

k i c a d b o m 2 s p e c

Руководство пользователя

Вариант для операционных систем MS Windows

Содержание

| | |
|---|----|
| Введение..... | 3 |
| 1. Установка необходимых программ и компонентов..... | 4 |
| 1.1 python..... | 4 |
| 1.2 wxPython..... | 8 |
| 1.3 odfpw..... | 12 |
| 1.4 opengostfont..... | 15 |
| 1.5 LibreOffice..... | 15 |
| 2. Работа с программой kicadbom2spec..... | 16 |
| 2.1 Графический пользовательский интерфейс..... | 16 |
| 2.1.1 Меню..... | 16 |
| 2.1.1.1 Файл..... | 17 |
| 2.1.1.2 Правка..... | 18 |
| 2.1.1.3 Вид..... | 19 |
| 2.1.1.4 Справка..... | 19 |
| 2.1.2 Панель инструментов..... | 20 |
| 2.1.3 Таблица полей элементов..... | 21 |
| 2.1.3.1 Сортировка элементов..... | 21 |
| 2.1.3.2 Выделение элементов..... | 21 |
| 2.1.3.3 Редактирование полей..... | 22 |
| 2.1.3.4 Поиск/замена текста..... | 22 |
| 2.2 Параметры командной строки..... | 23 |
| 2.2.1 Первый позиционный аргумент schematic..... | 23 |
| 2.2.2 Второй позиционный аргумент spec..... | 23 |
| 2.2.3 Параметр -h, --help..... | 23 |
| 3. Подключение kicadbom2spec к EESchema в качестве плагина..... | 24 |
| 3.1 Настройка редактора схем EESchema..... | 24 |
| 3.2 Создание и настройка плагина..... | 25 |
| 4. Редактирование настроек программы..... | 28 |
| 4.1 Основные..... | 28 |
| 4.2 Значения полей..... | 30 |
| 5. Пример создания перечня элементов..... | 31 |

Введение

kicadbom2spec — программа, написанная на языке python и предназначена для создания файлов перечня элементов в виде таблиц *.ods в формате Open Document Format, оформленных согласно требованиям ЕСКД, для схем выполненных с помощью программы *EEschema* из пакета программ САПР *KiCAD*.

Кроме создания перечня элементов, с помощью этой программы можно редактировать необходимые пользовательские поля элементов как в файлах схем (*.sch) так и в файлах библиотек (*.lib).

Примечание: в данном руководстве указаны версии программ и файлов доступные на момент написания данного руководства и могут отличаться от существующих.

1. Установка необходимых программ и компонентов

Программа *kicadbom2spec* написана на языке *python* и требует для своей работы наличие интерпретатора языка *python*, который будет считывать команды из файла, строку за строкой, и выполнять их.

Графический пользовательский интерфейс создан на базе библиотеки *wxWidgets* и для связи данной библиотеки с программным кодом *python* используется прослойка *wxPython*.

Файл перечня элементов имеет формат Open Document Format и сохраняется в виде электронной таблицы отформатированной таким образом, чтобы соответствовать требованиям ЕСКД. Для облегчения работы с файлами в данном формате используется модуль *odfpy*, который устанавливается как расширение интерпретатора языка *python*.

Шаблон перечня элементов был создан с помощью свободного офисного пакета *LibreOffice* с применением открытого чертежного шрифта *opengostfont*. Наличие данных элементов не обязательно, но крайне желательно, дабы получить перечень элементов с корректным форматированием. Известно, что MS Office Word неправильно воспроизводит форматирование при открытии сгенерированного файла спецификации.

О том как установить нужные элементы для корректной работы *kicadbom2spec* будет рассказано далее на примере операционной системы MS Windows XP.

1.1 python

В данном случае нужен интерпретатор языка *python* версии 2. Устанавливается он как обычное приложение:



Рисунок 1. Ссылка на скачивание python2

- для начала нужно скачать файл установки с официального сайта www.python.org. На главной странице можно найти ссылку для скачивания, например, как показано на рис.1.
- после нажатия на ссылку появится диалоговое окно (рис. 2). Если нажать на кнопку «Выполнить», файл сохранится во временную папку и после завершения загрузки начнется процесс установки. А если нажать на кнопку «Сохранить» - появится диалог сохранения файла, в котором нужно указать место, куда нужно сохранить загружаемый файл и после окончания загрузки он не будет выполняться автоматически (его можно будет запустить вручную).

