

Altium Designer Summer09

Практические подходы к организации библиотек и структуры проектов. Библиотека графических изображений компонентов

Автор в ряде статей [1, 2] уже рассказывал, как создавать графические компоненты. Однако в этих публикациях были рассмотрены ранние версии пакета, а само описание создания графики в них не было систематизировано и ограничивалось рассмотрением нескольких примеров. В этой статье приведено более общее описание процесса создания графических компонентов, однако и здесь все равно обсуждаются только те вопросы, которые актуальны для библиотеки, ориентированной для включения в базу данных.

Владимир Пранович,
к. т. н.

v.pranovich@gmail.com

Итак, в [3] мы показали, как сделать новый проект и наполнить его существующими библиотеками. Если вы хотите добавить новую библиотеку графических изображений к проекту, откройте его и примените команду **File=>New=>Library=>Schematic Library**. Так как в примере мы использовали библиотеку от старых проектов, то покажем, как их можно быстро модифицировать для оптимального применения в сочетании с базой данных. Подскажем, как установить глобальные настройки, и отметим нюансы, которые следует учесть при создании сложных или специфических графических элементов.

Особо отметим, что в приведенных библиотеках автор применяет типовой шаг между выводами компонентов, который используется в программе, а при выводе на печать электрических схем устанавливает масштаб 1:2. Это позволяет применять собственные библиотеки для работы на зарубежный рынок и в то же время использовать многие готовые графические изображения компонентов, поставляемые с пакетом, для максимального приближения к требованиям ГОСТ в странах СНГ.

Создание простого графического изображения компонента

Итак, рассмотрим создание простого графического изображения компонента. Представим это на примере компонента для контрольной точки. Такой компонент имеет посадочное место, но не входит в перечень элементов, так как контрольная точка будет создана топологическим способом.

Для этих целей, как описано в [3], мы включили в проект библиотеку **Micelangelo.SchLib** и теперь в ней создадим указанный компонент (рис. 1):

1. Откроем библиотеку **Micelangelo.SchLib**. Затем откроем панель **SCH Library** и настроим ее положение. Это можно сделать через панель **Access**, находящуюся в правом нижнем углу рабочего окна, и используя ее закладку **SCH**, в которой следует

выбрать строку **SCH Library**. Более подробно работа по вызову и настройке панелей описана в [1].

2. В панели **SCH Library** в окне отображения имен графических изображений, используя кнопку **Add**, добавим в нашу библиотеку новый компонент и дадим ему название, например **PIN**.

3. В результате в окне отображения имен графических изображений появится имя нового компонента **PIN**. Двойным щелчком левой кнопки указателя по данной надписи вызовем свойства нужного компонента.

4. В левом верхнем окне определим значения следующих параметров:

A. В поле **Default Designator** введем начальные буквы, которые будут присваиваться компоненту на схеме в качестве позиционного номера. Обязательно поставьте в конце обозначения вопросительный знак (в будущем при помещении графического изображения из библиотеки на схему компилятор не будет выдавать предупреждения при совпадении позиционных обозначений, а при аннотации вопросительный знак будет заменен на первый незанятый порядковый номер в соответствии с выбранным типом нумерации компонентов на схеме).

B. Обязательно установим флаг **Visible** для параметра **Default Designator** — это означает, что позиционный номер данного компонента будет отображен на схеме.

C. Флаг **Locked** снимем, что позволит при аннотации схемы изменять (присваивать новый) порядковый номер **Default Designator** для данного компонента.

D. Поле **Comment** можем оставить незаполненным. Однако это единственный параметр (кроме параметра **Designator**), который всегда передается из схемы в PCB, и его удобно использовать для промежуточной передачи туда значения иных пользовательских параметров. Например, таким параметром может быть наименование компо-

нента или значение номинала резистора. Для этих целей мы введем специальный параметр **ValueSCH**, а здесь, для передачи в будущем именно этого значения параметра в топологический редактор, в поле **Comment** следует ввести значение **=ValueSCH**.

- E. Флаг **Visible** для параметра **Comment** рекомендуем снять. В связи с тем, что такой параметр может и не иметь прямого отношения к схеме, его не следует отображать видимым по умолчанию.
- F. Сейчас мы создали простой компонент, состоящий из одной неделимой части графического изображения, при этом кнопки «<<, <, >, >>» будут недоступны, а единственная часть всегда отображается под номером **Part1/1**. Соответственно, для данной части флаг **Locked** не имеет смысла, и его снимем. Назначение этого флага покажем на примере сложных составных компонентов далее.
- G. Поле **Description** можно оставить незаполненным, так как этот параметр мы будем вводить, используя параметры базы данных.
- H. В поле **Type** следует указать тип данного компонента. В нашем случае это компонент для электрической схемы, который будет иметь посадочное место, но не должен входить в перечень элементов, и мы выбираем значение **Standard (No BOM)**. Программа поддерживает следующие типы компонентов:
- **Standard** — стандартный (типовой) компонент для электрической схемы, который имеет посадочное место и должен включаться в перечень элементов.
 - **Mechanical** — компонент для изображения элементов, не имеющих электрических связей. Имеет посадочное место и включается в перечень элементов.
 - **Graphical** — компонент для изображения элементов, не имеющих ни электрических связей, ни посадочного места. Как правило, используется для улучшения «читаемости» схемы.
 - **Net Tie (in Bom)** — компонент для изображения элементов, объединяющих различные электрические связи. В качестве примера такого элемента может служить перемычка.
 - **Net Tie** — такой же компонент при выполнении перемычки топологическим способом, который не требует включения данного компонента в перечень элементов.
 - **Standard (No BOM)** — стандартный (типовой) компонент для электрической схемы, который, однако, следует исключить из перечня элементов.

