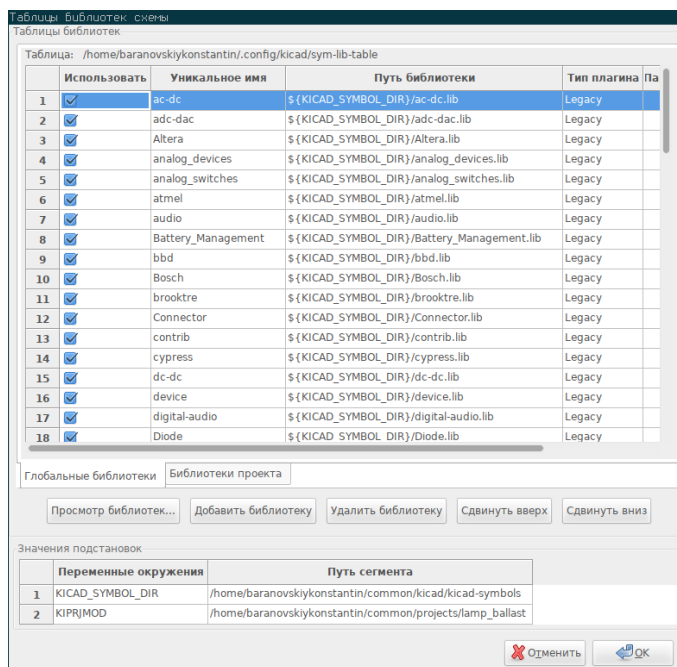
Сохранить текущий лист схемы как...	Сохранить текущий лист схемы под новым именем.
Настройки страницы	Настроить параметры листа и заполнить основную надпись.
Печать	Распечатать проект схемы (см. раздел Печать и черчение).
Чертить	Экспорт в PDF, PostScript, HPGL или SVG (см. раздел Печать и черчение).
Заккрыть	Прервать выполнение приложения.

3.2 Меню "Настройки"



Менеджер библиотек компонентов	Добавить или удалить библиотеки компонентов.
Настроить пути	Установить пути для поиска по умолчанию.
Общие настройки	Основные параметры (ед.измерения, шаг сетки, названия полей и т.д.).
Установить язык	Выбрать язык для интерфейса пользователя.
Параметры иконок	Настроить видимость иконок.
Импорт и Экспорт	Сохранить/загрузить параметры в/из файла.

3.2.1 Менеджер библиотек компонентов



В Eeschema используются две таблицы для хранения перечня доступных библиотек компонентов, которые различаются областью применения:

- Глобальные библиотеки

Библиотеки, перечисленные в общей таблице, доступны в любом проекте. Эта таблица хранится в файле **sym-lib-table** в домашнем каталоге пользователя (в общем, путь зависит от операционной системы; увидеть путь можно под таблицей).

- Библиотеки проекта

Библиотеки, перечисленные в таблице проекта, доступны только в открытом в данный момент проекте. Эта таблица хранится в файле **sym-lib-table** в каталоге проекта (путь отображается под таблицей).

Увидеть любую из таблиц можно переходя по вкладкам "Глобальные библиотеки" или "Библиотеки проекта" под таблицей библиотек.

3.2.1.1 Добавление новой библиотеки

Добавить библиотеку можно либо нажатием кнопки **Просмотр библиотек...**, с последующим выбором файла, либо нажатием кнопки "Добавить библиотеку", с последующим указанием пути к файлу библиотеки. Выбранная библиотека будет добавлена в открытую в данный момент таблицу (общую или таблицу проекта).

3.2.1.2 Удаление библиотеки

Для удаления необходимо выбрать одну или более библиотек и нажать кнопку **Удалить библиотеку**.

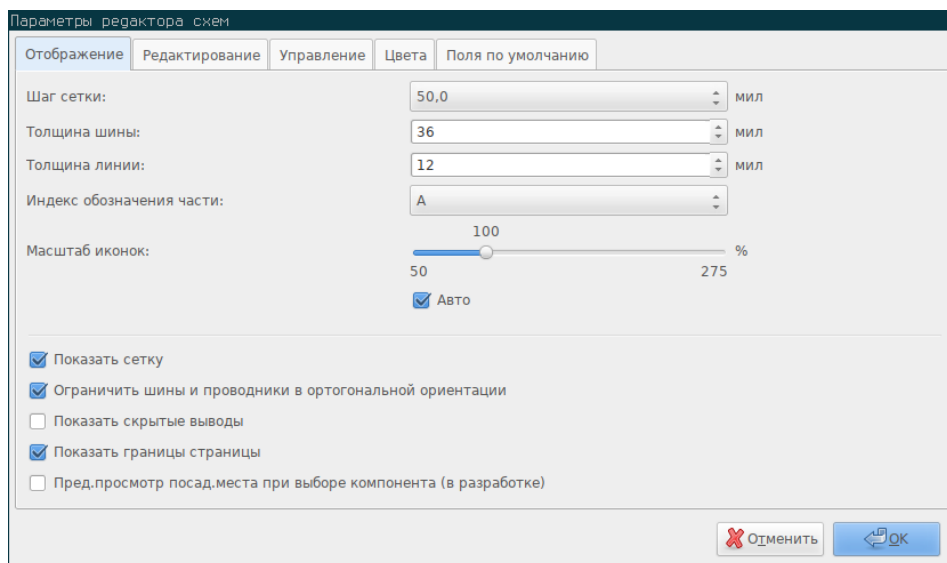
3.2.1.3 Настройка библиотек

Каждая строка в таблице содержит несколько полей, описывающих библиотеку:

Использовать	Использовать или не использовать библиотеку. Это помогает временно уменьшить количество загружаемых библиотек.
Уникальное имя	Уникальное имя - это короткий, уникальный идентификатор библиотеки, который используется в свойствах компонента. Компонента обозначается в виде строки <Уникальное имя библиотеки>:<Имя компонента>.

3.2.2 Общие настройки

3.2.2.1 Отображение



Шаг сетки	Выбор шага сетки. Рекомендуется работать с нормальным шагом сетки (0.050 дюймов или 1,27 мм). Меньший шаг используется при создании компонентов.
Толщина шины	Размер пера для черчения шин.
Толщина линии	Размер пера для черчения объектов, которые не имеют специальных настроек толщины.
Индекс обозначения части	Тип суффикса для компонентов, состоящих из нескольких частей (U1A, U1.A, U1-1, и т.д.)
Масштаб иконок	Настройка размера иконок панели инструментов.
Показать сетку	Параметр отображения сетки.
Ограничить шины и проводники в ортогональной ориентации	Если отмечено, шины и проводники прокладываются только горизонтально или вертикально. В противном случае, шины и проводники прокладываются под любым углом.
Показать скрытые выводы	Показать невидимые (<i>скрытые</i>) выводы, как обычно, выводы питания.
Показать границы страницы	Если отмечено, на экране будут показаны границы листа.

Пред.просмотр посад.места при выборе компонента

Отображать панель предварительного просмотра посадочного места с возможность выбора в диалоге добавления компонента.

Примечание: могут возникать задержки, используйте на своё усмотрение.

3.2.2.2 Редактирование

Параметры редактора схем

Отображение Редактирование Управление Цвета Поля по умолчанию

Единицы измерения: дюймы

Гориз. шаг повторения элементов: 0 мил

Верт. шаг повторения элементов: 100 мил

Инкремент повторения меток: 1

Размер текста по умолчанию: 50 мил

Интервал времени автосохранения: 0 минут

Автоматически разместить поля компонентов

Разрешить при авторазмещении поля изменять выравнивание

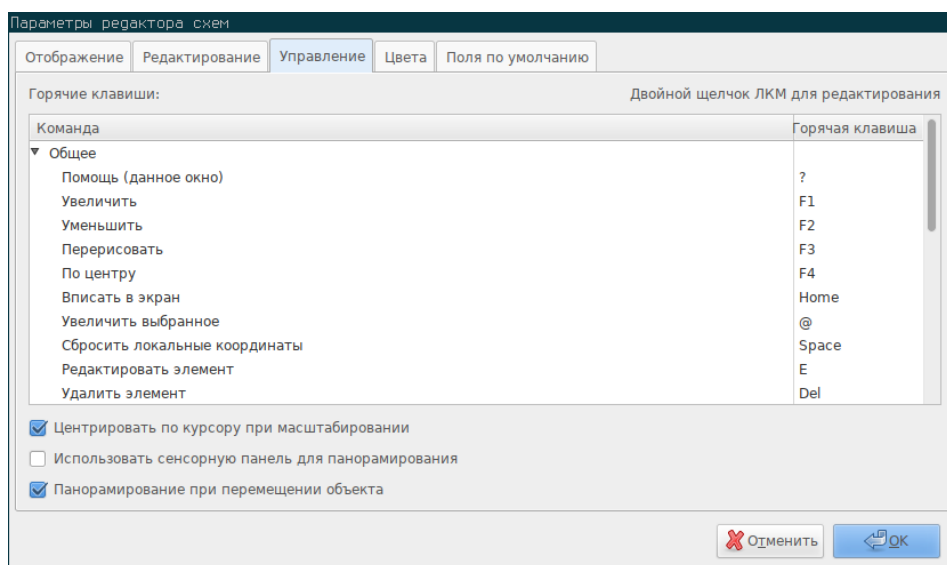
Всегда выравнивать авторазмещенные поля по сетке 50 мил

Отменить OK

Единицы измерения	Выбор единиц измерения для отображения и координат курсора (дюймы или миллиметры).
Гориз. шаг повторения элементов	Смещение по оси X при дублировании элемента (по умолчанию 0) (после расположения такого элемента как компонент, метка или проводник, можно создать его дубликат с помощью клавиши <i>Insert</i>).
Верт. шаг повторения элементов	Смещение по оси Y при дублировании элемента (по умолчанию 0.100 дюйма или 2,54 мм).
Инкремент повторения меток	Величина, на которую увеличивается значение метки с числом в конце (например, номер шины), при дублировании (обычно 1 или -1).
Размер текста по умолчанию	Размер текста для новых текстовых полей и меток.
Интервал времени автосохранения	Интервал, в минутах, для создания резервных копий.
Автоматически размещать поля компонентов	Если отмечено, то поля компонента (такие как значение и обозначение) при его добавлении будут расположены так, чтобы не пересекаться с остальными элементами.
Разрешить при авторазмещении поля изменять выравнивание	Дополнение к параметру <i>Автоматически размещать поля компонентов</i> . Разрешает подстраивать выравнивание текста полей компонента при его добавлении.
Всегда выравнивать авторазмещенные поля по сетке 50 мил	Дополнение к параметру <i>Автоматически размещать поля компонентов</i> . Если отмечено, при автоматическом размещении полей будет использоваться сетка с шагом 50 мил, иначе — они будут размещены произвольно.

3.2.2.3 Управление

Редактирование горячих клавиш и настройка взаимодействия с интерфейсом пользователя.



Новая горячая клавиша выбирается двойным щелчком по строке с командой или через контекстное меню:

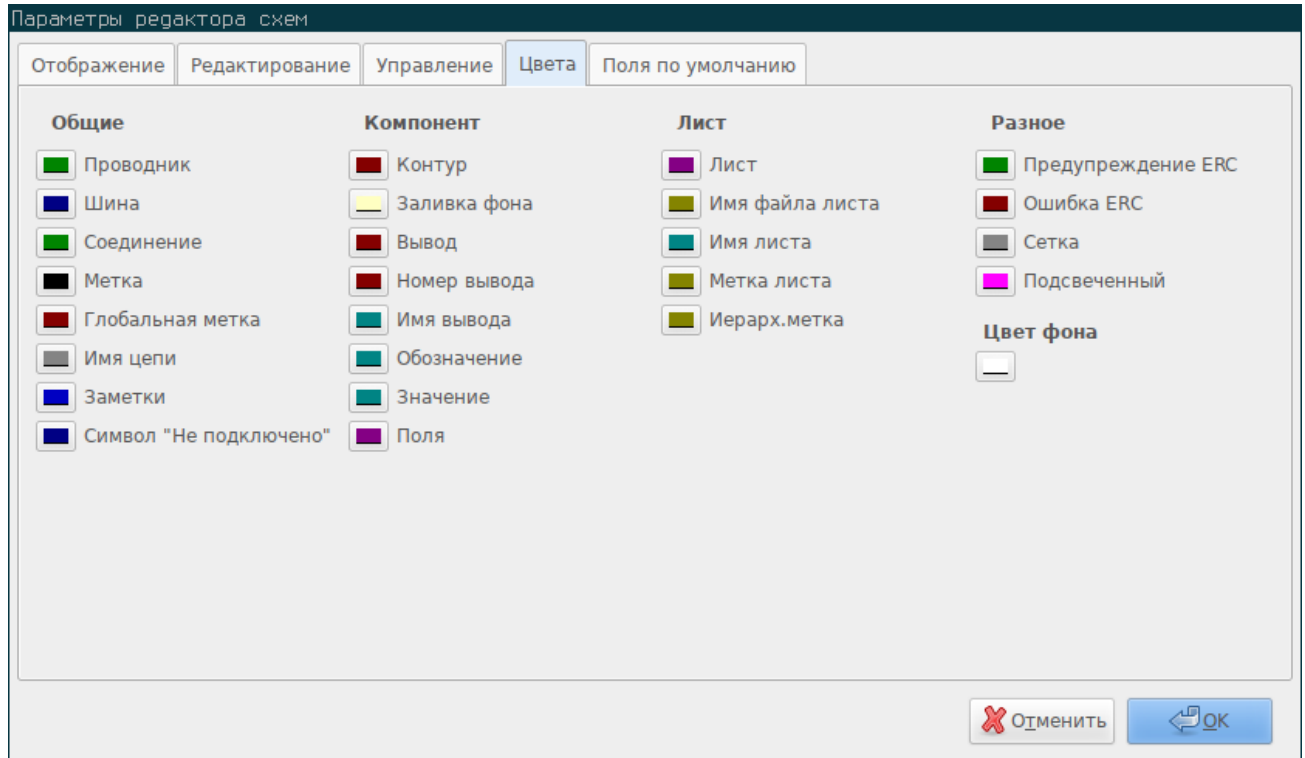
Править	Указать новую горячую клавишу для команды (тоже, что и двойной щелчок).
Отменить изменения	Отменить последние изменения горячей клавиши для текущей команды.
Значение по умолчанию	Вернуть исходное значение горячей клавиши для текущей команды.
Отменить все изменения	Отменить последние изменения горячих клавиш для всех команд.
Все значения по умолчанию	Вернуть исходные значения горячих клавиш для всех команд.

Описание параметров:

Центрировать по курсору при масштабировании	Если отмечено, то при масштабировании точка под курсором будет перемещена в центр экрана.
Использовать сенсорную панель для панорамирования	Если отмечено, изображение прокручивается с помощью колёс мыши (или областей прокрутки сенсорной панели), а масштабирование выполняется при нажатой клавише Ctrl. В противном случае колесо мыши управляют масштабированием, а клавиши Ctrl/Shift позволяют прокручивать изображение.
Панорамирование при перемещении объекта	Если отмечено, при выходе курсора за пределы окна в процессе черчения или перемещения окно будет автоматически прокручиваться.

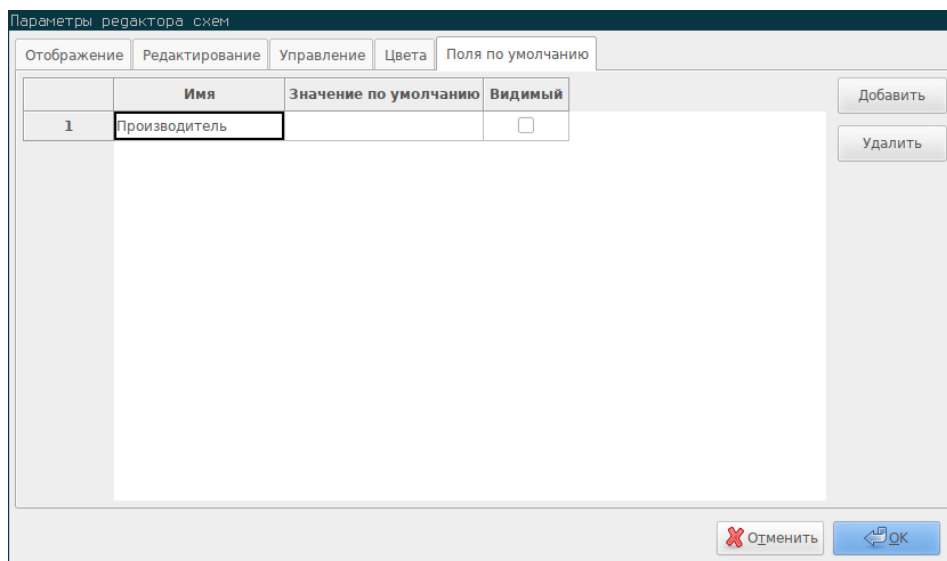
3.2.2.4 Цвета

Цветовая схема для разных графических элементов. Для выбора нового цвета какого-нибудь элемента щёлкните левой кнопкой мыши по кнопке с образцом текущего цвета.



3.2.2.5 Поля по умолчанию

Определение дополнительных пользовательских полей и их значений, которые будут добавлены при размещении новых компонентов.



3.3 Меню "Справка"

Предоставляет доступ к данному документу справки и детальному руководству пользователя KiCad.

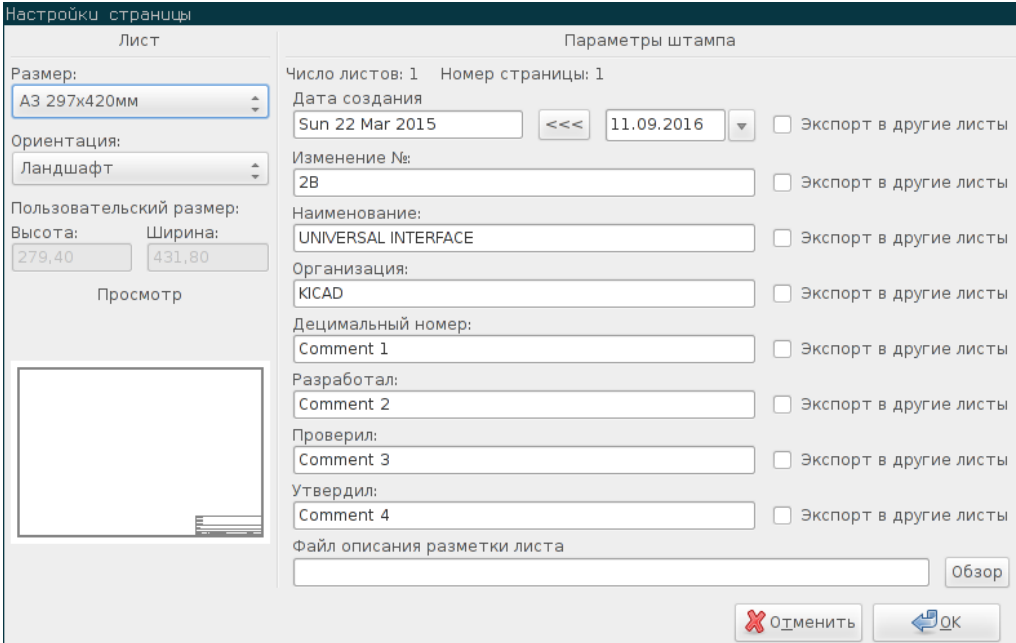
Используйте кнопку "Копировать информацию о версии" при составлении отчета об ошибке для уточнения используемой сборки и операционной системы.

Глава 4

Верхняя панель инструментов


4.1 Настройки страницы

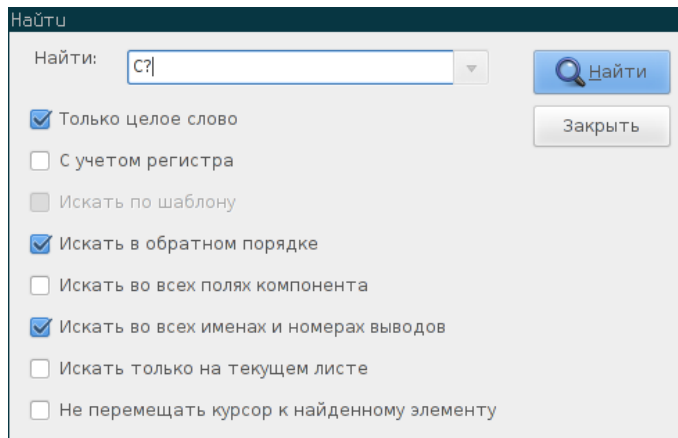
Кнопка "Настройки страницы"  открывает окно, в котором можно настроить размер листа и заполнить поля основной надписи.



Нумерация листов выполняется автоматически. Можно вставить сегодняшнюю дату с помощью кнопки со стрелками справа от "Даты создания". Дата автоматически не обновляется.

4.2 Инструмент поиска

Чтобы воспользоваться инструментом поиска нужно нажать кнопку "Поиск компонента или текста" 



Выполнить поиск можно по обозначениям, значениям и текстовым полям на текущем листе или же по всей иерархии листов. Как только совпадение будет найдено, курсор переместится к найденному элементу в соответствующем листе.

4.3 Инструмент формирования списка цепей



Кнопка "Сформировать список цепей" вызывает инструмент создания списка цепей.

Этот инструмент создаёт файл, в котором описываются все соединения во всей иерархии листов.

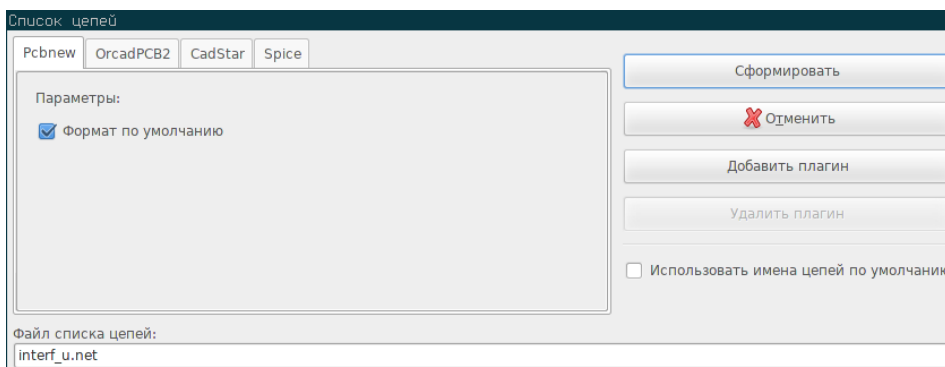
В сложных схемах (содержащих несколько листов) любая локальная метка видна только внутри того листа, на котором она расположена. Например: метка АБВ с листа №3 не соединена с меткой АБВ из листа №5 (если они не соединены намерено другим способом). Это связано с тем, что внутри каждой локальной метки содержится полное имя листа, к которому она относится.

Замечание

Несмотря на то что в Eeschema длина имён меток не ограничивается, имейте в виду, что при загрузке сформированного списка цепей в других приложения могут возникнуть проблемы.

Замечание

Избегайте пробелов в метках, так как слова могут восприниматься по отдельности в сформированном файле. Eeschema этого не запрещает, но многие форматы списков цепей предполагают, что метки будут без пробелов.



Параметры:

Формат по умолчанию	Установите отметку, чтобы выбрать формат Pcbnew в качестве формата по умолчанию.
---------------------	--

Можно использовать и другие форматы:


- Orcad PCB2
- CadStar
- Spice (симуляторы)

Можно использовать прочие инструменты для создания списков цепей в других форматах (далее будет показано как это сделать на примере PadsPcb).

Больше информации можно получить в разделе [Создание списка цепей](#).

4.4 Инструмент обозначения элементов



Кнопка  запускает инструмент обозначения компонентов. Этот инструмент присваивает элементам буквенно-цифровые обозначения.

К частям составных компонентов (таких как 7400 ТТЛ, состоящих из 4-х частей), также, добавляется суффикс (в случае с 7400 ТТЛ с обозначением DD3, части будут обозначены как DD3.1, DD3.2, DD3.3 и DD3.4).

Можно обновить обозначения всех компонентов или же только новых, т.е. тех что ещё не обозначены.

Обозначить

По всей схеме	Обозначить компоненты на всех листах (по умолчанию).
Только текущий лист	Обозначить компоненты только на текущем листе(этот вариант используется только в крайних случаях, например, чтобы определить количество резисторов на текущем листе).
Сохранить существующие обозначения	Будут обозначены только новые компоненты (по умолчанию).
Сбросить существующие обозначения	Все компоненты будут заново обозначены (этот вариант будет полезным при наличии компонентов с одинаковыми обозначениями).
Сбросить, но не менять аннотированные части компонентов	Подобен предыдущему пункту, но сохраняет суффиксы всех составных компонентов (типа DD2.1, DD2.2) без изменений.


Порядок обозначения

Выбор порядка, в котором будут обозначены компоненты (горизонтально или вертикально).

Выбор варианта обозначений

Выбор необходимого формата позиционных обозначений.

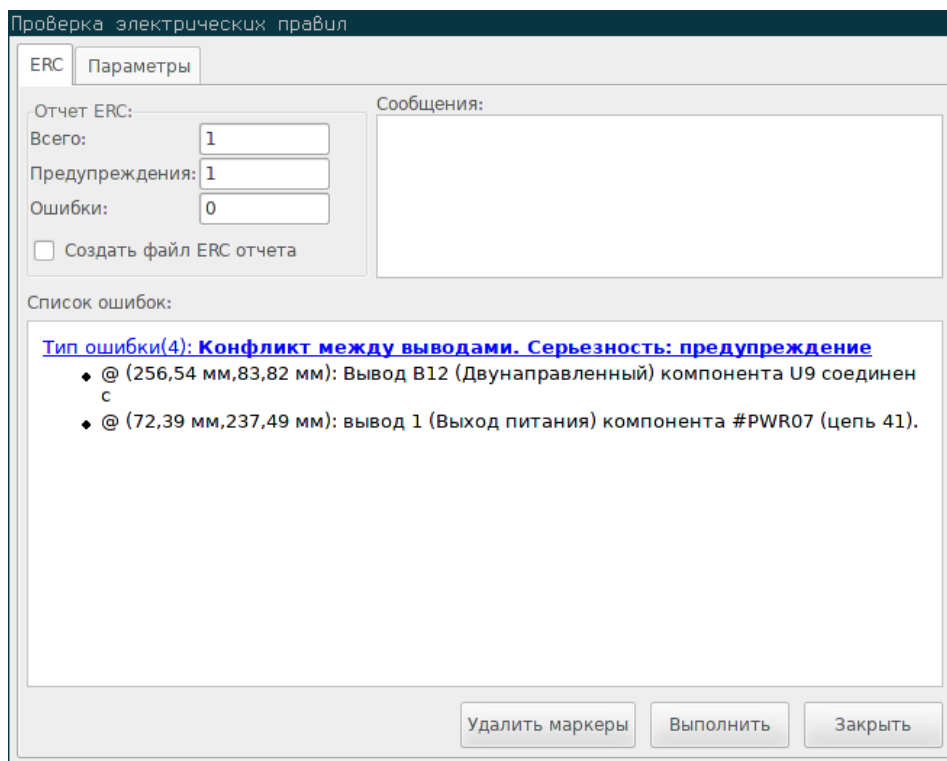
4.5 Инструмент проверки электрических правил проектирования

Кнопка  запускает инструмент проверки электрических правил проектирования (ERC).

Этот инструмент выполняет проверку правил проектирования и позволяет определить отсутствующие или неверные соединения.

После проверки правил, Eeschema размещает метки для указания проблемных мест. Чтобы увидеть описание ошибки, нужно нажать левой кнопкой мыши на метке. Список всех найденных проблем можно записать в файл.

4.5.1 Диалоговое окно проверки электрических правил проектирования



В диалоговом окне содержится следующая информация:

- Общее количество ошибок и предупреждений.
- Количество ошибок.
- Количество предупреждений.

Параметры:

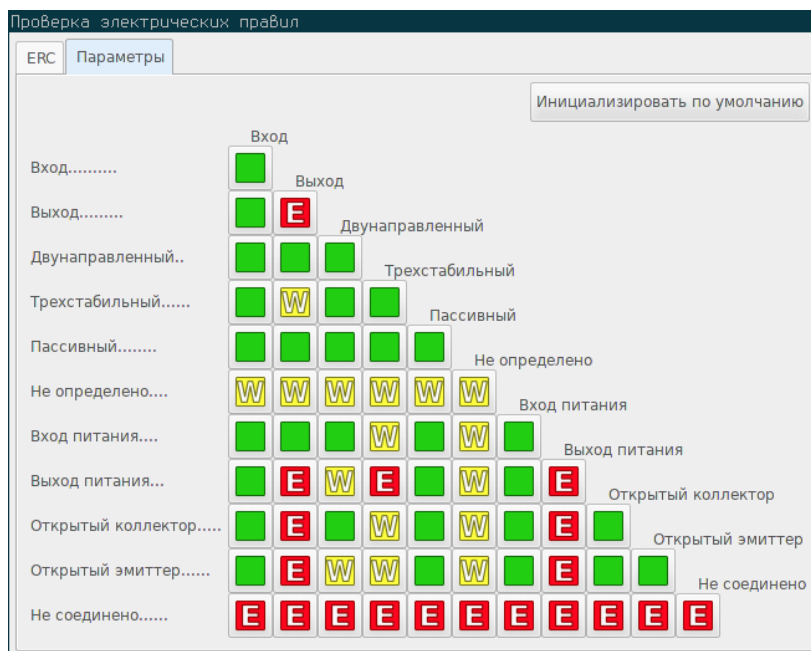
Создать файл ERC отчета	Если отмечено, будет создан файл со списком найденных ошибок и предупреждений.
-------------------------	--

Команды:

Удалить маркеры	Удалить все маркеры ошибок/предупреждений ERC.
Выполнить	Начать проверку правил проектирования.
Закреть	Закреть диалоговое окно.

- При выборе какого-нибудь сообщения об ошибке, курсор переместится к соответствующей метке на схеме.

4.5.2 Настройка электрических правил проектирования



На этой вкладке можно установить правила соединения выводов. Доступны три варианта для каждого параметра:

- Без ошибок или предупреждений
- Формировать предупреждение
- Формировать ошибку

Каждый элемент матрицы можно редактировать с помощью мыши.

Параметры:


Тестировать одинаковые метки	Сообщать о наличии меток, которые отличаются лишь регистром букв (например: метка, Метка, MeTka.) Имена цепей чувствительны к регистру, следовательно, указывают на разные цепи.
Тестировать единственность глобальных меток	Сообщать о глобальных метках, которые встречаются лишь один раз, образуя часть цепи. Обычно требуется как минимум две метки для создания цепи.

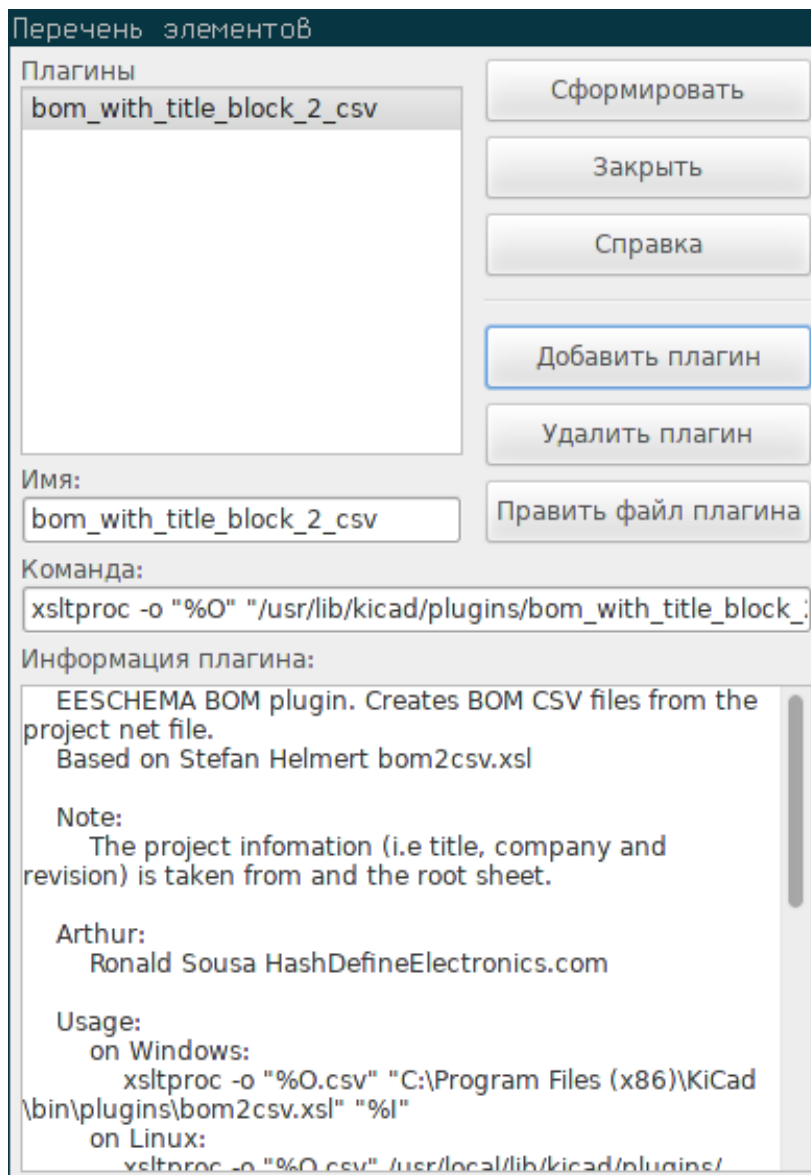
Команды:

Инициализировать по умолчанию	Восстановить начальное значение.
-------------------------------	----------------------------------

4.6 Инструмент формирования перечня элементов



Кнопка  запускает инструмент генерации перечня элементов (BOM). Этот инструмент позволяет создавать файлы, содержащие список компонентов и/или иерархических соединений (глобальных меток).



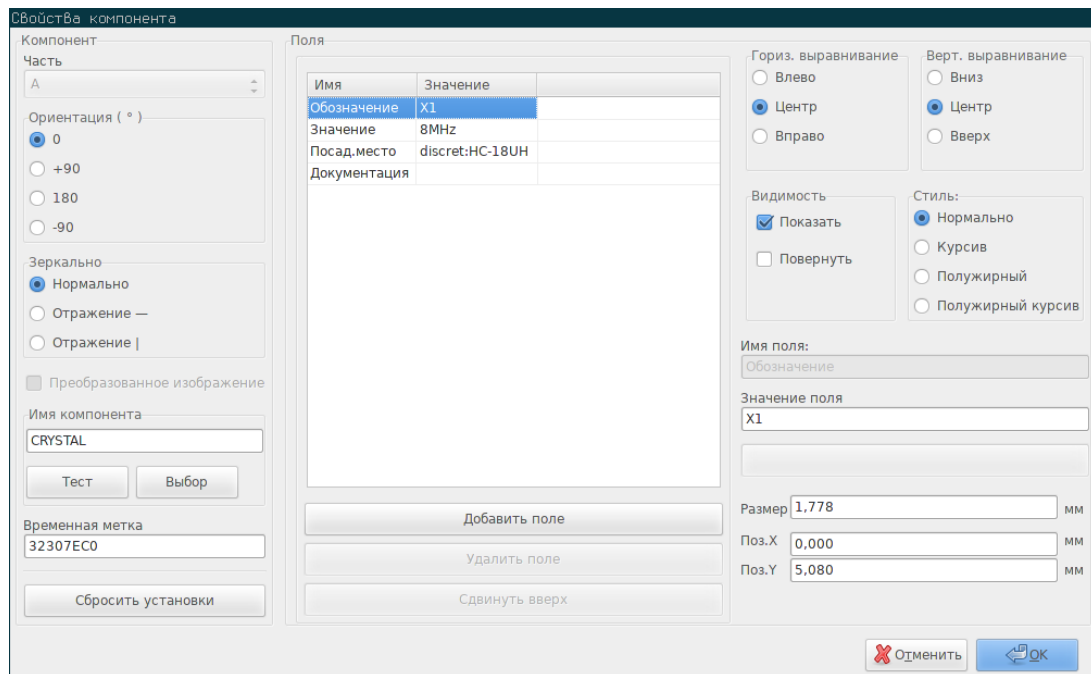
Инструмент генерации перечней элементов в Eeschema использует внешние приложения в качестве плагинов, в основном это XSLT или Python. В каталоге установки KiCad присутствует несколько демонстрационных скриптов.

Для составления перечня элементов можно использовать следующий набор полей:

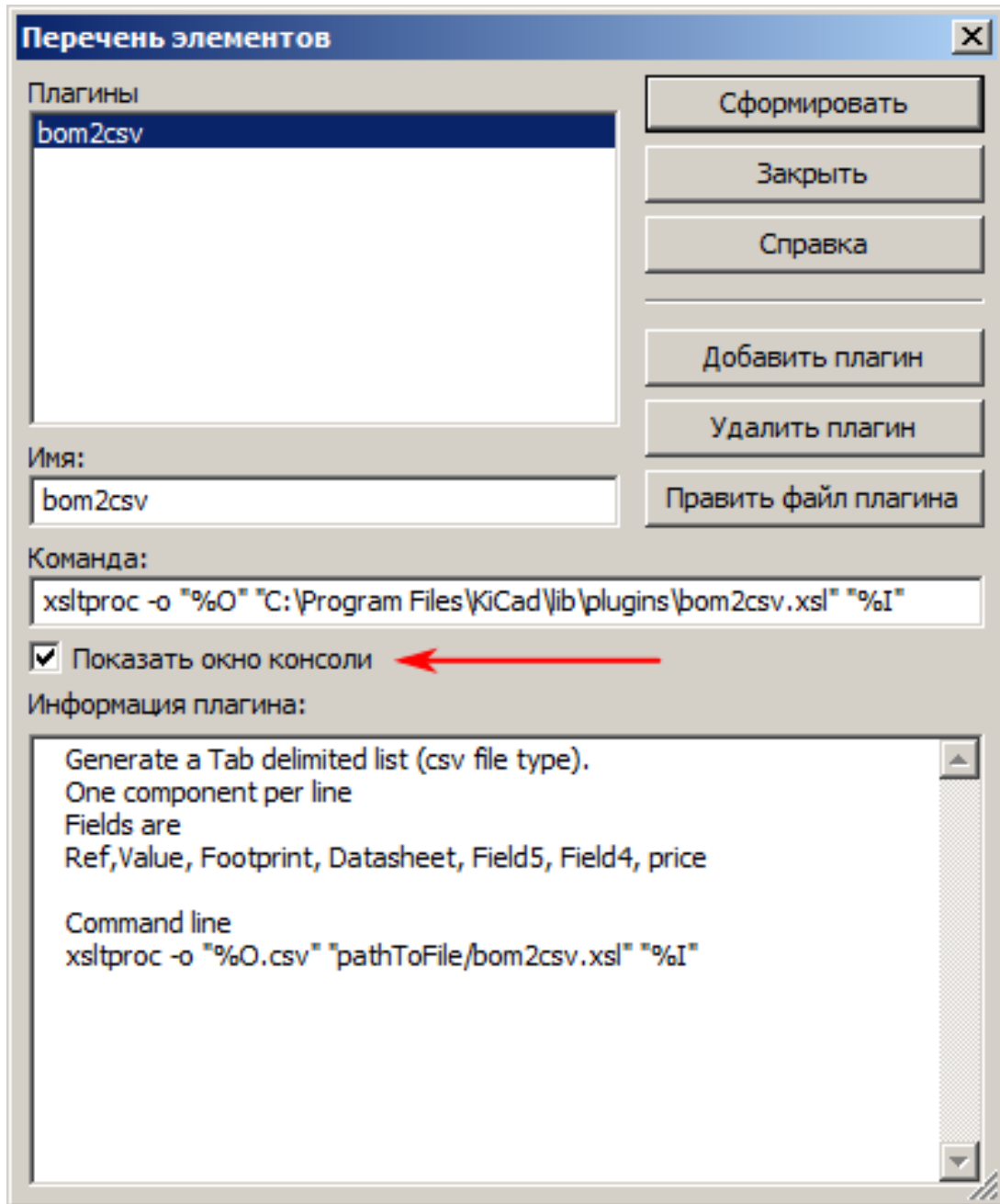
- Значение - уникальное имя для каждого элемента.
- Посадочное место - введенное вручную или импортировано (см. ниже).
- Поле1 - название производителя.