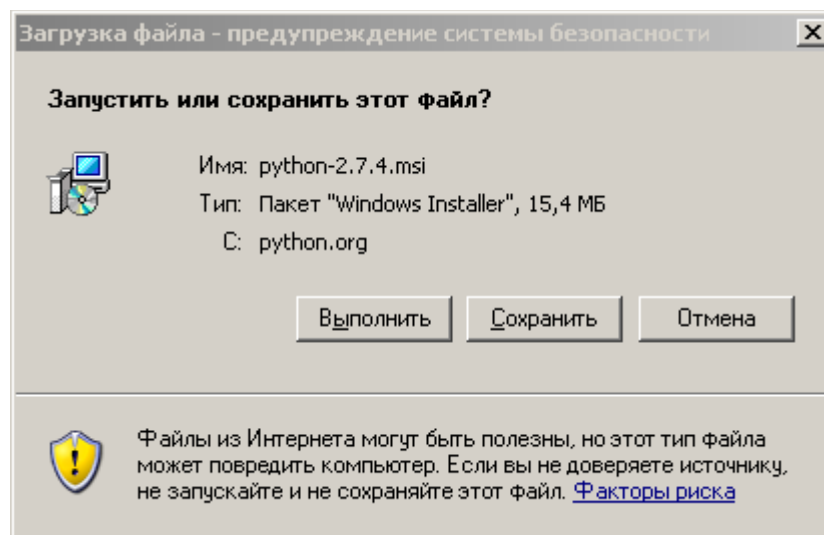


Рисунок 2. Диалоговое окно

Нажимаем кнопку «Выполнить».

- далее начинается загрузка файла установки (рис. 3).

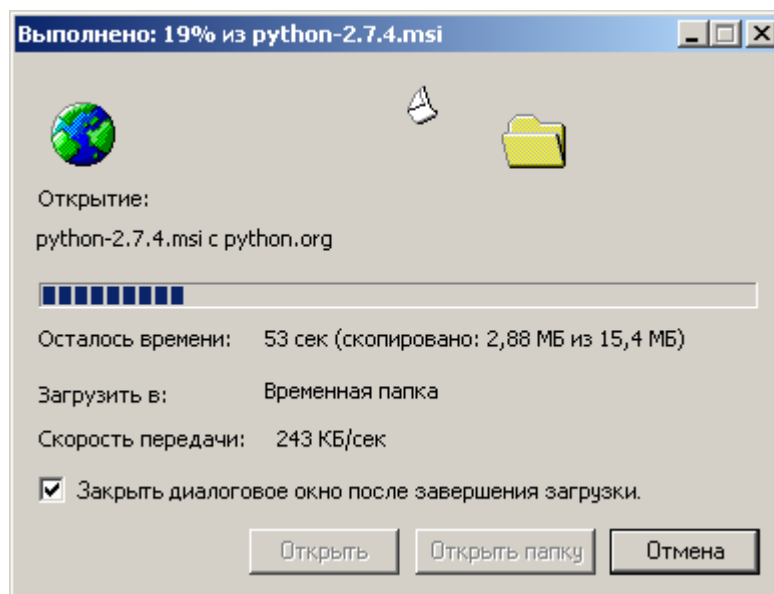


Рисунок 3. Загрузка файла

- После завершения загрузки откроется мастер установки интерпретатора *python* (рис. 4). Настройки по умолчанию нас устраивают, поэтому в процессе установки ничего не изменяем, а

просто нажимаем кнопку «Next >».