5. Окно **Library Link**, параметр **Symbol Reference**. Здесь при необходимости мы можем изменить имя нашего компонента. Заметим, что мы описываем создание библиотечного компонента для библиотеки из базы данных. В базе данных необходимо будет в соответствующем поле указать именно это значение, если вы хотите подключить его

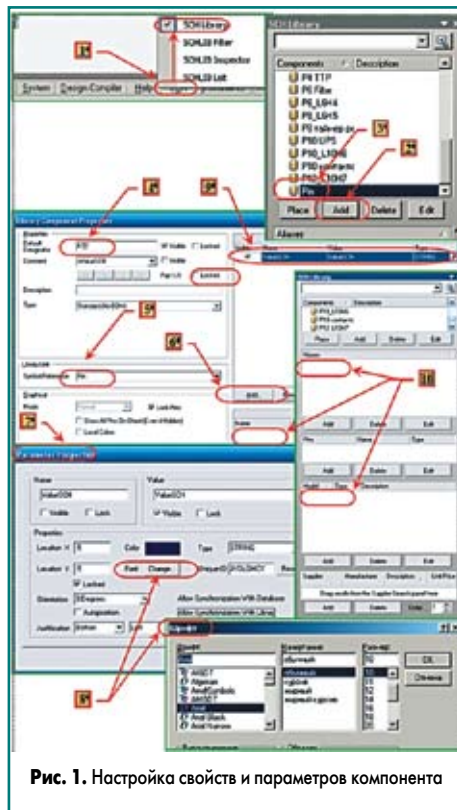


Рис. 1. Настройка свойств и параметров компонента

графическое изображение. Окно **Graphical mode**. Здесь оставим все, как указано по умолчанию, а именно:

- A. Параметр **Mode** для этого простого компонента недоступен. Он станет доступен только при создании альтернативных графических изображений компонента.
- B. Флаг **Lock Pin** установлен, что означает запрет редактирования свойств выводов компонента на схеме.
- C. Флаг **Show All Pins On Sheet (Even if Hidden)** снят, что означает необходимость отображения всех **Pin** (выводов) компонента, даже если они имеют свойство **Hidden** (скрытый или невидимый). В нашем случае компонент имеет единственный видимый вывод, и нет необходимости его устанавливать.
- D. Флаг **Local Color** снят. В этом примере мы не будем указывать, как настраивать цветовую гамму для выводов.
6. Перейдем в окно пользовательских параметров компонента и добавим к компоненту параметры, для которых следует определить положение, формат надписи или значение по умолчанию:
- **ValueSCH** — для отображения на схеме названия компонента или значения величины его основного параметра.
 - **NoteSCH** — для отображения на схеме, если требуется, вспомогательного параметра компонента.
 - **ValueBOM** — для формирования значения записи в графе «Наименование» перечня элементов.

Других параметров для компонентов библиотеки определять не будем, так как все необходимые параметры будут добавлены на схему через базу данных.

7. Для каждого параметра следует настроить вид его отображения на схеме, что покажем на примере параметра **ValueSCH**. Такую на-

стройку удобно сделать, используя панель **Parameter Properties** («Свойства параметра»), которая открывается при добавлении параметра или вызывается двойным щелчком на строке с нужным параметром. В данном окне для указанного параметра сделаем следующие настройки:

- A. В окне **Name** вводим или редактируем название параметра. Нет смысла отображать на схеме название самого параметра (**ValueSCH**), и поэтому флаг **Visible** не устанавливаем. Защитим имя нашего параметра от изменений, установив флаг **Lock**.
- B. Соответственно, в окне **Value** определим значение параметра по умолчанию (например, просто укажем то же имя, что у самого параметра **ValueSCH**); установим флаг **Visible**, так как эта величина всегда будет отображаться на схеме; снимем флаг **Lock**, что позволит при необходимости редактировать параметр непосредственно на схеме.
- C. В окне **Property** («Свойства») положение параметра не будем указывать, это проще сделать при визуальном перемещении параметра в нужную точку. Здесь настроим только следующие свойства:

- Снимем флаг **Locked**, тем самым разрешим редактирование параметра в части перемещения, изменения шрифта и т.д.
- Снимем флаг **Auto position**, тем самым запретив автоматическое размещение параметра. Такая установка позволяет корректировать положение параметра на схеме вручную (так всегда можно указать оптимальное его расположение), и это положение сохраняется относительно графики компонента при перемещении последнего.
- Установим направление надписи от точки привязки: **Justification=Bottom+Left** (точка привязки будет находиться снизу и слева от текста параметра).
- Установим флаг **Allow Synchronization With Database**. Отметим, что данный параметр всегда получает значение из базы данных сразу же при помещении его на схему из библиотеки. Однако, поскольку мы создаем библиотеку для базы данных, рекомендуем все же этот флаг установить.
- Снимем флаг **Allow Synchronization With Library**. Мы делаем библиотеку для базы данных. Непосредственный в библиотеке графических изображений хранить параметры не будем. По этой причине необходимо запретить синхронизацию этого параметра со значением аналогичного параметра в библиотеке **SCH Library**, так как там мы ввели значение по умолчанию (**ValueSCH**), а оно никак не связано с реальным значением этой величины на схеме.

8. Теперь установим шрифт для отображения значения параметра, для чего следует нажать кнопку **Change** напротив надписи **Font**. В окне «Шрифт» выберем тип, на-

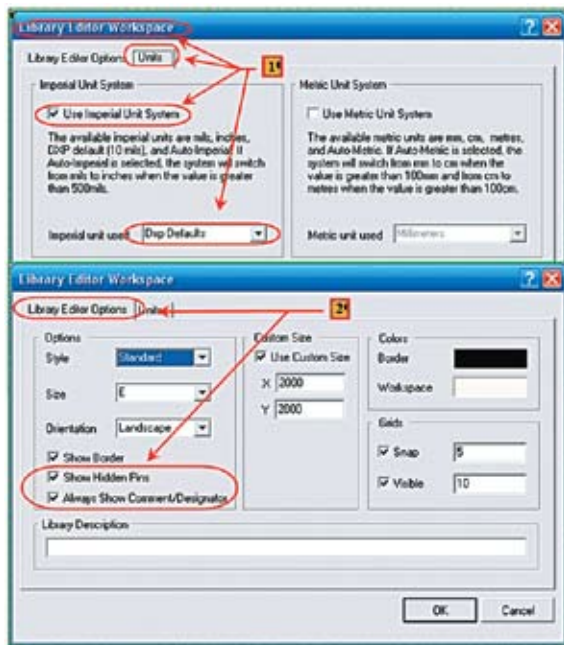


Рис. 2. Параметры библиотеки

чертание и размер шрифта. Практика показала: чтобы использовать стандартные настройки пакета и при этом применять созданные графические компоненты, как для стран СНГ, так и для зарубежья, удобно установить размер шрифта равным 10, а саму схему печатать в масштабе 1:2. В этом случае размер надписей будет соответствовать требованиям ГОСТ.