- Поле2 - код по каталогу производителя.
- Поле3 - код по каталогу поставщика.


Например:

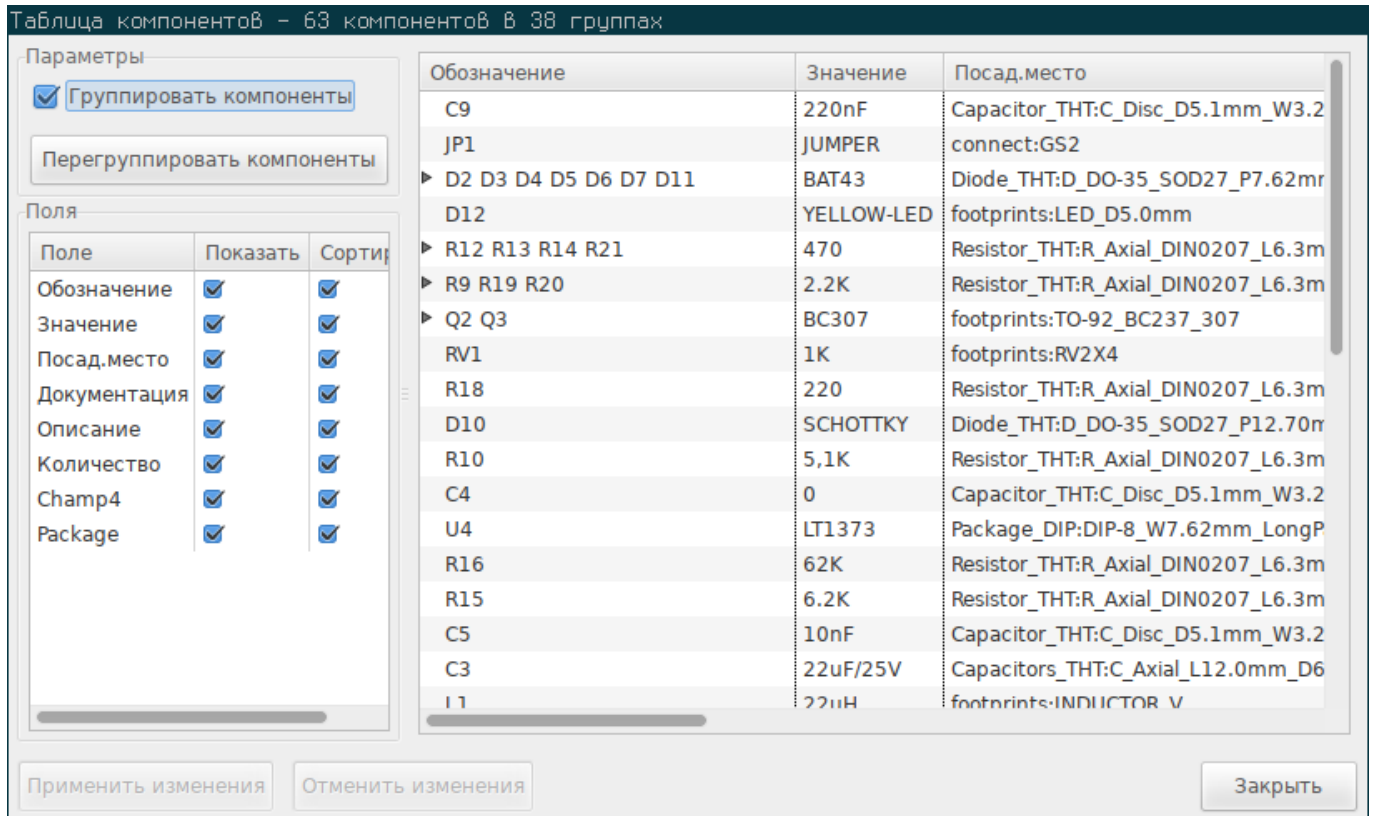


В операционных системах **MS Windows** диалоговое окно формирования перечня элементов (BOM) имеет дополнительный параметр (на изображении указан красной стрелкой), который управляет отображением окна внешнего приложения. По умолчанию, приложение, которое генерирует перечень элементов, выполняется со скрытым окном, а вывод перенаправляется в поле *Информация плагина*. Чтобы увидеть окно выполняющегося приложения, необходимо установить отметку возле указанного параметра. Это может потребоваться для плагинов, имеющих графический интерфейс.



4.7 Инструмент редактирования полей

Кнопка  вызывает таблицу для просмотра и редактирования значений полей всех компонентов.



После изменения значений полей, их необходимо сохранить нажатием кнопки *Применить* или отменить нажатием кнопки *Отменить*.

4.7.1 Приёмы быстрого заполнения полей

Для таблиц имеется несколько специальных приёмов копирования-вставки. Они будут полезны при вводе одинаковых значений в поля нескольких компонентов одновременно.

Эти приёмы показаны ниже.

Копирование (Ctrl+C)	Выделение	Вставка (Ctrl+V)

Замечание

Данные методы также доступны и в таблицах других диалоговых окон.

4.8 Инструмент импорта значений посадочных мест

4.8.1 Доступ



Кнопка **BACK** запускает инструмент импорта значений посадочных мест.

Этот инструмент позволяет импортировать изменения посадочных мест из Pcbnew в поля "Посад.место" компонентов в Eeschema.

Глава 5

Менеджер библиотек компонентов

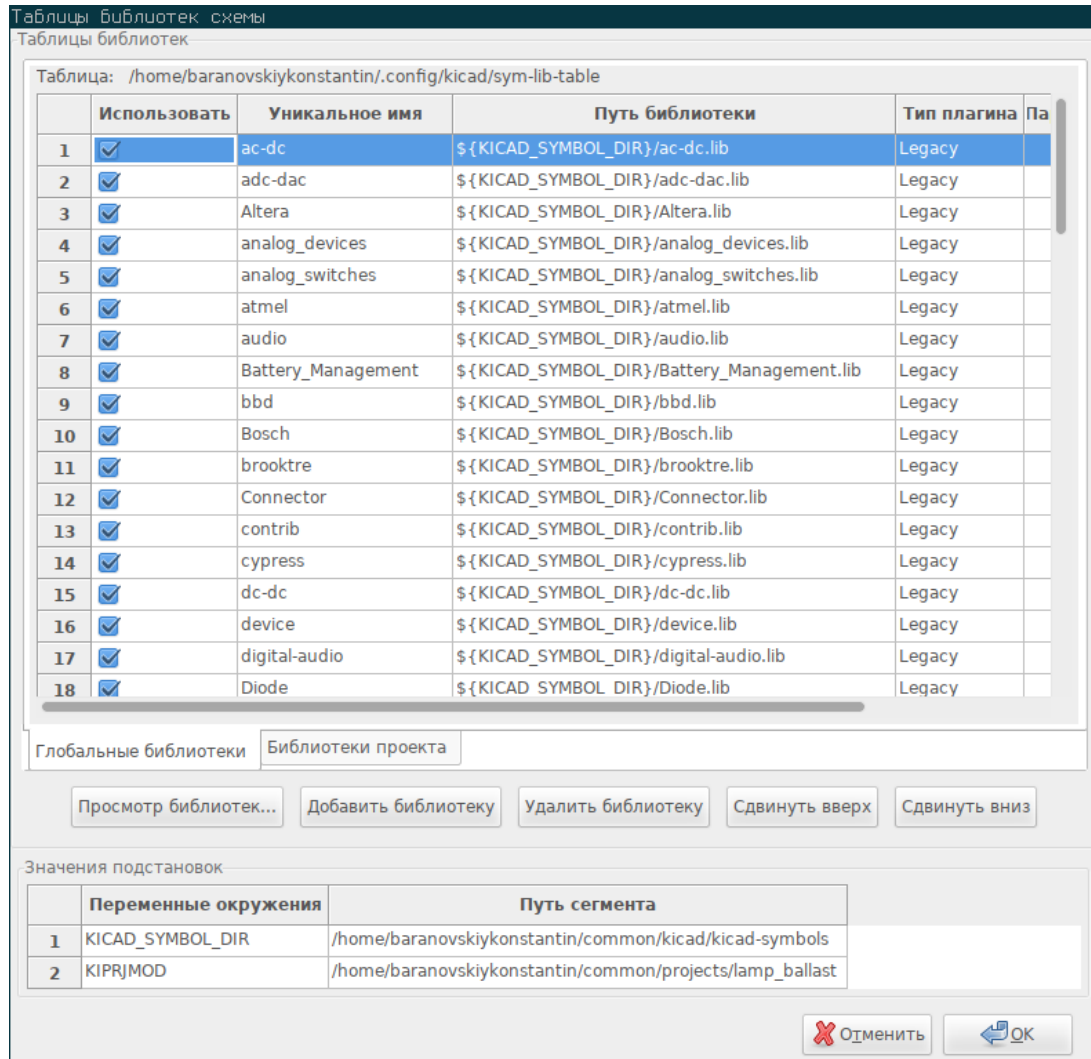
Библиотеки компонентов содержат наборы компонентов, используемые при создании схем. Каждый компонент схемы однозначно обозначен полным наименованием, которое состоит из уникального имени библиотеки и названия компонента. Например `Audio:AD1853`.

5.1 Таблица библиотек компонентов

Таблица библиотек компонентов содержит перечень всех файлов библиотек, о которых должен знать KiCad. Таблица библиотек компонентов строится на основе файла глобальной (общей) таблицы библиотек и файла таблицы библиотек проекта.

После загрузки компонента, Eeschema использует уникальное имя библиотеки (`Audio` в нашем случае) для определения расположения библиотеки, указанного в таблице библиотек.

На изображении ниже, показано диалоговое окно редактора таблиц библиотек компонентов, которое можно открыть с помощью команды «Менеджер библиотек компонентов...» в меню «Настройки».



5.1.1 Глобальная таблица библиотек компонентов

Глобальная таблица библиотек компонентов содержит перечень библиотек, которые доступны всегда, независимо от загруженного в данный момент проекта. Эта таблица хранится в файле `sym-lib-table` в домашнем каталоге пользователя. Расположение данного каталога зависит от используемой операционной системы.

5.1.2 Таблицы библиотек компонентов проекта

Таблица библиотек компонентов проекта содержит перечень библиотек, которые доступны только для загруженного в данный момент проекта. Данную таблицу можно редактировать только после загрузки файла проекта. Если проект не загружен или в каталоге проекта нет файла таблицы библиотек, будет создана пустая таблица, которую можно редактировать и в дальнейшем сохранить рядом с файлом проекта.

5.1.3 Начальная настройка

При первом запуске Eeschema, файл таблицы библиотек компонентов **sym-lib-table** отсутствует в домашнем каталоге пользователя, поэтому Eeschema выполнит попытку копировать файл стандартной таблицы библиотек `sym-lib-table` из систем-

ного каталога шаблонов KiCad в домашний каталог пользователя. Если стандартный файл `sym-lib-table` не удастся найти в каталоге шаблонов, будет показано диалоговое окно с запросом указать возможное расположение файла `sym-lib-table`. Если же файл `sym-lib-table` так и не удастся найти или запрос в диалоговом окне будет отменён, Eeschema создаст пустую таблицу библиотек компонентов в домашнем каталоге пользователя. В таком случае, пользователь должен либо самостоятельно скопировать `sym-lib-table`, либо настроить таблицу вручную.

Замечание

Стандартная таблица библиотек компонентов содержит все библиотеки компонентов, которые устанавливаются вместе с KiCad. Они могут быть полезными или нет, в зависимости от нужд и быстродействия системы. Время загрузки библиотек компонентов прямо зависит от их количества в таблице. Если библиотеки загружаются очень долго, удалите ненужные или редко используемые библиотеки из общей таблицы и добавляйте их в случае необходимости в таблицу проекта.

5.1.4 Добавление элементов таблицы

Для использования библиотеки компонентов, её сначала нужно добавить или в общую таблицу, или в таблицу проекта. Таблица проекта доступна только при загруженном файле проекта.

Каждая библиотека должна иметь уникальное имя.

Это имя может отличаться от имени библиотеки. Запрещается использовать символы двоеточия `:` и косой черты `/` в любом месте уникального имени. В каждом элементе таблицы библиотек должен быть верно указан путь и/или имя файла, в зависимости от типа библиотеки. Пути можно указывать как абсолютно, так и относительно, или с применением переменных окружения (см. следующий раздел).

Чтобы библиотека была загружена верно, нужно указать соответствующий тип плагина. На данный момент KiCad поддерживает только собственный формат библиотек компонентов.

Кроме того, каждый элемент таблицы имеет поле описания, в которое можно добавить дополнительную информацию о библиотеке. Поле параметров на данный момент не используется, так что его содержимое никак не повлияет на загрузку библиотек.

- Примите к сведению, что дублирование уникальных имён в пределах одной таблицы недопустимо. Тем не менее, одно и то же уникальное имя может быть одновременно использовано в общей таблице и в таблице проекта.
- Элемент в таблице проекта имеет больший приоритет чем в элемент с таким же уникальным именем в общей таблице.
- Элементы, определённые в таблице проекта, будут записаны в файл `sym-lib-table`, который хранится в том же каталоге, что и открытый в данный момент файл проекта.

5.1.5 Подстановка переменных окружения

Одно из наиболее мощных особенностей таблицы библиотек компонентов — использование переменных окружения. С их помощью можно указывать путь к месту хранения библиотек компонентов. Переменные окружения указываются в пути библиотеки с помощью специальной записи: `$(ИМЯ_ПЕРЕМЕННОЙ_ОКРУЖЕНИЯ)`.

По умолчанию, при запуске KiCad создаются две переменные окружения:

- **KIPRJMOD** — содержит путь к каталогу текущего проекта. Эту переменную нельзя изменить.
- **KICAD_SYMBOL_DIR** — содержит путь к стандартным библиотекам компонентов, которые устанавливались вместе с KiCad.

Переменную **KICAD_SYMBOL_DIR** можно переопределить, создав свою переменную окружения с таким же именем с помощью меню Настройки→Настройка путей..., что позволит использовать свои собственные библиотеки вместо стандартных, установленных вместе с KiCad.

KIPRJMOD позволяет хранить библиотеки в каталоге проекта и не использовать абсолютные пути к библиотекам в таблице проекта (которые при перемещении проекта могут меняться).

5.1.6 Варианты использования

Библиотеки компонентов можно определять как в общей таблице, так и в таблице загруженного в данный момент проекта. Библиотеки из общей таблицы доступны всегда, а файл самой таблицы `sym-lib-table` хранятся в домашнем каталоге пользователя. Таблица проекта доступна только для загруженного проекта.

При использовании каждой из таблиц имеются свои достоинства и недостатки. Если определить все библиотеки в общей таблице, то они будут доступны всегда и в любом проекте. Недостатком этого способа является увеличенное время загрузки.

Если же определить только необходимые библиотеки в таблице проекта, то это существенно снизит время загрузки. Недостатком этого способа является то, что нужно постоянно помнить перечень библиотек и добавлять нужные к каждому новому проекту.

Наилучшим вариантом будет объединить два предыдущих и в общей таблице указать библиотеки с компонентами, которые используются часто, а в таблицу проекта заносить только недостающие библиотеки компонентов для текущего проекта. Единых рекомендаций о том, какие библиотеки в какую таблицу вносить — нет. Каждый пользователь должен решить это самостоятельно, исходя из своих предпочтений.

5.1.7 Загрузка устаревших проектов

При загрузке схем, созданных до внедрения таблиц библиотек компонентов, Eeschema выполнит попытку переопределить библиотеки компонентов, связанных со схемой, к библиотекам, имеющимся в таблице. Успех выполнения данной процедуры зависит от нескольких факторов:

- оригинальные библиотеки, используемые в схеме, должны быть доступны и не иметь изменений с тех пор, как их компоненты были добавлены в схему;
- необходимо убедиться что недостающие компоненты были "спасены" и сохранены в отдельной библиотеке (`rescue`), а если эта библиотека существует, убедиться чтобы её содержимое было обновлено;
- содержимое кешированной библиотеки компонентов проекта не будет повреждено;



Внимание

В процессе переопределения библиотек, в каталоге проекта будет создан каталог с резервными копиями файлов, которые подлежат изменению. Но, в любом случае, следует сделать резервную копию всего проекта на случай, если что-то пойдёт не так.

**Внимание**

Операция по "спасению" компонентов выполняется даже если эта функция отключена. Это необходимо для выявления подходящих компонентов. Не отменяйте эту операцию, иначе процесс переопределения библиотек не сможет успешно завершиться. Все недостающие или ошибочные библиотеки необходимо будет поправить вручную.

Замечание

В случае, когда изначальная библиотека удалена и спасти компоненты не удаётся, можно воспользоваться кешированной библиотекой для восстановления. Для этого нужно скопировать её и переименовать. Затем, поместить эту библиотеку в верх списка используемых библиотек с помощью устаревшей версии Eeschema, в которой ещё не используются таблицы библиотек компонентов.

Глава 6

Создание и редактирование схем

Введение

Схема может быть представлена на одном листе или на нескольких, если проект достаточно большой.

Сложные схемы, представленные на нескольких листах, являются иерархическими. Все листы (каждому из которых соответствует отдельный файл) составляют единый проект схемы в Eeschema. Как работать с иерархическими листами будет рассказано в разделе [Иерархические листы](#).

6.1 Основные положения

Схема, разработанная в Eeschema, является не просто графическим представлением электронного устройства. В общем случае, она является отправной точкой в цикле разработки и позволяет выполнять следующее:

- [Проверку электрических правил проектирования](#) для выявления ошибок и упущений.
- Автоматически формировать перечень элементов ([BOM](#)).
- [Формировать список цепей](#) для приложений симуляции работы электрических схем, таких как SPICE.
- [Формировать список цепей](#) для передачи в приложение трассировки печатной платы.

Схема, в основном, состоит из компонентов, проводников, меток, соединений, шин и символов питания. Помимо этого, можно размещать и графические элементы, такие как выводы соединения с шиной, текстовые комментарии и графические линии.


6.2 Цикл разработки

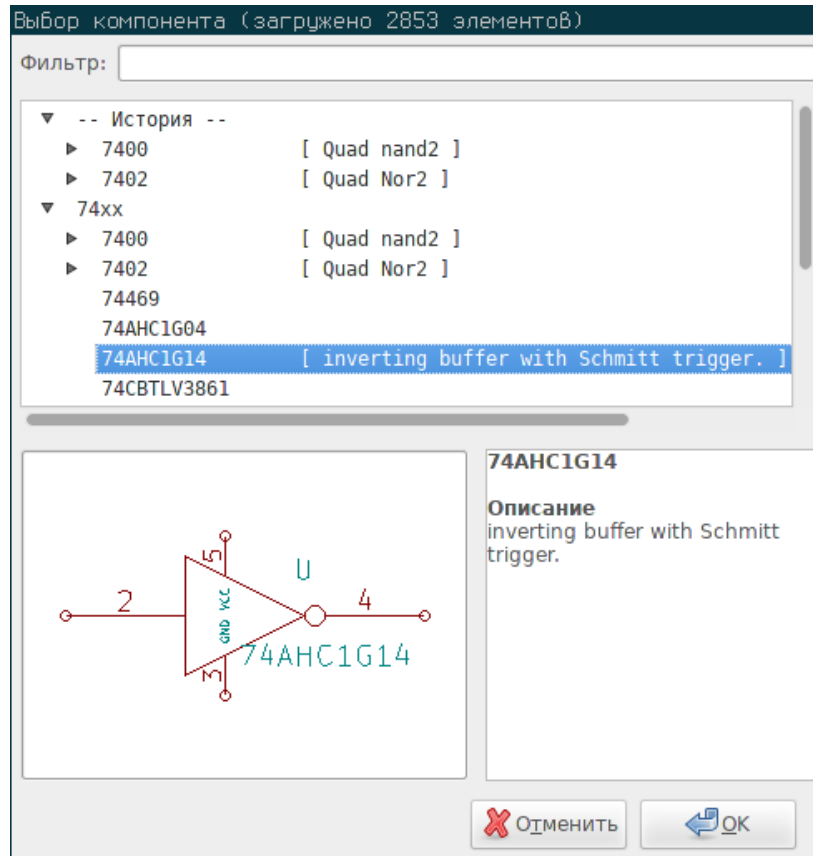


Компоненты добавляются в схему из библиотеки. Затем, когда схема готова, создаётся список цепей, который позже используется для импорта соединений и посадочных мест в PcbNew.

6.3 Размещение и редактирование компонентов

6.3.1 Поиск и размещение компонента

Чтобы добавить компонент на схему, нужно нажать кнопку . Диалоговое окно позволяет найти нужный компонент по имени.



Диалог выбора отфильтрует компоненты по имени, ключевым словам или описанию, согласно запросу, введенном в текстовое поле. Для более гибкого поиска можно использовать специальные фильтры:

- **Шаблоны:** используйте символы ? и * чтобы указать "любой символ" и "множество любых символов", соответственно.
- **Метки:** если в описании элементов библиотеки или в ключевых словах содержатся метки в формате "Key:123", можно выполнить относительный поиск по ним набрав "Key>123" (больше) или "Key<123" (меньше). К числу может прибавляться один из следующих не чувствительных к регистру суффиксов:

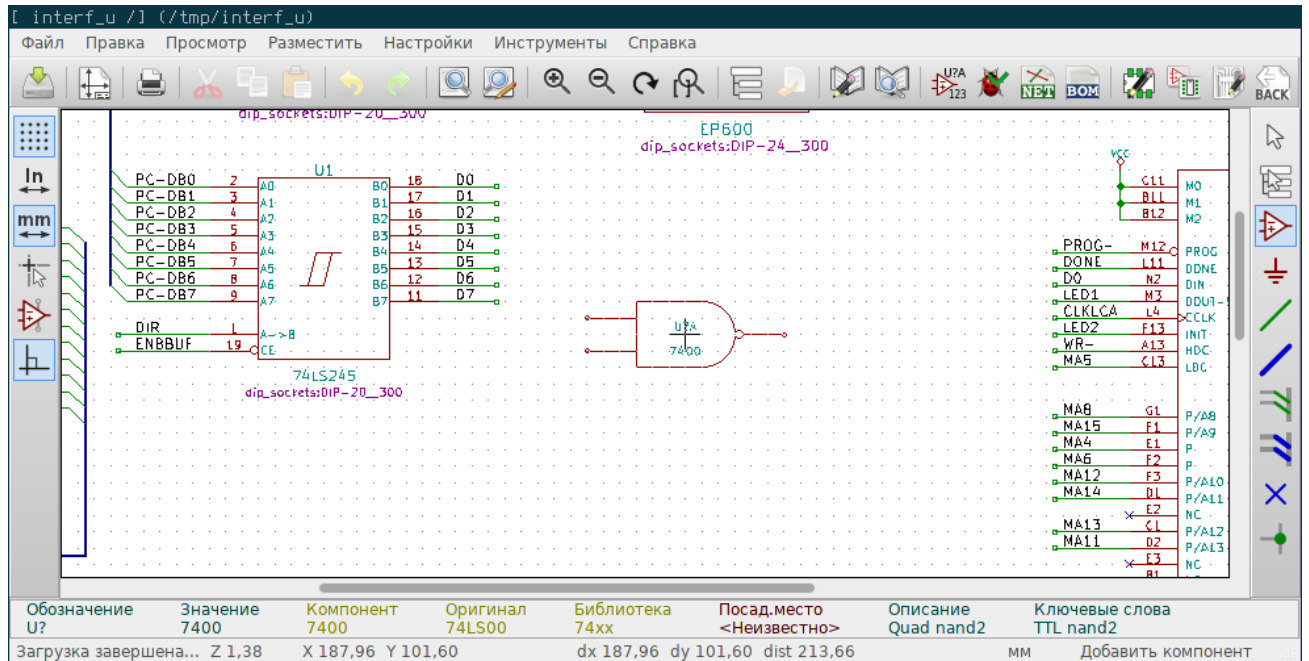
p	n	u	m	k	meg	g	t
10^{-12}	10^{-9}	10^{-6}	10^{-3}	10^3	10^6	10^9	10^{12}

ki	mi	gi	ti
2^{10}	2^{20}	2^{30}	2^{40}


- **Регулярные выражения:** если имеется опыт работы с регулярными выражениям, то их можно применять тоже. Используется формат регулярных выражений **wxWidgets Advanced Regular Expression style**, подобный регулярным выражениям Perl.

Прежде чем расположить компонент на схему можно его вращать, отражать и редактировать значения полей с помощью горячих клавиш или контекстного меню. Эти же операции можно выполнить и после расположения.

Так выглядит компонент в процессе расположения:



6.3.2 Символы питания

Символ питания представляет из себя компонент (все символы питания собраны в библиотеке “power”), поэтому его можно добавить с помощью диалога выбора компонента. Но, так как эти символы добавляются часто, существует отдельный инструмент . Этот инструмент аналогичен добавлению компонента, только список состоит из символов питания.

6.3.3 Редактирование размещённых компонентов

Редактировать компоненты можно двумя путями:

- Редактирование самого компонента: позиция, ориентация, выбор части в составных компонентах.
- Редактирование одного из полей компонента: обозначение, значение, посадочное место и т.д.

Как только компонент был размещен, можно задать его значение (в частности для резисторов, конденсаторов и т.д.). Но не стоит сразу указывать порядковый номер в обозначении или номер части (кроме случая, когда нужная часть выбирается вручную), так как это выполняется автоматически с помощью функции обозначения.

6.3.3.1 Изменение компонента

Для изменения какого-нибудь свойства компонента расположите курсор над ним и затем выполните:

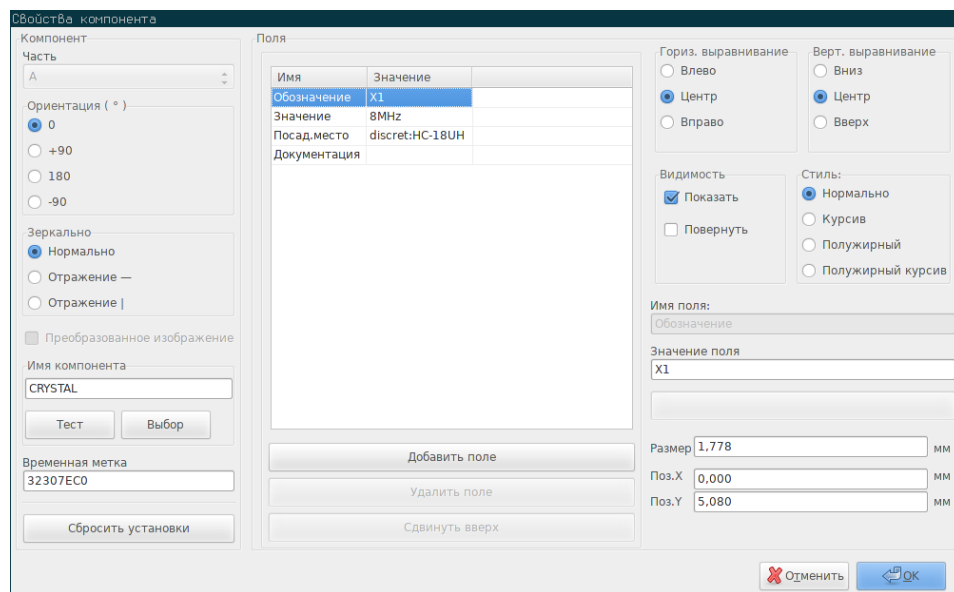
- Двойной щелчок левой кнопкой мыши для вызова редактора свойств компонента.
- Щелчок правой кнопкой мыши для вызова контекстного меню и выбор одного из вариантов: Переместить, Ориентировать, Править, Удалить и т.д.

6.3.3.2 Изменение значения полей

Можно изменять значение, позицию, ориентацию, размер текста и видимость полей:

- Двойной щелчок левой кнопки мыши на текстовом поле для начала редактирования.
- Щелчок правой кнопкой мыши для вызова контекстного меню и выбор одной из команд: Переместить, Повернуть, Править, Удалить и т.д.

Чтобы получить больше опций или если нужно добавить поля, выполните двойной щелчок по компоненту для вызова диалога редактирования свойств компонента.



Каждое поле можно показать или спрятать, отобразить горизонтально или вертикально. Указанная позиция всегда соответствует нормально расположенному компоненту (без поворотов и отражения) и относительно точки привязки компонента.

Опция “Сбросить установки” устанавливает ориентацию в нормальное положение и сбрасывает свойства, размер и позицию для каждого поля. Но при этом, значения полей не изменяются, так как это может разрушить схему.

6.4 Проводники, шины, метки, символы питания

Введение

Все эти элементы черчения можно добавить с помощью инструментов на правой панели.

Вот эти элементы:

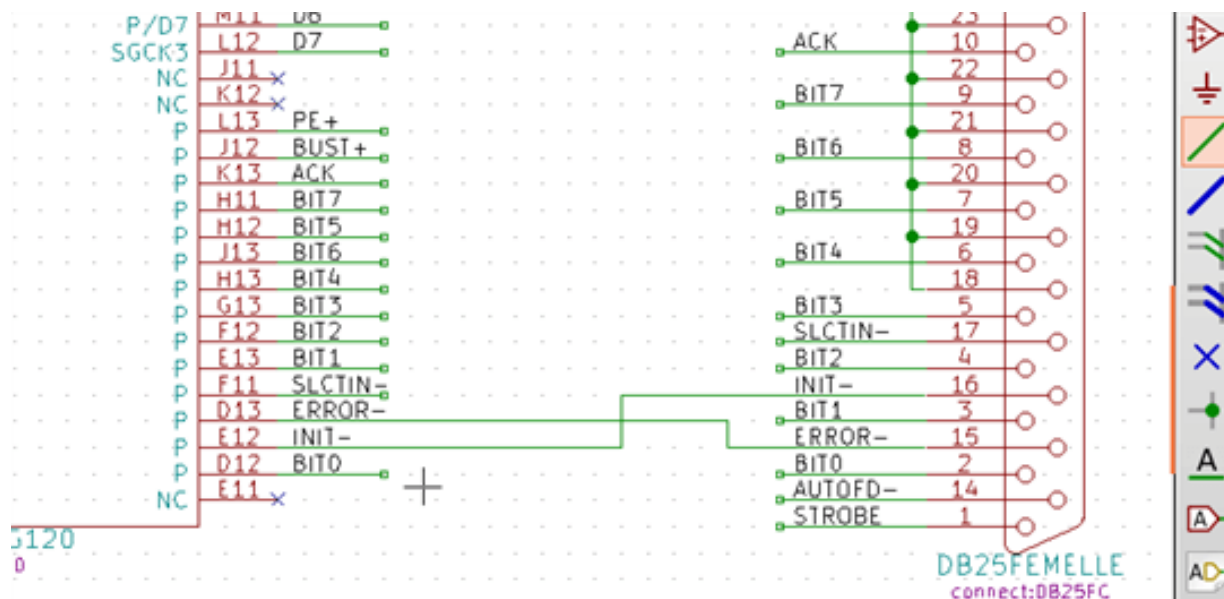
- **Проводник:** большинство соединений между компонентами.
- **Шины:** графическое соединение меток шины.
- **Графические линии:** черчение графических элементов.
- **Соединения:** создание соединения между пересекающимися проводниками или шинами.
- **Ввод шины:** служит для соединения проводника с шиной.
- **Метки:** для обозначения или создания соединений.
- **Глобальные метки:** для соединений между листами.
- **Текст:** для комментариев и надписей.
- **Флаг "Не подключено":** для обозначения выводов, которые не нужно ни куда подключать.
- **Иерархический лист** и его выводы для подключения.

6.4.1 Соединения с помощью проводников и меток

Есть два пути для создания соединения:

- Проводник от вывода к вывод.
- Метки.

Следующее изображение показывает оба метода:



Примечание 1:

Место “контакта” метки — нижний левый край её первого символа. Если метка не подключена, эта точка отображается в виде маленького квадрата.

Данная точка должна располагаться на проводнике или накладываться на конец вывода, чтобы принять вид подключенной.

Примечание 2:

Для образования соединения, сегменты должны подключаться концами друг к другу или к выводам.

При пересечении (если проводник проходит над выводом, но не соединяется с его концом) соединение не создается.

Примечание 3:

Пересекающиеся проводники сами по себе не соединяются. Их нужно соединять явно, с помощью точки соединения, если это необходимо.

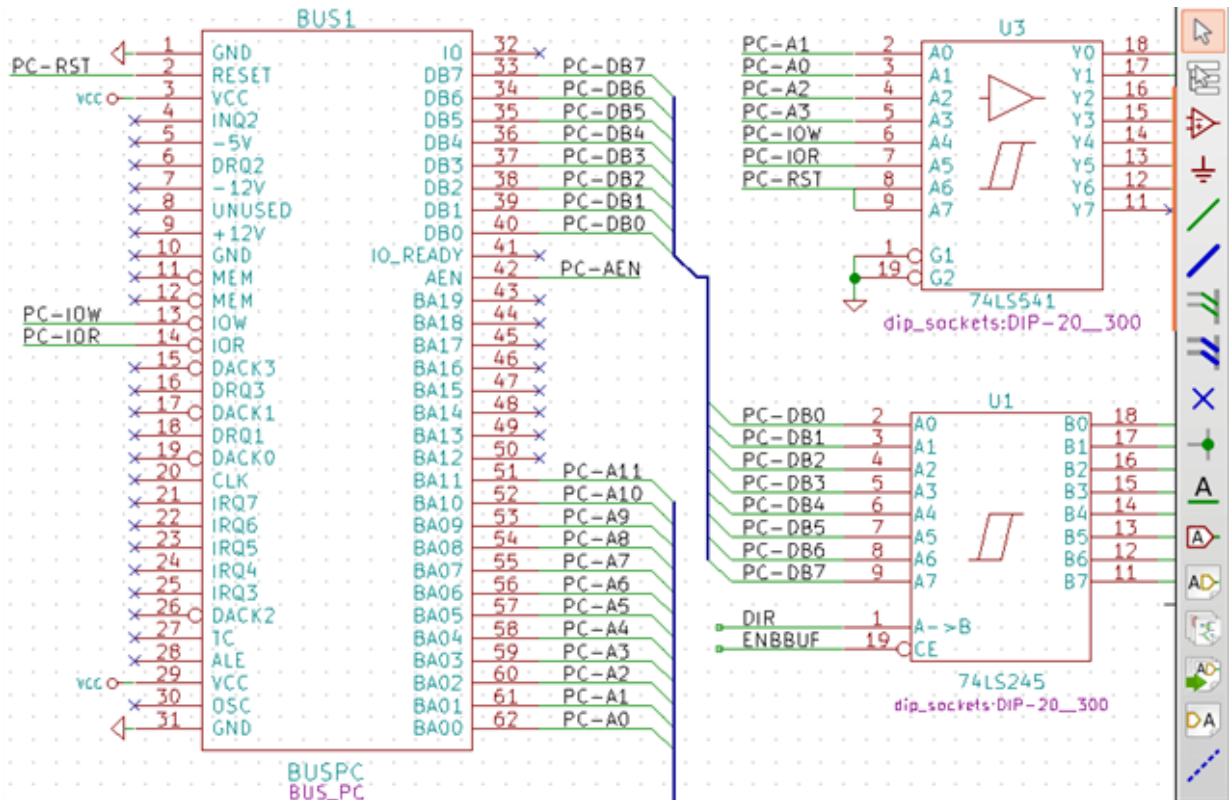
На предыдущем изображении показано случай, когда используются точки для соединения (проводники подключённые к выводам 22, 21, 20, 19 компонента DB25FEMALE).

Примечание 4:

Если две разные метки расположены на одном проводнике, то они соединяются вместе и становятся эквивалентными: все прочие элементы, подключенные к одной или другой метке соединяются между собой.

6.4.2 Соединения с помощью шин

На следующей схеме множество выводов подключено к шине.



6.4.2.1 Сигналы шины

С точки зрения схемы, шина — это совокупность проводников, метки которых имеют одинаковый префикс и заканчиваются порядковым номером. Например, PCA0, PCA1 и PCA2 являются сигналами шины PCA.

Законченная шина обозначается как PCA[N..m], где N и m — номера первого и последнего проводника этой шины. Таким образом, если шина PCA имеет 20 сигналов от 0 до 19, то, в итоге, шина будет обозначена как PCA[0..19]. Совокупность сигналов типа PCA0, PCA1, PCA2, WRITE, READ нельзя соединить в одну шину.

6.4.2.2 Соединение сигналов шины

Выводы, подключенные к одинаковым сигналам шины, соединяются по средствам меток. Невозможно подключить вывод напрямую к шине, такие соединения игнорируются в Eeschema.

На приведённом выше примере, соединения выполнены с помощью меток, расположенных на проводниках, которые подключены к выводам. Ввод шины (отрезок проводника под углом 45 градусов) всего лишь графический элемент и не требуется для создания логической связи.

На практике, использование команды повторения (клавиша *Insert*) позволяет значительно ускорить процесс создания соединений следующим образом, если шаг выводов соответствует шагу повторения (обычно, при работе с такими компонентами как память, микропроцессор и пр.):

- Расположить первую метку (например, PCA0)
- Использовать команду повторения столько раз, сколько нужно для создания всех сигналов. Eeschema автоматически создаст следующие метки (PCA1, PCA2, ...), выровненные по вертикали и, предположительно, на уровне остальных выводов.

- Начертить проводник под первой меткой. Затем использовать команду повторения для размещения остальных проводников под остальными метками.
- Если необходимо, расположите вводы в шину таким же образом (сначала расположите первый ввод, затем остальные с помощью команды повторения).

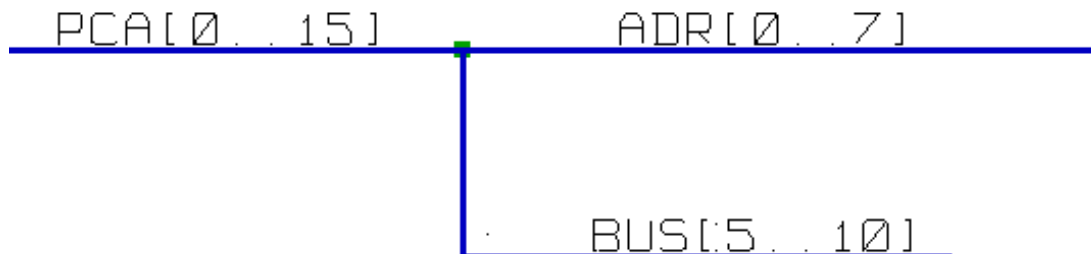
Замечание

В меню Настройки/Параметры редактора схем можно установить параметры повторения:

- Шаг по вертикали
 - Шаг по горизонтали
 - Инкремент метки (которая может увеличиваться на 2, 3 или уменьшаться).
-

6.4.2.3 Глобальные соединения между шинами

Иногда, может потребоваться соединить шины между собой, чтобы связать две шины с разными именами или, в случае иерархических схем, чтобы создать соединения между разными листами. Сделать это можно следующим способом.



Шины `PCA [0..15]`, `ADR [0..7]` и `BUS [5..10]` соединены вместе (заметьте, точка соединения используется для соединения вертикального отрезка шины с серединой горизонтального).

При этом, соответствующие сигналы тоже соединяются между собой: `PCA0` с `ADR0`, `PCA1` с `ADR1`, ..., `PCA7` с `ADR7`, и т.д.

Более того, `PCA5`, `BUS5` и `ADR5` тоже соединяются (также как и `PCA6`, `BUS6`, `ADR6` или `PCA7`, `BUS7`, `ADR7`).

И `PCA8` соединён с `BUS8` (как и `PCA9` с `BUS9`, и `PCA10` с `BUS10`).

6.4.3 Подключение символов питания


Если выводы питания компонента видимы, то они должны подключаться также как и любые другие выводы.

Такие компоненты как логические элементы или триггеры могут иметь скрытые выводы питания. С ними нужно работать аккуратно, потому что:

- К ним нельзя подключить проводники, так как они невидимы.
- Их имена неизвестны.

Более того, не стоит делать их видимыми и соединять как обычные выводы, так как схема при этом становится нечитаемой и не соответствует общим рекомендациям.