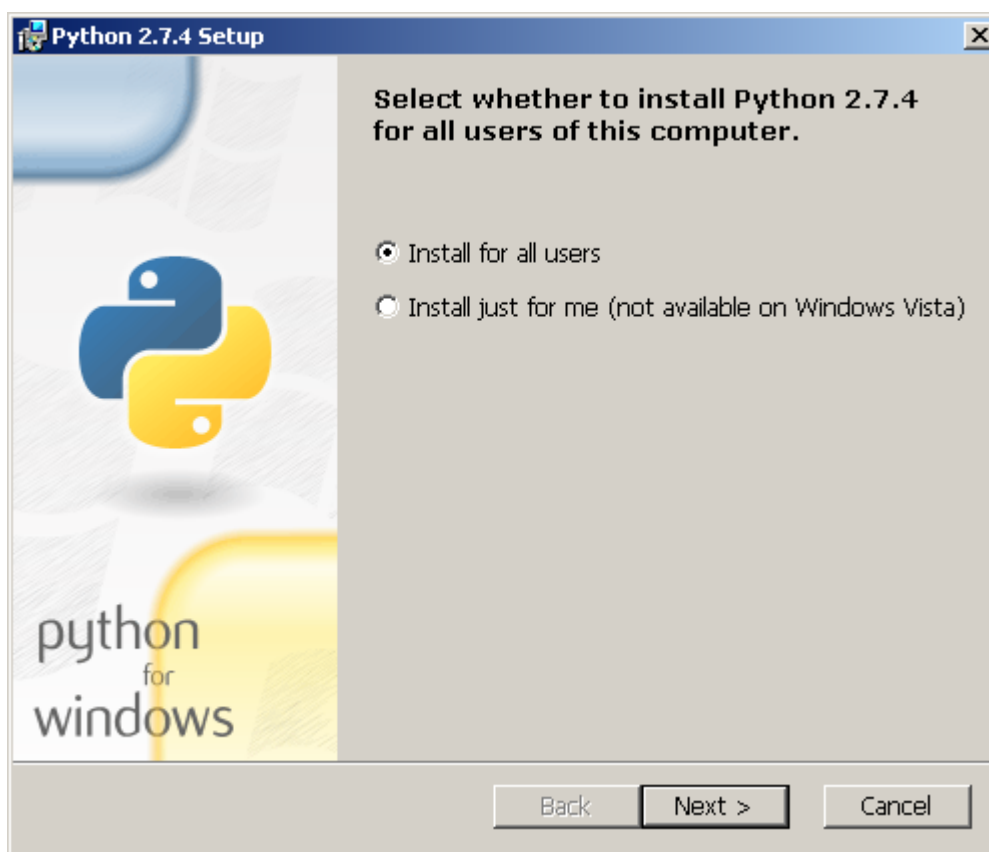


Рисунок 4. Начало установки

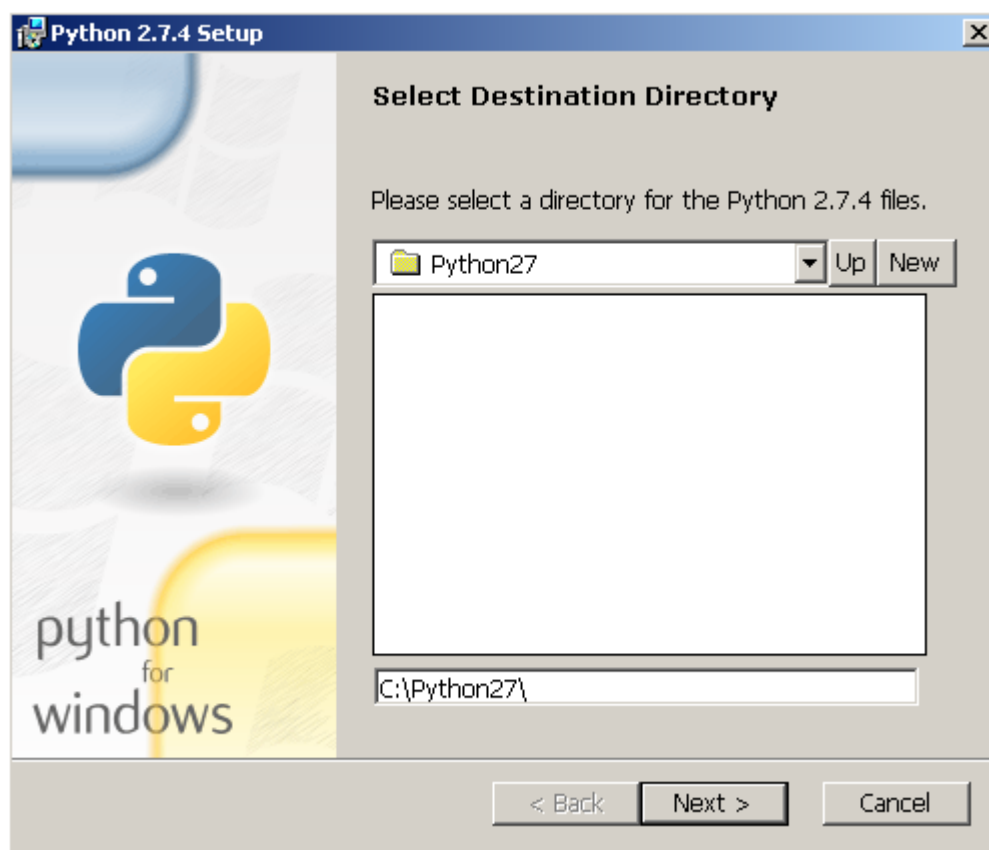


Рисунок 5. Выбор места установки

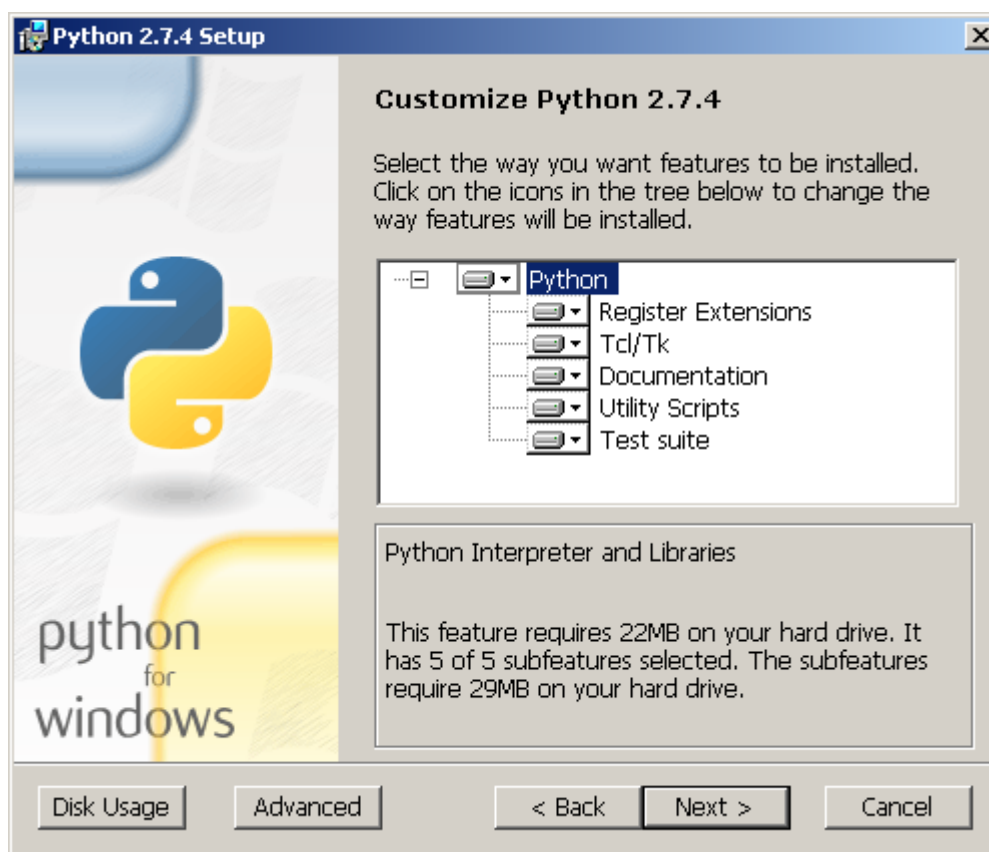


Рисунок 6. Выбор элементов установки

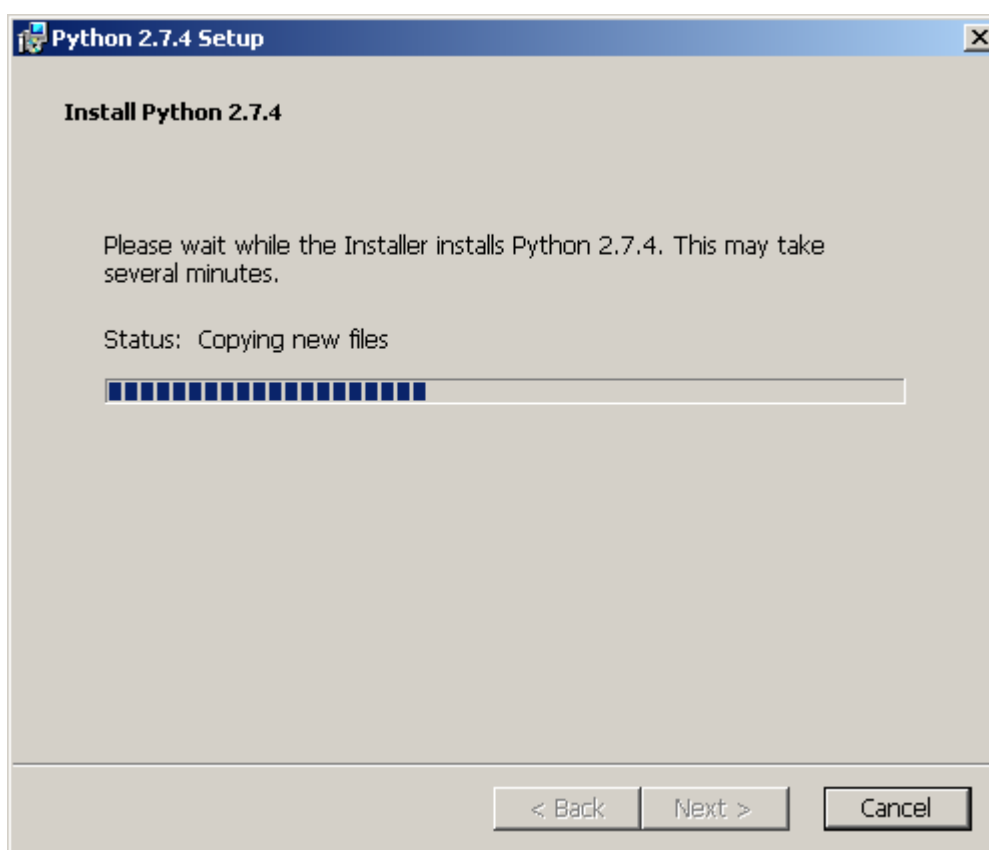


Рисунок 7. Процесс установки



Рисунок 8. Завершение установки

- если в процессе установки не возникло ошибок, окно мастера установки примет вид как показано на рис. 8. Для завершения нужно нажать кнопку «Finish».
- Готово, интерпретатор языка *python* установлен и готов к работе.