9. После указанных действий добавленный параметр будет размещен в окне пользовательских параметров. Укажем отличия от предыдущего в настройках двух других параметров:

- **NoteSCH.** Здесь вводим значение по умолчанию **NoteSCH**, а поскольку параметр по умолчанию не будет отображаться на схеме, флаг **Visible** лучше снять. Если все же возникнет необходимость отобразить этот параметр на схеме, там же можно установить этот флаг и ввести требуемое значение параметра.
- **ValueBOM.** Здесь иная особенность: значение этого параметра должно содержать сведения для столбца «Наименование» перечня элементов, которые должны включать краткое описание комплектующего изделия, полное его наименование и указание производителя. Для данного компонента это неактуально, однако значение данного параметра тоже введем, и в этом случае оно будет постоянно для всех компонентов во всех библиотеках. Обозначим еще одну проблему. На момент создания графического компонента мы не знаем ничего ни о его производителе, ни о его обозначении. Эти сведения будут содержаться в базе данных как значения нескольких параметров, здесь же их нужно объединить в одном. Решение этой проблемы такое: в качестве значения параметра введем текст `=ShortDescription+',' +VendorPN+',' +Vendor`. То есть значение

данного параметра будет состоять из текста трех других параметров, разделенных запятыми. Это параметры базы данных: **ShortDescription** — краткое описание компонента; **VendorPN** — обозначение компонента по производителю, **Vendor** — название производителя комплектующего изделия. Соответственно, флаг **Visible** для значения параметра снят, флаг **Allow Synchronization With Database** тоже снят, поскольку такого параметра в базе данных не будет, флаг **Allow Synchronization With Library** установлен, что позволит обновлять его из библиотеки. Заметим, что при синхронизации с библиотекой обновляться будет только сама запись, указанная выше, сами же параметры будут подставляться те, что указаны в записи.

10. Окно подключения моделей к графическому изображению компонента. Данное поле оставляем незаполненным, так как все ссылки на модели будем указывать только в базе данных. Соответственно, данное поле останется пустым и в панели **SCH Library**. Теряет также смысл и создание ссылки (**Alias**) на данное изображение.

Итак, свойства и параметры компонента определены, теперь следует создать его графическое изображение. Вначале следует настроить параметры самой библиотеки (рис. 2). Откроем (**Tool>>Document Option**) окно **Library Editor WorkSpace**. На вкладке **Units** определим размерность единиц, которая будет использоваться при создании графики. Рекомендуем использовать значение **Default DXP units**, что соответствует **10 mils** (2,54 мм). Именно эту сетку используют все библиотеки, которые доступны на сайте [4], и в этом случае для многих компонентов можно брать готовую графику оттуда.

1. На вкладке **Library Editor Option** установим или изменим следующие свойства:

А. В окне **Option** параметры **Style**, **Size**, **Orientation** не изменяем. Они влияют только на размер и вид поля для отображения компонента. Это касается и флага **Show Border** отображения границ листа. Если вид компонента нужно вывести на печать в рамке, вы сможете настроить эти параметры по аналогии, как это мы укажем в продолжении, при рассмотрении схемного редактора.

В. Рекомендуем установить флаг **Show Hidden Pin**. Это позволит легко задать в библиотеке местоположение скрытых выводов компонента. Желательно эти выводы расположить внутри внешних границ компонента: несмотря на то, что они будут скрыты на схеме, может возникнуть ситуация, при которой их следует отобразить, а такое их положение позволит легко найти расположение скрытого вывода.

С. В окне **Custom Size** установим флаг **Use Custom Size**. Это позволит определить собственные размеры листа для библиотечного элемента. Сами значения размеров рекомендуем изменить в том случае, если графика компонента не помещается на существующем листе.

Д. В окне **Color** вы можете определить собственные цвета для границы (**Border**) листа и его рабочей зоны (**WorkSpace**).

Е. В окне **Grid** следует задать сетку (**Grid**) для привязки элементов компонента и сделать отображение (**Visible**) сетки на рабочей зоне. Здесь рекомендуем поставить значения 5 и 10 (это в **Default DXP units = 10 mils** или, соответственно, 1,27 и 2,54 мм). Такая установка позволяет легко строить компоненты с шагом выводов 2,54 мм, не используя перенастройку сетки. При печати в масштабе 1:2 мы получим шаг выводов, близкий к 5 мм, что соответствует требованиям ГОСТ. В то же время именно эту сетку используют все библиотеки, которые доступны на сайте [4].

Ф. Соответственно, в окне **Library Description** вы можете дать краткое описание библиотеки.

2. Настроим все библиотеки так, чтобы на листе отображались такие параметры компонента, как **Designator** («Позиционный номер») и **Comment**.

Перейдем непосредственно к графике компонента (рис. 3).

1. Командой **Place>>Pin** расположим на рабочем поле единственный вывод нашего компонента (контрольная точка). Откроем окно свойств вывода и установим следующие настройки:

А. В данном случае и номер **PIN** (вывода), и его имя будут иметь одинаковое значение — мы присвоим значение «Единица».

В. Так как у данного компонента вывод один, нет смысла отображать и номер, и вывод, флаг отображения видимости напротив них снимем.