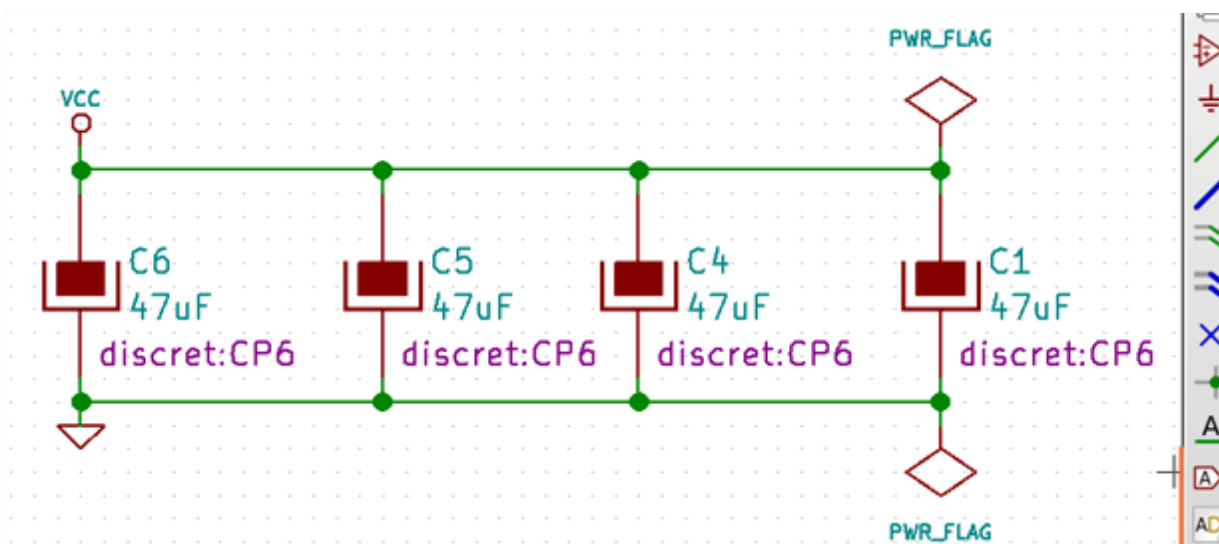
Замечание

Чтобы скрытые выводы стали видны, нужно установить параметр ``Показать скрытые выводы`` через меню Настройки/Параметры редактора схем или с помощью кнопки  на левой панели инструментов.

Eeschema автоматически соединяет невидимые выводы питания с одноимёнными силовыми линиями. Если нужно их подключить к линиям с другими именами (например, "GND" для компонентов TTL-логики, а "VSS" для КМОП-компонентов) — используйте символы питания.

Не рекомендуется использовать метки для соединения выводов питания. Они имеют только локальную область действия и не соединяются со скрытыми выводами.

На рисунке ниже показан пример подключения символов питания.




В этом примере земля (GND) соединена с выводом питания VSS, а символ питания VCC соединён с VDD.

Здесь видно два флага PWR_FLAG. Они указывают на то, что оба символа питания VCC и GND подключены к источнику питания. Без этих флагов, при проверке электрических правил, будет показано сообщение: *Предупреждение: символы питания не запитаны.*

Все эти символы — компоненты библиотеки «power».


6.4.4 Флаг «Не подключено»

Эти символы очень полезны при контроле схемы с помощью электрических правил проектирования. С их помощью можно указать что выводы оставлены без соединений намеренно.

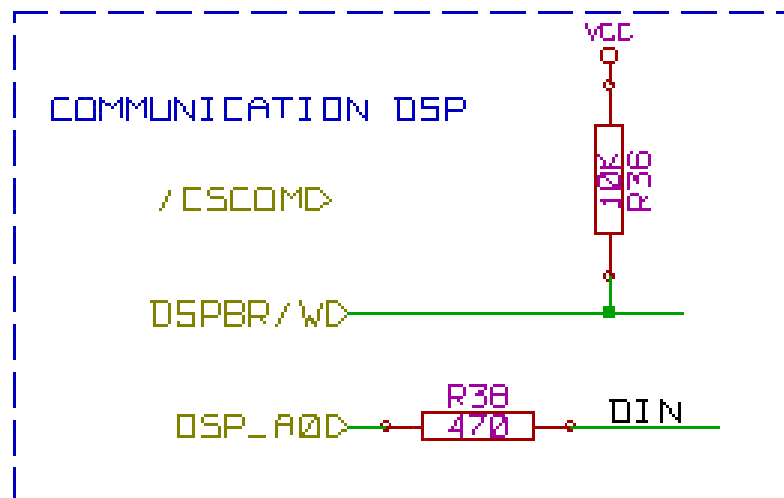
Так, если выводы нужно оставить не подключенными, расположите флаг "Не подключено" на их концах с помощью инструмента . Эти символы никак не влияют на создаваемый список соединений.

6.5 Чертежные элементы


6.5.1 Текстовые комментарии

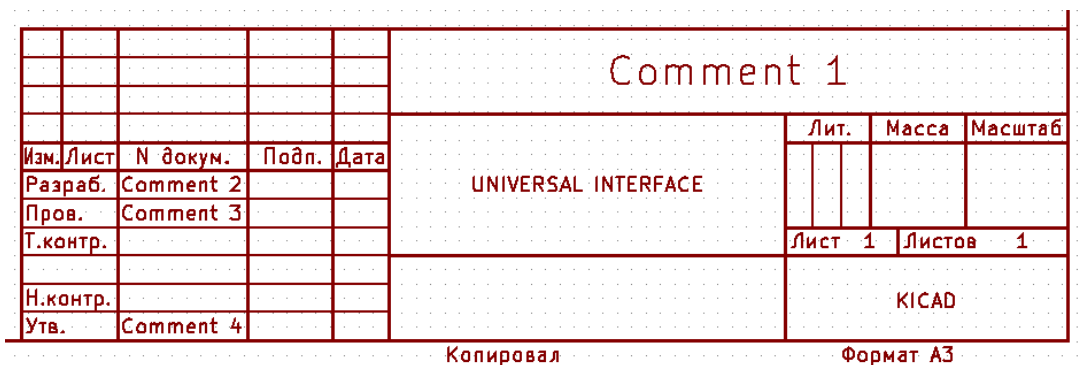
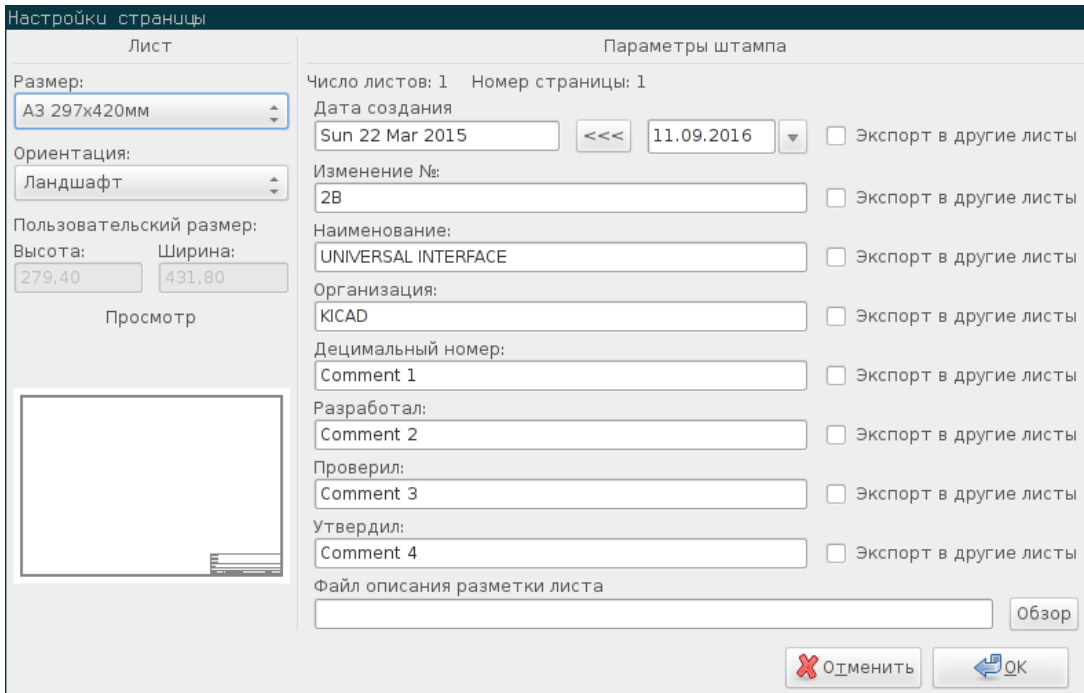
Может быть полезным (для уточнений в схеме) размещение пояснений в виде текста или рамок. Текстовые поля (кнопка **T**) и графические линии (кнопка ) предназначены для этих целей, в отличие от меток и проводников, которые служат для создания соединений.

Здесь представлен пример рамки с текстовым комментарием.



6.5.2 Основная надпись

Содержимое основной надписи изменяется с помощью инструмента .

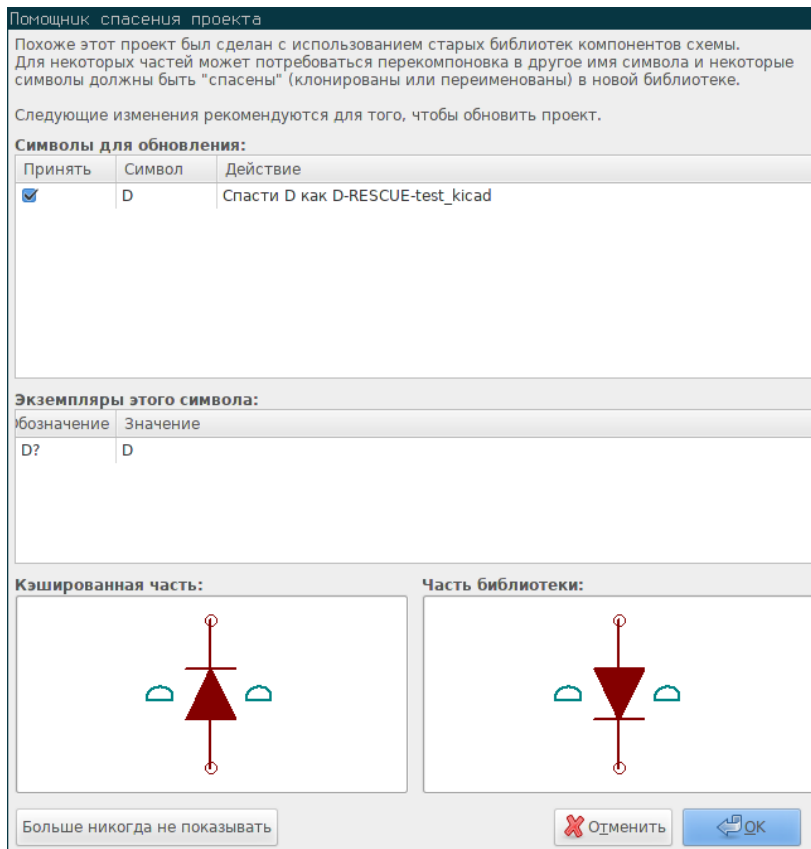


Количество листов (Лист/Листов) подсчитывается автоматически.

6.6 Восстановление кэшированных компонентов

По умолчанию, Eeschema загружает компоненты из библиотек согласно порядку настроенных путей. Это может создать проблемы при загрузке старых проектов: если компонент библиотеки изменился с момента использования его в проекте, то он будет автоматически обновлён до новой версии. Обновлённый компонент может иметь другие размеры или ориентацию, что в свою очередь испортит схему.

При сохранении проекта рядом с ним создается кэш библиотеки компонентов, в котором сохраняются все используемые компоненты. Это позволяет распространять проект без предоставления всех используемых библиотек. Если при загрузке проекта компонент существует в обеих библиотеках, кэшированной и системной, то Eeschema выполнит поиск конфликтов между ними. Все найденные проблемы будут показаны в таком диалоговом окне:



Из данного примера видно, что проект изначально использовал диод с катодом вверх, но теперь в библиотеке он расположен катодом вниз. Такое изменение разрушит проект! Если нажать кнопку ОК, старый компонент будет сохранён в специальной «rescue» библиотеке под новым именем, что решает проблему конфликта с новой версией компонента.

Если нажать кнопку Cancel, никаких действий по восстановлению не будет предпринято и Eeschema загрузит новую версию компонента из библиотеки. Если на данном этапе сохранить схему, кэш будет перезаписан и старые компоненты станут невозможно восстановить. Если схема была сохранена, всё равно, можно вернуться и запустить процесс восстановления заново с помощью пункта меню "Инструменты" → "Спаси старые компоненты".

Чтобы этот диалог больше не отображался нужно нажать кнопку "Больше никогда не показывать". По умолчанию, никакие действия не будут предприниматься и будут загружаться новые версии компонентов. Это поведение можно изменить (вернуть обратно) в диалоговом окне настроек библиотек.

Глава 7


Иерархические схемы

Введение

Иерархическое представление схемы, в общем случае, является хорошим решением для проектов из нескольких листов. Использование проектов данного типа понадобится при:

- Больших форматах листов, что приводит к проблемам при печати и в использовании.
- Множестве листов, которые лучше привести к иерархической структуре.

Законченная схема состоит из основного листа, называемого корневым (root), и вложенных листов, которые формируют иерархию. При этом, умелое разделение проекта на отдельные листы часто повышает читаемость схемы.

Из корневого листа должны быть доступны все вложенные. Управлять иерархическими схемами с помощью Eeschema очень просто благодаря встроенному "Навигатору по иерархии", который вызывается с помощью кнопки  на верхней панели инструментов.

Всего существует два типа иерархий: первый и наиболее распространённый только что был рассмотрен. Суть второго состоит в том, чтобы создать компонент в библиотеке, который внешне похож на обычный компонент схемы, но при этом связан со схемой, которая описывает его внутреннюю структуру.

Второй тип используется при разработке микросхем (интегрированных схем) и при этом нужно использовать библиотеки функций в проектируемых схемах.

Eeschema на данный момент не охватывает иерархии второго типа.

Иерархия может быть:

- простой: содержит схемы, которые используются только единожды
- сложной: содержат схемы, которые используются несколькими листами (имеют несколько экземпляров)
- плоской: простая иерархия, но без указания соединений между листами


Eeschema может работать со всеми этими иерархиями.

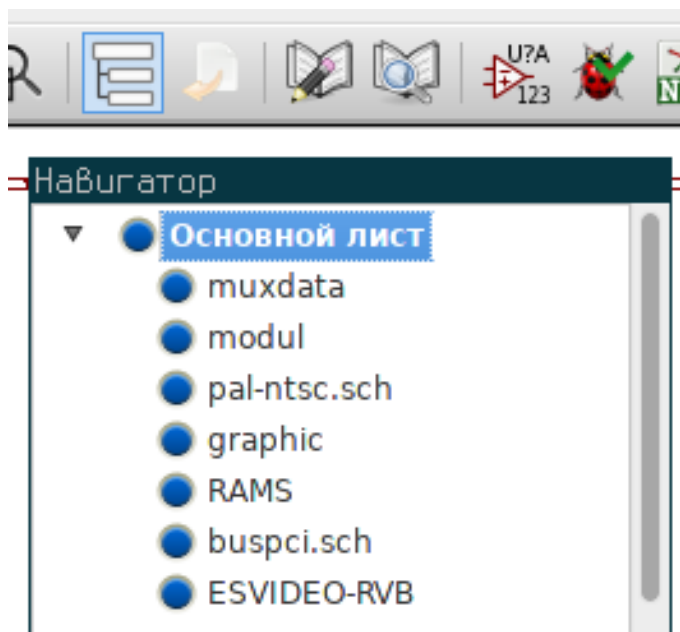
Создать иерархическую схему просто, вся иерархия строится начиная с корневого листа, так же как и обычная схема, состоящая из одного листа.

Далее, нужно изучить две важные операции:

- Создание вложенных листов.
 - Построение электрических соединений между вложенными листами.
-

7.1 Навигация по иерархии

Перемещаться между листами очень легко благодаря инструменту навигации, который вызывается с помощью кнопки  на верхней панели инструментов.






К каждому листу можно получить доступ с помощью щелчка левой кнопкой мыши по его имени. Для быстрого доступа, нажмите правой кнопкой мыши на имени листа и выберите "Войти в лист" или дважды щёлкните левой кнопкой мыши внутри символа листа.

Также, чтобы покинуть текущий лист и перейти к родительскому листу, нажмите правую кнопку мыши на любом пустом месте схемы, где нет никаких объектов, и левой кнопкой мыши выберите пункт контекстного меню "Покинуть лист" или нажмите клавиши Alt+Backspace.

7.2 Локальные, иерархические и глобальные метки

7.2.1 Свойства

Локальные метки, инструмент , соединяет сигналы только внутри листа. Иерархические метки (инструмент ) соединяет сигналы только между самим листом и иерархическими выводами, расположенными на родительском листе.

Глобальные метки (инструмент ) соединяет сигналы внутри всей иерархии. Выводы питания (типа *вход питания* и *выход питания*) скрыты, как и глобальные метки, потому что они соединены между собой во всей иерархии.

Замечание

В иерархии (простой или сложной) можно использовать как иерархические так и глобальные метки.

7.3 Создание иерархических схем

Нужно:

- Разместить символ иерархического листа на основной схеме.
- Войти в новый лист схемы (вложенный) с помощью навигатора и начертить её как и любую другую схему.
- Разместить электрические соединения между двумя схемами используя глобальные или иерархические метки внутри вложенного листа и выводы иерархических листов с такими же именами на корневом листе. Эти иерархические выводы добавляются к символу листа на корневой схеме. Проводники подключаются к ним также как и к обычным выводам компонента.

7.4 Символ листа

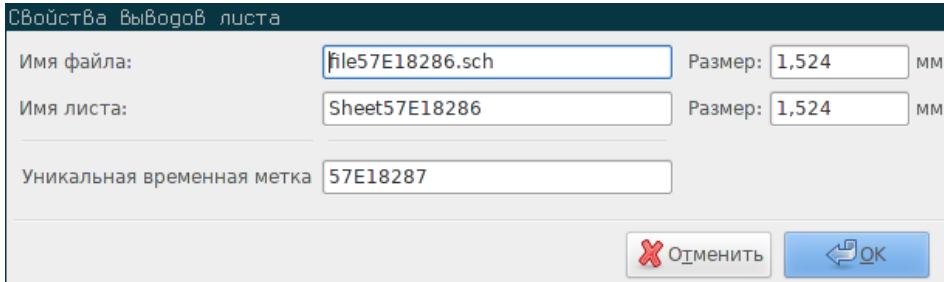
Чертите прямоугольник по двум диагональным точкам для создания символа вложенного листа.

Размер этого прямоугольника должен позволять позже разместить специальные метки, иерархические выводы, соответствующие глобальным и иерархическим меткам из вложенного листа.

Эти метки подобны обычным выводам компонентов. Используйте инструмент .

Нажмите левой кнопкой мыши для расположения верхнего левого угла. Затем, нажмите еще раз для расположения нижнего правого угла, образуя достаточно большой прямоугольник.

После этого появится запрос на ввод имени файла схемы и имени листа (в основном для доступа к соответствующей схеме из навигатора).



Нужно ввести, как минимум, имя файла схемы. Если имя листа оставить пустым, то в его качестве будет использовано имя файла (обычно так и поступают).

7.5 Иерархические выводы

Здесь будет рассказано как создать точки соединения (иерархические выводы) символа листа, который был создан ранее.


Эти точки соединения подобны обычным выводам компонентов, но с возможностью подключения целой шины к одному выводу.

Создать их можно двумя способами:

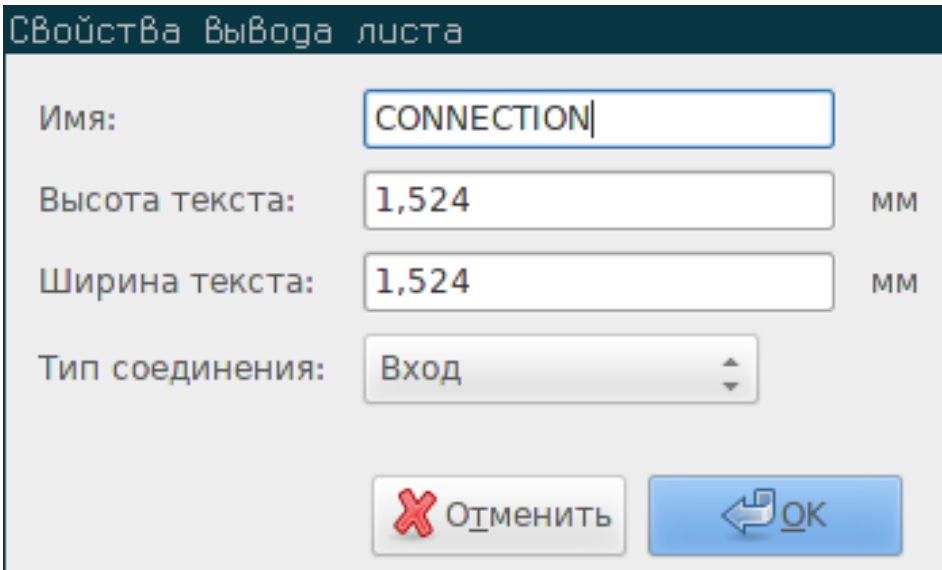
- Разместить разные выводы перед черчением вложенной схемы (ручное размещение).
- Разместить разные выводы после черчения вложенной схемы и глобальных меток (полуавтоматическое размещение).

Второе решение более предпочтительно.

Ручное размещение:

- Выберите инструмент .
- Нажмите на символ листа, на котором хотите расположить вывод.


Ниже приведён пример создания иерархического вывода с именем "CONNECTION":



Можно задать имя, размер и тип вывода в процессе его создания или позже, используя правую кнопку мыши для вызова контекстного меню и выбора "Править вывод листа".


На вложенной схеме иерархические метки должны называться точно так же, как и иерархические выводы. За этим нужно самостоятельно следить, именно из-за этого второй способ, описанный ниже, более предпочтителен.

Автоматическое расположение:

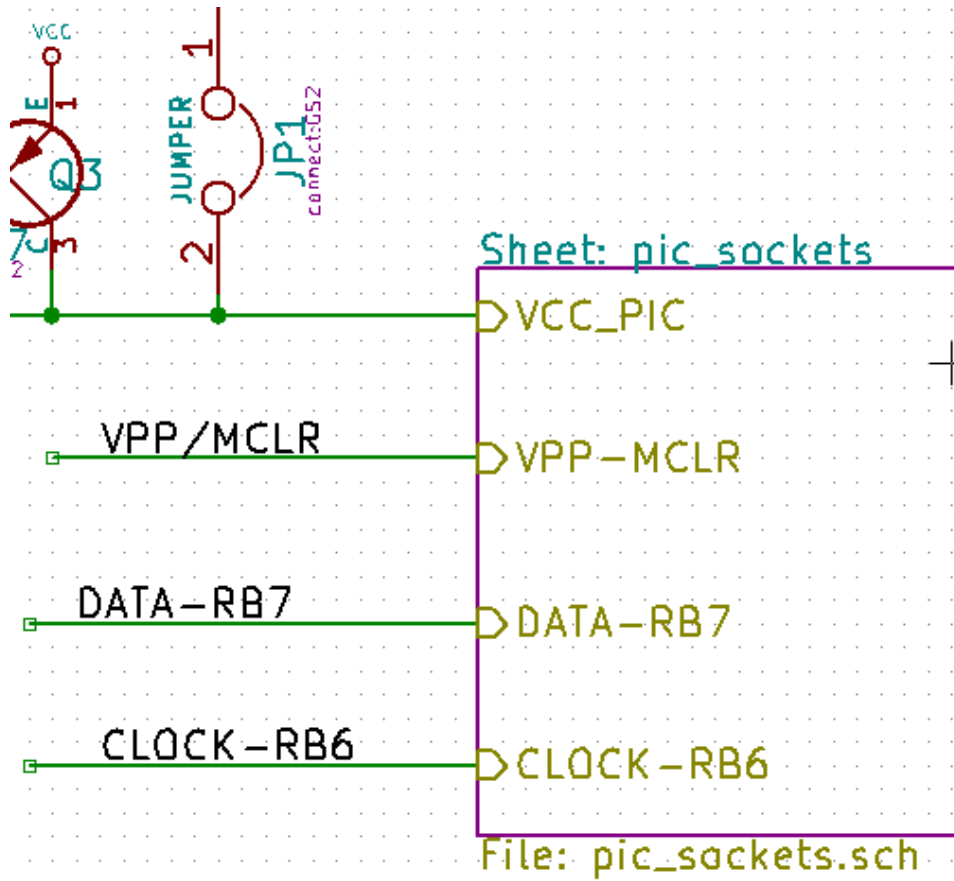
- Выберите инструмент .
- Нажмите левой кнопкой мыши на символе иерархического листа, из которого нужно импортировать выводы соответствующие глобальным меткам, расположенными на вложенной схеме. Иерархический вывод добавляется только для новой глобальной метки, т.е. той, у которой ещё нет соответствующего иерархического вывода.
- Нажмите на месте, где нужно расположить этот вывод.

Таким образом можно расположить все необходимые выводы быстро и без ошибок. И все они будут соответствовать связанным глобальным меткам.

7.6 Иерархические метки

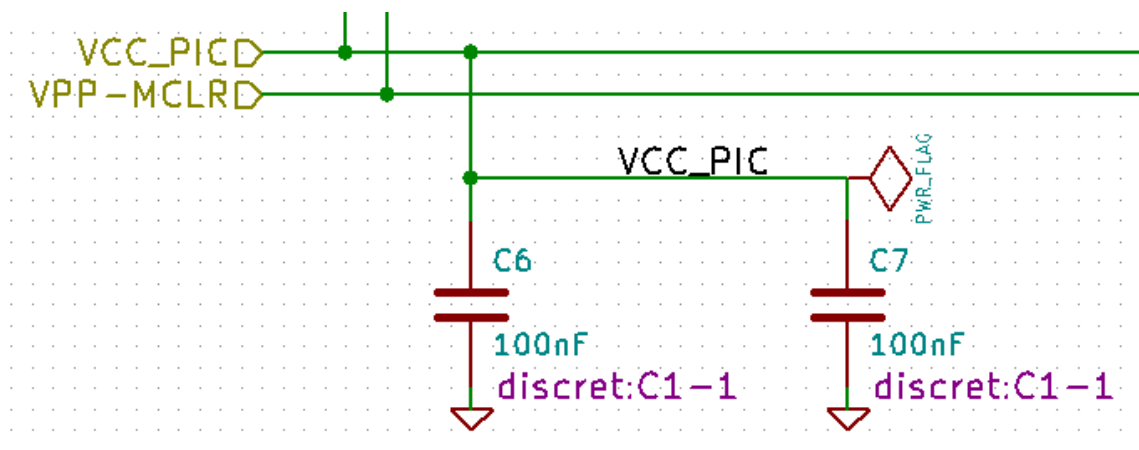
Каждый вывод, добавленный на символ листа, должен соответствовать метке на вложенном листе. Такие метки называют иерархическими. Иерархические метки подобны локальным, но образуют соединение между вложенным листом и корневым. Внешнее представление обоих иерархических элементов (вывода и метки) похоже. Иерархические метки добавляются с помощью инструмента .

Ниже, для примера, показан участок корневого листа:



Заметьте, вывод VCC_PIC подключён к разъему JP1.

Вот как выглядят соответствующие соединения на вложенном листе:



Можно увидеть две соответствующие иерархические метки, образующие соединение между двумя иерархическими листами.

Замечание

Можно использовать иерархические метки и выводы для соединения двух шин, соответствующих синтаксису (Шина [N..m]), описанному ранее.

7.6.1 Локальные, иерархические, глобальные метки и скрытые выводы питания

Здесь приведены примечания для разных способов образования соединений без использования проводников.

7.6.1.1 Обычные метки

Обычные метки действуют локально, т.е. ограничены листом схемы, в котором расположены. Это связано с тем, что:

- Каждый лист имеет свой номер.
- Локальная метка связана с номером листа.

Таким образом, если расположить метку "АБВ" на листе №3, в итоге получим "АБВ_3". Также, если расположить метку "АБВ" на листе №1 (корневом листе), то в итоге получим "АБВ_1", что отличается от "АБВ_3". Это справедливо всегда, даже если лист всего один.

7.6.1.2 Иерархические метки

Все что сказано о простых метках, относится и к иерархическим меткам.

Так, на одном листе, иерархическая метка "АБВ" соединяется с соответствующей локальной меткой "АБВ", но не соединяется с иерархическими или локальными метками "АБВ" из других листов.

Но, в то же время, иерархическая метка может быть подключена к соответствующему иерархическому выводу, размещенном на символе листа корневой схемы.

7.6.1.3 Скрытые выводы питания

Как уже было показано, скрытые выводы питания соединяются вместе, если они имеют одинаковые имена. Таким образом, выводы питания объявленные "скрытыми" и обозначены VCC — соединяются и формируют единую цепь VCC, на листе, в котором расположены.

Это значит что, если разместить метку VCC на вложенном листе, то она не будет соединена с выводами VCC, так как эта метка примет вид VCC_n, где n — номер листа.

Если, все же, нужно соединить метку VCC с цепью выводов питания VCC, необходимо явно подключить скрытый вывод питания к символу питания VCC.

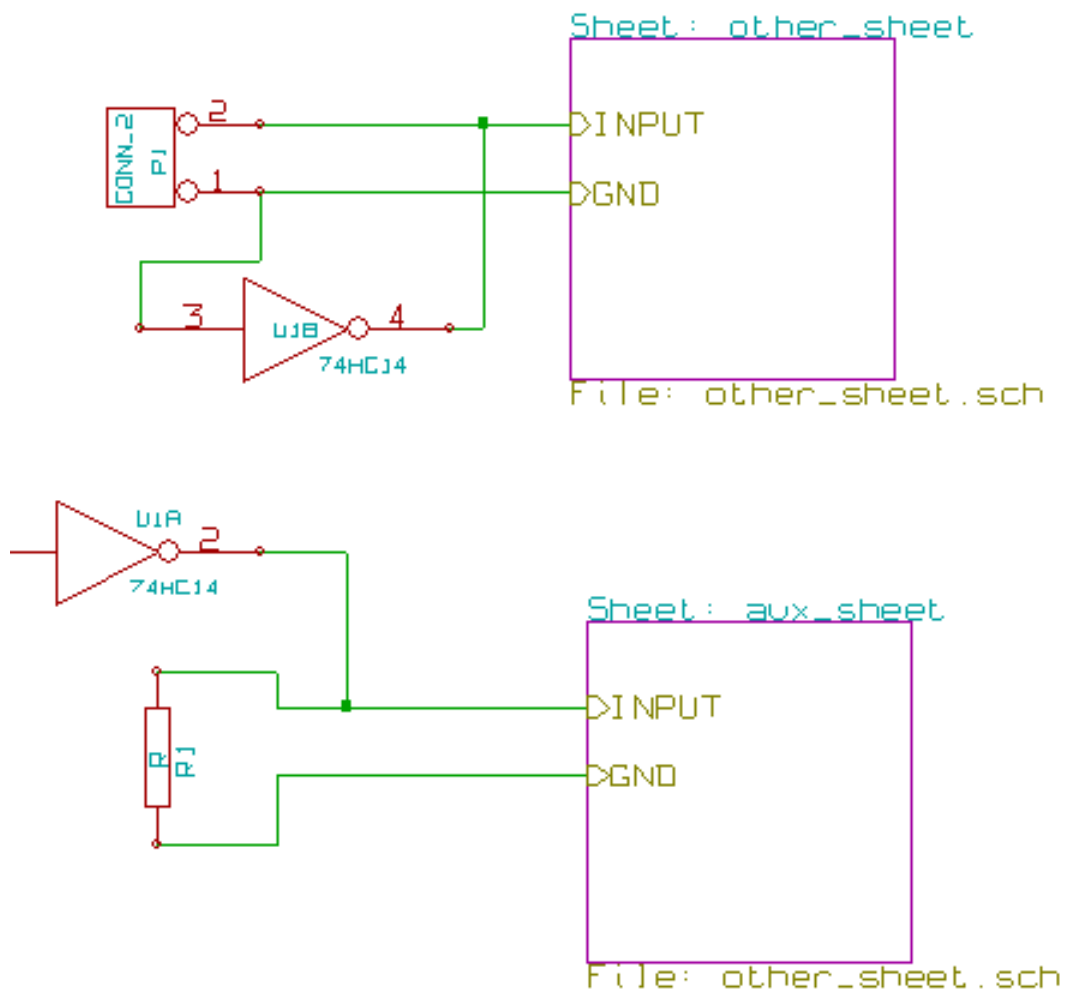
7.6.2 Глобальные метки

Глобальные метки, имеющие одинаковые имена, соединяются между собой во всей иерархии.

(метки питания, такие как VCC, — это тоже глобальные метки)

7.7 Сложные иерархии

Рассмотрим такой пример. Одна и та же схема используется дважды (существует две инстанции). Два листа ссылаются к одной и той же схеме, так как имя файла одинаковое у обоих листов («other_sheet.sch»). Имена листов должны быть уникальными.



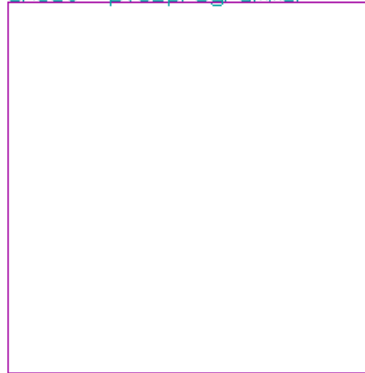
7.8 Плоские иерархии

Можно создавать проекты, использующие множество листов, которые не имеют между собой явных соединений (плоская иерархия) если выполняются следующие правила:

- Создан корневой лист, содержащий остальные листы, и служащий связующим между ними.
- Непосредственные соединения не нужны.
- Используйте глобальные метки вместо иерархических во всех листах.

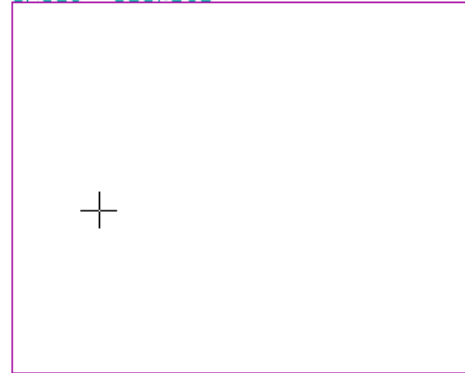
Здесь показан пример такого корневого листа.

Sheet: pic_programmer



File: pic_programmer.sch

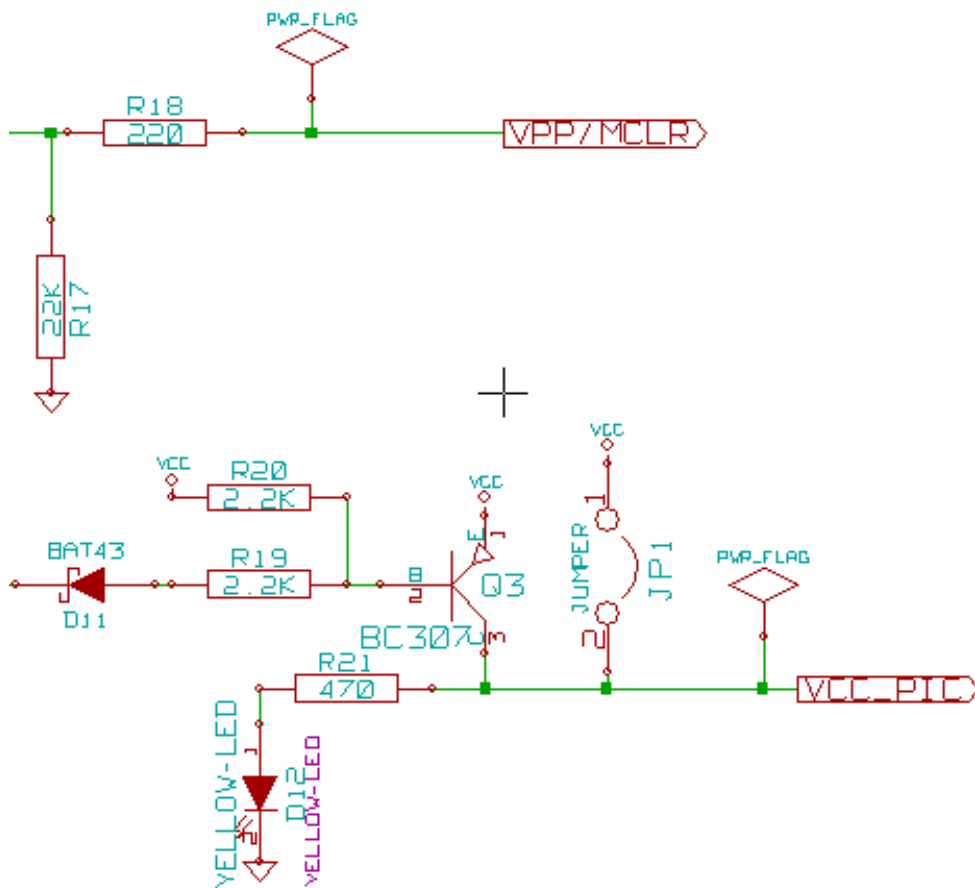
Sheet: sockets



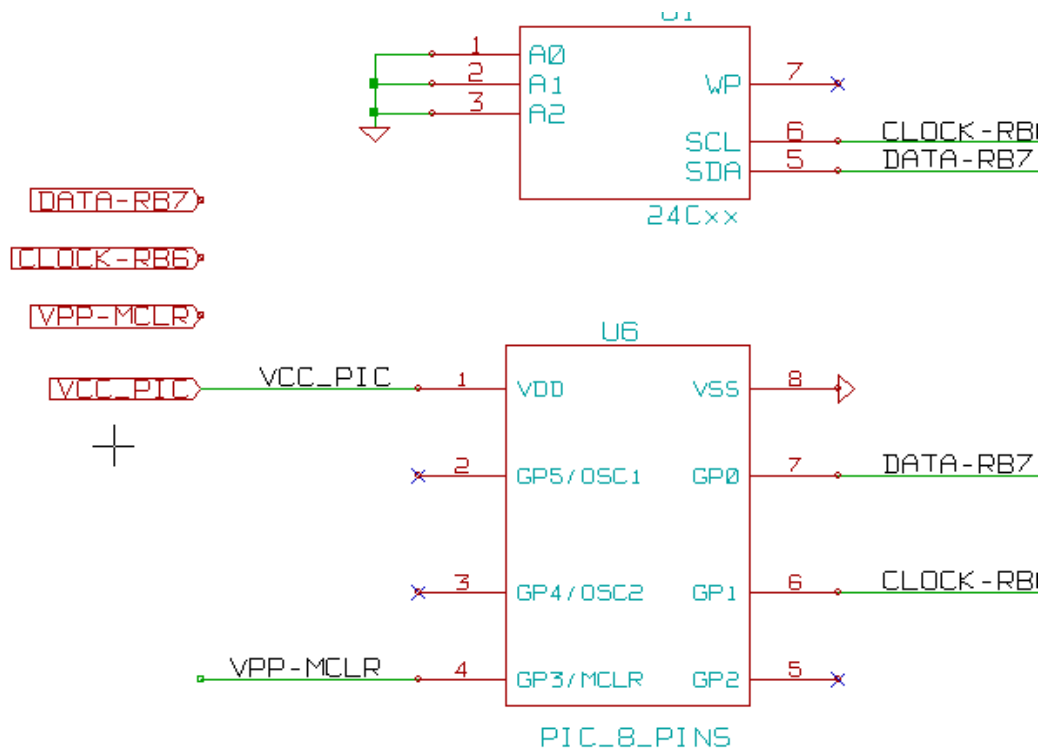
File: pic_sockets.sch

На нём два вложенных листа, соединённых посредством глобальных меток.

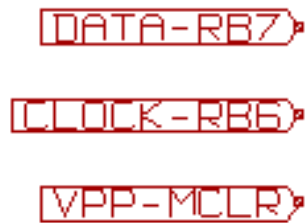
Так выглядит pic_programmer.sch.



А так pic_sockets.sch.




Обратите внимание на глобальные метки.