1.2 wxPython

Для отображения элементов пользовательского интерфейса необходимо установить *wxPython*. Загрузить инсталлятор можно со страницы www.wxpython.org/download.php выбрав ссылку *wxPython2.8-win**-unicode-py27*.¹

Процесс загрузки и запуска проходит подобно тому как это было с *python*.

Итак, после загрузки и запуска появится окно мастера установки (рис. 9). В процессе установки можно изменять разные параметры, но значения по умолчанию нас устраивают и поэтому нужно просто нажимать кнопку «Next >».

На втором этапе будет показано лицензионное соглашение, с которым нужно согласится выбрав «I accept the agreement», после чего станет доступна кнопка «Next >».

После завершения установки в последнем окне мастера установки (рис. 14) нужно снять флажок «View README.win**.txt»¹ и нажать кнопку «Finish». После этого ненадолго появится окно командной строки для завершения настройки установленного *wxPython*.

1. ** — 32 или 64 - разрядность установленной версии *python* (именно *python*, а не операционной системы или процессора).

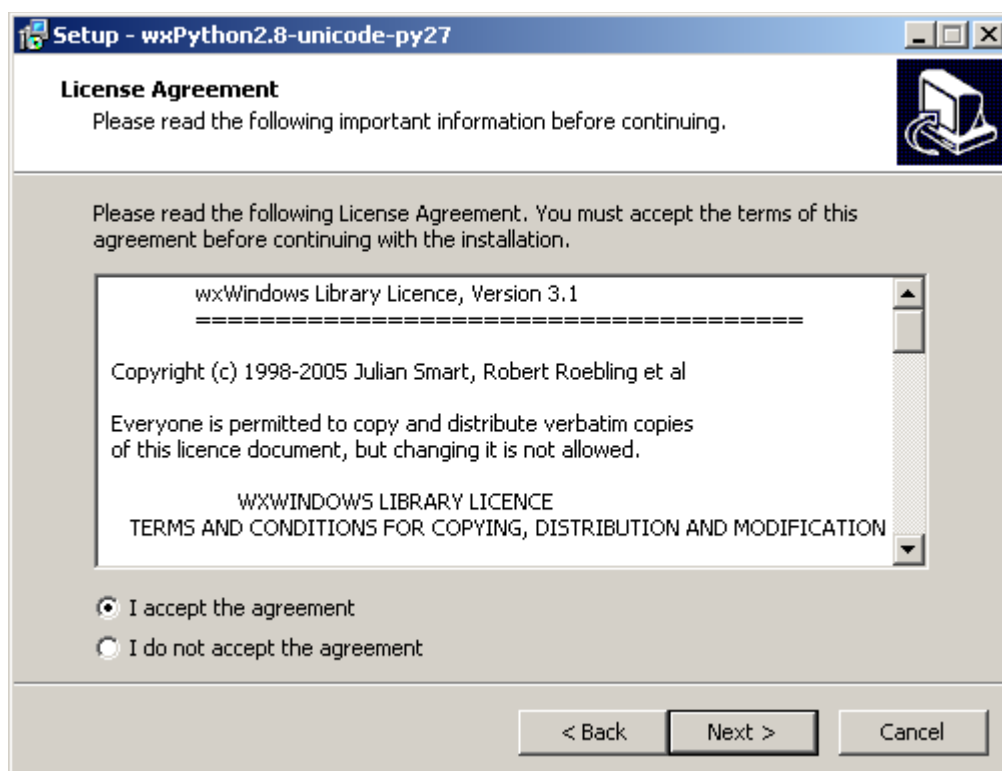
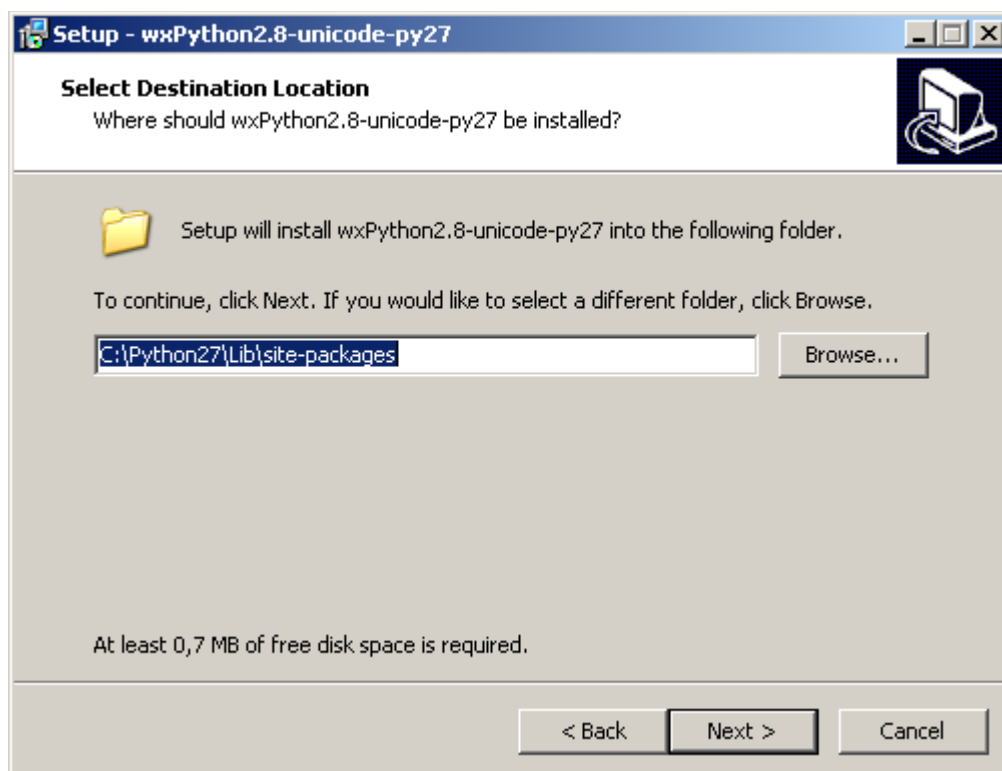
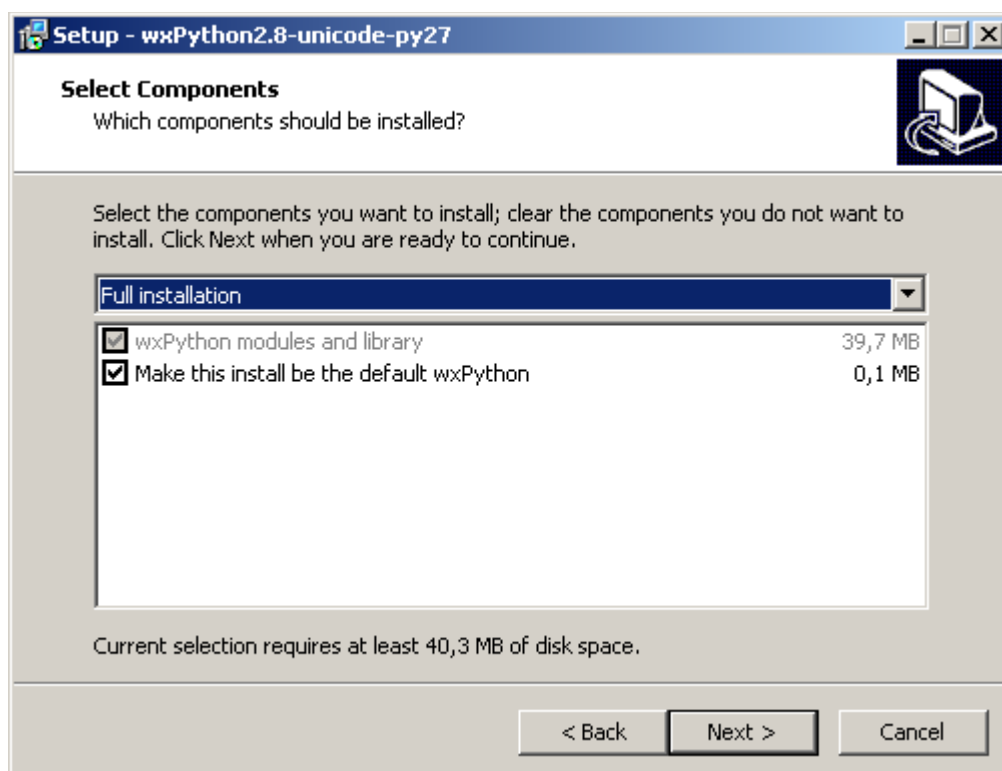
Рисунок 9. Мастер установки *wxPython*