С. В поле **Electrical Type** установим тип **Passive**. Практически во всех случаях будет стоять именно этот тип, и если вы не знаете, какой тип назначить, рекомендуем использовать именно такой.

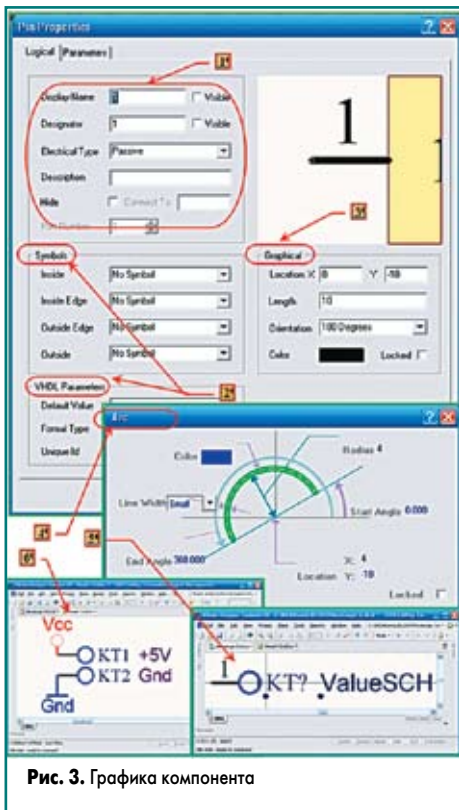


Рис. 3. Графика компонента

- D. Поле **Description** оставим незаполненным.
- E. Флаг **Hide** (скрытый вывод) снимем, соответственно, поле **Connect to** (чтобы указать подключение скрытого вывода к электрической цепи) станет недоступным.
- F. Поле **Part number** также нам недоступно, поскольку мы создали простую графику компонента, состоящую из единственной неделимой части.
- В области **Symbol** находятся поля, которые определяют вид вывода для данного компонента. Здесь во всех полях установим значение **No Symbol**, что означает простой вывод. Поля в области **Vhdl Parameter** оставим незаполненными (эти поля определяют параметры для аппарата средств симуляции электрических сигналов на схеме, а мы этот аппарат использовать не будем).
 - В области **Graphical** поля **Location** и **Orientation** заполнять не следует, так как положение вывода мы зададим его перемещением в нужную зону. Длину (**Length**) вывода рекомендуем установить равной 10. Это удобно, когда нет отображения номера вывода или номер состоит из одного символа, для двухсимвольного обозначения — 20, трехсимвольного — 30 (напомним, выше мы установили единицу измерения в библиотеке **Default DXP units = 10 mils**). Здесь же можно задать цвет вывода и зафиксировать его положение от перемещения и редактирования на схеме.
 - Командой **Place>>Arc** введем графическое изображение тела компонента, которое будет представлять в данном случае замкнутую окружность. Открыв окно свойств окружности, установим следующие ее параметры:
 - Цвет окружности.
 - Ширину линии. Здесь рекомендуем установить из 4 доступных значений величину **Small**, что будет наиболее адекватно выглядеть на схеме.

- Начальный (0.0) и конечный (360.0) угол для разворота окружности.
 - Значение радиуса (4). Такое значение позволит расположить в сетке 10 два рядом стоящих подобных компонента, и они не будут касаться друг друга.
 - Положение центра окружности (хотя это можно сделать и переместив окружность на то место, где она должна находиться в компоненте).
 - Ну и наконец, следует зафиксировать ее — для защиты от случайного перемещения.
5. Теперь расставим отображаемые параметры и саму графику компонента. Предвосхищая создание будущих компонентов, имеющих несколько графических изображений или состоящих из нескольких частей, заметим, что положение данных параметров, к сожалению, не устанавливается индивидуально для любой составной части, а едино для всех, относительно точки привязки листа. Поэтому рекомендуем сначала, например, в центре, разместить обозначение компонента и другие параметры, а затем относительно него расположить графику.
6. Для примера на рис. 3 показан внешний вид данного компонента уже на схеме. Заметим, что здесь:
- Установлены настройки листа схемы, при которых неотображаемые параметры выводов не видны.
 - В качестве значения параметра **ValueSCH** из базы данных введен пробел, и это не видно на схеме.
 - Значения параметра **NoteSCH** введены непосредственно на схеме.
 - Схема подготовлена к печати, и чтобы не отображалась точка привязки параметров, в настройках **Tools>>Schematic**

Preference>>Graphical Editor снят флаг **Mark Manual Parameter**.

Типовое графическое изображение компонента

Теперь покажем процесс создания типового графического изображения компонента для электрической схемы. Для примера выберем транзистор MC13202 для ZigBee-приложений. Сначала ищем описание данного транзистора [5] и на странице 15 открываем таблицу **Pin Function Description**, где указаны номер и имя выводов компонента, а также их назначение. Этой информации достаточно для создания графического изображения компонента. Не станем приводить все действия — они перечислены выше при описании процесса создания простого изображения, а отметим только новые и специфические, свойственные данному компоненту. Итак, аналогично вышеописанному, добавим в библиотеку новый компонент, однако установим в поле **Type** значение **Standard**, что означает создание типового компонента. Введем для нового компонента такие же пользовательские параметры, какие мы установили при создании простого графического изображения выше. Далее поступим следующим образом (рис. 4):

- При добавлении выводов компонента первым присвоим значение «Единица» — и для номера, и для имени. В случае если у вас в настройках указан инкремент данных значений (**Tools>>Schematic Preference>>General: Auto-increment During Placement**), каждый следующий вывод получит новый порядковый номер, который будет больше предыдущего на величину, указанную в данной настройке.
- Аналогично расставим выводы компонента, например, так, как указано на рис. 4, и сделаем графику компонента, используя команду **Place Polygon**.