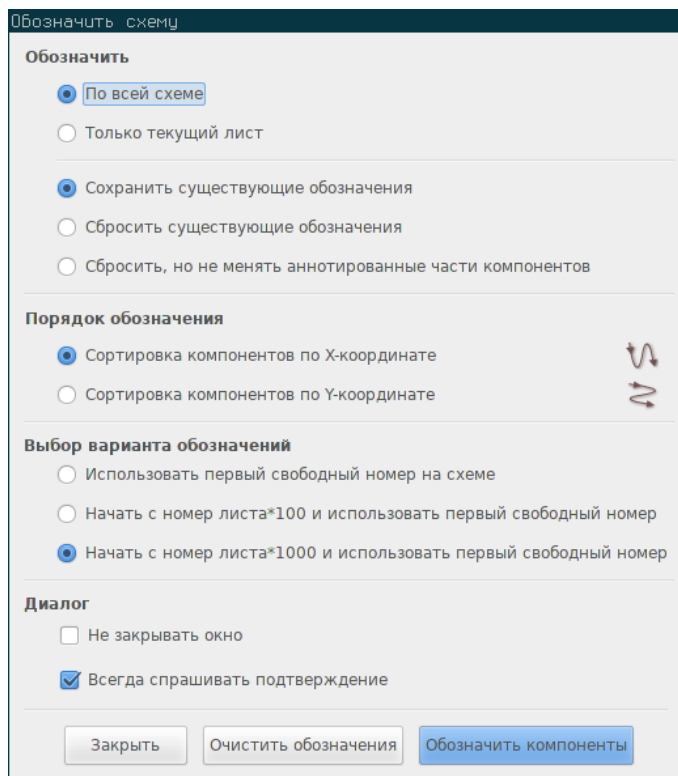


Глава 8

Инструмент обозначения элементов

Введение

Инструмент обозначения компонентов позволяет автоматически присвоить уникальный порядковый номер каждому компоненту в схеме. В составных компонентах суффиксы присваиваются частям таким образом, чтобы использовать как можно меньше целых компонентов. Инструмент обозначения компонентов вызывается с помощью кнопки . Ниже показано его диалоговое окно.



Доступны следующие варианты обозначения компонентов:

- Обозначить все компоненты (параметр: Сбросить существующие обозначения)
- Обозначить все компоненты, но не изменять назначенные ранее суффиксы частей.
- Обозначить только необозначенные компоненты. Необозначенным компонентом называется компонент, порядковый номер которого обозначен вопросительным знаком ?.
- Обозначить всю иерархию листов (параметр: По всей схеме).

- Обозначить только текущий лист (параметр: Только текущий лист).

Параметр «Сбросить, но не менять аннотированные части компонентов» позволяет сохранить все присвоенные суффиксы частям составных компонентов. Таким образом, если имеются компоненты U2A и U2B, то их можно заново обозначить как U1A и U1B, соответственно, но никак не U1A и U2B или U2B и U2A. Это будет полезным в случае, когда уже спланировано где каждая часть будет расположена и нужно сохранить соответствие выводов.

Выбранный порядок обозначения будет использоваться на каждом листе иерархии.

Исключением является случай, когда вся схема уже обозначена (на всех листах) и добавляются новые компоненты, которые обозначаются отдельно, чтобы не нарушать уже присвоенные обозначения.

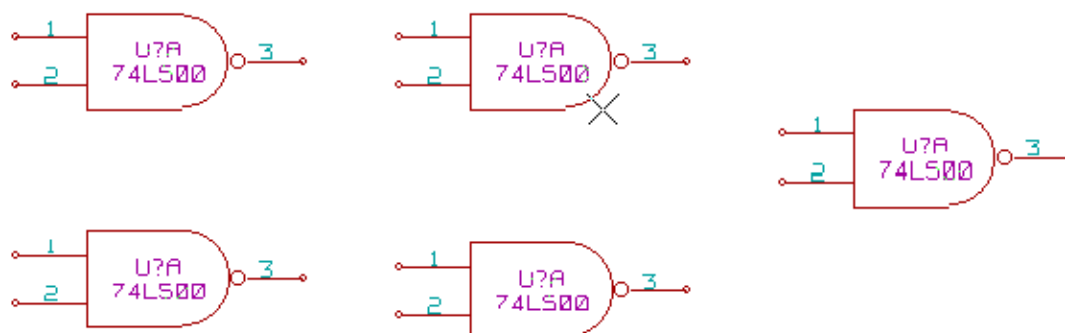
Выбор варианта обозначения позволяет использовать разные методы для образования порядкового номера:

- Использовать первый свободный номер в схеме: компоненты обозначаются от 1 (для каждого типа элемента). Если присутствуют уже обозначенные компоненты, будут использоваться не занятые номера.
- Начать с номер_листа*100 и использовать первый свободный: порядковые номера начинаются с 101 для первого листа, с 201 для второго и т.д. Если имеется более 99 элементов одного типа (DA, R, ...) внутри первого листа то инструмент обозначения будет использовать порядковый номер 200 и более, а обозначения на втором листе начнутся со следующего свободного.
- Начать с номер_листа*1000 и использовать первый свободный. Порядковые номера начинаются с 1001 на первом листе, с 2001 на втором и т.д.

8.1 Примеры работы инструмента

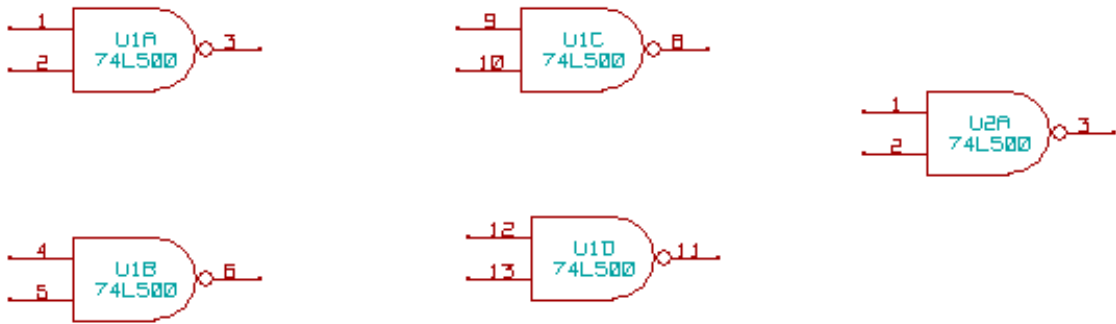
8.1.1 Порядок обозначения

На этом примере показано 5 элементов без обозначения.

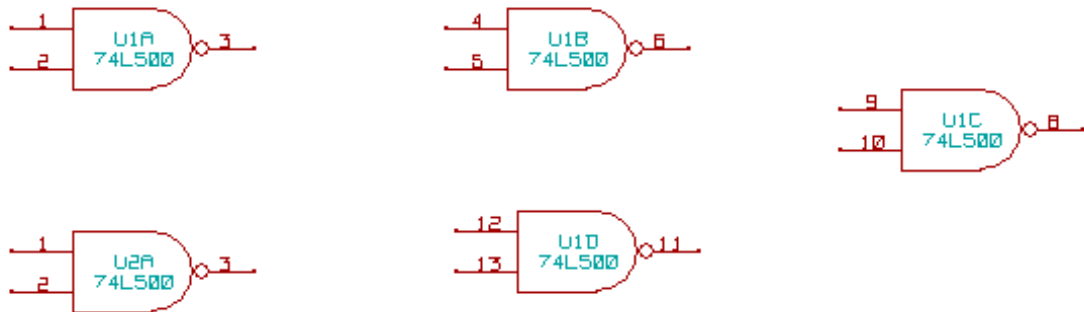


После работы инструмента обозначения компонентов будет получен следующий результат.

Сортировка компонентов по X-координате.



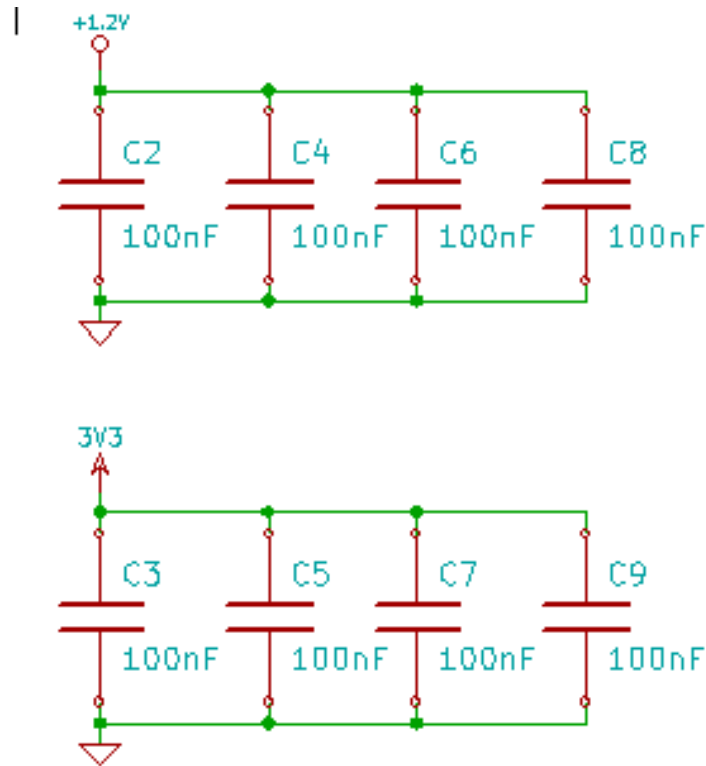
Сортировка компонентов по Y-координате.



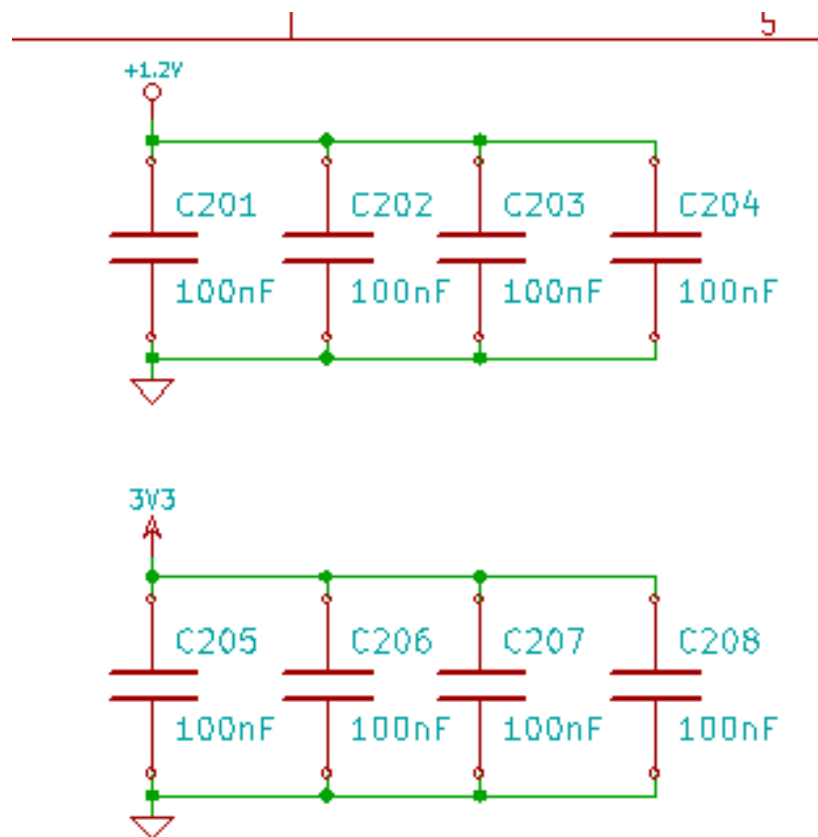
Как можно заметить, четыре части элемента 74LS00 образуют один компонент U1, а пятый — U2.

8.1.2 Выбор варианта обозначений

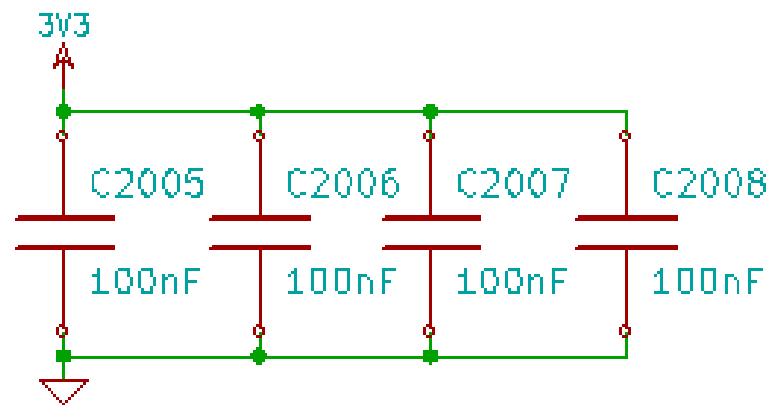
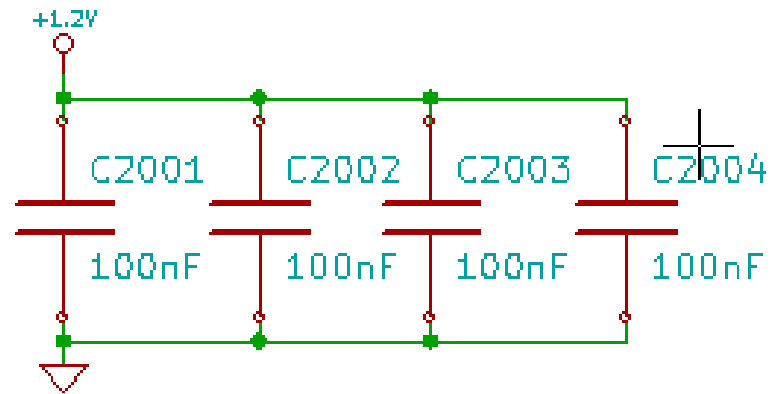
Далее приведён пример обозначения на листе №2 с применением параметра "Использовать первый свободный номер на схеме".



Применение параметра "Начать с номер листа*100 и использовать первый свободный номер" даст следующий результат.



Применение параметра "Начать с номер листа*1000 и использовать первый свободный номер" даст следующий результат.



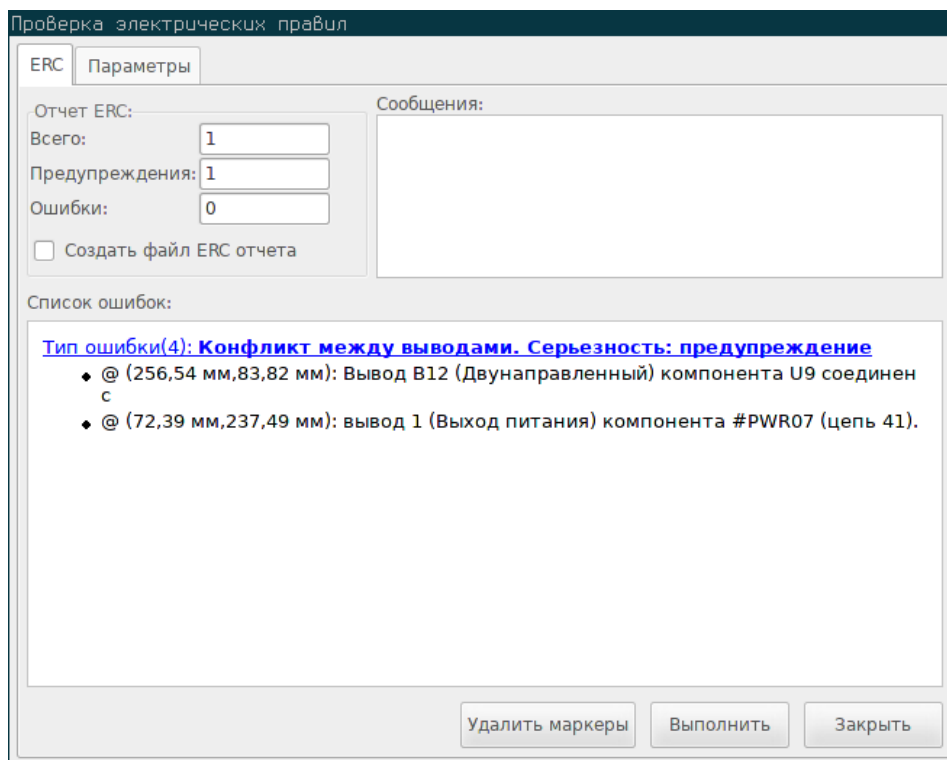
Глава 9

Проверка электрических правил проектирования


Введение

Инструмент проверки электрических правил (ERC) выполняет автоматический контроль схемы. Он обнаружит любые ошибки в схеме, такие как не подключенные выводы компонентов, иерархических листов или неверные соединения. Вообще-то, автоматическая проверка правил не без изъянов, подпрограмма, отвечающая за поиск всех возможных ошибок еще не на все 100% завершена. Тем не менее, этот инструмент очень полезен, потому что позволяет определить оплошности и мелкие ошибки.

На деле, все обнаруженные ошибки должны быть проверены и исправлены прежде чем двигаться далее. Качество проверки напрямую зависит от того, насколько точно установлен тип каждого вывода при проектировании библиотеки компонентов. В процессе проверки электрических правил сообщения выдаются как «ошибки» или «предупреждения».



9.1 Как производить проверку правил

Проверку правил проектирования начинают с нажатия кнопки .

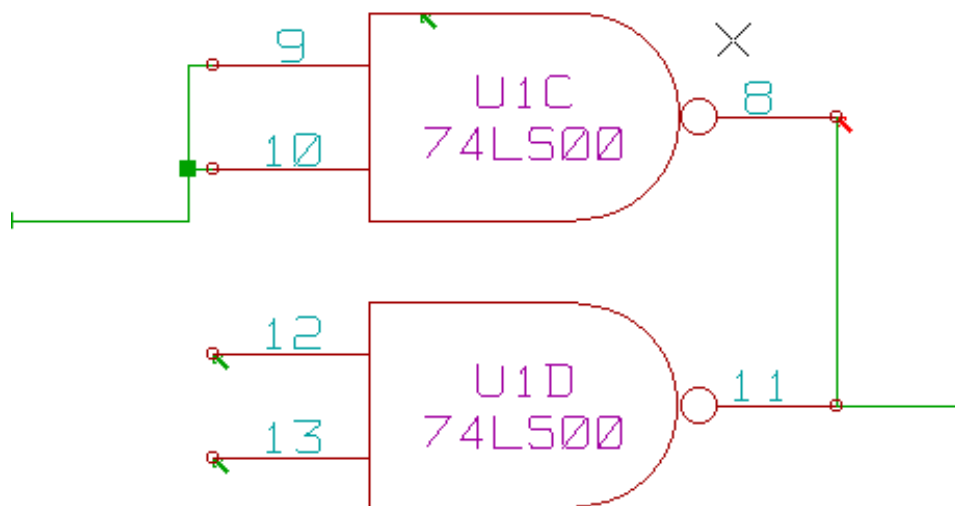
Предупреждения располагаются на элементах схемы в месте обнаружения ошибки (выводы или метки).

Замечание

- В диалоговом окне, с помощью нажатия на сообщение об ошибке можно перейти к соответствующему маркеру на схеме.
 - В схеме, щелчок правой кнопки мыши на маркере предоставляет доступ к соответствующему поясняющему сообщению.
-

Также, из диалогового окна можно удалять маркеры ошибок.

9.2 Пример выполнения проверки правил

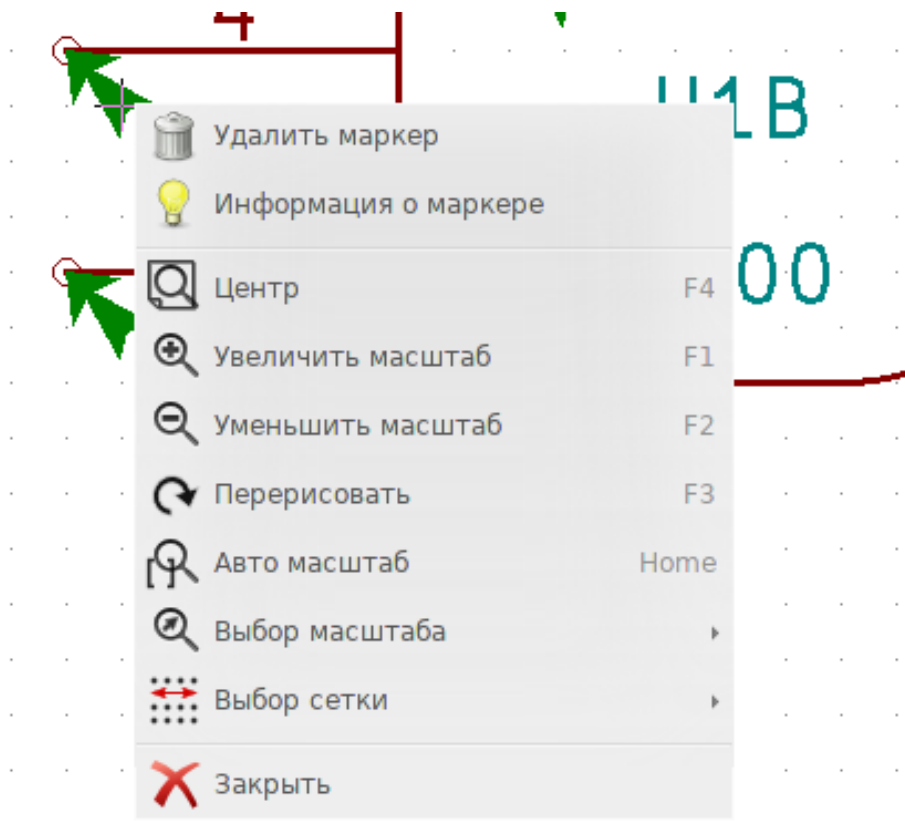


Здесь показано четыре ошибки:

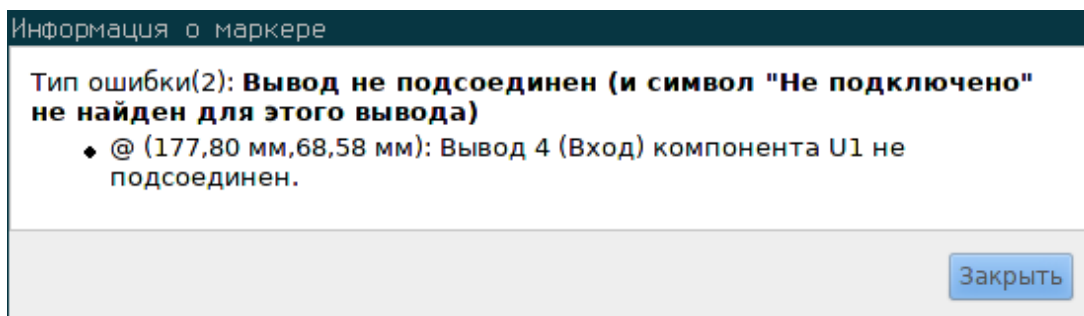
- Два выхода были ошибочно соединены вместе (красная стрелка).
- Два входа оставлены не подключенными (зелёные стрелки).
- Ещё одна ошибка на скрытом выводе питания из-за отсутствия флага питания (зелёная стрелка вверх).

9.3 Получение информации об ошибках

Щелчок правой кнопки мыши на маркере вызывает контекстное меню, позволяющее открыть информационное окно об ошибке.



И если выбрать пункт "Информация о маркере" — появится сообщение с описанием ошибки.

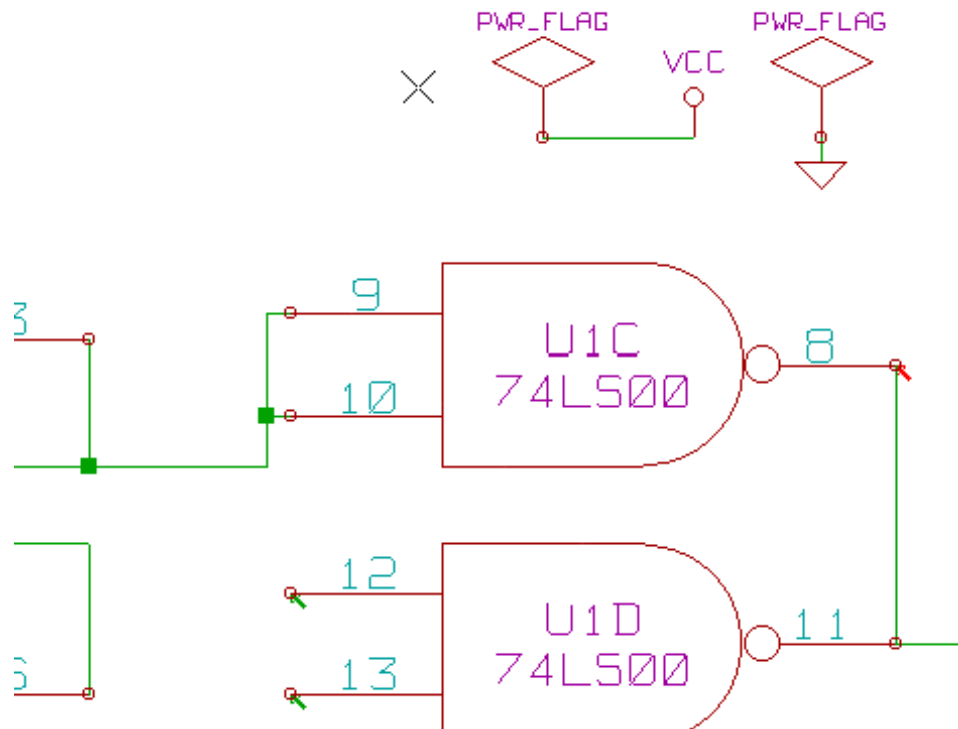


9.4 Выводы питания и флаги питания

Очень часто появляются ошибки или предупреждения на выводах питания, хотя все выглядит нормально, как на примере выше. Это происходит потому, что в большинстве проектов питание подаётся через разъемы, которые, сами по себе, не являются источниками (в отличие от регуляторов, в которых выход обозначен как "выход питания").

Если инструмент проверки электрических правил обнаружит цепь питания, в которой нет ни одного выхода питания, то будет создано сообщение о том, что выводы, подключенные к данной цепи — не запитаны.

Дабы избежать подобных сообщений, нужно разместить флаг "PWR_FLAG" на соответствующий символ питания. Посмотрите на следующий пример:

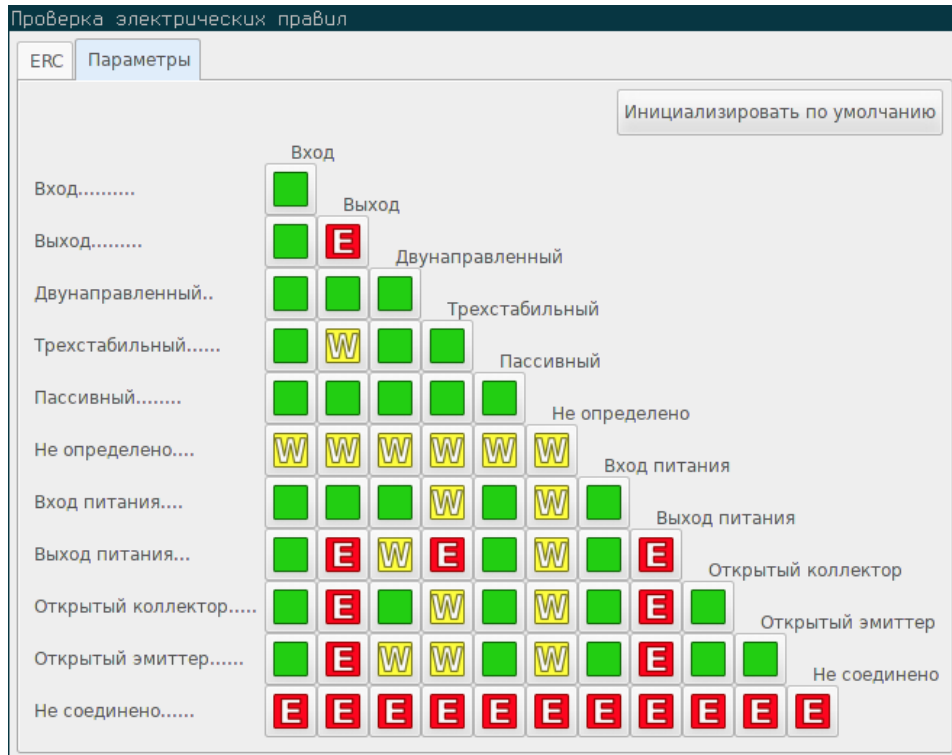


Метки ошибок исчезнут при следующей проверке правил.

В большинстве случаев, флаг PWR_FLAG должен быть подключён к GND, так как регуляторы имеют только вывод выхода обозначенный как выход питания, а общий вывод — нет (он обозначается как вход питания). Таким образом, общий провод всегда будет оставаться не запитанным, если не использовать PWR_FLAG.

9.5 Настройка

Вкладка *Параметры* позволяет настроить правила соединений и установить уровень сообщений (ошибка или предупреждение).



Правила настраиваются путём нажатия левой кнопки мыши на квадратах матрицы, циклически перебирая варианты: нормально, предупреждение, ошибка.

9.6 Отчёт проверки правил

Если отмечена опция "Создать файл ERC отчета", то будет создан и сохранён файл с информацией о проверке правил проектирования. Расширение таких файлов — .erc. Далее приведен пример содержимого файла отчёта.

```
b''Ob''b''tb''b''cb''b''eb''b''tb'' ERC (b''Cb''b''бb'' 24 b''cb''b''eb''b''nb'' 2016 ←
11:50:12, b''Kb''b''ob''b''дб''b''иб''b''pb''b''об''b''вб''b''кб''b''аб'' UTF8 )

***** Sheet /
ErrType(4): b''Kb''b''ob''b''nb''b''fb''b''лб''b''иб''b''кб''b''tb'' b''mb''b''eb''b''жб''b ←
''дб''b''yb'' b''вб''b''ыб''b''вб''b''об''b''дб''b''аб''b''мб''b''иб''. b''Cb''b''eb''b ←
''pb''b''ьб''b''eb''b''зб''b''nb''b''об''b''cb''b''tb''b''ьб'': b''пб''b''рб''b''eb''b ←
''дб''b''yb''b''пб''b''рб''b''eb''b''жб''b''дб''b''eb''b''nb''b''иб''b''eb''
@ (256,54 b''mb''b''mb'',83,82 b''mb''b''mb''): b''Vb''b''ыб''b''вб''b''об''b''дб'' В12 ←
(b''Дб''b''вб''b''yb''b''nb''b''аб''b''пб''b''рб''b''аб''b''вб''b''лб''b''eb''b ←
''nb''b''nb''b''ыб''b''йб'') b''кб''b''об''b''мб''b''пб''b''об''b''nb''b''eb''b''nb'' ←
b''tb''b''аб'' U9 b''cb''b''об''b''eb''b''дб''b''иб''b''nb''b''eb''b''nb'' b''cb''
@ (72,39 b''mb''b''mb'',237,49 b''mb''b''mb''): b''вб''b''ыб''b''вб''b''об''b''дб'' 1 ( ←
b''Vb''b''ыб''b''xb''b''об''b''дб'' b''пб''b''иб''b''tb''b''аб''b''nb''b''иб''b'' ←
''яб'') b''кб''b''об''b''мб''b''пб''b''об''b''nb''b''eb''b''nb''b''tb''b''аб'' #PWR07 ←
(b''цб''b''eb''b''пб''b''ьб'' 41).
```

>> b''Cb''b''ob''b''ob''b''бb''b''щб''b''eb''b''nb''b''иб''b''яб'' ERC: 4

Глава 10

Создание списка соединений

10.1 Обзор

Список соединений — это файл, содержащий описание электрических соединений между компонентами. В нём находятся:

- Перечень компонентов.
- Глобальные соединения между шинами

Существует несколько форматов списка соединений. Иногда, список компонентов и список эквипотенциальных связей хранятся в отдельных файлах. Список соединений является основой при использовании приложений проектирования схем, так как с его помощью осуществляется связь с другим программным обеспечением САПР:

- Приложения трассировки печатных плат.
- Симуляторы схем и переходных процессов.
- Компиляторы ПЛИС и других программируемых микросхем.

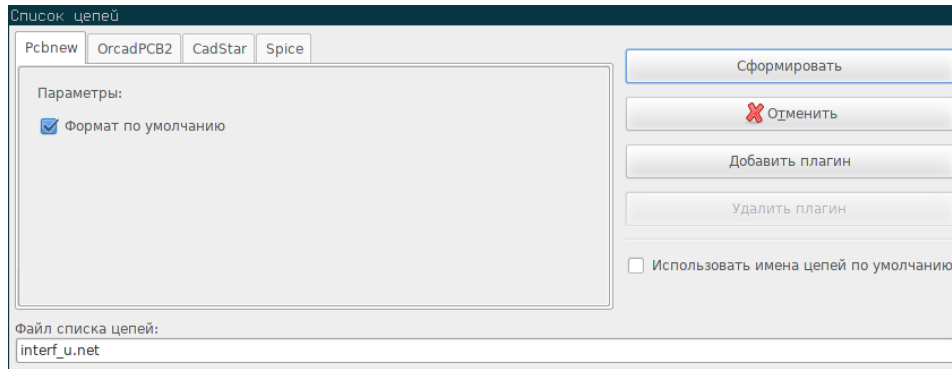
Eeschema поддерживает несколько форматов списков соединений.

- Формат PCBNEW (печатные платы).
- Формат ORCAD PCB2 (печатные платы).
- Формат CADSTAR (печатные платы).
- Формат Spice (различные симуляторы).

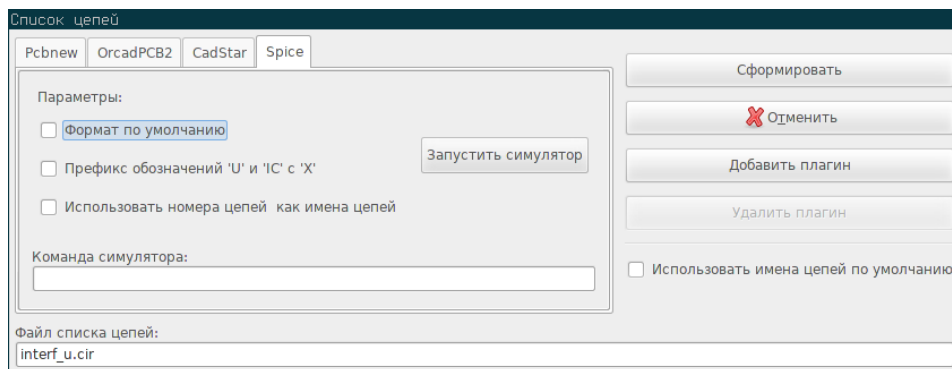
10.2 Форматы списков соединений

Нажмите кнопку , чтобы открыть диалоговое окно формирования списка соединений.

Выбран формат Pcbnew



Выбран формат Spice



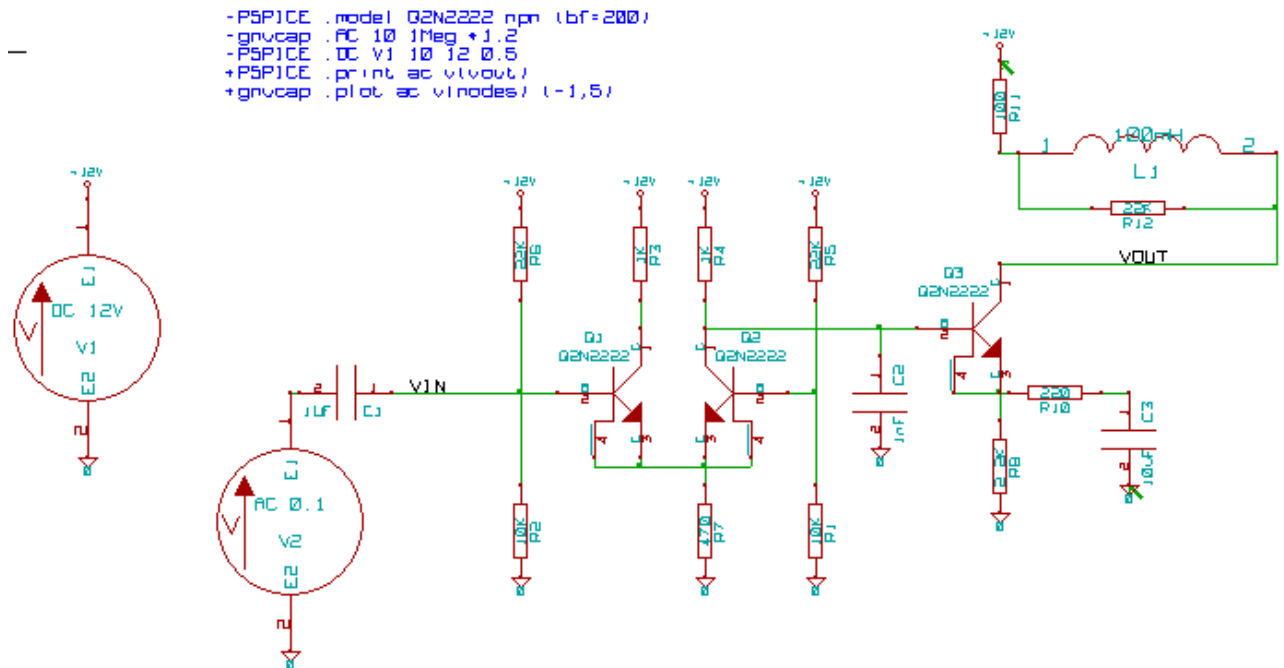
С помощью разных вкладок можно выбрать желаемый формат. В формате Spice можно формировать список соединений с упрощенными именами цепей, что делает SPICE-файлы более читабельными, или нумерованными, которые использовались в старых версиях формата Spice. При нажатии кнопки Сформировать будет запрошено имя файла для списка цепей.

Замечание

Формирование списка соединений для больших схем может занять несколько минут.

10.3 Примеры списков соединений

Ниже показан проект схемы, использующий библиотеку PSPICE:



Пример содержимого файла списка цепей в формате PCBNEW:

```

# Eeschema Netlist Version 1.0 generee 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
```

В формате PSPICE список цепей выглядит так:

```
* Eeschema Netlist Version 1.1 (Spice format) creation date: 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
```

10.4 Замечания о списках соединений

10.4.1 Предостережение об именовании элементов списка соединений

Множество приложений, которые используют списки соединений, не поддерживают пробелы в именах компонентов, выводов, цепей и прочих данных. Избегайте пробелов в метках, полях обозначений и значений компонентов или их выводов для обеспечения максимальной совместимости.

По той же причине, специальные символы, отличные от латинских букв и цифр, могут вызывать проблемы. Заметьте, это ограничение не связано с Eeschema, оно нужно для приложений, которые используют особый формат списка цепей.

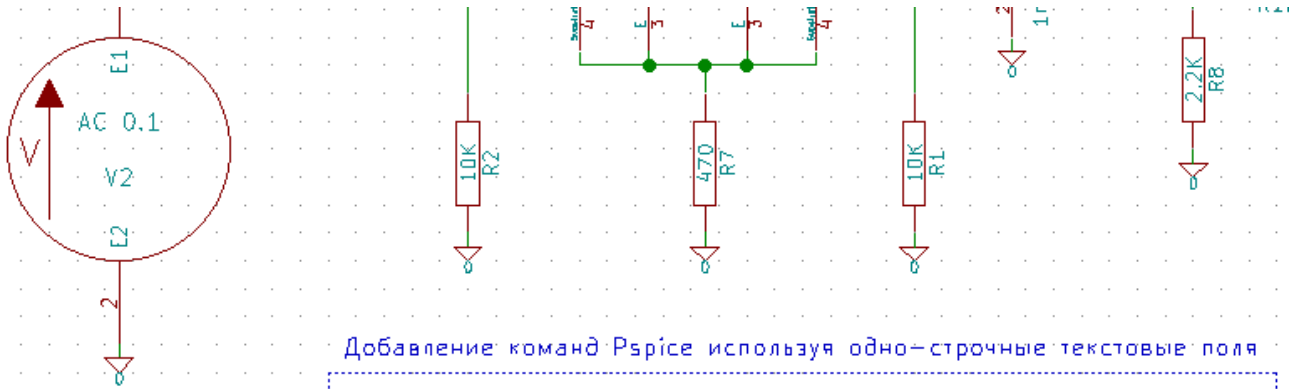
10.4.2 Списки соединений в формате PSPICE

Для симулятора Pspice нужно, помимо всего прочего, включать в список соединений ещё и некоторые командные строки (.PROBE, .AC, и т.д.).