Рисунок 10. Лицензионное соглашение

Рисунок 11. Путь установки *wxPython*Рисунок 12. Выбор компонентов установки *wxPython*

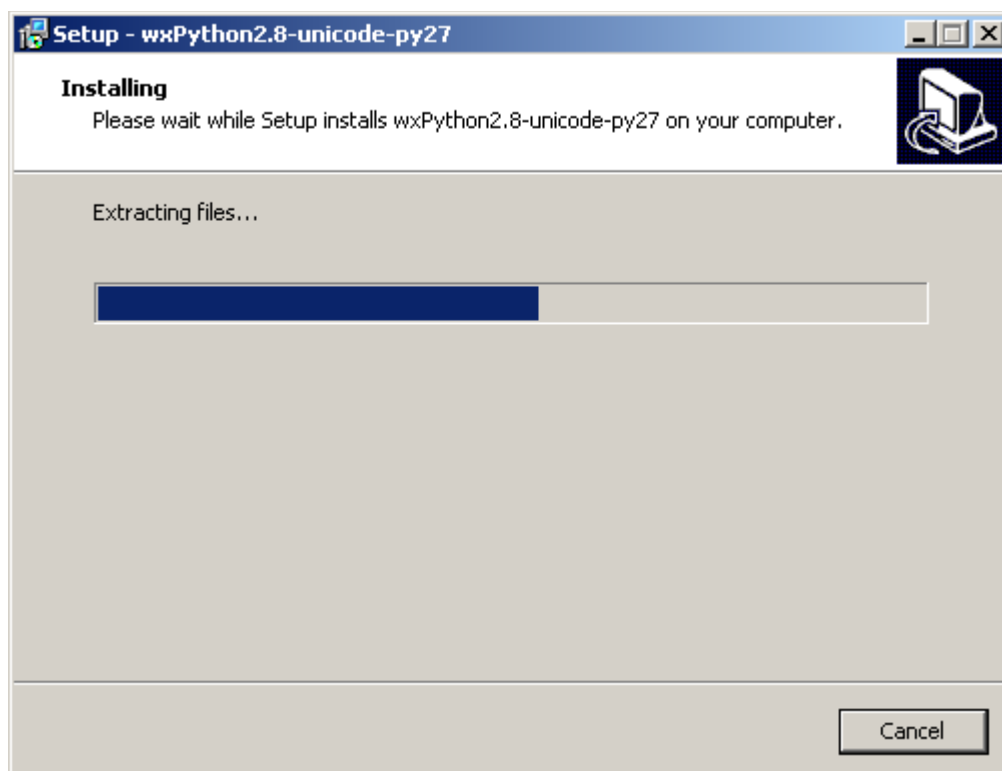


Рисунок 13. Процесс установки wxPython

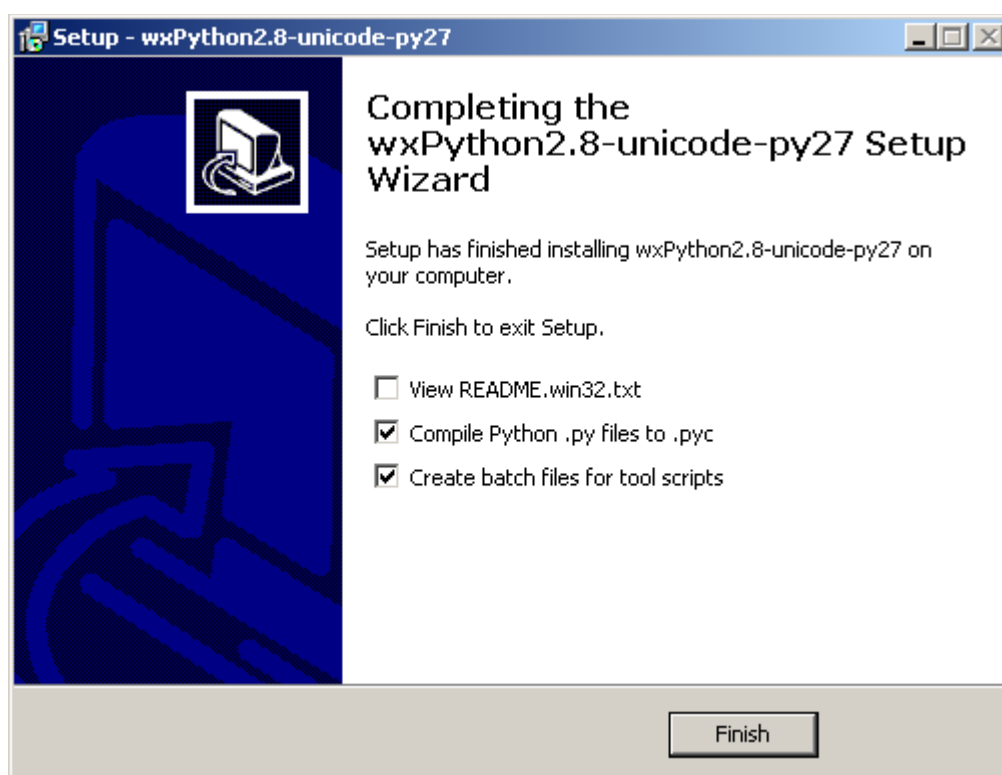


Рисунок 14. Завершение установки wxPython

wxPython содержит в себе необходимые элементы библиотеки wxWidget и ее не нужно устанавливать отдельно.