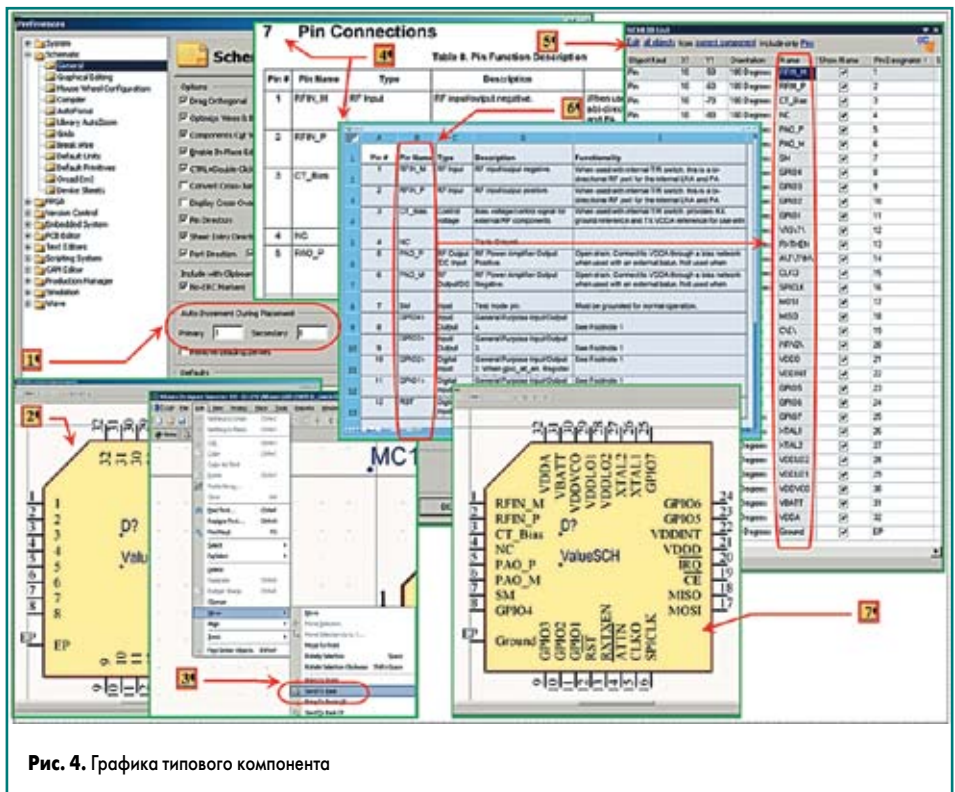


Рис. 4. Графика типового компонента

3. Отметим, что вам доступна правка контура, границы заливки и прозрачности полигона. Однако использовать прозрачный контур полигона следует с осторожностью, так как не все печатающие устройства поддерживают такую функцию. Это, в частности, проявляется при подготовке документов в формате **Acrobat**. В случае использования сплошной заливки полигона или других фигур следует обязательно определить порядок слоев, иначе на схеме некоторые надписи или выводы могут быть скрыты полигоном и не отображаться.

4. Теперь следует ввести имена всех выводов микросхемы, а желательно еще и их назначение. Первое можно сделать так же, как было показано выше для простого примера, то есть путем ввода текста, однако гораздо проще скопировать это название из текста описания (MC13202.pdf) и вставить в соответствующем окне свойств вывода. Для предложенного компонента это возможно, но в случае, когда число выводов компонента больше сотни, этот процесс вызывает определенное неудобство, поэтому покажем, как это можно сделать быстро, сразу для всех выводов и безошибочно. В случае если таблица выводов в описании документа проста и небольшого размера, достаточно выделить в ней нужные столбцы и скопировать их в буфер. В противном случае рекомендуем таблицу предварительно преобразовать и сохранить, например, в формате Excel.

5. Теперь откроем панель **SCHLibList**. Настроим список отображаемых компонентов. Для этого в верхней командной строке панели установим следующие значения:

- **Edit all object** (возможность редактирования всех объектов списка).
- **From current component** (в список войдут объекты открытого компонента библиотеки).
- **Include only Pins** (отображаемый список будет включать только объекты типа **Pins**, а именно выводы компонента).

После этого найдем в таблице столбец **Pin Designator** и, нажав указателем на данный заголовок, отсортируем объекты (выводы компонента) по порядковому номеру.

6. Копируем в буфер содержимое столбца **Pin Name** из таблицы Excel, устанавливаем указатель в первую строку столбца **Name** на панели **SCHLibList** и вставляем содержимое буфера. В итоге все имена выводов получат необходимое значение. Таким же образом можно перенести и содержимое других столбцов. Заметим: если в таблице на панели **SCHLibList** расположить столбцы в таком порядке, чтобы они соответствовали столбцам из таблицы Excel, то можно скопировать часть или всю таблицу сразу.

7. Здесь представлен конечный вид компонента. Указанный подход существенно облегчает создание многовыводных компонентов, а самое главное — минимизирует ошибки, которые неизбежно возникают при ручном наборе всех значений.