Любая строка текста, расположенная на схеме и начинающаяся с ключевого слова **-pspice** или **-gnucap** будет вставлена (без этих ключевых слов) в начало списка цепей.

Любая строка текста, расположенная на схеме и начинающаяся с ключевого слова **+pspice** или **+gnucap** будет вставлена (без этих ключевых слов) в конец списка цепей.

Здесь представлен пример использования нескольких одно-строчных текстовых полей и одного много-строчного:



Добавление команд Pspice используя одно-строчные текстовые поля

```
-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)
```

Добавление команд Pspice используя много-строчное текстовое поле

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

К примеру, если набрать следующий тест (без использования меток!):

```
-PSPICE .PROBE
```

то строка `.PROBE` будет вставлена в список соединений.

В предыдущем примере три строки были вставлены в начало списка цепей, а две — в его конец, по этому методу.

При использовании много-строчного текста, ключевые слова **+pspice** или **+gnucap** нужно указывать только один раз:

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

при этом будет создано четыре строки:

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

Также, заметьте, что цепь GND должна называться 0 (ноль) для Pspice.

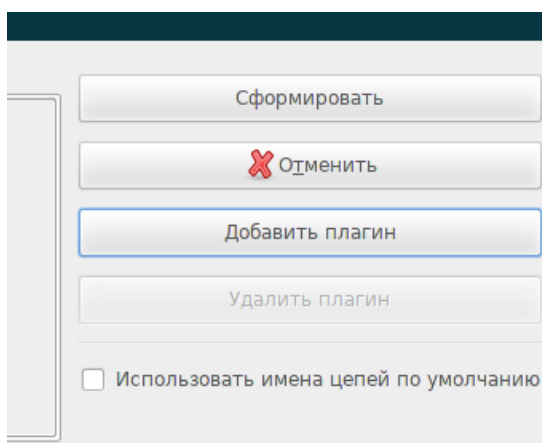
10.5 Прочие форматы

Для формирования списков соединений в других форматах можно использовать конвертеры в виде плагинов. Эти конвертеры автоматически запустятся из Eeschema. В разделе 14 даны некоторые пояснения и примеры таких конвертеров.

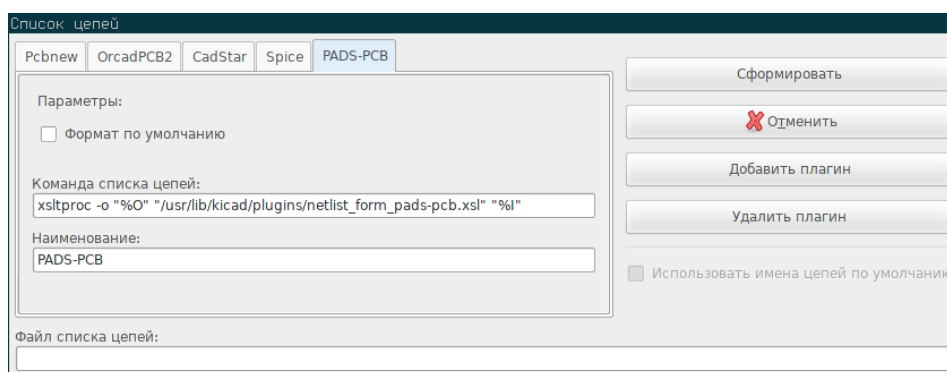
Конвертер — это текстовый файл в xls-формате, хотя могут использоваться и другие языки, такие как Python. Когда используется xls формат, приложение (xsltproc.exe или xsltproc) считывает временный файл, созданный Eeschema, и файл конвертера для создания выходного файла. В данном случае, файл конвертера (таблица стилей) очень мал его легко писать.

10.5.1 Подготовка диалогового окна

Добавить новый плагин для создания списка соединений можно с помощью кнопки "Добавить плагин."



Здесь показано окно настройки плагина PadsPcb:



Для настройки нужно указать:

- Наименование (например, имя формата списка цепей).
- Команду для запуска плагина.

При формировании списка соединений:

1. Eeschema создает временный файл *.tmp, например, test.tmp.
2. Eeschema запускает плагин, который считывает test.tmp и создает test.net.

10.5.2 Формат командной строки

Далее приведён пример, в котором используется `xsltproc.exe` в качестве инструмента преобразования файлов `.xsl` и `netlist_form_pads_pcb.xsl` в качестве конвертера:

```
f:/kicad/bin/xsltproc.exe -o %O.net f:/kicad/bin/plugins/netlist_form_pads_pcb.xsl %I
```

где:

<code>f:/kicad/bin/xsltproc.exe</code>	Приложение для преобразования файлов xsl.
<code>-o %O.net</code>	Выходной файл: %O определяет имя выходного файла.
<code>f:/kicad/bin/plugins/netlist_form_pads_pcb.xsl</code>	Имя файла конвертера (таблицы стилей в формате xsl).
<code>%I</code>	Будет заменено именем временного файла (*.tmp).

Для схемы с именем `test.sch` командная строка будет построена так:

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

10.5.3 Конвертер и таблица стилей (плагин)

Это очень простая часть программы, потому что её задача только преобразовать входной текстовый файл (временный текстовый файл) в другой текстовый файл. Более того, из временного текстового файла можно создать перечень элементов.

При использовании `xsltproc` в качестве инструмента преобразования нужно создать только таблицу стилей.

10.5.4 Формат временного файла списка соединений

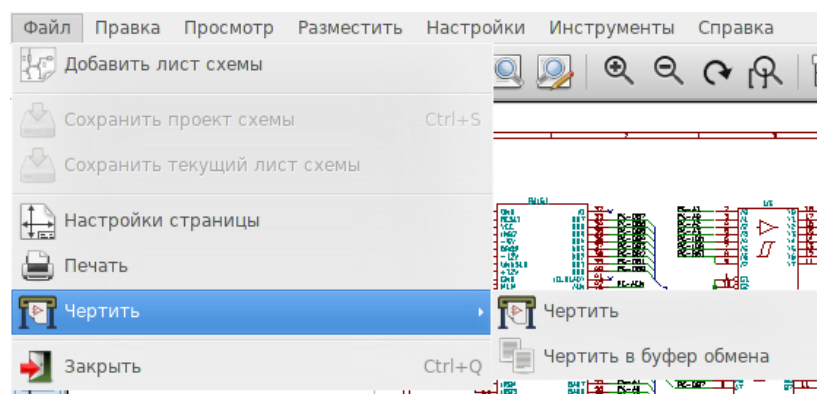
Смотрите раздел 14, чтобы получить больше подробностей о `xsltproc`, формате временного файла и некоторых примеров таблиц стилей для конвертеров.

Глава 11

Черчение и печать

Введение

Доступ к обоим командам, печати и черчения, можно получить через меню "Файл".



Поддерживаются следующие выходные форматы: Postscript, PDF, SVG, DXF и HPGL. Также, можно напрямую печатать на принтере.

11.1 Основные команды печати

Чертить текущий лист

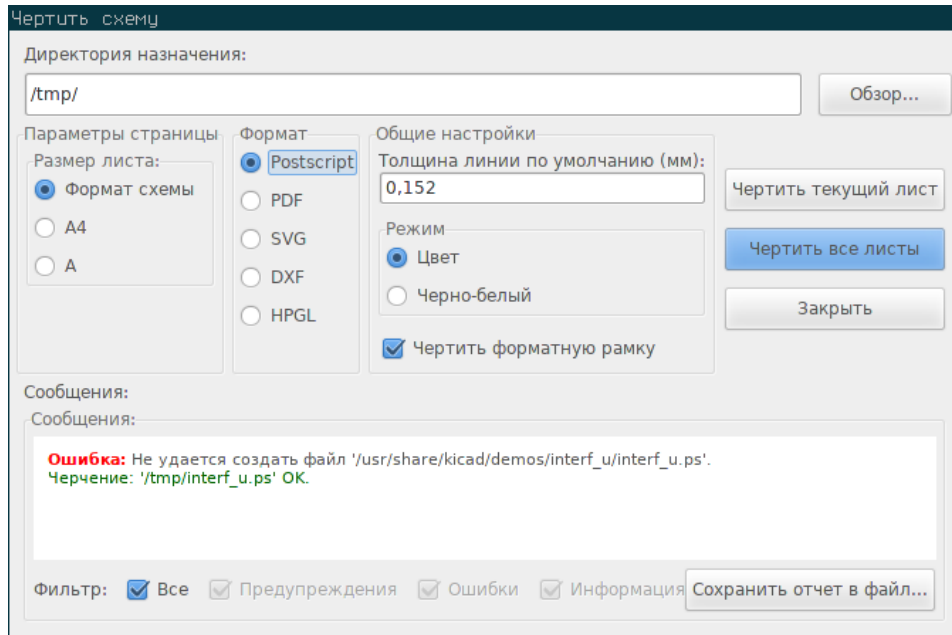
создаст один файл только для текущего листа.

Чертить все листы

позволяет начертить всю иерархию (по одному файлу для каждого листа).

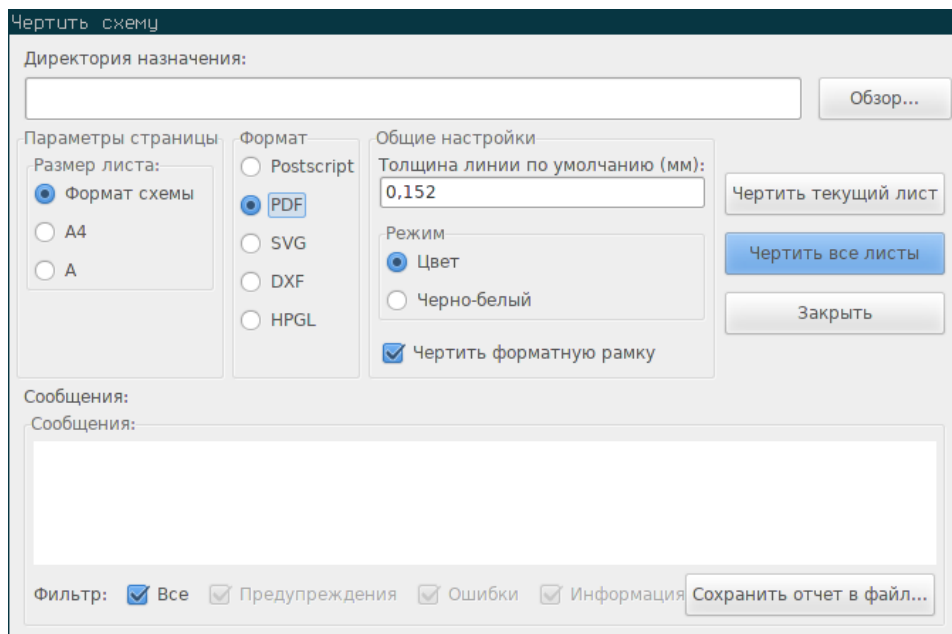
11.2 Чертить в формате Postscript

Этот параметр позволяет создавать PostScript файлы.



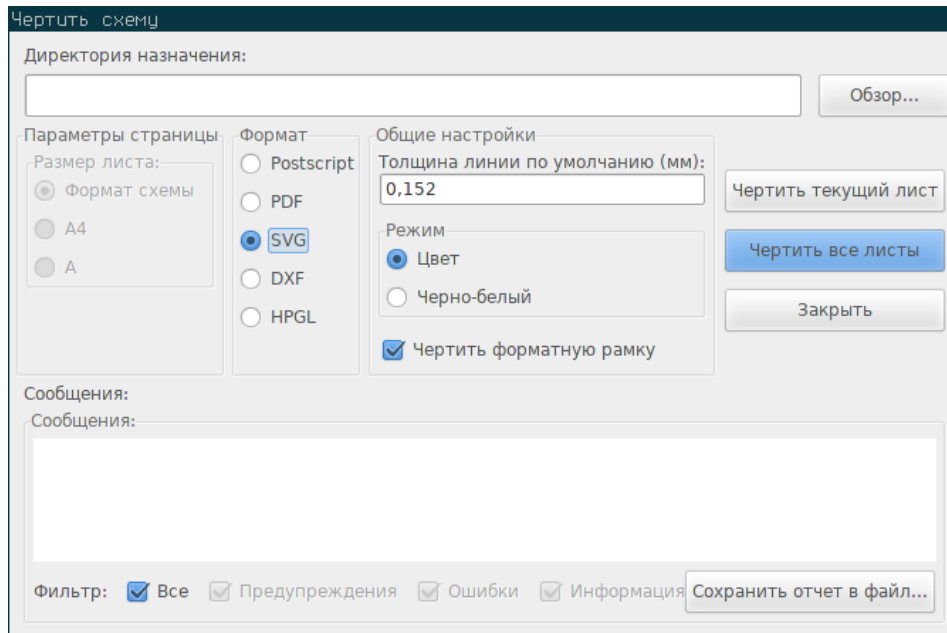
В качестве имени файла используется имя листа с расширением .ps. Можно отключить опцию "Чертить форматную рамку". Это будет полезно при последующем преобразовании в Encapsulated PostScript (формат .eps) для передачи полученных чертежей в текстовые редакторы. Поле сообщений содержит список созданных файлов.

11.3 Чертить в формате PDF



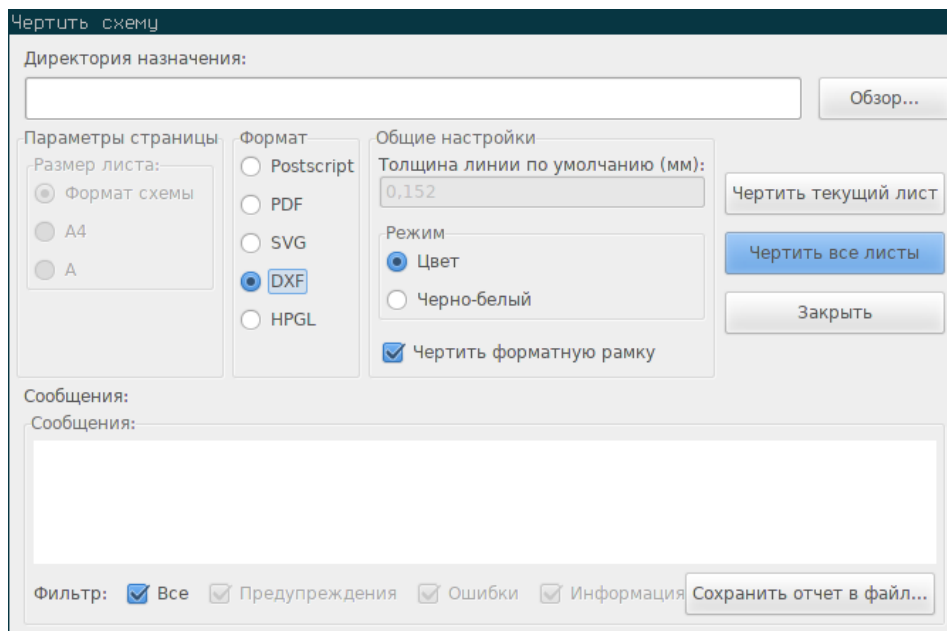
Позволяет создавать файлы чертежей используя формат PDF. Имя файла образуется из имени листа с добавлением расширения .pdf.

11.4 Чертить в формате SVG



Позволяет создавать файлы чертежей используя формат SVG. Имя файла образуется из имени листа с добавлением расширения .svg.

11.5 Чертить в формате DXF



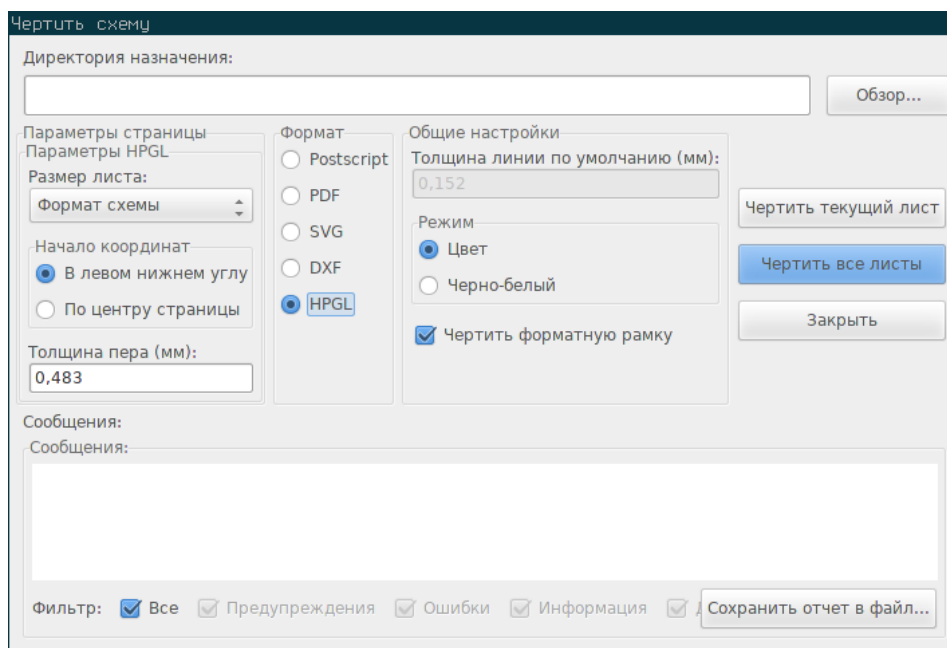
Позволяет создавать файлы чертежей используя формат DXF. Имя файла образуется из имени листа с добавлением расширения .dxf.

11.6 Чертить в формате HPGL

Позволяет создавать файлы чертежей используя формат HPGL. В этом формате можно задать:

- Размер листа.
- Начало координат.
- Толщину пера (в мм).

Диалоговое окно черчения выглядит как показано далее:



Имя файла образуется из имени листа с добавлением расширения .plt.

11.6.1 Установка размера листа

По умолчанию, выбран формат схемы. При этом, будет использован размер листа, установленный в настройках страницы, и масштаб, равный 1. Если выбран другой размер листа (A4 при размере A0 или A при размере E), то масштаб будет подобран так, чтобы заполнить страницу.

11.6.2 Настройка смещения

Для всех стандартных размеров, можно подстроить смещение центра чертежа так точно, как только это возможно. Дело в том, что плоттеры могут чертить или относительно центра, или относительно нижнего левого угла листа, и необходимо задать верное смещение для правильной постройки чертежа.

Другими словами:


- Для плоттеров с опорной точкой в центре листа — смещение должно быть отрицательным и равным половине размера листа.

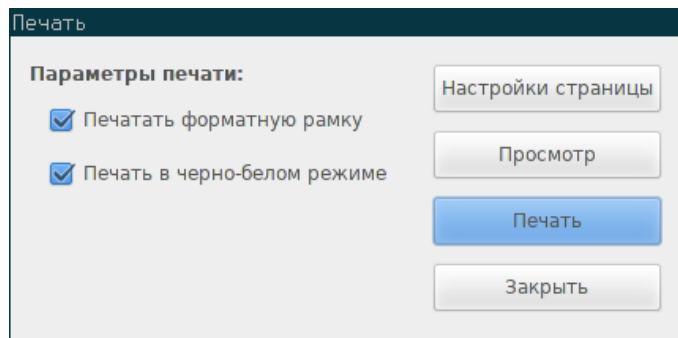
- Для плоттеров с опорной точкой в нижнем левом углу — смещение должно быть равным 0.

Для установки смещения:

- Выберите размер листа.
- Установите смещение по координатам X и Y.
- Подтвердите установленное смещение.

11.7 Печать на бумаге

Команда, доступная через кнопку , позволяет просматривать и создавать файлы печати для обычного принтера.



Параметр "Печатать форматную рамку" разрешает или запрещает печать форматной рамки и основной надписи.

Параметр "Печать в черно-белом режиме" переводит печать в монохромный режим. Этот параметр, в основном, нужен при использовании чёрно-белых лазерных принтеров, так как цвета печатаются в полутонах, из-за чего часто схемы становятся не читаемыми.

Глава 12

Просмотр библиотек компонентов

12.1 Общая информация о библиотеках компонентов

Компонент — это элемент схемы, который содержит условное графическое обозначение (УГО), электрические соединения и поля, определяющие компонент. Компоненты, используемые в схеме, хранятся в библиотеках. Eeschema содержит инструмент для создания библиотек компонентов, который позволяет создавать новые библиотеки, добавлять, удалять и перемещать компоненты между библиотеками, экспортировать их в файлы и импортировать обратно. Инструмент редактирования библиотек предоставляет простой путь к управлению файлами библиотек компонентов.

12.2 Обзор библиотеки компонентов

Библиотека компонентов состоит из одного или нескольких компонентов. Обычно, компоненты объединяют по функциональному назначению, типу, и/или производителю.

Компонент состоит из:

- Графических элементов (линий, окружностей, дуг, текста и пр.), определяющих его графическое представление.
- Выводов, которые имеют как графические свойства (линия, динамический, инверсный, тактовый по заднему фронту и пр.) и электрические свойства (вход, выход, двунаправленный и т.д.), которые используются для проверки электрических правил проектирования (ERC).
- Полей, таких как обозначение, значение, назначенное посадочное место для проекта печатной платы и другие.
- Псевдонимов, используемых для связывания подобных компонентов, таких как 7400 и его аналогов 74LS00, 74HC00 и 7437. Все эти псевдонимы ссылаются к одному и тому же компоненту.

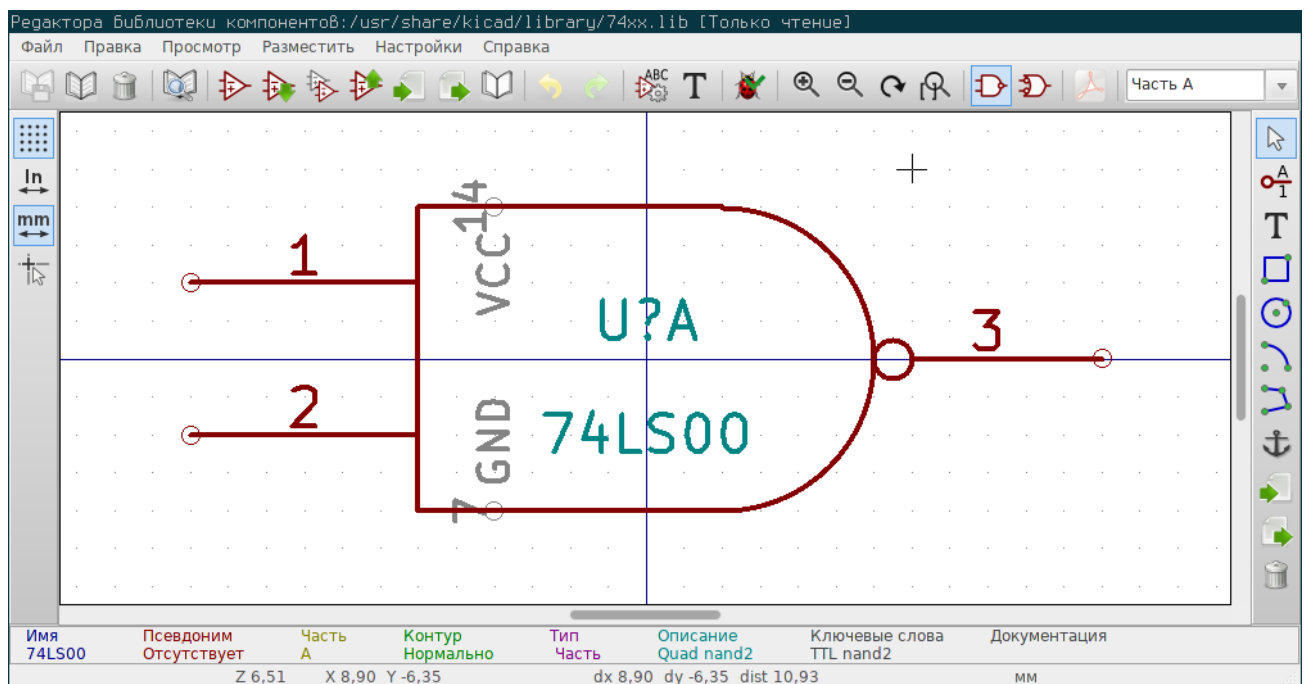
Для создания правильного компонента необходимо:

- Определить, состоит ли компонент из нескольких частей.
 - Определить, имеет ли компонент альтернативное начертание, также известное как дополнительное обозначение по де Моргану.
-

- Начертить его графическое обозначение используя линии, прямоугольники, окружности, полигоны и текст.
- Добавить выводы, внимательно назначая каждому графическое представление, имя, номер и электрический тип (вход, выход, трехстабильный, вывод питания и т.п.).
- Добавить псевдонимы, если другие компоненты имеют такое же обозначение и расположение выводов, или удалить их, если компонент был создан на основе другого.
- Добавить дополнительные поля, такие как имя посадочного места, используемого при проектировании печатной платы, и/или установить их видимость.
- Документировать компонент, путём заполнения строки описания, указания ссылки на справочную документацию и пр.
- Сохранить его в спроектированной библиотеке.

12.3 Обзор редактора библиотеки компонентов

Основное окно редактора библиотек компонентов показано ниже. Оно состоит из троих панелей инструментов для быстрого доступа к основным командам и поля для просмотра и редактирования. На панелях инструментов расположены не все доступные команды, остальные можно найти в меню.



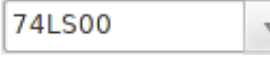




12.3.1 Основная панель инструментов

Основная панель инструментов, как обычно, расположена в верхней части основного окна и состоит из команд управления библиотекой, отмены/повторения действий, масштабирования и вызова диалогов настройки компонента, как показано ниже.













	Сохранить выбранную в данный момент библиотеку. Кнопка не активна, если не выбрана ни одна из библиотек или не было сделано никаких изменений в текущей библиотеке.
	Выбрать библиотеку для редактирования.
	Удалить компонент из текущей библиотеки или другой, определённой в проекте, если ни одна из библиотек не выбрана.
	Открыть просмотрщик компонентов для выбора библиотеки или компонента для дальнейшего редактирования.
	Создать новый компонент.
	Загрузить компонент из текущей библиотеки для редактирования.
	Создать новый компонент из выбранного в данный момент.
	Сохранить изменения текущего компонента в памяти. Файл библиотеки остается без изменений.
	Импортировать один компонент из файла.
	Экспортировать текущий компонент в файл.
	Создать новую библиотеку, содержащую текущий компонент. Примечание: новая библиотека не добавляется в проект автоматически.
	Отменить последнее изменение.
	Повторить предыдущее изменение.
	Редактировать свойства текущего компонента.
	Редактировать поля текущего компонента.
	Проверить текущий компонент на наличие ошибок проектирования.
	Увеличить масштаб.
	Уменьшить масштаб.
	Обновить изображение.
	Масштабировать так, чтобы изображение компонента полностью поместилось на экране.
	Выбрать нормальное обозначение компонента. Эта кнопка недоступна, если текущий компонент не имеет дополнительных обозначений (по де Моргану).
	Выбрать дополнительное обозначение компонента. Эта кнопка недоступна, если текущий компонент не имеет дополнительных обозначений (по де Моргану).

	Показать документацию для текущего компонента. Кнопка не активна, если файл документации не указан в свойствах компонента.
	Выбор части для отображения. Выпадающий список не активен, если текущий компонент не состоит из нескольких частей.
	Выбор псевдонима. Выпадающий список не активен, если текущий компонент не имеет псевдонимов.
	Редактировать выводы: редактирование формы и положения отличающихся выводов в компонентах с несколькими частями и/или дополнительными обозначениями.
	Показать таблицу выводов.

12.3.2 Панель инструментов чертёжных элементов





Вертикальная панель инструментов, расположенная с правой стороны основного окна, позволяет размещать все необходимые элементы обозначения компонента. В таблице ниже приведено описание каждой кнопки.

	Инструмент выделения. Щелчок правой кнопки мыши на выделенном элементе открывает контекстное меню для объекта под курсором. При нажатии левой кнопки мыши, на информационной панели, внизу основного окна, будут показаны свойства объекта, находящегося под курсором. Двойной щелчок левой кнопки мыши откроет диалоговое окно свойств объекта, что находится под курсором.
	Добавить вывод. Щелчок левой кнопки мыши для добавления нового вывода.
	Добавить текст. Щелчок левой кнопки мыши для добавления нового текстового элемента.
	Добавить прямоугольник. Щелчок левой кнопки мыши для расположения первого угла прямоугольника. Ещё один щелчок для расположения второго, противоположного, угла прямоугольника.
	Добавить окружность. Щелчок левой кнопки мыши для начала черчения новой окружности от центра. Ещё один щелчок для установки радиуса окружности.
	Добавление дуги. Щелчок левой кнопки мыши для начала черчения новой дуги, начиная из центра. Ещё один щелчок для установки первой конечной точки. Последний щелчок для установки второй конечной точки.
	Добавление полигона. Щелчок левой кнопки мыши для начала черчения полигональной линии для текущего компонента. Щелчок левой кнопки мыши для добавления следующего отрезка полигональной линии. Двойной щелчок левой кнопки мыши для завершения полигона.
	Точка привязки компонента. Щелчок левой кнопки мыши для установки точки привязки компонента.
	Импорт компонента из файла.


	Инструмент удаления. Щелчок левой кнопки мыши для удаления объекта из текущего компонента.
---	--

12.3.3 Панель инструментов параметров



Вертикальная панель инструментов, расположенная с левой стороны основного окна, позволяет установить некоторые параметры редактирования. В таблице ниже приведена таблица с описанием каждой кнопки.

	Переключить видимость сетки (вкл. или выкл.)
	Установить в качестве единиц измерения — дюймы.
	Установить в качестве единиц измерения — миллиметры.
	Переключить режим отображения курсора.

12.4 Выбор и управление библиотекой

Выбрать текущую библиотеку можно с помощью кнопки  , которая показывает все доступные библиотеки и позволяет выбрать одну из них. Когда компонент загружен или сохранён, он будет помещён в эту библиотеку. В качестве имени компонента в библиотеке принимается содержимое его поля "Значение".


Замечание

- Сначала нужно загрузить библиотеку в Eeschema, чтобы потом получить доступ к её содержимому.
- Содержимое текущей библиотеки можно сохранить после внесённых изменений с помощью кнопки  из главной панели инструментов.
- Компонент можно удалить из библиотеки, нажав на кнопку  .

12.4.1 Выбор и сохранение компонента

При редактировании компонента, изменения, на самом деле, применяются не к компоненту из библиотеки, а к копии, расположенной в памяти компьютера. Любые операции редактирования можно легко отменить. Компонент можно загрузить из локальной библиотеки или из существующего компонента.

12.4.1.1 Выбор компонента

При нажатии на кнопку  из основной панели инструментов, отобразится перечень доступных компонентов, которые можно выбрать и загрузить из текущей библиотеки.

Замечание

Если выбран псевдоним какого-нибудь компонента, то в заголовке окна будет показано имя оригинального компонента, вместо псевдонима. Список псевдонимов всегда загружается вместе с компонентом и его можно редактировать. Можно создать новый компонент, выбрав один из псевдонимов текущего компонента из



. Первый элемент списка псевдонимов является оригинальным компонентом.


Замечание


Также, нажатие кнопки  позволяет загрузить компонент, который ранее был сохранён с помощью кнопки




12.4.1.2 Сохранение компонента

После применённых изменений, компонент можно сохранить в текущей библиотеке, новой библиотеке или экспортировать в файл резервной копии.

Для сохранения изменённого компонента в текущей библиотеке, нажмите . Пожалуйста заметьте, что команда обновления лишь сохранит изменения в выделенной памяти. Об этом нужно помнить до сохранения библиотеки.


Чтобы непосредственно сохранить изменения компонента в файл библиотеки, нажмите кнопку , которая перезапишет существующий файл библиотеки новым содержимым со всеми изменениями.

Если нужно создать новую библиотеку, содержащую текущий компонент, нажмите . Будет выдан запрос на ввод имени новой библиотеки.

Замечание

Новые библиотеки не добавляются в текущий проект автоматически.





Нужно добавлять все новые, необходимые для схемы, библиотеки в таблицу библиотек проекта с помощью [соответствующего диалогового окна](#) в Eeschema.

Нажмите , чтобы создать файл, содержащий только текущий компонент. Этот файл будет стандартной библиотекой, которая содержит всего один компонент. Его можно использовать для импорта компонента в другую библиотеку. Таким образом, команда создания новой библиотеки и команда экспорта, в основном, одинаковы.

12.4.1.3 Перемещение компонента в другую библиотеку

Можно, достаточно просто, скопировать компонент из одной библиотеки в другую используя следующие команды:

- Выбрать исходную библиотеку, нажав кнопку .
-


- Загрузить компонент, который нужно переместить, нажав кнопку . Этот компонент будет показан в области редактирования.
- Выбрать конечную библиотеку, нажав кнопку .
- Сохранить текущий компонент в выделенной памяти новой библиотеки, нажав кнопку .
- Сохранить компонент в файл библиотеки, нажав кнопку .

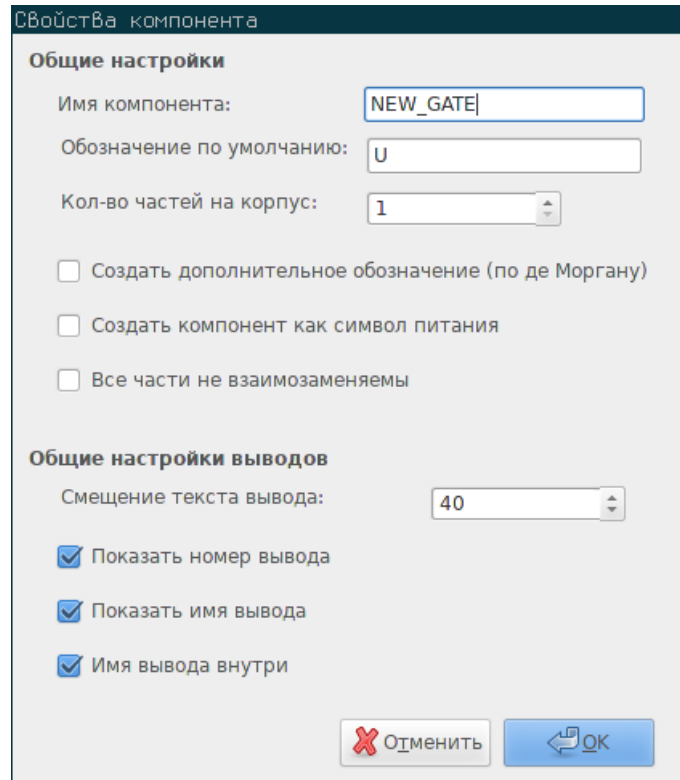
12.4.1.4 Отмена изменений компонента

При редактировании компонента, изменения выполняются только над рабочей копией настоящего компонента библиотеки. Это значит, что пока изменения не сохранены в памяти, компонент можно перезагрузить заново, отменив, тем самым, все сделанные изменения. Если изменения уже записаны в память, но не записаны в файл библиотеки, то всегда можно выйти из редактора и начать заново. Eeschema отменит все изменения.

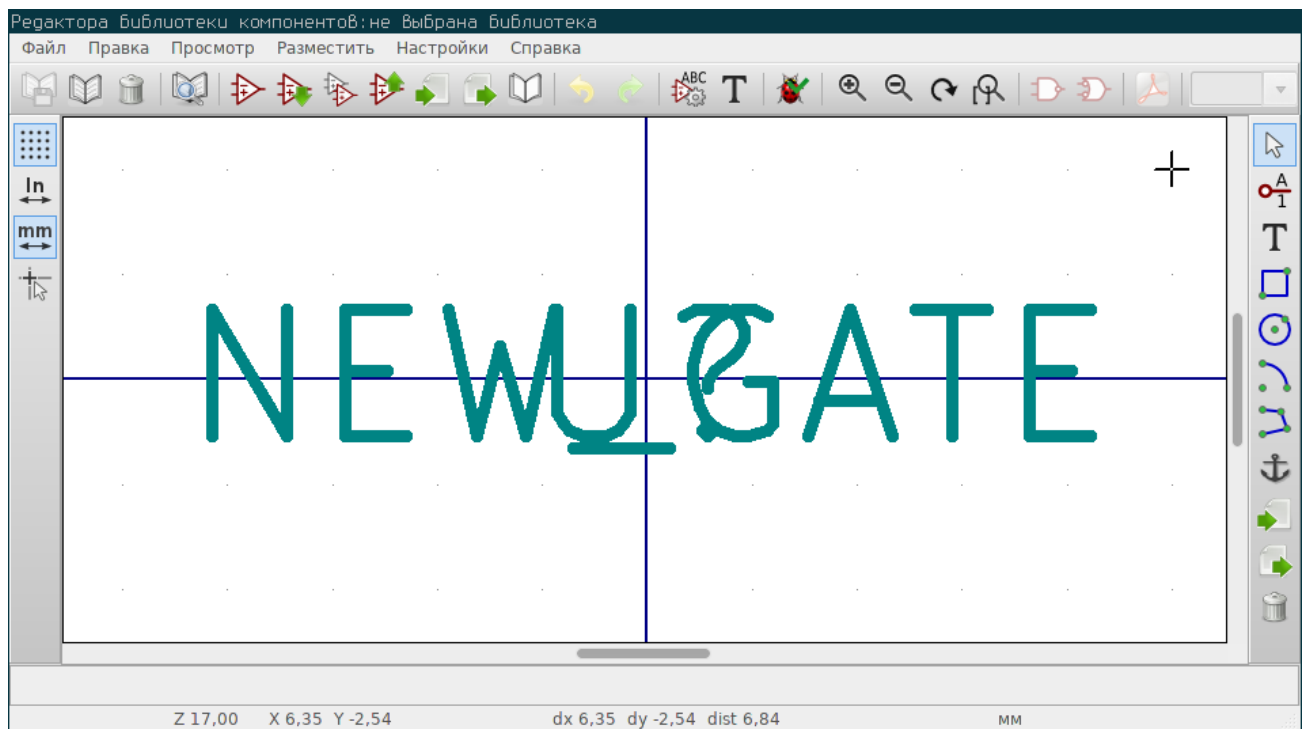
12.5 Создание компонентов библиотеки

12.5.1 Создание нового компонента

Новый компонент можно создать, нажав кнопку . Будет запрошено имя компонента (это имя будет использовано в качестве значения его поля "Значение" в редакторе схем), префикс обозначения (DA, VT, R, ...), количество частей (например, 7400 состоит из четырёх частей) и, если нужно, дополнительное начертание (также известное как обозначение по де Моргану). Если префикс обозначения не указан, будет использовано значение по умолчанию "U". Все эти параметры можно позже изменить, но, все же, лучше сразу задавать верные значения.








Будет создан новый компонент с указанными выше параметрами и представлен в редакторе, как показано ниже.




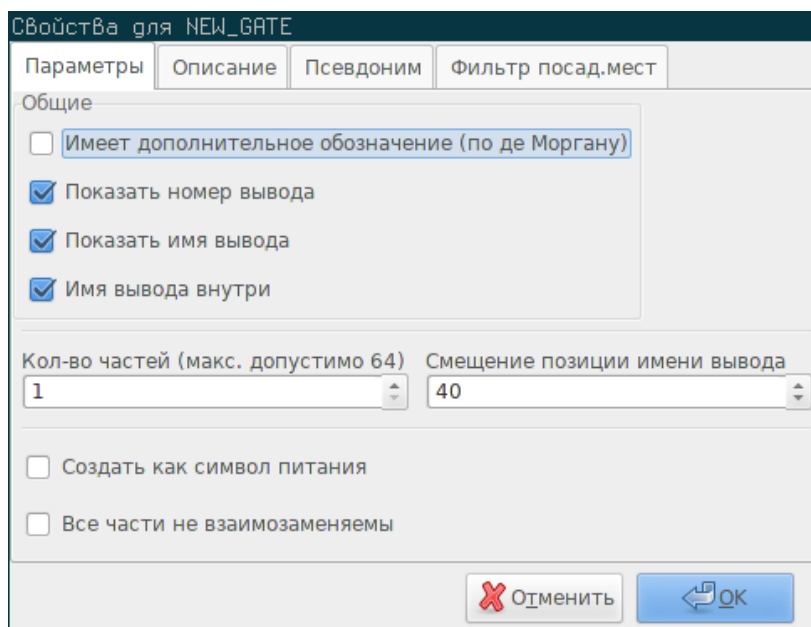
12.5.2 Создание компонента на основе другого

Часто компонент, который нужно сделать, похож на один из уже существующих в библиотеке. В этом случае легко загрузить и поправить существующий компонент.

- Загрузите компонент, который нужно взять за основу.
- Нажмите кнопку  или изменив его имя, используя щелчок правой кнопки мыши на поле значения и измените его текст. При нажатии кнопки дублирования, будет запрошено новое имя для компонента.
- Если компонент имеет псевдонимы, будет показано сообщение с предложением удалить их из нового компонента, чтобы устранить конфликт в библиотеке. Если отказаться от этого, создание нового компонента будет отменено. В библиотеке компонентов не может быть дубликатов имён или псевдонимов.
- Выполните все необходимые изменения в компоненте.
- Обновите новый компонента в текущей библиотеке, нажав кнопку  или сохраните его в новой библиотеке, нажав кнопку  или, если нужно, сохраните этот новый компонент в другой библиотеке, выбрав для этого другую библиотеку с помощью  и сохранив в неё новый компонент.
- Сохраните файл текущей библиотеки на диске с помощью кнопки .

12.5.3 Свойства компонента

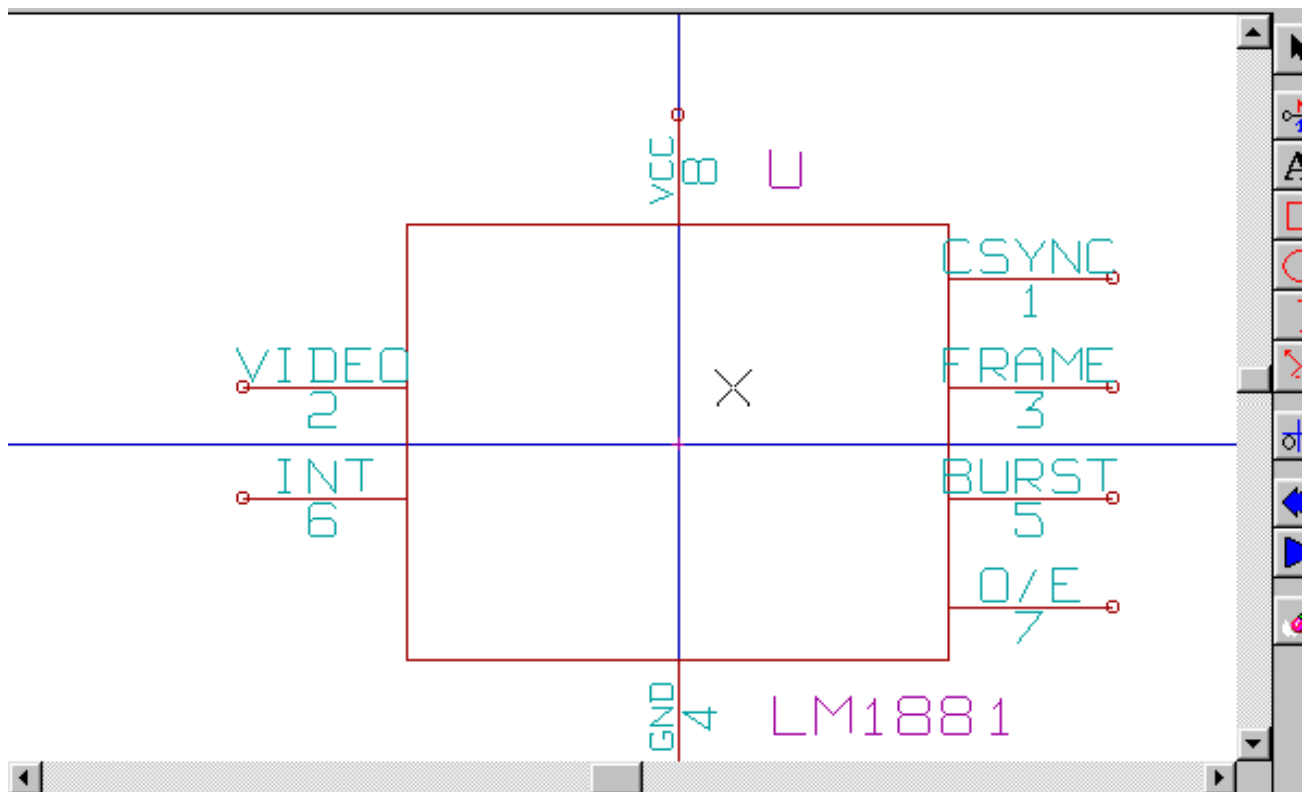
Свойства компонента должны быть внимательно заданы в процессе его создания, иначе, они будут унаследованы от скопированного компонента. Чтобы изменить свойства компонента, нажмите кнопку , будет показано диалоговое окно, как на рисунке ниже.



Очень важно, чтобы было правильно указано количество частей и наличие дополнительного начертания, потому что, редактирование или создание выводов затронет каждую из частей. Если количество частей изменено уже после создания и редактирования выводов, то нужно дополнительно обработать выводы и обозначение новых частей.


Параметры графического характера "Показать номер вывода" и "Показать имя вывода" определяют видимость номера и имени выводов. Эти надписи будут отображаться, если соответствующий параметр отмечен. Опция "Имя вывода внутри" определяет положение имени вывода по отношению к его обозначению. Эта надпись будет показана внутри контура компонента, если параметр отмечен. В таком случае, параметр "Смещение позиции имени вывода" задаёт смещение текста от края вывода. Приемлемое значение от 30 до 40 (в 1/1000 дюйма).


На примере ниже показан компонент с выключенным параметром "Имя вывода внутри". Обратите внимание на положение имени и номера выводов.

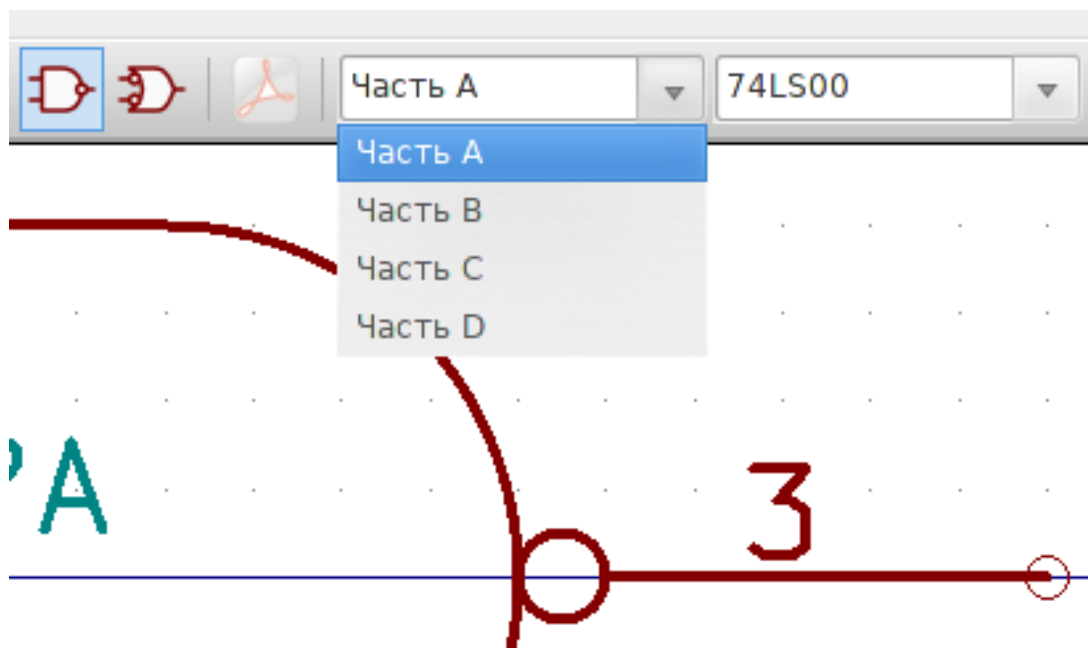


12.5.4 Компоненты с альтернативным начертанием

Если компонент имеет больше чем одно графическое обозначение, необходимо выбрать нужное для его редактирования.

Чтобы редактировать нормальное начертание, нажмите кнопку .

Чтобы редактировать дополнительное начертание, нажмите кнопку . Используйте , как показано ниже, чтобы выбрать нужную часть компонента для редактирования.



12.6 Графические элементы

С помощью графических элементов создают контур элемента, они не несут никакой электрической информации. Создаются с помощью следующих инструментов:

- Линии и полигоны, задаются начальной и конечной точками.
- Прямоугольники, задаются двумя противоположными углами.
- Окружности, задаются центром и радиусом.
- Дуги, задаются начальной и конечной точками дуги и её центром. Дуга может быть в пределах от 0° до 180° .

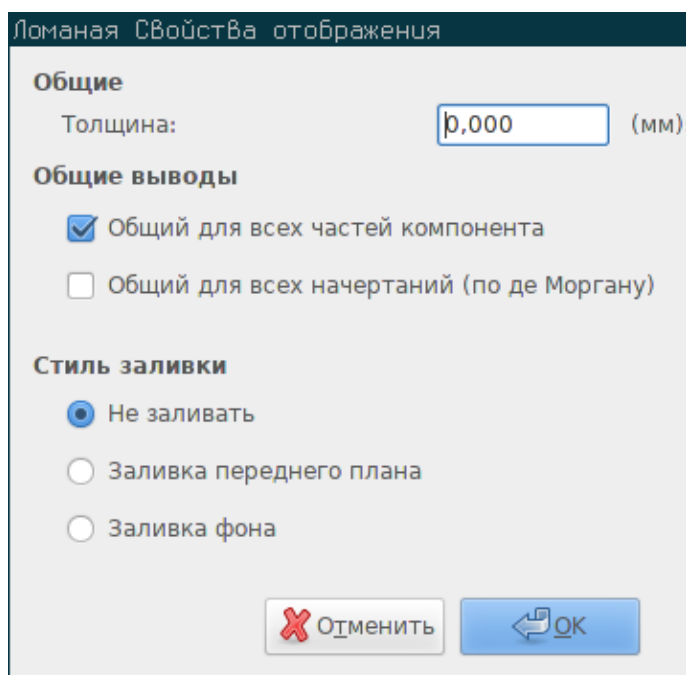
Вертикальная панель инструментов с правой стороны основного окна позволяет размещать все графические элементы, необходимые для построения условного графического обозначения компонента.

12.6.1 Графические элементы в частях и дополнительных начертаниях

Каждый графический элемент (линия, дуга, окружность, и пр.) можно определить как общий для всех частей и/или дополнительных начертаний или отдельный для данной части и/или начертания. К параметрам элемента можно легко получить доступ через щелчок правой кнопки мыши на нём для вызова контекстного меню. Ниже показано контекстное меню для линии.



Также, можно дважды щёлкнуть на элементе для изменения его параметров. Ниже приведён диалог настройки элемента полигона.



Свойства графического элемента:

- Толщина определяет толщину линии элемента в текущих единицах измерения.
- "Общий для всех частей компонента" определяет, принадлежит ли нарисованный графический элемент каждой части компонента, состоящего из нескольких частей, или только к текущей части.
- "Общий для всех начертаний (по де Моргану)" определяет, принадлежит ли нарисованный графический элемент каждому обозначению компонента, содержащего альтернативное начертание, или только текущему обозначению.
- Стилем заливки определяется каким образом будет закрашен графический элемент: без заливки, заливкой переднего плана, заливкой заднего фона.

12.6.2 Графический текст

Команда **T** позволяет создавать графический текст. Он всегда остаётся читаемым, даже когда компонент отражён. Обратите внимание, что графический текст — это не поле.

12.7 Компоненты из нескольких частей и с дополнительными начертаниями

Компонент может иметь два графических представления (обычное и дополнительное, часто называемое в честь "де Моргана") и/или более одной части (например, логические элементы). Некоторые компоненты могут иметь несколько частей и при этом отличаться друг от друга обозначением и расположением выводов.

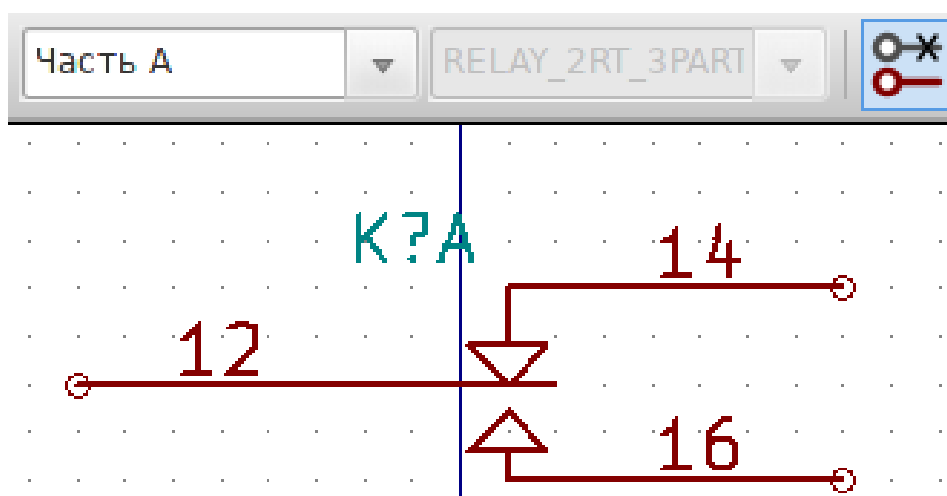
Рассмотрим реле с двумя контактами — оно может быть представлено как компонент с тремя разными частями: катушки, контакта 1 и контакта 2. Проектирование компонентов, состоящих из нескольких частей и/или начертаний, выполняется достаточно гибко. Вывод или часть обозначения может быть общим для всех частей или уникальным для текущей части, либо они могут быть общими для всех начертаний или уникальными только для текущего.

По умолчанию, выводы являются уникальными для каждого начертания и для каждой части, так как номера выводов отличаются. Если вывод должен быть общим для всех начертаний и частей, его нужно создать всего лишь раз, общим для всех частей и начертаний (обычно так поступают с выводами питания). Это касается и графических элементов обозначения и текста, которые также могут быть общими для всех частей (но, обычно, разными для каждого начертания).

12.7.1 Пример компонента из нескольких частей и с несколькими начертаниями:

Это пример реле, состоящего из трёх частей: контакта 1, контакта 2 и реле:

Параметр: не связанные выводы. Можно добавлять или редактировать выводы в каждой из частей отдельно без изменения остальных частей.



Все части не взаимозаменяемы, это нужно указать.

Свойства для RELAY_2RT_3PART

Параметры | Описание | Псевдоним | Фильтр посад.мест

Общие

Имеет дополнительное обозначение (по де Моргану)

Показать номер вывода

Показать имя вывода

Имя вывода внутри

Кол-во частей (макс. допустимо 64) Смещение позиции имени вывода

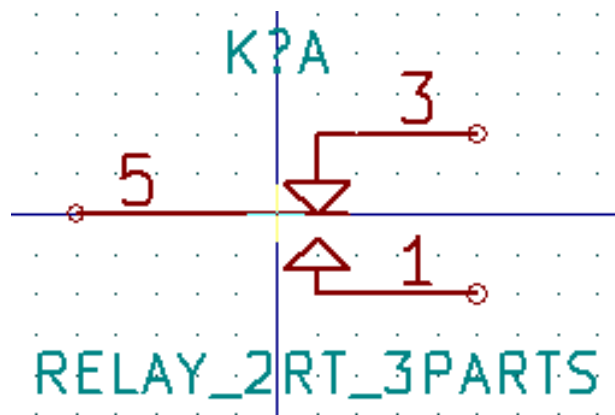
3 40

Создать как символ питания

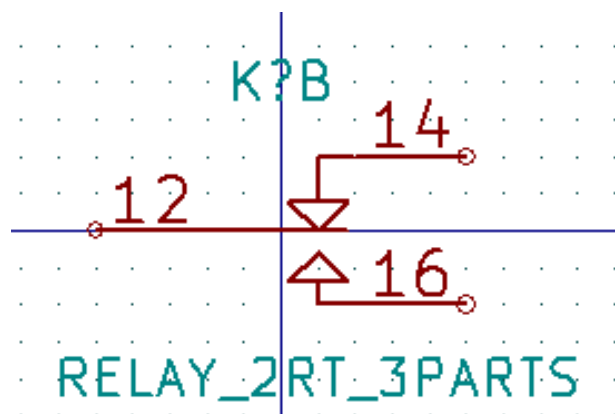
Все части не взаимозаменяемы

Отменить OK

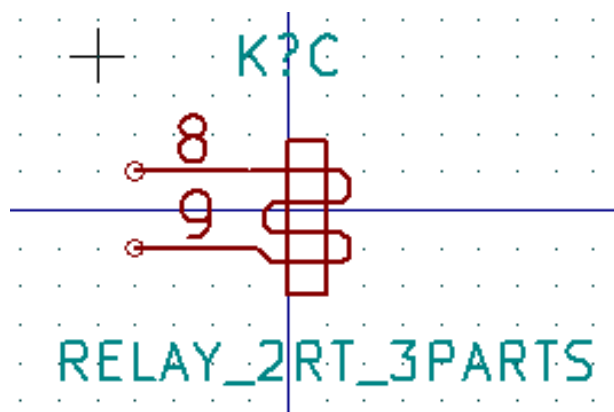
Часть 1



Часть 2



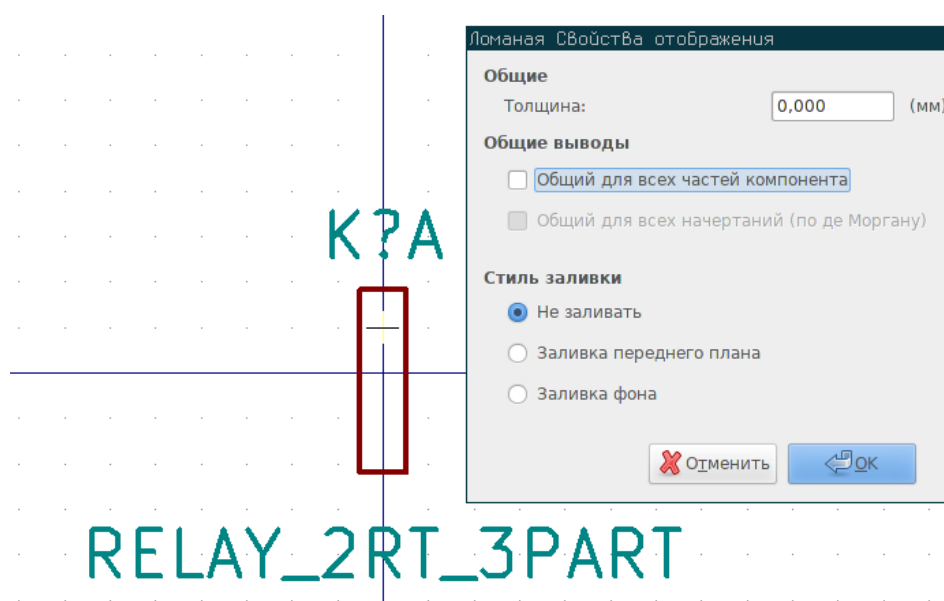
Часть 3




Она отличается обозначением и расположением выводов, следовательно, не может быть заменена ни частью 1, ни частью 2.

12.7.1.1 Элементы графического обозначения

Ниже показаны свойства графического контура компонента. Из примера с реле, показанного выше, три части отличаются графическим обозначением. Следовательно, каждая часть создавалась отдельно и у графических элементов должен быть отключен параметр "Общий для всех частей компонента".



12.8 Создание и редактирование выводов

Чтобы создать и добавить новый вывод нужно нажать кнопку . Редактирование всех свойств вывода выполняется с помощью двойного щелчка мыши по нему или через контекстное меню, используя правую кнопку мыши. Выводы нужно создавать внимательно, потому что любая ошибка скажется, в итоге, на печатной плате. Любой, уже добавленный, вывод можно отредактировать, удалить и/или переместить.

12.8.1 Общие сведения о выводах

Вывод объединяет в себе графическое представление, имя и "номер". "Номер" вывода может содержать до четырех букв и/или цифр. Чтобы использовать инструмент проверки электрических правил проектирования (ERC), нужно также корректно указать тип вывода (вход, выход, трехстабильный и т.д.). Если же тип указан не точно, результат проверки ERC в схеме будет не верным.

Важные замечания:

- Не используйте пробелы в именах и номерах выводов.
- Чтобы задать имя выводу с обозначением инверсии (подчёркнутым сверху), используйте символ ~ (тильда). Следующий символ ~ завершит подчёркивание. Например, слово ~FO~O будет отображено как FO O.
- Если имя вывода уменьшится до одного символа, то вывод будет считаться безымянным.
- Имена выводов, начинающиеся с #, зарезервированы для символов питания.
- "Номер" вывода состоит из 1-4 букв и/или цифр. 1,2,..9999 — допустимые значения. A1, B3, Anod, Wire и т.д. — тоже допустимые значения.
- Не допускается наличие выводов с одинаковыми "номераами" в одном компоненте.

12.8.2 Свойства выводов

Свойства вывода

Имя вывода:	<input type="text" value="C1"/>	Размер шрифта имени:	<input type="text" value="1,016"/>	мм
Номер вывода:	<input type="text" value="5"/>	Размер шрифта номера:	<input type="text" value="1,016"/>	мм
Ориентация:	<input type="text" value="Вправо"/>	Длина:	<input type="text" value="4,318"/>	мм
Электр.тип:	<input type="text" value="Пассивный"/>			
Графический стиль:	<input type="text" value="Линия"/>			

Общие выводы

Общий для всех частей компонента

Общий для всех начертаний (по де Моргану)

Свойства схемы

Видимый

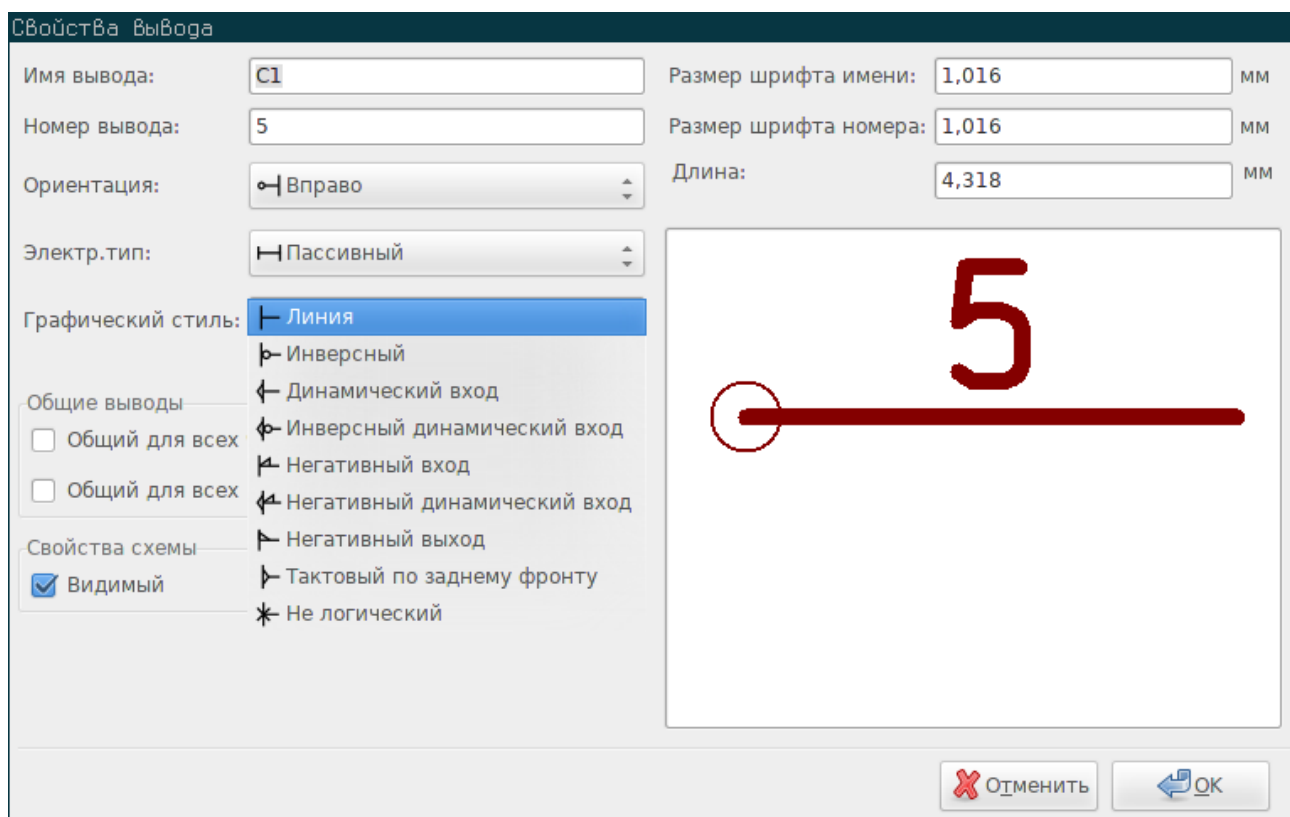
Отменить OK

Диалог свойств вывода позволяет редактировать все его параметры. Этот диалог автоматически отображается при создании вывода или при двойном щелчке на существующем выводе. Данный диалог позволяет изменять:

- Имя и размер текста имени.
- Номер и размер текста номера.
- Длину.
- Электрический и графический тип.
- Принадлежность к части или альтернативному начертанию.
- Видимость.

12.8.3 Графическое представление выводов

На изображении ниже показаны разные стили начертания вывода. Выбор внешнего вида вывода никак не отразится на его электрическом типе.



12.8.4 Электрический тип выводов

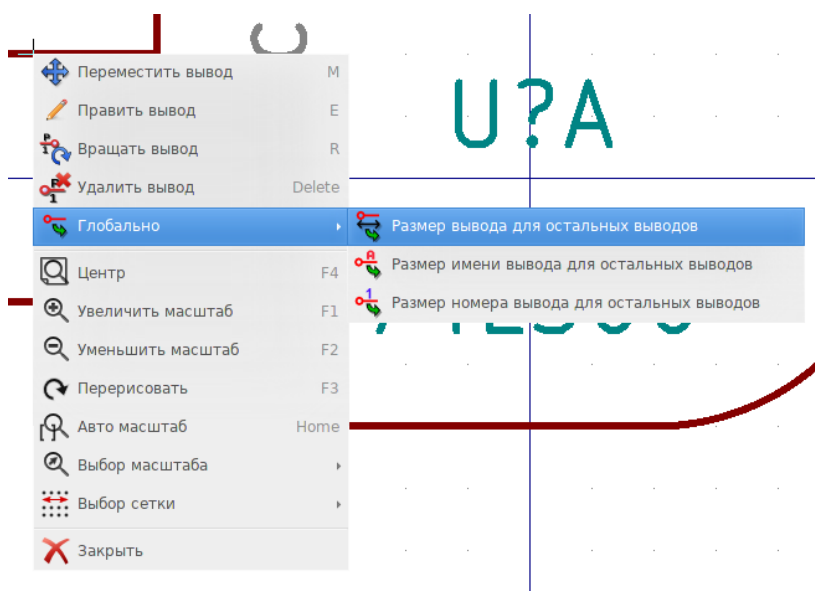
Правильное назначение электрического типа важно для инструмента проверки ERC в схеме. Электрические типы определены как:

- Двухнаправленный — указывает на двухнаправленный обмен данными между входом и выходом (шина данных микропроцессора, к примеру).
- Трехстабильный — выход с тремя состояниями.

- Пассивный — используется для обозначения выводов в пассивных компонентах, таких как резисторы, разъемы, и т.п.
- Не определено — может использоваться в случае, когда проверка ERC не нужна.
- Вход питания — используется для обозначения выводов питания компонентов. Выводы питания автоматически соединяются с остальными входами питания с такими же именами.
- Выход питания — используется для обозначения выходов стабилизаторов напряжения.
- Открытый эмиттер и открытый коллектор — можно использовать для обозначения выходов логических элементов.
- Не подсоединен — используется когда компонент содержит вывод, не имеющий внутреннего подключения (изолирован).


12.8.5 Глобальные свойства выводов

Можно изменять длину или размер текста имени и/или номера всех выводов сразу, используя пункты глобальных команд из контекстного меню вывода. Щёлкните левой кнопкой мыши по параметру, который хотите изменить и введите новое значение, которое будет применено ко всем выводам текущего компонента.



12.8.6 Установка выводов в компонентах с частями и альтернативными начертаниями

Компоненты, состоящие из нескольких частей и/или содержащие альтернативные начертания, имеют некоторые сложности при создании и редактировании выводов. Большинство выводов устанавливаются отдельно для каждой части (так как номер вывода уникален для каждой из них) и каждого альтернативного начертания (из-за различий в форме и положении). Создание и редактирование выводов может быть усложнённым для компонентов, которые одновременно состоят из нескольких частей и содержат альтернативное начертание. Редактор библиотек позволяет создавать выводы для всего одновременно. По умолчанию, изменения вывода применяются ко всем частям сложного компонента и обоих начертаний, если компонент содержит альтернативное начертание.


Единственное исключение при этом — это графический тип вывода и его имя. Эта зависимость была установлена, чтобы упростить создание и редактирование выводов в большинстве случаев. Данную зависимость можно отключить с помощью кнопки  из главной панели инструментов. Это позволит создавать выводы для каждой части и начертания отдельно.


Компонент может иметь два графических начертания (по де Моргану) и применяться к каждой из частей, как в случае с логическими элементами. Некоторые компоненты могут содержать части с разными графическими обозначениями и разными выводами. Подобно примеру с реле из [предыдущего раздела](#), которое представлено тремя частями: катушкой, переключающими контактами 1 и 2.

Управление компонентами из нескольких частей и с несколькими начертаниями довольно гибко. Вывод может быть общим или уникальным для разных частей. Также, вывод может быть общим для обоих начертаний или уникальными для каждого из них.

По умолчанию, выводы устанавливаются отдельно для каждой части и для каждого начертания, так как их номера для каждой из частей и внешний вид для каждого из начертаний различные. Если вывод общий для всех частей, его достаточно установить один раз, как в случае с выводами питания.

Возьмём для примера выходной вывод микросхемы 7400 — это четыре логических элемента ИЛИ-НЕ. Так как он состоит из четырех частей и двух графических начертаний, то нужно установить восемь уникальных выходных выводов для каждого. При редактировании компонента 7400, в редакторе библиотек будет показана часть А в нормальном начертании.

Чтобы изменить внешний вид альтернативного начертания, нужно сначала на него переключиться с помощью кнопки  из панели инструментов. Для редактирования номера выводов каждой из частей, нужно выбрать соответствующую часть

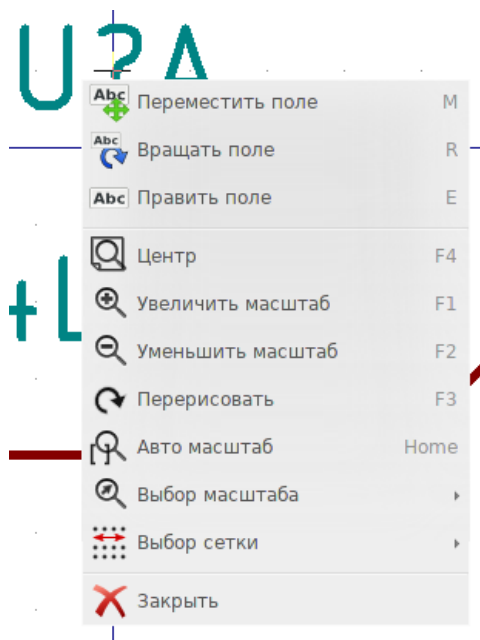
с помощью выпадающего меню .

12.9 Поля компонентов

Все компоненты библиотеки определяются по четырём стандартным полям. Обозначение, значение, посадочное место и файл документации — эти поля создаются всегда при создании или копировании компонента. Обязательными для заполнения являются только два поля — обозначение и значение. Для существующих полей можно вызывать контекстное меню с помощью правой кнопки мыши. Компоненты, содержащиеся в библиотеке, обычно, содержат эти четыре стандартных поля. Дополнительные поля, такие как производитель, каталожный номер, цена и пр. можно добавить в компонент библиотеки, но, обычно, это выполняют в редакторе схем, так как поля можно добавлять любому компоненту схемы.

12.9.1 Редактирование полей компонентов

Чтобы изменить существующие поле, нужно вызвать контекстное меню с помощью правой кнопки мыши, как показано на рисунке ниже.



Чтобы изменить пустое поле, добавить новое или удалить одно из них, используйте **T** из главной панели инструментов, чтобы открыть диалог свойств полей, который показан ниже.

Имя	Значение	Гориз. выравнивание	Верт. выравнивание
Обозначение	U	<input type="radio"/> Влево	<input type="radio"/> Вниз
Значение	74LS00	<input checked="" type="radio"/> Центр	<input checked="" type="radio"/> Центр
Посад.место		<input type="radio"/> Вправо	<input type="radio"/> Вверх
Документация			

Видимость	Стиль:
<input checked="" type="checkbox"/> Показать	<input checked="" type="radio"/> Нормально
<input type="checkbox"/> Повернуть	<input type="radio"/> Курсив
	<input type="radio"/> Полужирный
	<input type="radio"/> Полужирный курсив

Имя поля:

Значение поля:

Размер: мм

Позиция по X: мм

Позиция по Y: мм

Поля — это текстовые свойства, связанные с компонентом. Не путайте их с текстом, расположенном на графическом обозначении этого компонента.

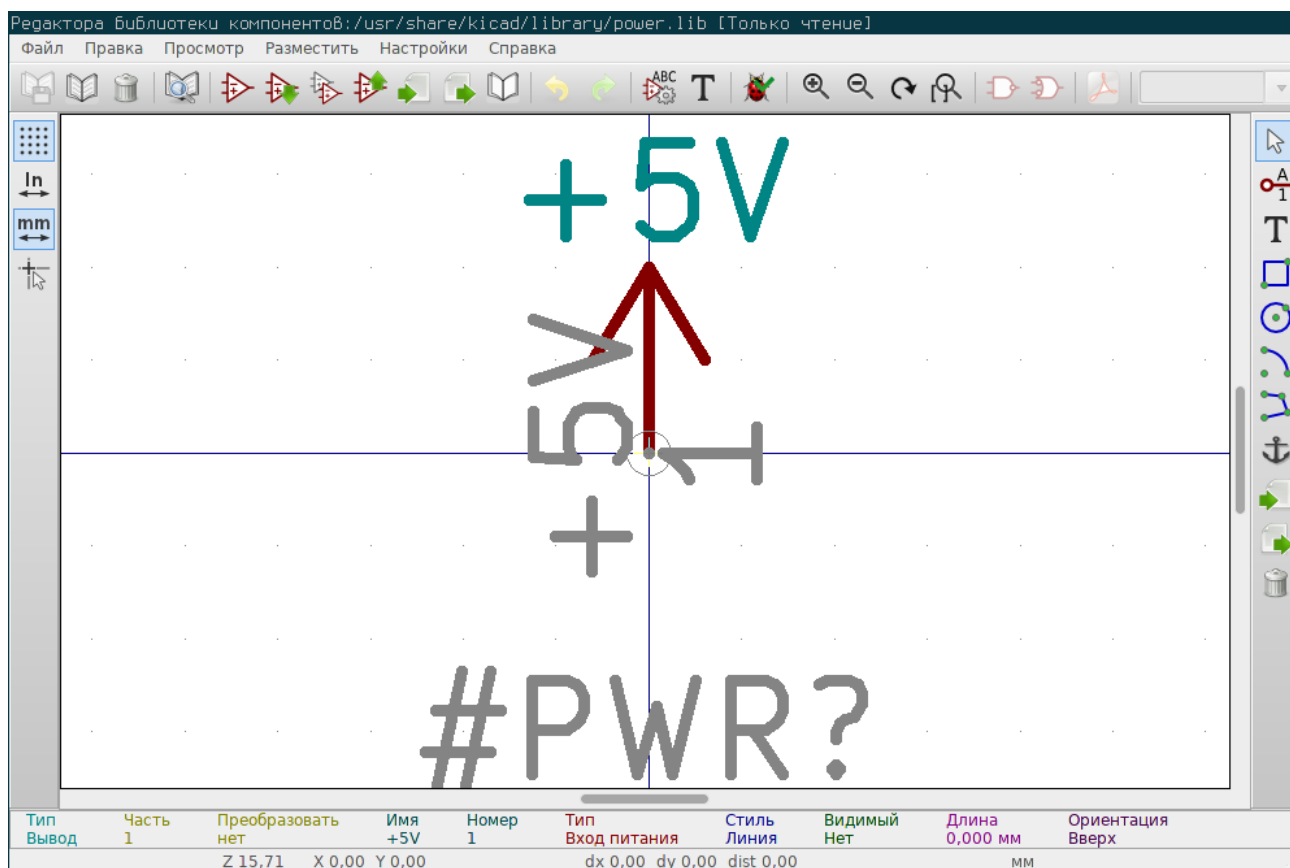
Важные замечания:

- Изменяя поле значения можно создавать новые компоненты, используя текущий в качестве шаблона для нового. При сохранении компонента в текущей библиотеке, в качестве имени будет использовано значение из поля значения.

- Диалог редактирования полей, показанный выше, должен использоваться при редактировании пустых или скрытых полей.
- Посадочное место указывается как абсолютный путь, используя формат LIBNAME:FPNAME, где LIBNAME — имя библиотеки посадочных мест из таблицы библиотек (см. раздел "Таблица библиотек посадочных мест" из руководства пользователя Pcbnew) и FPNAME — имя посадочного места из библиотеки LIBNAME.

12.10 Символы питания

Символы питания создаются также как и обычные компоненты. Желательно помещать их в специальную библиотеку, такую как power.lib. Символы питания состоят из графического обозначения и вывода "скрытый вход питания". Они обрабатываются также как и остальные компоненты в редакторе схем. Есть несколько существенных мер предосторожности. Ниже показан пример символа питания +5V.



Для создания символа питания применяйте следующие шаги:

- Добавьте вывод типа "Вход питания" с именем +5V (это важно, так как это имя будет использовано для соединения с цепью +5V), номером 1 (значение номера не важно), длиной 0 и графическим стилем "Линия".
- С помощью графических элементов изобразите внешний вид символа питания, как показано на рисунке.
- Установите точку привязки на выводе.
- Значение компонента — +5V.

- Обозначение компонента — $\#+5V$. Текст обозначения не имеет значения, за исключением первого символа, которым обязательно должен быть $\#$, чтобы указать, что компонент является символом питания. Условно, каждый компонент, в котором поле обозначения начинается с $\#$, не передаётся в список компонентов или список цепей и само обозначение устанавливается невидимым.

Более простой способ создания нового символа питания, используя другой как шаблон:

- Загрузить существующий символ питания.
 - Изменить имя вывода на имя нового символа питания.
 - Изменить поле значения на то же имя, что и у вывода, если хотите, чтобы оно было показано.
 - Сохраните новый компонент.
-

Глава 13

LibEdit - Редактор компонентов

13.1 Обзор

Компонент состоит из следующих элементов

- Графическое обозначение (графические линии, текст).
- Выводы.
- Поля или связанный текст, который используется генераторами списка цепей и перечня элементов.

Два поля должны быть заполнены: обозначение и значение. Название документации на компонент, имя назначенного посадочного места и прочие поля — свободны, их, в общем случае, можно оставить пустыми и заполнять при построении схемы.

Но с другой стороны, назначение документации для любого из компонентов облегчает изучение, использование и управление библиотеками. Документация состоит из:

- Строки комментария.
- Строки ключевых слов, таких как TTL CMOS NAND2, разделяемых пробелами.
- Имени приложенного файла (например, пример использования или pdf-файл).