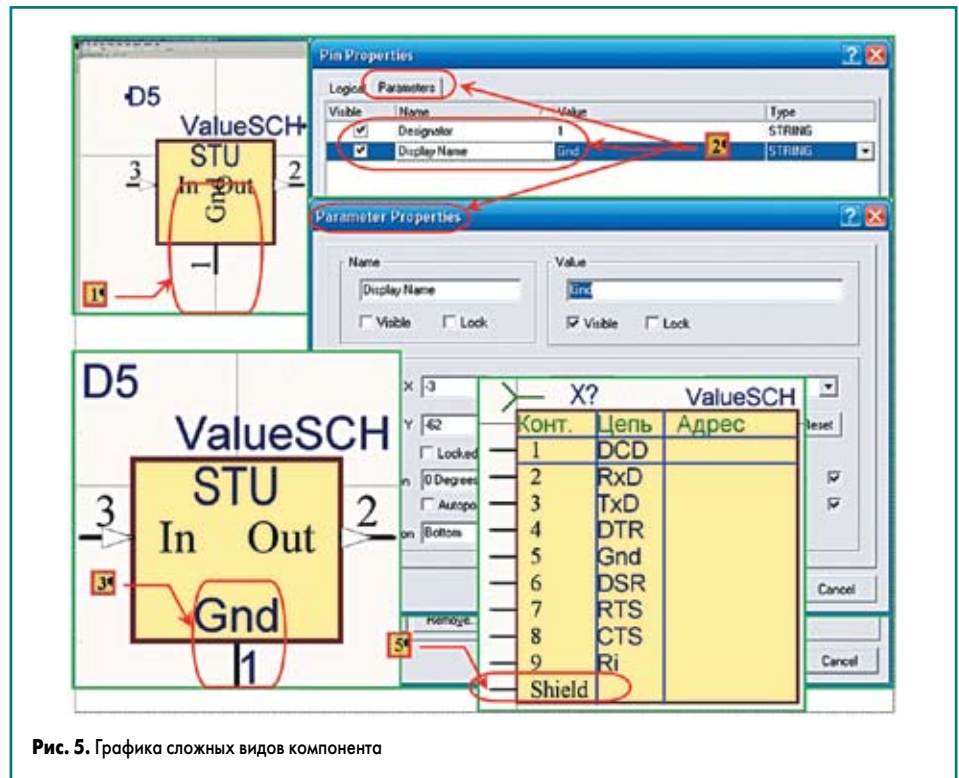


Рис. 5. Графика сложных видов компонента

Сложное графическое изображение компонента

Существует несколько вариантов сложных графических изображений, требующих иных подходов, например, такие, где требуется:

- Необходимость в вертикальном расположении вывода, однако при этом следует надпись номера вывода и его отображение расположить горизонтально.
- Возможность применения на схеме надписей с возможной ориентацией текста во всех четырех направлениях.
- Особый случай графики соединителя, требующий табличного представления дополнительной информации.
- Возможность при изображении простых полупроводниковых элементов номера выводов присваивать не по номеру соответствующей контактной площадки, а по названию самого вывода в графическом обозначении. Решение этой проблемы потребует дополнительно и создания специальных посадочных мест. Вопрос создания таких посадочных мест будет описан в соответствующем разделе продолжения данной статьи.
- Иные нестандартные решения графических компонентов.

Вначале рассмотрим графическое изображение линейного стабилизатора (рис. 5):

1. Очевидно, что надписи общего вывода данного компонента желательно развернуть на 90°. Настройки по отображению не дают такой возможности, поэтому мы сделаем это иным способом. Настроим редактор таким образом, чтобы разрешить положение текстовых надписей во всех четырех направлениях. Это можно сделать, установив флаг **Tools>>Schematic Preference>>Graphical Editor: Display String As Rotated**. Такая настройка обеспечит идентичное расположе-

ние надписей как при установленном, так и при снятом флаге.

2. Редактор графических изображений компонентов имеет только одну настройку положения надписи номера и имени вывода компонента: **Tools>>Schematic Preference>>General: Pin Margin**. Это позволяет настроить отступ начала соответствующего текста. Более того, это единая настройка для всех компонентов библиотеки и схемы. Рекомендуем в таком случае поступить следующим образом:

А. Во-первых, в параметрах для данного вывода (окно **Pin Property**, вкладка **Logical**) снимем флаг **Visible** (сноска 1, рис. 3) и таким образом скроем отображение имени и номера вывода.

В. Во-вторых, откроем вкладку **Parameter** и добавим два параметра, которые имеют значения, идентичные номеру вывода и его имени, например, **Designator** = «1», **Display Name** = «GND». Соответственно, для каждого параметра настроим шрифт, цвет и остальные свойства так, как показано на сноске 7 рис. 1.

3. Расположим новые параметры на графике компонента так, как нам удобно.

4. Здесь представлено изображение соединителя, выполненного совместно с таблицей в соответствии с пунктом 3.41 ГОСТ.2.702, причем все значения таблицы введены в виде параметров вывода соединителя так, как это было сделано выше. В качестве примера использован соединитель (который монтируется на печатную плату) для подключения к COM-порту компьютера. Отметим, что номера выводов в данном компоненте скрыты, при этом имена выводов совпадают с номером, а позиционно находятся в соответствующих строках столбца «Конт.». Надписи столбца «Цепь» — это дополнительные параметры соответствующего

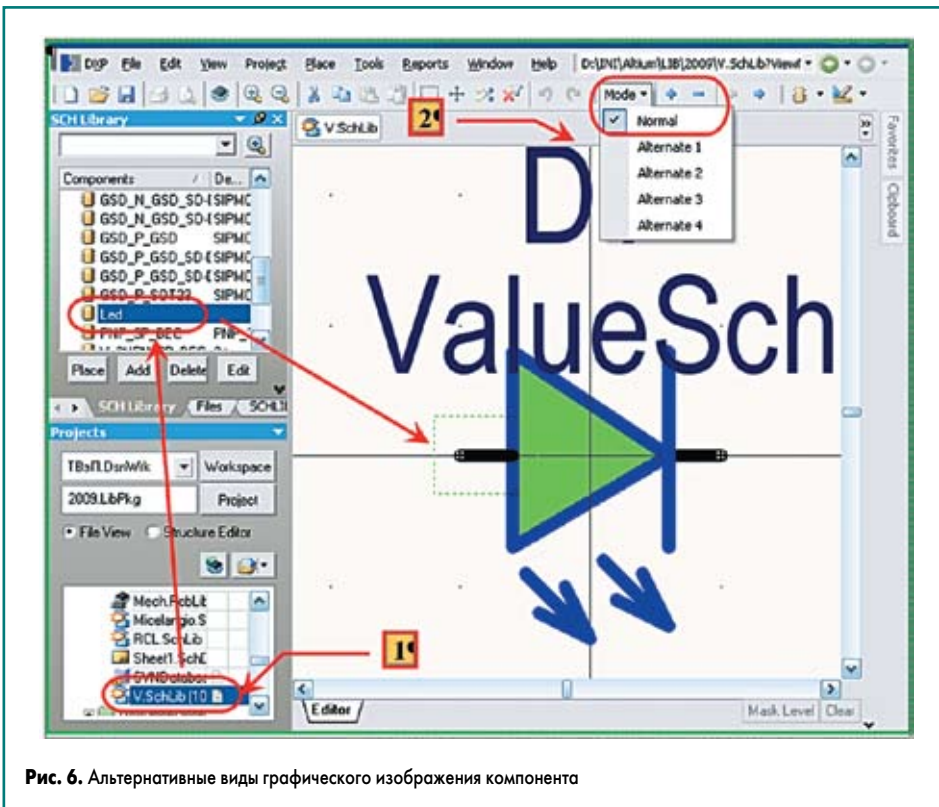


Рис. 6. Альтернативные виды графического изображения компонента

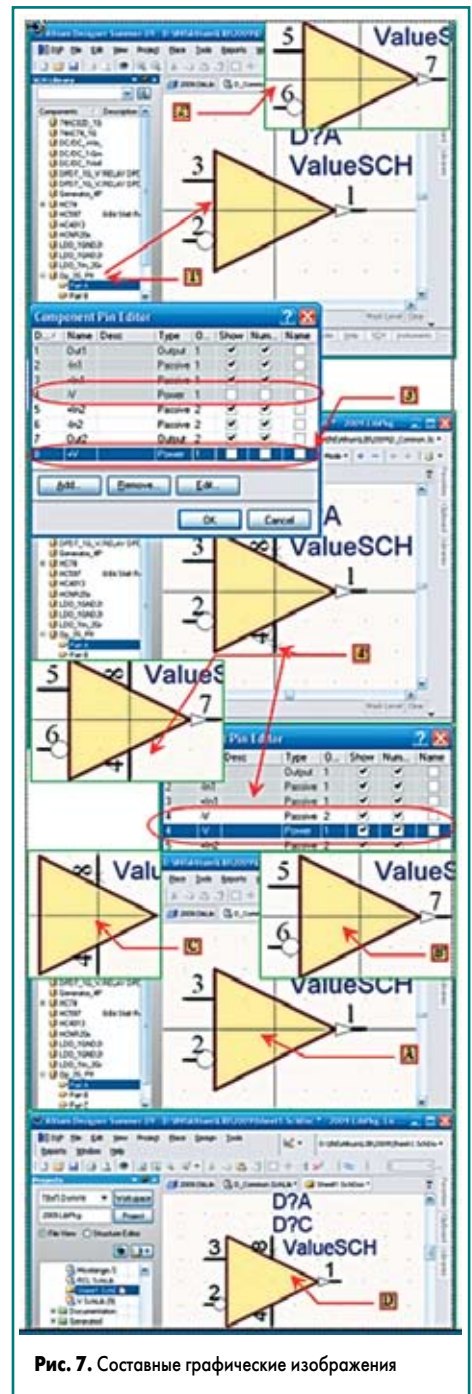


Рис. 7. Составные графические изображения

вывода. Поля столбца «Адрес» должны быть оформлены аналогично, однако в библиотеке мы их не устанавливаем, так как это лучше сделать после установки такого компонента на схему и уже в редакторе схем.

- Обратите особое внимание на то, что данный соединитель должен иметь два отверстия для крепления на печатной плате, а последние могут быть подключены, например, к полигону. Такое подключение должно быть представлено на схеме, а значит, компонент должен содержать соответствующие выводы. Однако в данном случае нет необходимости указывать оба вывода, достаточно один. Отдельно этот вопрос мы также будем рассматривать далее на примере создания посадочного места для подобных компонентов. Заметим: подобный вопрос также актуален, например, для клемм, тактовых кнопок, других крепежных компонентов, имеющих несколько контактов, электрически замкнутых в самом компоненте. Применение одного вывода для подключения всех контактных площадок, имеющих электрическое соединение в самом компоненте, — весьма удобный механизм для повышения «читаемости» схемы.

Альтернативные виды изображения

Перейдем к созданию альтернативных графических изображений компонентов. Как правило, такая необходимость возникает, когда желательно иметь возможность выбора различных вариаций изображения компонента для достижения наиболее удобного вида электрической схемы. Процедура добавления альтернативных изображений проста, и мы продемонстрируем ее на примере вариаций изображения светодиода. Однако предварительно отметим,

что при выборе компонента через библиотеку на основе базы данных отображаться на панели **Library** в окне просмотра будет тот вид изображения, которой редактировался последним. Изменить вид изображения на иной альтернативный можно только после размещения компонента на электрической схеме.

Итак (рис. 6):

- Откроем в библиотеке существующий компонент, к которому следует добавить альтернативное изображение. В нашем примере это компонент **Led** из библиотеки **V.SHLIB**.
- Данный компонент уже имеет альтернативные изображения, и список доступных изображений можно посмотреть, используя команду **Tools>>Mode**. В этом же меню находятся команда «Добавить/удалить» альтернативное изображение. При давлении альтернативного изображения рекомендуем скопировать одно из существующих, а затем добавить и отредактировать его в альтернативном виде.

Примечание. Следует иметь в виду, что настройки отображения и расположения параметров действуют на все альтернативные изображения.

Составные графические изображения

И, наконец, рассмотрим вопросы создания альтернативных графических изображений компонентов. Добавление новой части составного компонента производится командой **Tools>>New Part**. Создание графики составной части компонента можно осуществить теми же приемами, что и основную часть. Следует только иметь в виду, что размещение параметров будет одинаковым для всех частей компонента, и это нужно учитывать при создании графики. Несмотря на всю про-

стоту этого процесса, покажем его на примере создания графики двоячного операционного усилителя, которую изобразим в 3 вариантах: из двух частей со скрытыми выводами питания; с отображением выводов питания; из трех частей, где в третьей части компонента будут находиться общие выводы компонента.