Каталог по умолчанию для приложенных файлов:

```
kicad/share/library/doc
```

Или:

```
kicad/library/doc
```

В Linux:

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

```
/usr/share/kicad/library/doc
```

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

Ключевые слова позволяют выборочно находить компоненты, в соответствии с разными критериями выбора. Комментарии и ключевые слова отображаются в разных меню и, в частности, при выборе компонента в библиотеке.

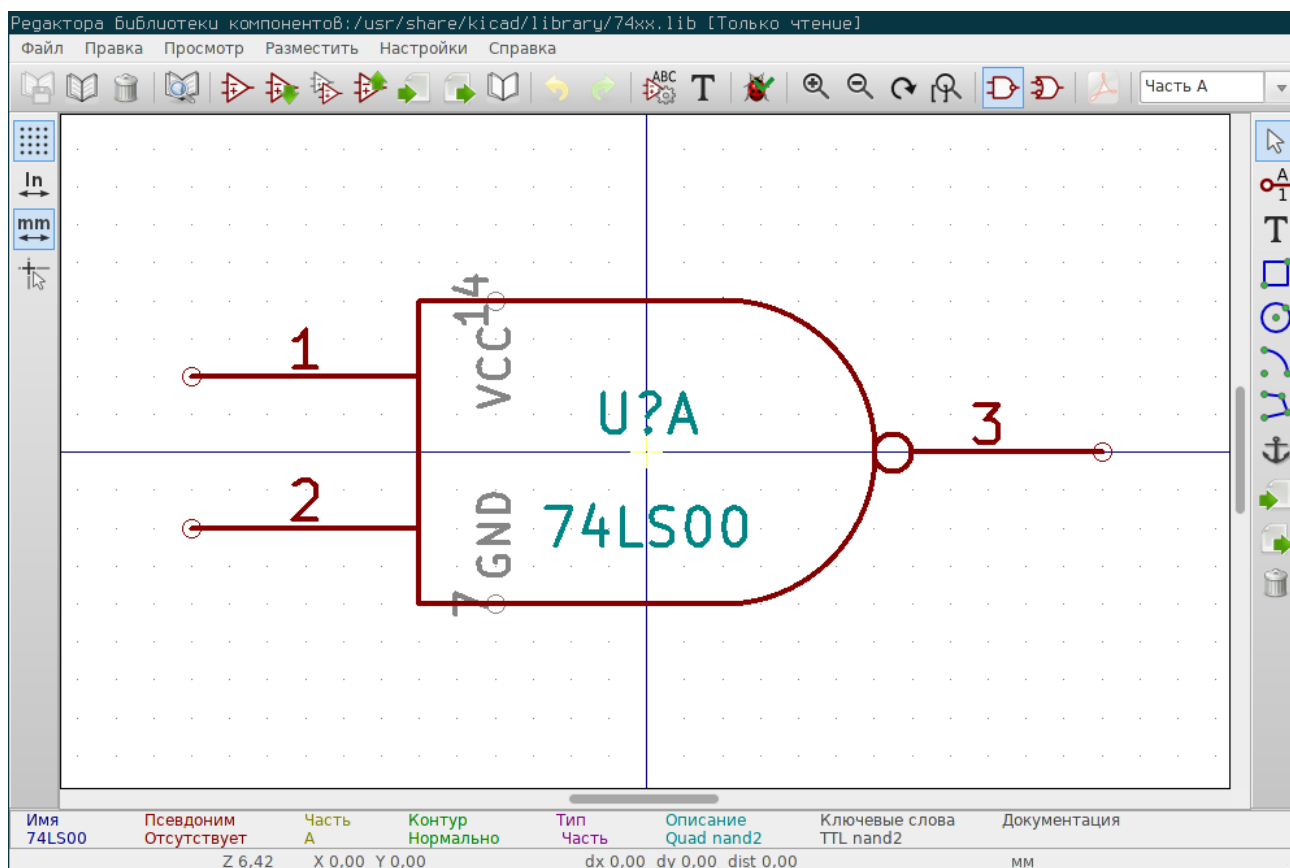
Компонент также имеет точку привязки. Поворот или отражение выполняется относительно этой точки и в процессе размещения, она используется в качестве опорной. Поэтому, располагать эту точку привязки нужно аккуратно.


Компонент может содержать псевдонимы, т.е. эквивалентные имена. Это позволяет значительно сократить количество компонентов, которые нужно создать (например, компоненту 74LS00 можно назначить псевдонимы 74000, 74HC00, 74HCT00, ...).

И последнее, компоненты распространяются в библиотеках (сгруппированы по назначению или производителю) упорядоченных так, чтобы облегчить работу с ними.

13.2 Установка точки привязки

Точка привязки имеет координаты (0,0) и отображается в виде синих осей на дисплее.




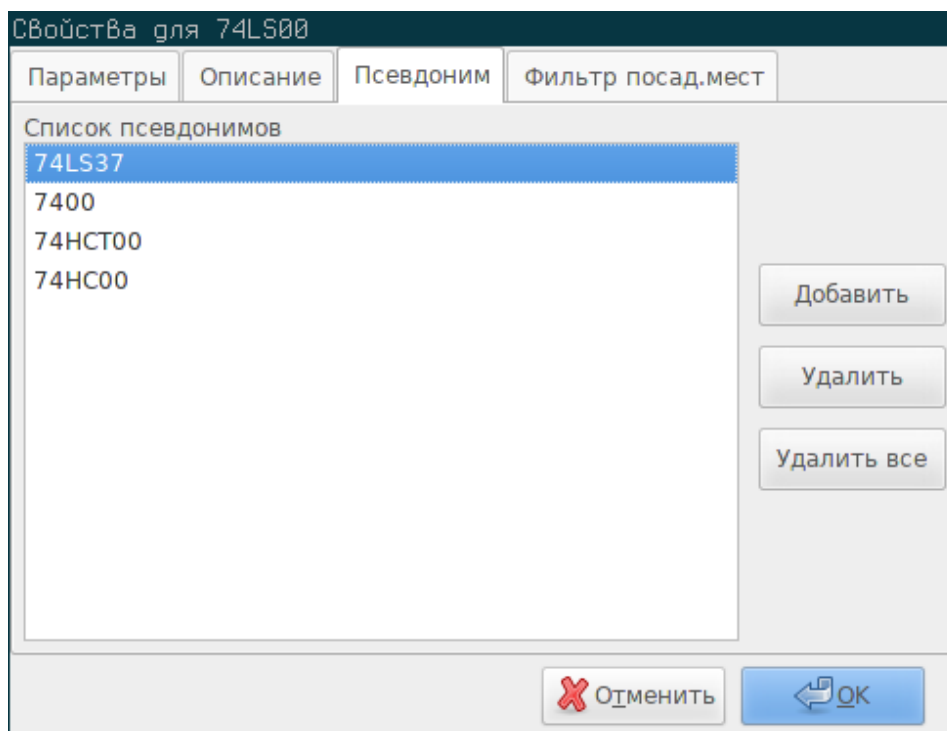
Точку привязки можно установить заново, выбрав  и нажав левой кнопкой мыши в точке её новой позиции. Изображение будет автоматически перерисовано с новым центром в указанной точке.

13.3 Псевдонимы

Псевдоним — это другое имя, связанное с одним и тем же компонентом библиотеки. Компоненты с подобным расположением выводов и графическим обозначением можно представить как один компонент с несколькими псевдонимами (как 7400 с псевдонимами 74LS00, 74HC00, 74LS37).

Применение псевдонимов позволяет строить целые библиотеки быстрее. Плюс ко всему, эти библиотеки будут намного компактнее и будут проще загружаться в KiCad.

Чтобы изменить список псевдонимов, нужно вызвать окно редактирования главных свойств с помощью  и выбрать вкладку псевдонимов.



Здесь можно добавлять или удалять нужные псевдонимы. Текущий псевдоним нельзя удалить, пока он загружен в редактор.

Чтобы удалить все псевдонимы, нужно сначала выбрать корневой компонент. Это первый компонент в списке псевдонимов из выпадающего списка на панели инструментов.

13.4 Поля компонента

Редактор полей вызывается с помощью кнопки .

В нём присутствуют четыре стандартных поля (текст, связанный с компонентом) и настраиваемые пользовательские поля.

Имя	Значение
Обозначение	U
Значение	74LS00
Посад.место	
Документация	

Гориз. выравнивание:
 Влево
 Центр
 Вправо

Верт. выравнивание:
 Вниз
 Центр
 Вверх

Видимость:
 Показать
 Повернуть

Стиль:
 Нормально
 Курсив
 Полужирный
 Полужирный курсив

Имя поля:
Обозначение

Значение поля:
U

Добавить поле
Удалить поле
Сдвинуть вверх


Размер: 1,270 мм
Позиция по X: 0,000 мм
Позиция по Y: -1,270 мм

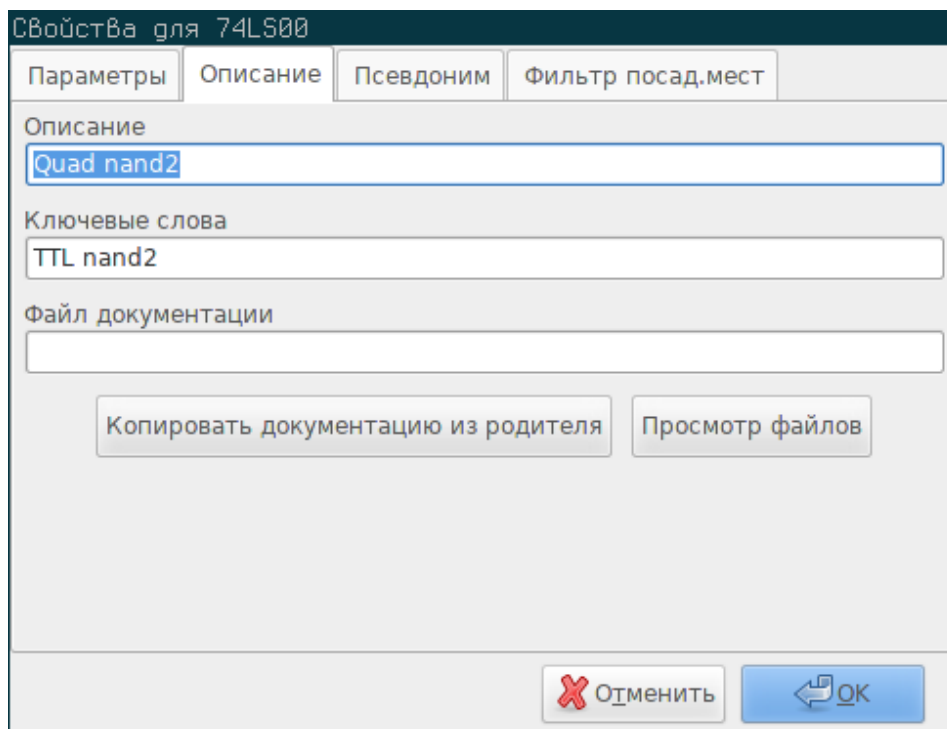
Отменить OK

Стандартные поля:

- Обозначение.
- Значение. Содержимое этого поля используется в качестве имени компонента в библиотеке и значения по умолчанию в схеме.
- Посадочное место. Здесь хранится имя посадочного места, используемого при проектировании печатной платы. При использовании CvPcb для назначения посадочных мест оно не очень полезно, но в противном случае — необходимо.
- Документация. Это зарезервированное поле и пока не используется.

13.5 Документация компонента

Для редактирования информации о компоненте нужно открыть окно главных свойств с помощью кнопки  и выбрать вкладку описания.



Нужно убедиться, что выбран нужный псевдоним или корневой компонент, так как это описание указывается отдельно для каждого из них. Кнопка "Копировать документацию из родителя" позволяет копировать информацию из корневого компонента в поля текущего.

13.5.1 Ключевые слова компонента

Ключевые слова позволяют вести выборочный поиск компонентов, в соответствии с заданными параметрами поиска (по функциональному назначению, технологии изготовления и пр.).

Инструмент поиска в Eeschema не чувствителен к регистру. В библиотеках в основном используются следующие ключевые слова:

- CMOS TTL — для микросхем стандартной логики
- AND2 NOR3 XOR2 INV... — для логических элементов (AND2 = элемент логического И с двумя входами, NOR3 = элемент логического ИЛИ-НЕ с тремя входами).
- JKFF DFF ... — для JK или D триггеров.
- ADC, DAC, MUX, ...
- OpenCol — для логических элементов с выходным каскадом в виде транзистора с открытым коллектором. Таким образом, если в редакторе схем в поле поиска компонента ввести ключевые слова NAND2 OpenCol, то Eeschema покажет список компонентов, которые их содержат.

13.5.2 Документация компонента

Строка описания (и ключевых слов) отображается в разных меню, в частности, при выборе компонента в списке библиотеки и в меню ViewLib.

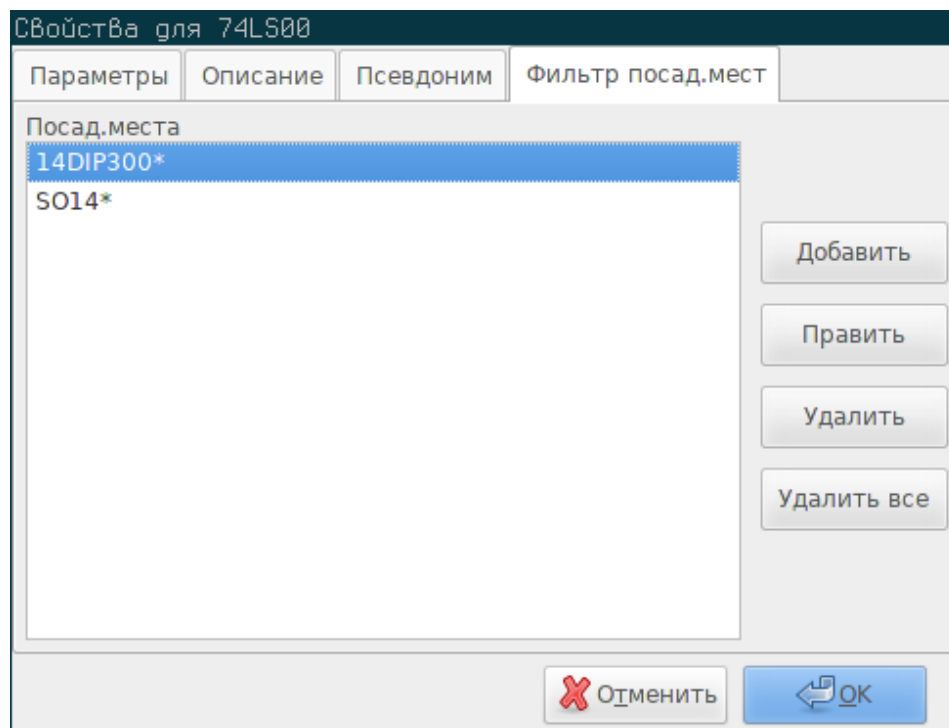
Если файл документации существует, то он будет доступен в редакторе схем из контекстного меню, которое открывается с помощью правой кнопки мыши.

13.5.3 Связанный файл документации

Указан файл (документации, примера применения), связанный с компонентом (pdf-файл, файл схемы или другое).

13.5.4 Фильтры посадочных мест для CvPcb

Можно ввести список подходящих посадочных мест для компонента. Этот список выступает в качестве фильтра, используемого в CvPcb для отображения только подходящих посадочных мест. Если список пуст — фильтрация не происходит.



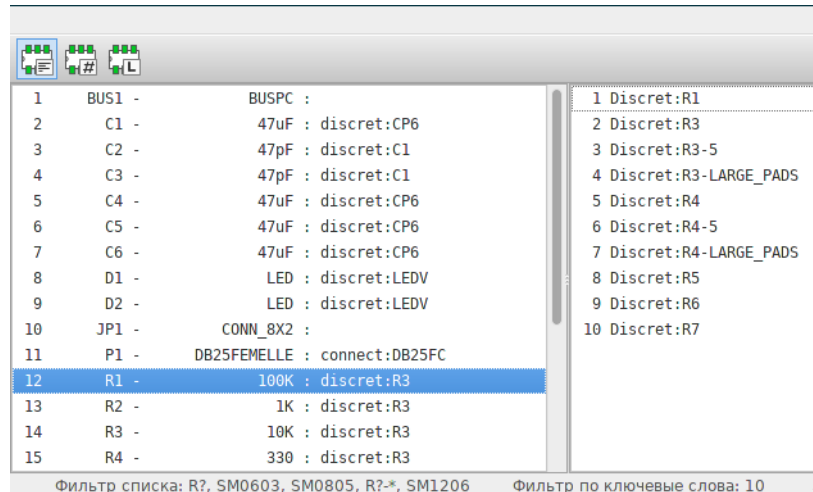
Допускается использование символов обобщения (масок, wild-card).

SO14* указывает CvPcb отобразить все посадочные места, имена которых начинаются с SO14.

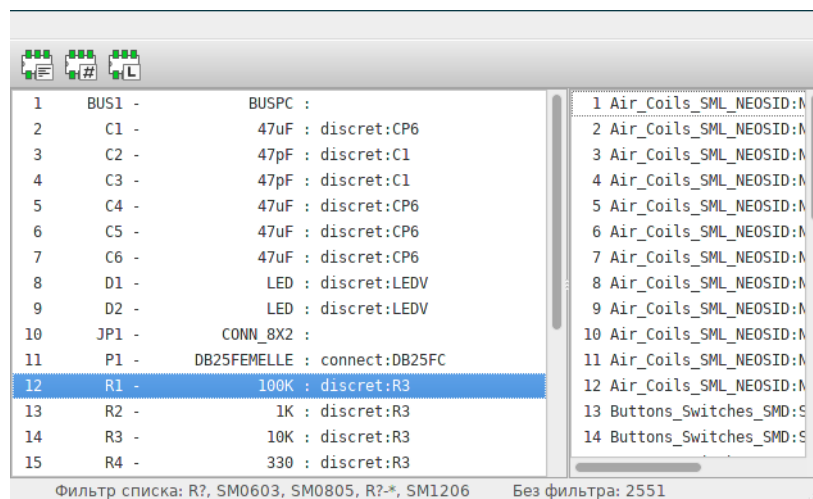
В случае с резисторами, R? покажет все посадочные места с именем из двух букв, первая из которых R.

Далее пример: с использованием фильтрации и без:

С фильтрацией



Без фильтрации




13.6 Библиотека символов


Можно достаточно просто создать файл графических символов, который будет содержать часто используемые символы. Они будут полезны при создании компонентов (треугольники, контуры логических элементов И, ИЛИ, исключающее ИЛИ и т.п.), для чего их сохраняют и, в последующем, используют.

Эти файлы, по умолчанию, хранятся в каталоге библиотек компонентов и имеют расширение *.sym*. Символы не собираются в библиотеки, как компоненты, потому что их не так много.

13.6.1 Экспорт (создание) символа

Компонент можно экспортировать как символ с помощью инструмента . Достаточно будет наличия только графического обозначения и, желательно, удалить все выводы, если они есть.


13.6.2 Импорт символа

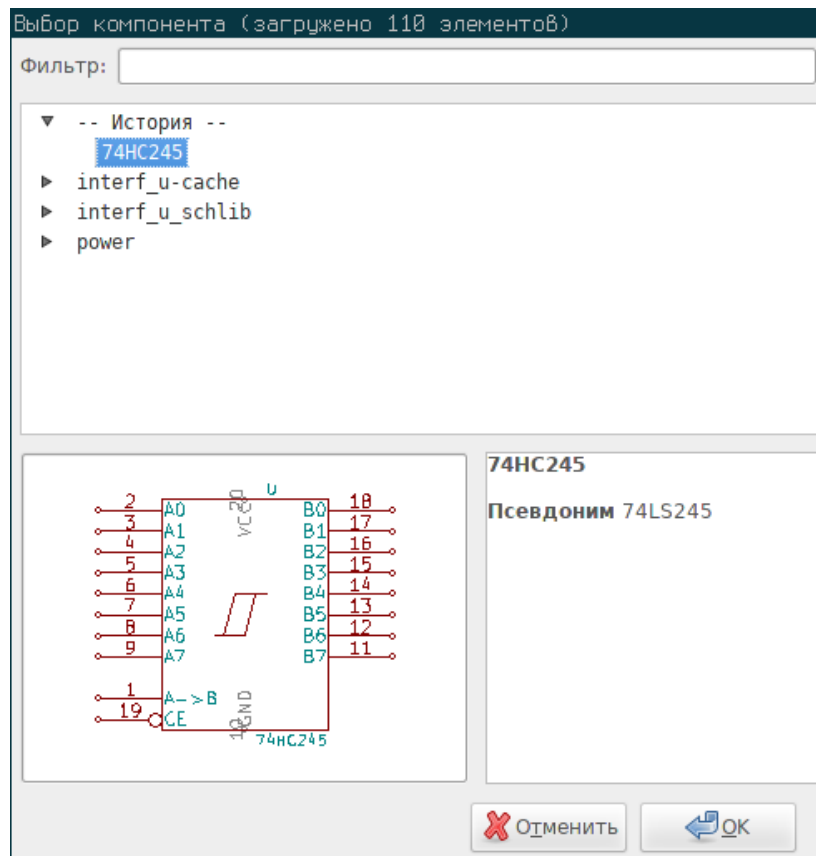
Импорт позволяет добавлять графические обозначения в редактируемый компонент. Символ загружается с помощью кнопки . Загруженное графическое обозначение размещается в том месте, в котором было создано перед экспортом.

Глава 14

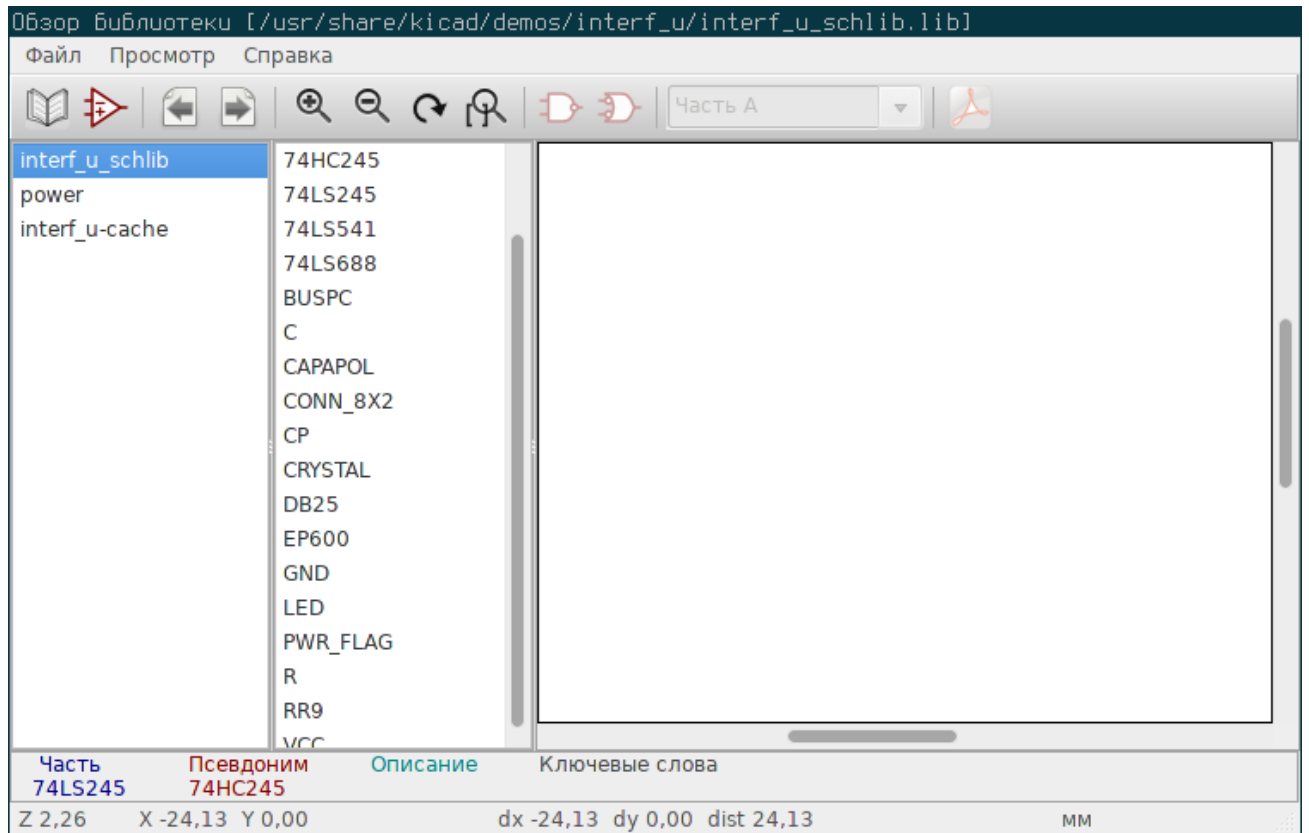
Просмотрщик библиотек компонентов

Введение

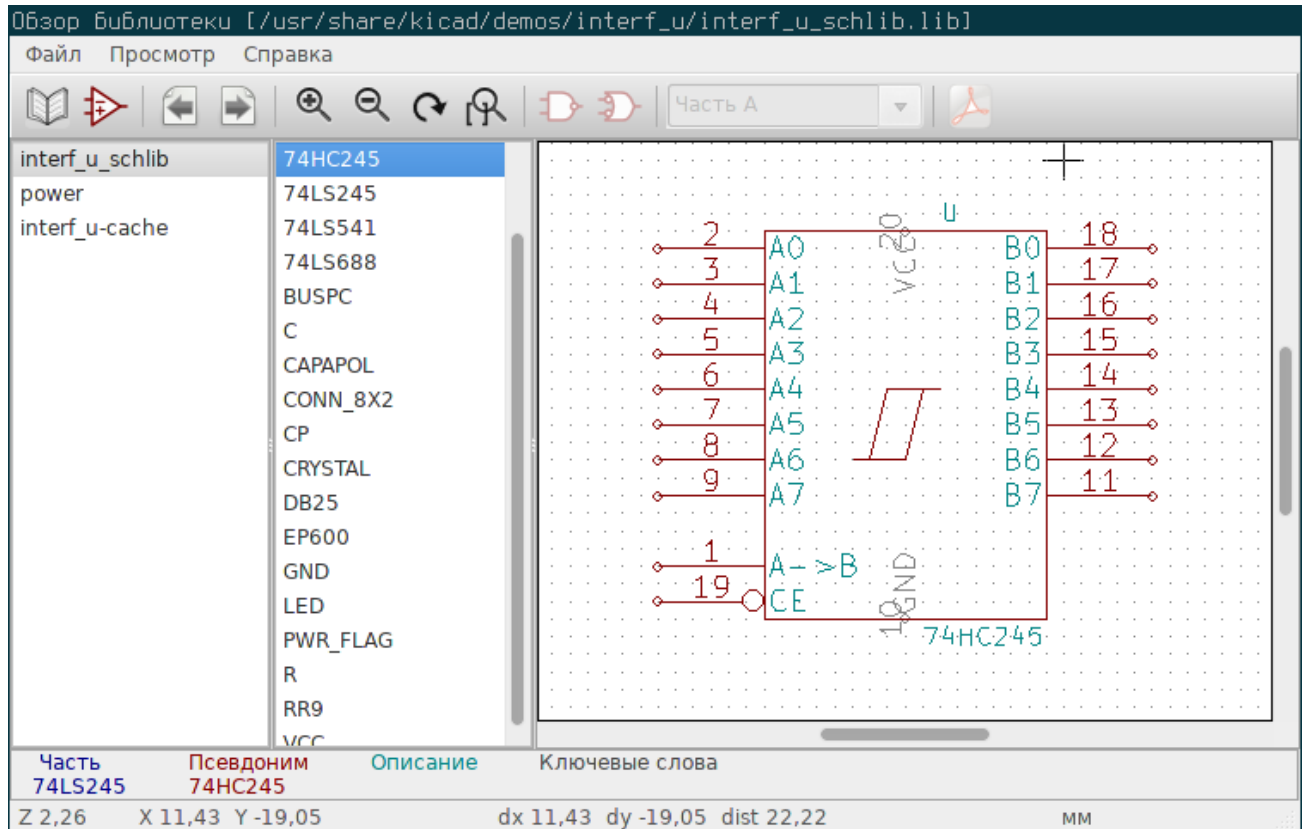
Просмотрщик библиотек компонентов позволяет быстро просмотреть содержимое библиотек. Он вызывается с помощью инструмента  из главной панели инструментов, выбором пункта "Просмотрщик библиотек" из меню "Вид" или двойным щелчком по изображению компонента в диалоге "Выбор компонента".



14.1 Главное окно



Чтобы просмотреть содержимое нужной библиотеки, нужно выбрать её в первой колонке слева. Доступные компоненты будут показаны во второй колонке, где можно выбрать нужный компонент для просмотра.





14.2 Верхняя панель инструментов просмотрщика компонентов библиотеки

Верхняя панель инструментов просмотрщика библиотеки компонентов показана ниже.



Доступны следующие команды:

	Выбор необходимой библиотеки, которую также можно выбрать из показанного списка.
	Выбор необходимого компонента, который также можно выбрать из показанного списка.
	Показать предыдущий компонент.
	Показать следующий компонент.
	Инструменты масштабирования.
	Выбор начертания (нормальное или альтернативное), если присутствует.
Unit A	Выбор части сложного компонента, который состоит из нескольких частей.

	<p>Если присутствует — показать назначенную документацию. Эта кнопка появляется, только при вызове из Eeschema для добавления компонента.</p>
	<p>Закрывает просмотрщик компонентов и располагает выбранный компонент в Eeschema. Эта кнопка отображается только в том случае, если просмотрщик был запущен из Eeschema (также, можно выбрать с помощью двойного щелчка по компоненту).</p>

Глава 15

Создание списков цепей и перечней элементов в различных форматах

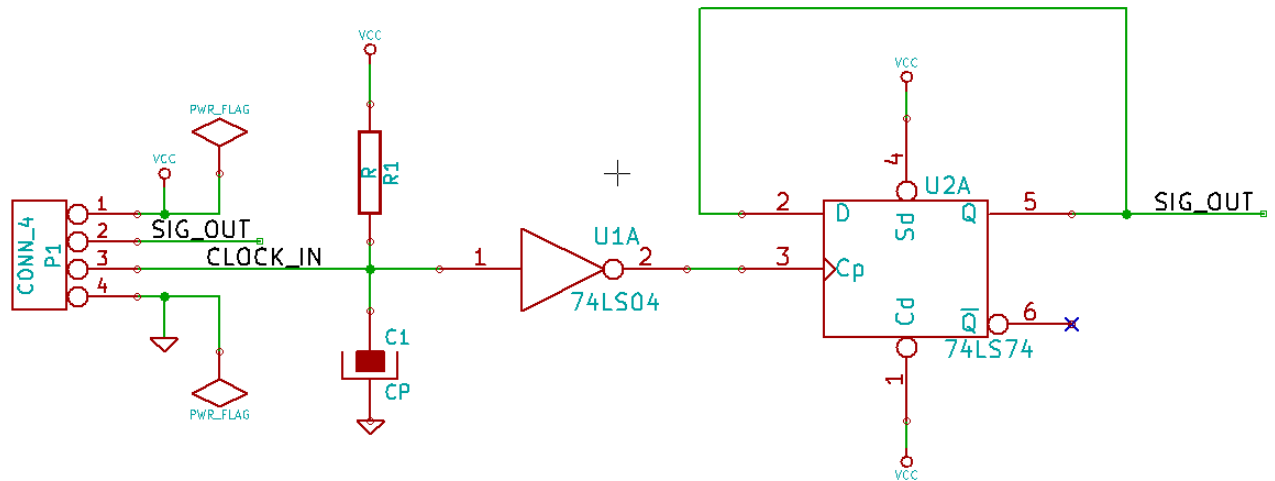
15.1 Промежуточный файл списка цепей

Перечни элементов и списки цепей можно создавать из промежуточного файла списка цепей, который создаёт Eeschema.

Этот файл построен с применением синтаксиса XML и называется промежуточным. Он содержит в себе большое количество данных о печатной плате и, благодаря этому, может быть использован для последующего преобразования в перечень элементов или файл другого формата.

В зависимости от выходного формата (перечень элементов или список цепей), будут использоваться различные области промежуточного файла, при генерации.

15.1.1 Пример схемы



15.1.2 Пример промежуточного файла списка цепей

Промежуточный файл списка цепей (в формате XML), соответствующий приведённой выше схеме, показан далее.

```
<?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="/" tstamp="/" />
      <tstamp>4C6E2141</tstamp>
    </comp>
    <comp ref="U2">
      <value>74LS74</value>
      <libsource lib="74xx" part="74LS74"/>
      <sheetpath names="/" tstamp="/" />
      <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>
      <fp>R?-*</fp>
      <fp>SM1206</fp>
    </footprints>
    <fields>
      <field name="Reference">R</field>
      <field name="Value">R</field>
    </fields>
    <pins>
```

```
<pin num="1" name="~" type="passive"/>
<pin num="2" name="~" type="passive"/>
</pins>
</libpart>
<libpart lib="conn" part="CONN_4">
  <description>Symbole general de connecteur</description>
  <fields>
    <field name="Reference">P</field>
    <field name="Value">CONN_4</field>
  </fields>
  <pins>
    <pin num="1" name="P1" type="passive"/>
    <pin num="2" name="P2" type="passive"/>
    <pin num="3" name="P3" type="passive"/>
    <pin num="4" name="P4" type="passive"/>
  </pins>
</libpart>
<libpart lib="74xx" part="74LS04">
  <description>Hex Inverseur</description>
  <fields>
    <field name="Reference">U</field>
    <field name="Value">74LS04</field>
  </fields>
  <pins>
    <pin num="1" name="~" type="input"/>
    <pin num="2" name="~" type="output"/>
    <pin num="3" name="~" type="input"/>
    <pin num="4" name="~" type="output"/>
    <pin num="5" name="~" type="input"/>
    <pin num="6" name="~" type="output"/>
    <pin num="7" name="GND" type="power_in"/>
    <pin num="8" name="~" type="output"/>
    <pin num="9" name="~" type="input"/>
    <pin num="10" name="~" type="output"/>
    <pin num="11" name="~" type="input"/>
    <pin num="12" name="~" type="output"/>
    <pin num="13" name="~" type="input"/>
    <pin num="14" name="VCC" type="power_in"/>
  </pins>
</libpart>
<libpart lib="74xx" part="74LS74">
  <description>Dual D FlipFlop, Set & Reset</description>
  <docs>74xx/74hc_hct74.pdf</docs>
  <fields>
    <field name="Reference">U</field>
    <field name="Value">74LS74</field>
  </fields>
  <pins>
```

```
<pin num="1" name="Cd" type="input"/>
<pin num="2" name="D" type="input"/>
<pin num="3" name="Cp" type="input"/>
<pin num="4" name="Sd" type="input"/>
<pin num="5" name="Q" type="output"/>
<pin num="6" name="~Q" type="output"/>
<pin num="7" name="GND" type="power_in"/>
<pin num="8" name="~Q" type="output"/>
<pin num="9" name="Q" type="output"/>
<pin num="10" name="Sd" type="input"/>
<pin num="11" name="Cp" type="input"/>
<pin num="12" name="D" type="input"/>
<pin num="13" name="Cd" type="input"/>
<pin num="14" name="VCC" type="power_in"/>
</pins>
</libpart>
</libparts>
<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>
  <library logical="74xx">
    <uri>F:\kicad\share\library\74xx.lib</uri>
  </library>
</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"/>
  </net>
</nets>
```

```
<node ref="U2" pin="3"/>
</net>
<net code="5" name="/SIG_OUT">
  <node ref="P1" pin="2"/>
  <node ref="U2" pin="5"/>
  <node ref="U2" pin="2"/>
</net>
<net code="6" name="/CLOCK_IN">
  <node ref="R1" pin="2"/>
  <node ref="C1" pin="1"/>
  <node ref="U1" pin="1"/>
  <node ref="P1" pin="3"/>
</net>
</nets>
</export>
```

15.2 Преобразование списка цепей в другой формат

Применяя сценарии преобразования, можно из промежуточного файла списка цепей получить упрощенный список цепей или перечень элементов. Так как выполняется преобразование из текстового файла в текстовый, то сценарий для преобразования можно составить на языках Python, XSLT или любом другом, умеющем работать с XML данными.

XSLT сам по себе построен на синтаксисе XML и очень удобен для работы с XML-файлами. Существует бесплатное приложение *xsltproc*, которое можно загрузить и установить. Приложение *xsltproc* можно использовать для считывания промежуточного файла списка цепей в формате XML, применить сценарий для преобразования полученных данных и сохранить результат в выходной файл. Использование *xsltproc* требует написания сценария на языке XSLT. Весь процесс преобразования контролирует Eeschema, нужно лишь один раз настроить её для запуска *xsltproc*.

15.3 Основы XSLT

Документ, объясняющий преобразования XSL (XSLT - XSL Transformations), доступен здесь:

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

15.3.1 Создание списка цепей в формате Pads-Pcb

Формат *pads-pcb* состоит из двух частей.

- Списка посадочных мест.
- Списка соединений: контактные площадки сгруппированы по именам цепей.

Сразу под этим, показан сценарий для преобразования промежуточного списка цепей в файл в формате *pads-pcb*:

```
<?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.

  How to use:
    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>
    <xsl:value-of select="@ref"/>
    <xsl:text>.</xsl:text>
    <xsl:value-of select="@pin"/>
    <xsl:text>&nl;</xsl:text>
</xsl:template>

</xsl:stylesheet>
```

А здесь, показано содержимое выходного файла в формате pads-pcb после работы xsltproc:

```
*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*
```

Командная строка для запуска этого преобразования:

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

15.3.2 Создание списка цепей в формате Cadstar

Формат Cadstar состоит из двух частей.

- Списка посадочных мест.
- Списка соединений: контактные площадки сгруппированы по именам цепей.

Далее показан файл сценария для выполнения необходимых преобразований:

```
<?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">
            <xsl:text>"</xsl:text>
            <xsl:choose>
                <xsl:when test = "@name != '' ">
                    <xsl:value-of select="@name"/>
                </xsl:when>
                <xsl:otherwise>
                    <xsl:text>N-</xsl:text>
                </xsl:otherwise>
            </xsl:choose>
        </xsl:variable>
        <xsl:apply-templates select="@pins"/>
    </xsl:if>
</xsl:template>

```

```
        <xsl:value-of select="@code"/>
    </xsl:otherwise>
</xsl:choose>
<xsl:text>"&nl;</xsl:text>
</xsl:variable>
<xsl:apply-templates select="node" mode="first"/>
<xsl:value-of select="$netname"/>
<xsl:apply-templates select="node" mode="others"/>
</xsl:if>
</xsl:template>

<!-- for each node -->
<xsl:template match="node" mode="first">
    <xsl:if test="position()=1">
        <xsl:text>.ADD_TER </xsl:text>
        <xsl:value-of select="@ref"/>
        <xsl:text>.</xsl:text>
        <xsl:value-of select="@pin"/>
        <xsl:text> </xsl:text>
    </xsl:if>
</xsl:template>

<xsl:template match="node" mode="others">
    <xsl:choose>
        <xsl:when test='position()=1'>
            </xsl:when>
        <xsl:when test='position()=2'>
            <xsl:text>.TER </xsl:text>
        </xsl:when>
        <xsl:otherwise>
            <xsl:text> </xsl:text>
        </xsl:otherwise>
    </xsl:choose>
    <xsl:if test="position()>1">
        <xsl:value-of select="@ref"/>
        <xsl:text>.</xsl:text>
        <xsl:value-of select="@pin"/>
        <xsl:text>&nl;</xsl:text>
    </xsl:if>
</xsl:template>

</xsl:stylesheet>
```

Это выходной файл в формате Cadstar.

```
.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
```

15.3.3 Создание списка цепей в формате OrcadPCB2

Этот формат имеет только одну часть, которая является списком посадочных мест. Каждое посадочное место содержит перечень его контактных площадок с указанием подсоединённых цепей.