Итак (рис. 7):

- Здесь представлена первая часть графического изображения компонента, которую создаем как обычно.
- Командой **Tools>>New Part** добавляем вторую часть, копируем изображение первой части во вторую и корректируем имена и названия выводов.
- Здесь приведена таблица выводов, где, в частности, скрытые выводы отнесены к первой части компонента.
- Автор рекомендует не использовать скрытые выводы — так схему можно оформить

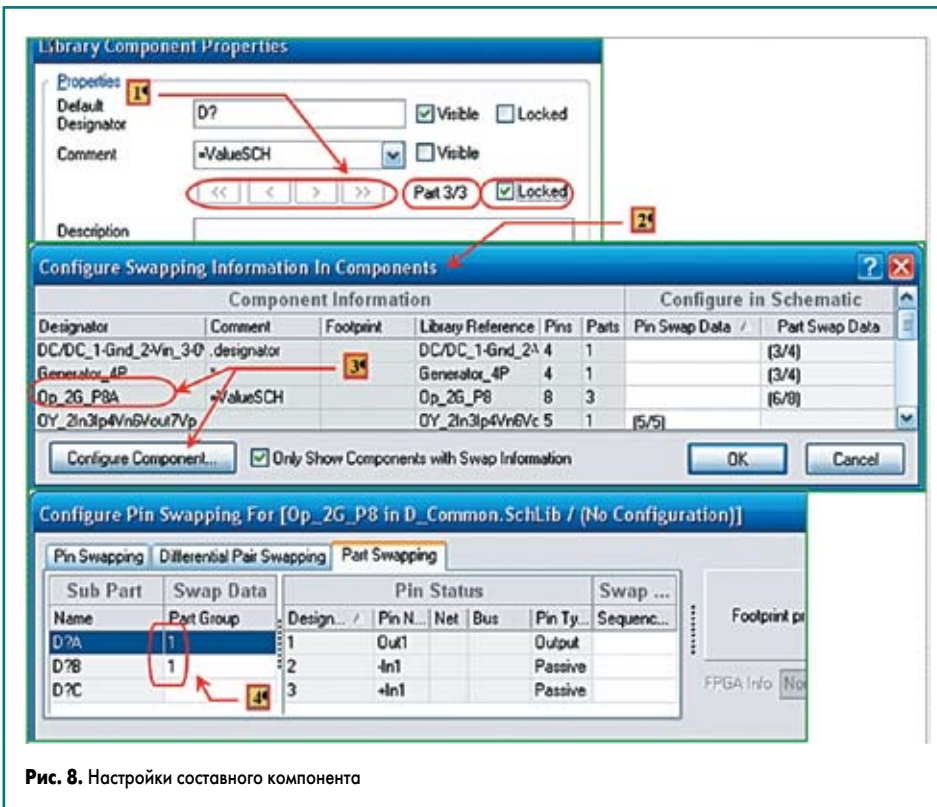


Рис. 8. Настройки составного компонента

2. Применим команду **Tools>>Configure Pin Swapping...** и откроем окно **Configure Swapping Information in Component**.
3. В таблице выберем строку с именем (столбец **Designator**) нашего компонента и нажмем кнопку **Configure Component** (эту операцию можно сделать и с помощью двойного нажатия указателем по нужной строке в столбце).

Примечание. Снимите флаг **Only Show Swap Information**, если вы не находите компонента.

4. В данном примере нас интересует только вкладка **Part Swapping**, где в столбце **Part Group** напротив секций А и В установим одинаковые значения, например «1». Наличие одинаковых значений (цифры и буквы) означает, что данные секции компонента взаимозаменяемы.

Продолжение следует

Литература

1. Пранович В. Система проектирования Altium Designer 6 // Технологии в электронной промышленности. 2006. № 5.
2. Пранович В. Altium Designer 6 в примерах // Технологии в электронной промышленности. 2007. № 5.
3. Пранович В. Altium Designer Summer09. Практические подходы к организации библиотек и структуры проектов. Структура и организация библиотеки на основе базы данных // Технологии в электронной промышленности. 2009. № 8.
4. www.altium.com
5. http://www.freescale.com/files/rf_if/doc/data_sheet/MC13202.pdf

более наглядно. На рис. 7 показано, как сделать выводы питания, отображаемые в двух частях компонента. Отметим, что в таблице **Component Pin Editor** каждый вывод питания продублирован и указан для всех секций компонента. Кстати, подход дублирования выводов на графическом компоненте можно применять и в других случаях, если это повышает наглядность схемы.

5. Часто питание усилителей производится через фильтры, а это явно требует указания данных выводов. Здесь есть проблема, так как, с одной стороны, необходимо отнести выводы питания к одной из секций, с другой — необходимо оставить возможность взаимозаменяемости секций. Эту проблему легко разрешить, создав трехсекционный компонент, причем в третьей секции будут размещены только выводы питания, и она легко совмещается с первыми двумя. На рис. 7 в данном варианте показаны:

- А. Вид секции А.
- В. Вид секции В.
- С. Вид секции С.
- Д. Совмещение изображений секции А и секции С. В целях читаемости обозначение третьей секции на схеме можно скрыть.

В составных компонентах следует обязательно настроить возможность замены при перенумерации компонентов на схеме. Заметим, в последнем созданном компоненте секции А и В идентичны и могут быть заменены одна другой, чего нельзя сказать о секции С. С этой точки зрения частично рассмотрим вопрос установки эквивалентности секций компонента. Полностью раскроем его после создания посадочных мест, так как некоторые настройки производятся непосредственно в схеме или в РСВ-редакторе.

Настройки составного компонента следует сделать такими (рис. 8):

1. Нажимая кнопки «<< < > >>», перейдем на третью секцию компонента (**Part**) и установим флаг **Locked**, таким образом мы запретим изменять номер секции при аннотации схемы.

Внимание! Этот флаг действует только на данную секцию и должен устанавливаться, при необходимости, для каждой секции компонента в отдельности.