Здесь приведён сценарий для выполнения необходимых преобразований:

```
<?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.

  How to use:
  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>
  <xsl:choose>
    <xsl:when test = "tstamp != '' ">
      <xsl:apply-templates select="tstamp"/>
    </xsl:when>
    <xsl:otherwise>
      <xsl:text>00000000</xsl:text>
    </xsl:otherwise>
  </xsl:choose>
  <xsl:text> </xsl:text>
  <xsl:choose>
    <xsl:when test = "footprint != '' ">
      <xsl:apply-templates select="footprint"/>
    </xsl:when>
    <xsl:otherwise>
      <xsl:text>$noname</xsl:text>
    </xsl:otherwise>
  </xsl:choose>
  <xsl:text> </xsl:text>
  <xsl:value-of select="@ref"/>
  <xsl:text> </xsl:text>
  <xsl:choose>
    <xsl:when test = "value != '' ">
      <xsl:apply-templates select="value"/>
    </xsl:when>
    <xsl:otherwise>
      <xsl:text>"~"</xsl:text>
    </xsl:otherwise>
  </xsl:choose>
  <xsl:text>&nl;</xsl:text>
  <xsl:call-template name="Search_pin_list" >
    <xsl:with-param name="cmplib_id" select="libsource/@part"/>
    <xsl:with-param name="cmp_ref" select="@ref"/>
  </xsl:call-template>
  <xsl:text> )&nl;</xsl:text>
</xsl:template>

<!--
  This template search for a given lib component description in list
  lib component descriptions are in /export/libparts,
  and each description start at ./libpart
  We search here for the list of pins of the given component
  This template has 2 parameters:
    "cmplib_id" (reference in libparts)
    "cmp_ref"   (schematic reference of the given component)
-->
<xsl:template name="Search_pin_list" >

```

```

<xsl:param name="cmplib_id" select="0" />
<xsl:param name="cmp_ref" select="0" />
  <xsl:for-each select="/export/libparts/libpart">
    <xsl:if test = "@part = $cmplib_id ">
      <xsl:apply-templates name="build_pin_list" select="pins/pin">
        <xsl:with-param name="cmp_ref" select="$cmp_ref"/>
      </xsl:apply-templates>
    </xsl:if>
  </xsl:for-each>
</xsl:template>

```

```
<!--
```

This template writes the pin list of a component
from the pin list of the library description
The pin list from library description is something like

```

<pins>
  <pin num="1" type="passive"/>
  <pin num="2" type="passive"/>
</pins>

```

Output pin list is (<pin num> <net name>)
something like

```

( 1 VCC )
( 2 GND )

```

```
-->
```

```

<xsl:template name="build_pin_list" match="pin">
  <xsl:param name="cmp_ref" select="0" />

  <!-- write pin numner and separator -->
  <xsl:text> ( </xsl:text>
  <xsl:value-of select="@num"/>
  <xsl:text> </xsl:text>

  <!-- search net name in nets section and write it: -->
  <xsl:variable name="pinNum" select="@num" />
  <xsl:for-each select="/export/nets/net">
    <!-- net name is output only if there is more than one pin in net
    else use "?" as net name, so count items in this net
    -->
    <xsl:variable name="pinCnt" select="count(node)" />
    <xsl:apply-templates name="Search_pin_netname" select="node">
      <xsl:with-param name="cmp_ref" select="$cmp_ref"/>
      <xsl:with-param name="pin_cnt_in_net" select="$pinCnt"/>
      <xsl:with-param name="pin_num"> <xsl:value-of select="$pinNum"/>
    </xsl:with-param>
    </xsl:apply-templates>
  </xsl:for-each>

```

```

    <!-- close line -->
    <xsl:text> )&nl;</xsl:text>
</xsl:template>

<!--
This template writes the pin netname of a given pin of a given component
from the nets list
The nets list description is something like
    <nets>
        <net code="1" name="GND">
            <node ref="J1" pin="20"/>
            <node ref="C2" pin="2"/>
        </net>
        <net code="2" name="">
            <node ref="U2" pin="11"/>
        </net>
    </nets>
This template has 2 parameters:
    "cmp_ref"    (schematic reference of the given component)
    "pin_num"    (pin number)
-->

<xsl:template name="Search_pin_netname" match="node">
    <xsl:param name="cmp_ref" select="0" />
    <xsl:param name="pin_num" select="0" />
    <xsl:param name="pin_cnt_in_net" select="0" />

    <xsl:if test = "@ref = $cmp_ref ">
        <xsl:if test = "@pin = $pin_num">
            <!-- net name is output only if there is more than one pin in net
                else use "?" as net name
            -->
            <xsl:if test = "$pin_cnt_in_net>1">
                <xsl:choose>
                    <!-- if a net has a name, use it,
                        else build a name from its net code
                    -->
                    <xsl:when test = "../@name != '' ">
                        <xsl:value-of select="../@name"/>
                    </xsl:when>
                    <xsl:otherwise>
                        <xsl:text>$N-0</xsl:text><xsl:value-of select="../@code"/>
                    </xsl:otherwise>
                </xsl:choose>
            </xsl:if>
            <xsl:if test = "$pin_cnt_in_net <2">
                <xsl:text>?</xsl:text>
            </xsl:if>

```



```
        </xsl:if>
    </xsl:if>

</xsl:template>

</xsl:stylesheet>
```

Здесь показан выходной файл в формате OrcadPCB2.

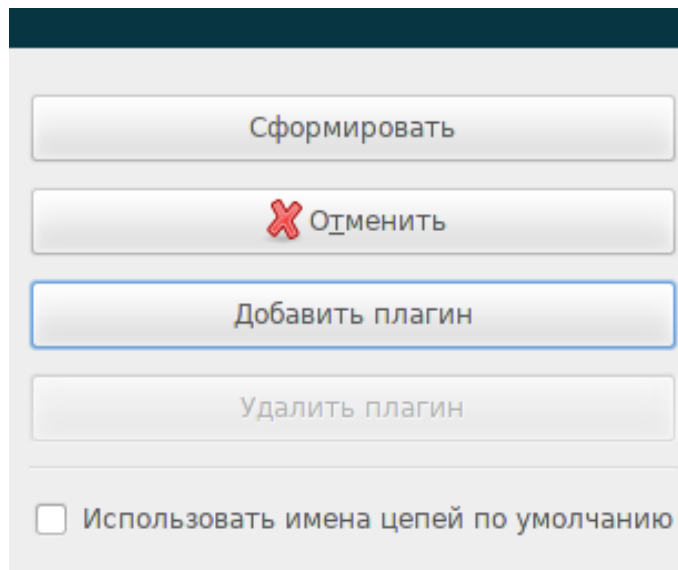
```
( { Eeschema Netlist Version 1.1 29/08/2010 21:07:51
eeschema (2010-08-28 BZR 2458)-unstable}
( 4C6E2141 $noname P1 CONN_4
( 1 VCC )
( 2 /SIG_OUT )
( 3 /CLOCK_IN )
( 4 GND )
)
( 4C6E20BA $noname U2 74LS74
( 1 VCC )
( 2 /SIG_OUT )
( 3 N-04 )
( 4 VCC )
( 5 /SIG_OUT )
( 6 ? )
( 7 GND )
( 14 VCC )
)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

15.3.4 Подключение плагинов в Eeschema

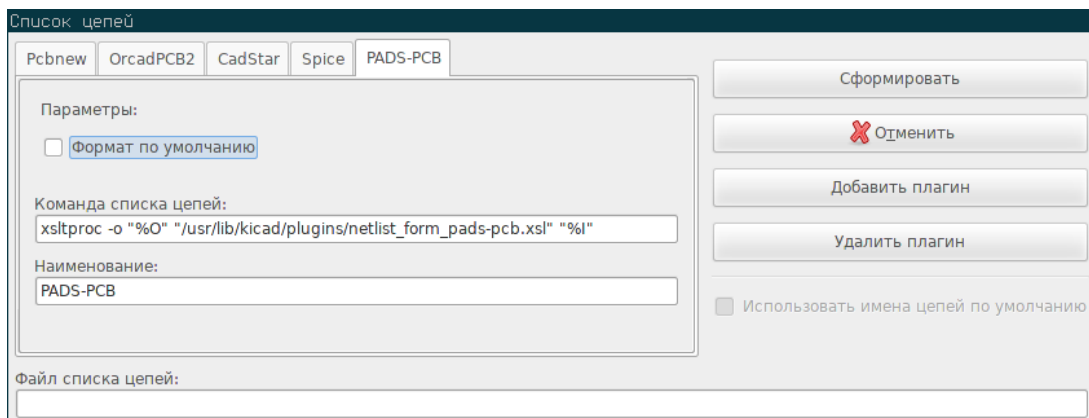
Инструменты преобразования промежуточного списка цепей можно автоматически запускать из Eeschema.

15.3.4.1 Настройка диалогового окна

Каждый может добавить вкладку для нового плагина генерации списка соединений с помощью кнопки "Добавить плагин".



Здесь показано как выглядит вкладка с параметрами нового плагина PadsPcb:



15.3.4.2 Параметры плагина

Диалоговое окно настройки плагина в Eeschema требует следующую информацию:

- Наименование: имя формата списка цепей, к примеру.
- Командная строка для запуска преобразования.

Как только будет нажата кнопка Сформировать, произойдет следующее:

1. Eeschema создаст промежуточный файл списка цепей *.xml, например, test.xml.
2. Eeschema запустит плагин, который считает test.xml и создаст test.net.

15.3.4.3 Создание списка цепей с помощью командной строки

Предположим, используется приложение *xsltproc.exe* для выполнения сценария по преобразованию промежуточного файла, тогда командная строка для запуска *xsltproc.exe* будет выглядеть так:

```
xsltproc.exe -o <имя выходного файла> <имя сценария> <имя входного XML файла для преобразования>
```

Если KiCad работает в операционной системе Windows, то командная строка будет иметь следующий вид:

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

В операционных системах на базе Linux командная строка будет такой:

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

Где *netlist_form_pads-pcb.xml* — файл сценария, который нужно применить. Не забывайте ставить кавычки вокруг имён файлов, это позволит Eeschema обрабатывать файлы с пробелами в именах.

Командная строка поддерживает команды для указания имен файлов:

Поддерживаемые параметры форматирования.

- *%B* ⇒ базовое имя — полное имя выходного файла без пути и расширения.
- *%I* ⇒ полное имя временного входного файла (промежуточный файл списка цепей).
- *%O* ⇒ полное имя выходного файла.

%I будет заменено актуальным именем промежуточного файла

%O будет заменено актуальным именем выходного файла.

15.3.4.4 Формат командной строки: пример для xsltproc

Формат командной строки для *xsltproc* следующий:

```
<путь к xsltproc> xsltproc <параметры для xsltproc>
```

В Windows:

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

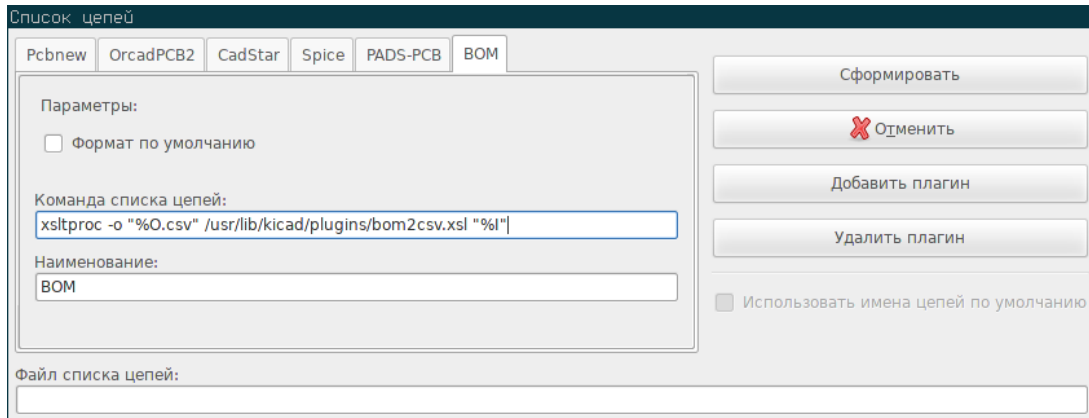
В Linux:

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

В приведённых выше примерах предполагается, что все файлы *xsltproc* на ПК с ОС Windows установлены в каталог *kicad/bin*.

15.3.5 Создание перечня элементов

Так как промежуточный файл списка цепей содержит всю информацию об использованных компонентах, то из него можно извлечь перечень элементов. Далее показано диалоговое окно с настроенным плагином (в Linux) для генерации перечня элементов (Bill Of Materials — BOM):



Путь к файлу сценария `bom2csv.xsl` зависит от используемой операционной системы. На данный момент, лучший сценарий XSLT для создания перечня элементов называется `bom2csv.xsl`. Если нужно, его всегда можно изменить и, если в результате получится что-то полезное для всех, попросить сделать новый сценарий частью проекта KiCad.

15.4 Формат командной строки: пример сценария на языке Python

Командная строка для скрипта на языке Python выглядит, приблизительно, так:

```
python <имя файла скрипта> <имя входного файла> <имя выходного файла>
```

В Windows:

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

В Linux:

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

Предполагается, что Python уже установлен на ПК.

15.5 Структура промежуточного файла списка цепей

Этот пример даёт представление о формате файла списка цепей.

```
<?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="/" tstamp="/" />
      <tstamp>4C6E2141</tstamp>
```

```
</comp>
<comp ref="U2">
  <value>74LS74</value>
  <libsource lib="74xx" part="74LS74"/>
  <sheetpath names="/" tstamp="/" />
  <tstamp>4C6E20BA</tstamp>
</comp>
<comp ref="U1">
  <value>74LS04</value>
  <libsource lib="74xx" part="74LS04"/>
  <sheetpath names="/" tstamp="/" />
  <tstamp>4C6E20A6</tstamp>
</comp>
<comp ref="C1">
  <value>CP</value>
  <libsource lib="device" part="CP"/>
  <sheetpath names="/" tstamp="/" />
  <tstamp>4C6E2094</tstamp>
<comp ref="R1">
  <value>R</value>
  <libsource lib="device" part="R"/>
  <sheetpath names="/" tstamp="/" />
  <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>
```

```
</net>
<net code="5" name="/SIG_OUT">
  <node ref="P1" pin="2"/>
  <node ref="U2" pin="5"/>
  <node ref="U2" pin="2"/>
</net>
<net code="6" name="/CLOCK_IN">
  <node ref="R1" pin="2"/>
  <node ref="C1" pin="1"/>
  <node ref="U1" pin="1"/>
  <node ref="P1" pin="3"/>
</net>
</nets>
</export>
```

15.5.1 Основные элементы файла списка цепей

Промежуточный файл насчитывает пять разделов.

- Раздел оглавления (design).
- Раздел компонентов (components).
- Раздел частей компонента (libparts).
- Раздел библиотек (libraries).
- Раздел цепей (nets).

Содержимое файла ограничивается разделителем <export>

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

15.5.2 Оглавление

Оглавление ограничивается разделителем <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>
```

Этот раздел может содержать подраздел с комментариями.

15.5.3 Компоненты

Раздел компонентов ограничивается разделителем <components>

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

Этот раздел содержит перечень компонентов используемых в схеме. Каждый компонент имеет следующее описание:

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

libsource	имя библиотеки, из которой был взят компонент.
part	имя компонента в библиотеке.
sheetpath	путь к листу иерархии: указан лист из всей иерархии схемы
tstamps (time stamps)	метка времени файла схемы.
tstamp (time stamp)	метка времени компонента.

15.5.3.1 Замечание о метке времени компонента

Для сопоставления компонентов из списка цепей, компонентам из печатной платы, используют метки времени в качестве уникального идентификатора для каждого компонента. Этот способ используется в KiCad как вспомогательный, для нахождения соответствующего компонента из списка цепей на печатной плате. Это позволяет заново обозначать компоненты в проекте схемы и не терять при этом связь с посадочным местом.

Метка времени — это уникальный идентификатор для каждого компонента или листа в проекте схемы. Но, в сложных иерархиях, где один файл схемы может использоваться несколькими листами, компоненты из этих листов имеют одни и те же метки времени.

Каждый лист сложной иерархии имеет свой уникальный идентификатор: его путь (sheetpath). Каждый компонент (из листа сложной иерархии), тоже, имеет уникальный идентификатор: путь листа + его метка времени.

15.5.4 Части компонента

Раздел частей компонента ограничивается разделителем <libparts> и содержит описания из библиотеки схемы. Этот раздел содержит:

- Допустимые имена посадочных мест (можно использовать маски) — разделитель <fp>.
- Поля, указанные в библиотеке — разделитель <fields>.
- Перечень выводов — разделитель <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>

```

Строки типа <pin num="1" type="passive"/> также содержат электрический тип вывода. Доступны следующие электрические типы:

Input	Вход — обычный входной вывод
Output	Выход — обычный выходной вывод
Bidirectional	Двухнаправленный — может быть как входом, так и выходом
Tri-state	Трехстабильный — ввод/вывод шины
Passive	Пассивный — обычный вывод пассивных компонентов
Unspecified	Не определено — не известный электрический тип
Power input	Вход питания — вход питания компонента
Power output	Выход питания — выход, на подобии выхода стабилизатора напряжения
Open collector	Открытый коллектор — часто используется в аналоговых компараторах
Open emitter	Открытый эмиттер — иногда используется в логических элементах
Not connected	Не подсоединен — должен остаться без подключения в схеме

15.5.5 Библиотеки

Раздел библиотек ограничивается разделителем <libraries>. Этот раздел состоит из списка библиотек компонентов, используемых в проекте схемы.

```

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

```



```

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

```

15.5.6 Цепи

Раздел цепей ограничивается разделителем <nets>. Этот раздел состоит из "подключений" в схеме.

```

<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>

```

В этом разделе перечисляются все цепи, присутствующие в схеме.

Обычно цепь состоит из следующего:

```

<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	внутренний идентификатор данной цепи
name	имя цепи
node	содержит ссылку на вывод компонента, подключенного к данной цепи

15.6 Более подробно о xsltproc

Посмотрите страницу: <http://xmlsoft.org/XSLT/xsltproc.html>

Введение

xsltproc — это консольное приложение для выполнения сценариев XSLT по преобразованию XML-файлов. На данный момент оно разрабатывается как часть проекта GNOME, но может использоваться отдельно от рабочего окружения GNOME.

xsltproc вызывается из командной строки (терминала) с указанием сценария и имени файла, к которому этот сценарий должен быть применён. Если нужно преобразовать стандартный ввод вместо файла, используют - (дефис).

Если сценарий содержится в самом XML-файле, то его не нужно указывать в командной строке. xsltproc автоматически определит наличие сценария в файле и применит его. По умолчанию, результат преобразования будет выведен на экран (*stdout*). Можно, вместо этого, указать имя выходного файла с помощью параметра -o.

15.6.1 Формат командной строки

```
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* ]
[ *....* ]
```

15.6.2 Параметры командной строки

-V или --version

Показать версию используемых библиотек libxml и libxslt.

-v или --verbose

Выводить описание каждого шага работы xsltproc в процессе применения сценария к документу.

-o или --output *имя_файла*

Направляет вывод в файл под именем *имя_файла*. При множественном выводе параметр -o *имя_каталога/* указывает вывести файлы в указанный каталог. Данный каталог должен существовать.

--timing

Отобразить время, затраченное на считывание сценария, выполнение преобразования и сохранение результата. Отображается в миллисекундах.

--repeat

Выполнить преобразование 20 раз. Используется для тестирования времени обработки.

--debug

Выводит дополнительный файл с древовидной XML-структурой в целях отладки.

--novalid

Пропустить загрузку элемента DTD из документа.

--noout

Не выводить результат.

--maxdepth значение

Настроить максимальную глубину стека, чтобы обеспечить выход из бесконечного цикла в libxslt. Значение по умолчанию — 500.

--html

Входной файл в формате HTML.

--param параметр значение

Передать параметр *параметр* со значением *значение* в сценарий. Можно передавать несколько параметров (до 32). Если в качестве значения будет передаваться строка, нужно использовать параметр *--stringparam*.

--stringparam параметр значение

Передать параметр *параметр* со значением *значение*, где *значение* — строка. (Примечание: строка должна быть в формате utf-8).

--nonet

Не использовать Интернет для получения DTD, объектов или документов.

--path пути

Использовать список (разделённый пробелами или точками с запятой) путей файловой системы, указанных в *пути* для загрузки DTD, объектов или документов.

--load-trace

Показать в стандартном выводе stderr все документы, загруженные в процессе работы.

--catalogs

Использовать каталог SGML, указанный в SGML_CATALOG_FILES, для установки расположения внешних объектов. По умолчанию, xsltproc ищет их в каталоге, указанном в XML_CATALOG_FILES. Если он не задан, используется /etc/xml/catalog.

--xinclude

Преобразовать входной документ используя спецификацию Xinclude. Больше информации об этом можно найти здесь: <http://www.w3.org/TR/xinclude/>

--profile --norman

Выводить профильную информацию с детальным описанием времени, потраченного на каждую часть сценария. Это будет полезным для оптимизации быстродействия сценария.

--dumpextensions

Вывести перечень всех зарегистрированных расширений на экран.

--nowrite

Запретить запись в любой файл или источник.

--nomkdir

Запретить создание каталогов.

--writesubtree каталог

Разрешить записывать файл только внутри указанного каталога.

--nodtdattr

Не применять значение по умолчанию для атрибута DTD в файле.

15.6.3 Возвращаемые значения xsltproc

xsltproc возвращает код результата, который может быть очень полезен при работе из командной строки.

0: норма

1: нет аргументов

2: слишком много параметров

3: неизвестный параметр

4: невозможно загрузить сценарий

5: ошибка в сценарии

6: ошибка в одном из документов

7: не поддерживаемый метод вывода xsl

8: строковый параметр содержит как одинарные, так и двойные кавычки

9: внутренняя ошибка обработки

10: работа была прервана внешним сигналом

11: невозможно записать результат в выходной файл

15.6.4 Больше подробностей о xsltproc

Страница о libxml: <http://www.xmlsoft.org/>

Страница о W3C XSLT: <http://www.w3.org/TR/xslt>

Глава 16

Симулятор

Eeschema имеет встроенный симулятор электрических схем, основанный на [ngspice](#).

При работе с симулятором, будут полезны официальные библиотеки *pspice*. Они содержат основные компоненты, используемые при симуляции, такие как источники напряжения/тока или транзисторы с нумерацией выводов, которая соответствует порядку узлов (node) в спецификации ngspice.

Также, имеется несколько демонстрационных проектов, показывающих возможности симулятора. Их можно найти в каталоге *demos/simulation*.

16.1 Назначение моделей

Перед началом симуляции, необходимо назначить компонентам Spice-модели.

Каждому компоненту можно назначить только одну модель, даже если компонент состоит из нескольких частей. В последнем случае, модель назначается первой части.

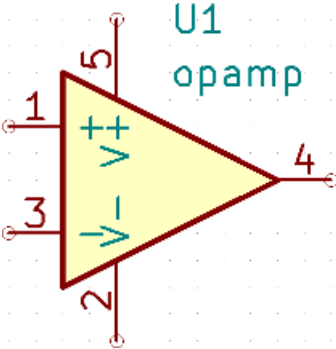
Пассивным компонентам, обозначения которых совпадают с типом элемента в Spice-записи (R^* для резисторов, C^* для конденсаторов, L^* для индуктивностей), модели будут назначены косвенно, а поле значения будет восприниматься как свойство модели.

Замечание

Имейте в виду, что в Spice множитель M обозначает мили, а Meg — мега. Если желаете использовать M в качестве множителя мега, установите соответствующий параметр в [диалоговом окне настройки симуляции](#).

Информация для spice-моделей хранится в виде текста в полях компонента и её можно легко указать с помощью редактора компонентов или непосредственно в редакторе схем. Откройте диалог редактирования свойств компонента и нажмите кнопку *Редактировать Spice модель* для вызова диалогового окна редактора Spice модели.

Редактор Spice моделей имеет три вкладки, которые соответствуют трём разным типам моделей. Также имеется два параметра, общих для всех типов:

Не симулировать компонент	Если отмечен, компонент исключается из симуляции.
Изменить последовательность выводов	<p>Позволяет переопределить выводы компонента для соответствия описанию узлов модели. Необходимо указать номера выводов в той последовательности, в которой их ожидает модель.</p> <p><i>Пример:</i></p> <pre>* connections: * 1: non-inverting input * 2: inverting input * 3: positive power supply * 4: negative power supply * 5: output .subckt tl071 1 2 3 4 5</pre>  <p>To match the symbol pins to the Spice model nodes shown above, one needs to use an alternate node sequence option with value: "1 3 5 2 4". It is a list of pin numbers corresponding to the Spice model nodes order.</p>

16.1.1 Пассивный

Вкладка *Пассивный* позволяет назначить модель (резистор, конденсатор или индуктивность) компоненту. Этот параметр используется редко, так как обычно пассивным элементам модели назначаются *косвенно*. Если же обозначение компонента не совпадает с необходимым типом модели, то нужно явно указать параметры на данной вкладке.

Замечание

Явно указанные модели имеют больший приоритет перед косвенно назначенными моделями. Это значит, что как только модель назначена вручную, поля обозначения и значения не принимаются во внимание при симуляции. Это может привести к путанице, когда значение модели не совпадает со значением, указанным на схеме.

Редактор Spice модели

Пассивный | Модель | Источник

Тип: Пассивный тип

Значение: Spice значение в симуляции

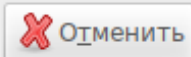

В Spice значениях десятичный разделитель - точка.
Значения могут использовать Spice символы приставок.

Spice приставки в значениях (нечувствительны к регистру):

f	фемто	1e-15
p	пико	1e-12
n	нано	1e-9
u	микро	1e-6
m	милли	1e-3
k	кило	1e3
meg	мега	1e6
g	гига	1e9
t	тера	1e12

Не симулировать компонент

Изменить последовательность выводов:

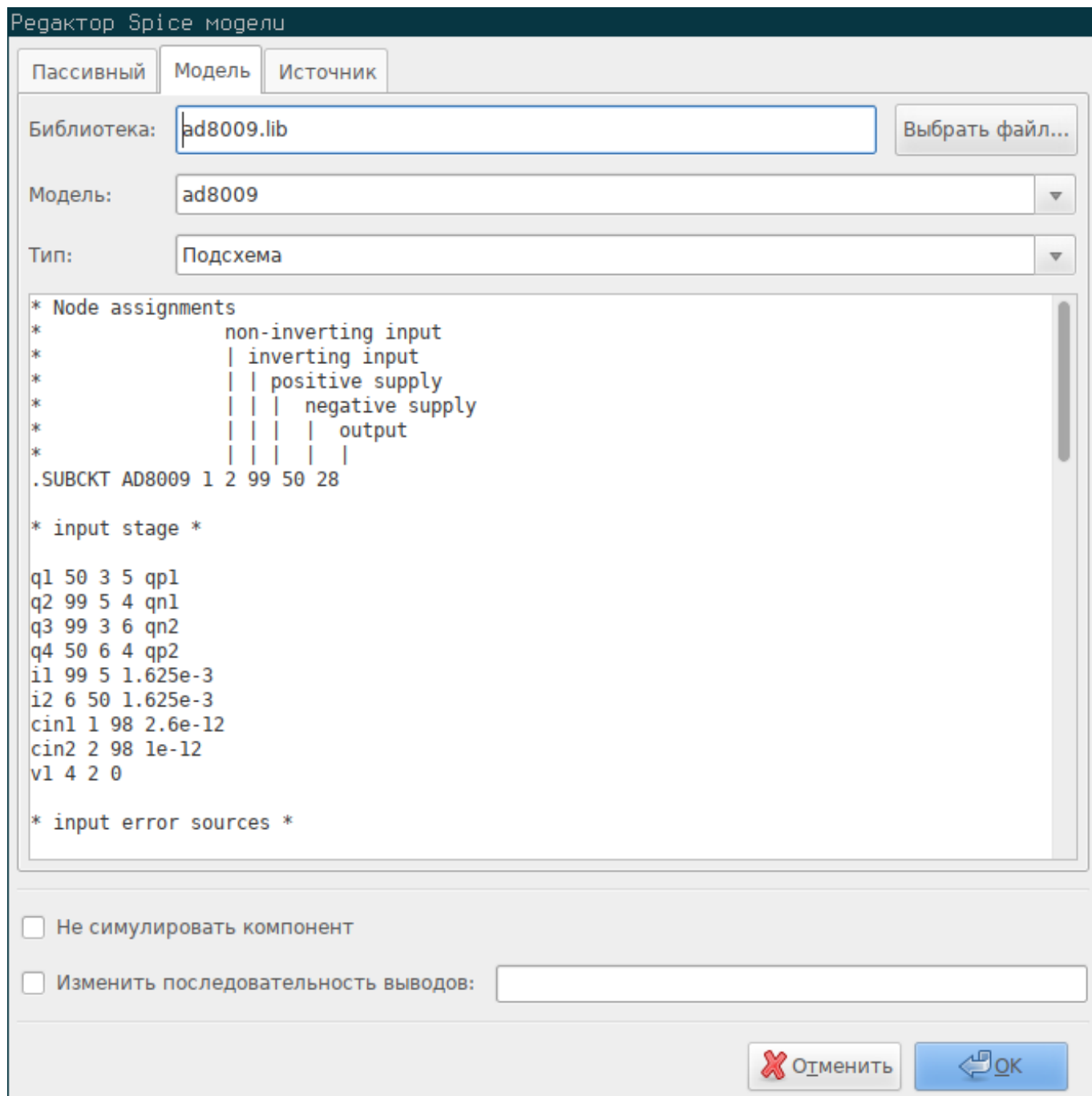
 

Тип	Выбор типа компонента (резистор, конденсатор или индуктор).
Значение	Определяет свойства компонента (сопротивление, ёмкость или индуктивность). В значении могут использоваться приставки, которые используются в описании Spice-моделей (они перечислены под текстовым полем ввода). В качестве десятичного разделителя необходимо использовать точку. Обратите внимание, Spice неверно воспринимает значения, в которых приставка выступает в роли десятичного разделителя (например: 1k5).

16.1.2 Модель

Вкладка *Модель* используется для назначения полупроводниковых или сложных моделей, которые определены в отдельном файле библиотеки. Библиотеки Spice-моделей часто предоставляются производителями компонентов.

В основном текстовом поле отображается содержимое выбранного файла библиотеки. Обычно в библиотеке дополнительно содержится описание моделей и порядок их узлов.



Библиотека	Путь к файлу Spice-библиотеки. Этот файл будет использоваться симулятором, он подключается с помощью директивы <i>.include</i> .
------------	--

Модель	Выбранная модель компонента. После выбора файла библиотеки, список заполняется доступными моделями для выбора.
Тип	Выбранный тип модели (макромодель (subcircuit), BJT, MOSFET или диод). Обычно тип устанавливается автоматически при выборе модели.

16.1.3 Источник

Вкладка *Источник* используется для настройки моделей питания или источников сигналов. Здесь имеется две секции: *Анализ DC/AC* и *Анализ переходных процессов*. В каждой задаются параметры источника для соответствующего типа симуляции.

Параметр *Тип источника* применяется ко всем типам симуляций.

Редактор Spice модели

Пассивный | Модель | **Источник**

Анализ DC/AC:

DC: В/А

Величина AC: В/А Фаза AC: рад

Анализ переходных процессов:

Импульс | Синусоидальный | Экспоненциальный | Кусочно-линейный

Начальное значение: В/А

Импульсное значение: В/А

Время задержки: с

Время нарастания: с

Время спада: с

Ширина импульса: с

Период: с

Тип источника:

Напряжение Ток

Не симулировать компонент

Изменить последовательность выводов:

Для получения более подробной информации об источниках обратитесь к разделу 4 (Источники напряжения и тока) документации [ngspice](#).

16.2 Директивы Spice

Существует возможность добавлять Spice-директивы, размещая их в текстовых полях на листе схемы. С помощью этого свойства удобно указывать текущий тип симуляции. Однако, функционал ограничен, можно использовать только Spice-директивы, которые начинаются с точки (например: ".tran 10n 1m"). Нельзя добавлять компоненты в виде их описания в текстовых полях.

16.3 Симуляция

Для запуска симуляции откройте окно *Spice симулятора* через меню редактора схем *Инструменты*→*Симулятор*.

Spice симулятор

Файл Симуляция Вид

Запустить/Остановить симуляцию Добавить сигналы Пробник Подстройка Установки

Plot1 X

Voltage Current

1.5V 1.2V 0.9V 0.6V 0.3V 0.0V -0.3V -0.6V -0.9V -1.2V -1.5V

1ms 2ms 3ms 4ms 5ms 6ms 7ms 8ms 9ms 10ms

Time

Сигналы

Сигнал

V(/rect_out)

V(/signal_in)

Курсоры

Сигнал	Time	Voltage / Cur
V(/rect_out)	4.33131m	802.12m

Подстройка

C1 X R2 X

200n 200k

100n 100k

50n 50k

Сохранить Сохранить

Circuit: KiCad schematic

Error on line 4 :

d1 net_d1-pad2_ /rect_out ln4148

Unable to find definition of model ln4148 - default assumed

Doing analysis at TEMP = 27.000000 and TNOM = 27.000000

Warning: v1: no DC value, transient time 0 value used

Initial Transient Solution

```
-----
Node                Voltage
-----
/signal_in          0
net_d1-pad2_        5,76787e-28
/rect_out           -5,76787e-26
v1#branch           -5,76787e-31
Reference value : 0,000000e+00
-----
```

Окно разделено на несколько секций:

- [Панель инструментов](#)
- [Панель графиков](#)
- [Сообщения консоли](#)
- [Перечень сигналов](#)
- [Перечень курсоров](#)
- [Панель подстройки](#)

16.3.1 Меню

16.3.1.1 Меню "Файл"

Новый график	Создать новую вкладку на панели графиков.
Открыть работу	Открыть перечень сигналов для построения графиков.
Сохранить работу	Сохранить перечень сигналов, построенных на графике.
Сохранить как изображение	Экспорт текущего графика в png-файл.
Сохранить как файл .csv	Экспорт данных текущего графика в csv-файл.
Выйти из симуляции	Закрыть окно симулятора.

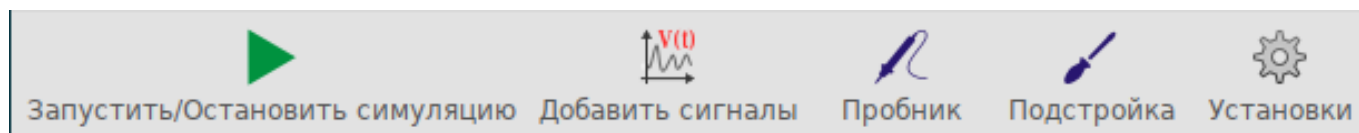
16.3.1.2 Симуляция

Запустить симуляцию	Выполнить симуляцию используя текущие настройки.
Добавить сигналы...	Открыть диалог для выбора сигналов, графики которых нужно построить.
Пробник из схемы	Запустить инструмент для работы со схемой Пробник .
Подстроить значение компонента	Запустить инструмент Подстройка .
Показать Spice список цепей...	Открыть окно, в котором будет показан список цепей для исследуемой схемы.
Настройки...	Открыть диалоговое окно настройки симуляции .

16.3.1.3 Меню "Вид"

Увеличить	Увеличить масштаб текущего графика.
Уменьшить	Уменьшить масштаб текущего графика.
Вписать в экран	Подстроить масштаб графика так, чтобы он полностью поместился на экране.
Показать сетку	Показать или скрыть сетку.
Показать надписи	Показать или скрыть надписи графика.

16.3.2 Панель инструментов



Эта панель предоставляет доступ к часто используемым командам.

Запустить/остановить симуляцию	Начать или остановить работу симулятора.
--------------------------------	--

Добавить сигналы	Открыть диалоговое окно для выбора сигналов, графики которых нужно построить.
Пробник	Активировать инструмент для работы со схемой Пробник .
Подстройка	Активировать инструмент Подстройка .
Установки	Открыть диалоговое окно настройки симуляции .

16.3.3 Панель графиков

Результат симуляции отображается в виде графиков. Можно создать несколько графиков на отдельных вкладках, но только текущий график будет обновляться при выполнении симуляции. Это позволяет сравнивать результаты симуляции при разных условиях.

Вид графиков можно настроить включая/выключая сетку или надписи через меню [Вид](#). Если надписи отображаются, их положение можно менять, перетаскивая левой кнопкой мыши.

Управление панелью графиков:

- колесо мыши увеличивает или уменьшает масштаб
- правая кнопка мыши открывает контекстное меню для настройки изображения
- выделите прямоугольную область для её увеличения во весь экран
- перетащите курсор для изменения координат

16.3.4 Сообщения консоли

Сообщения консоли — это сообщения от симулятора. Рекомендуется следить за этими сообщениями и проверять наличие ошибок или предупреждений.

16.3.5 Перечень сигналов

Содержит список сигналов, которые отображены на графике.

Управление перечнем сигналов:

- щелчок правой кнопки мыши открывает контекстное меню для скрытия сигналов или включения/выключения курсора
- двойной щелчок мыши скроет сигнал

16.3.6 Перечень курсоров

Содержит список курсоров и их координаты. Для каждого сигнала может быть показан свой курсор. Видимость курсора включается/выключается в перечне [Сигналов](#).

16.3.7 Панель подстройки

Здесь содержатся компоненты, выбранные с помощью инструмента [Подстройка](#). Панель подстройки позволяет пользователю быстро изменять параметры компонента и наблюдать влияние на результат симуляции — каждый раз при изменении значения компонента, симуляция запускается заново и график обновляется.

У каждого компонента на панели имеется несколько элементов управления:

- В верхнем текстовом поле устанавливается максимальное значение.
- В среднем текстовом поле устанавливается текущее значение.
- В нижнем текстовом поле устанавливается минимальное значение.
- Ползунок позволяет пользователю плавно менять значение компонента.
- Кнопка *Сохранить* записывает текущее значение, выбранное с помощью ползунка, в поле значения компонента на схеме.
- Кнопка *X* удаляет компонент из панели подстройки и восстанавливает его начальное значение.

В трёх текстовых полях можно использовать множители единиц измерения в формате Spice.

16.3.8 Подстройка

Инструмент подстройки позволяет пользователю выбрать компоненты для настройки его значения.

Чтобы выбрать компонент для подстройки, после активации инструмента щёлкните левой кнопкой мыши на компоненте в редакторе схем. Выбранные компоненты появятся на панели [подстройка](#). Подстраивать можно только пассивные компоненты.

16.3.9 Пробник

Инструмент "Пробник" предоставляет простой путь для выбора сигналов, которые нужно отобразить на графике.

Чтобы добавить сигнал на график, активируйте инструмент и нажмите левой кнопкой мыши на интересующем проводнике в редакторе схем.

16.3.10 Параметры симуляции

Настройки симулятора

AC DC переход Переходной процесс Пользовательская

Временной шаг: с

Конечное время: с

Начальное время: с (опционально; по умолчанию 0)

Подстроить значения пассивных компонентов (например М -> Meg; 100 nF -> 100n)

Добавить полный путь для директивы .include библиотека

Отменить ОК

Диалоговое окно параметров симуляции позволяет пользователю выбрать тип симуляции и её параметры. Здесь имеется четыре вкладки:

- AC
- DC переход
- Переходной процесс
- Пользовательская

На первых трёх вкладках имеются формы, в которых можно указать параметры симуляции. Последняя вкладка позволяет пользователю настроить процесс симуляции путём ручного ввода Spice-директив. Более подробную информацию о типах симуляции и их параметрах можно найти в разделе 1.2 [документации ngspice](#).

Альтернативный путь настройки процесса симуляции — ввести [Spice-директивы](#) в текстовых полях на схеме. Любые директивы указанные в текстовых полях и связанные с типом симуляции переопределяются параметрами, выбранными в диалоге настроек. Это значит, что после ввода параметров в диалоге настроек, эти параметры будут использоваться вместо директив указанных на схеме до тех пор, пока процесс симуляции не будет запущен заново.

Имеется два параметра, общих для всех типов симуляции:

Подстроить значения пассивных компонентов	Заменяет значения пассивных компонентов, путём преобразования общего формата записи значений в формат Spice.
Добавить полный путь для директивы <code>.include</code> библиотека	Параметр следит за тем, чтобы файл библиотеки Spice-моделей был сохранён в виде полного пути. Обычно ngspice требует использования полного пути для доступа к файлам библиотек.