1.3 odfpy

Этот модуль предоставляет функции для работы с документами Open Document Format и в данном случае используется для создания и редактирования электронных таблиц *.ods.

Обычно модули для интерпретатора языка python предоставляются в виде пакета исходных кодов, которые нужно собрать (с помощью того же интерпретатора) и установить к уже имеющимся стандартным модулям. Процесс построения и установки обычно выполняется с помощью командной строки (о том как это сделать, можно прочитать в файле README из пакета исходных кодов), но так как для обычного пользователя это может оказаться затруднительным здесь будет рассмотрен другой, более привычный, способ.

Итак, для того, чтобы установить *odfpy* без использования командной строки необходимо выполнить следующее:

- первым делом нужно скачать пакет с исходным кодом с сайта pypi.python.org/pypi/odfpy. На этой странице ссылка выполнена в виде кнопки «Download» (рис. 15).

» Package Index > odfpy > 0.9.6

odfpy 0.9.6

Python API and tools to manipulate OpenDocument files

Download
odfpy-0.9.6.tar.gz

Odfpy is a library to read and write OpenDocument v. 1.1 files. The main focus has been to prevent the programmer from creating invalid documents. It has checks that raise an exception if the programmer adds an invalid element, adds an attribute unknown to the grammar, forgets to add a required attribute or adds text to an element that doesn't allow it.

Рисунок 15. Загрузка odfpy

- после нажатия на эту кнопку появится предупреждающее сообщение (рис. 16).

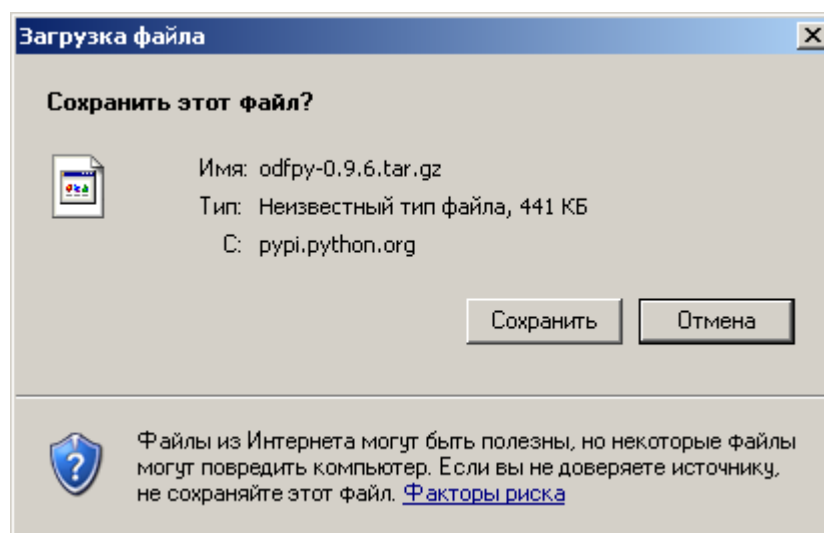


Рисунок 16. Загрузка файла

- нажимаем кнопку «Сохранить», в результате чего откроется диалог сохранения файла (рис. 17), в котором нужно указать куда должен быть загружен архив.
- сохраняем пакет с исходными кодами *odfpy* на Рабочем столе (можно сохранить и в любом другом месте, что не столь важно, так как после установки модуля этот архив можно удалить).
- далее нужно распаковать содержимое архива на Рабочий стол. В результате этого на Рабочем столе должна появиться новая папка «odfpy-0.9.6».
- открываем папку «odfpy-0.9.6». Внутри содержится много вложенных папок и несколько файлов. Файл отвечающий за построение и установку модуля называется «setup.py» - это скрипт на языке *python*. При простом запуске двойным щелчком он ничего не выполняет. Для того чтобы собрать модуль из исходных кодов, нужно запустить данный скрипт с параметром «build». А для того чтобы установить собранный модуль, нужно его запустить с параметром «install».
- чтобы передать параметр в «setup.py» при запуске без применения командной строки будем использовать ярлыки.

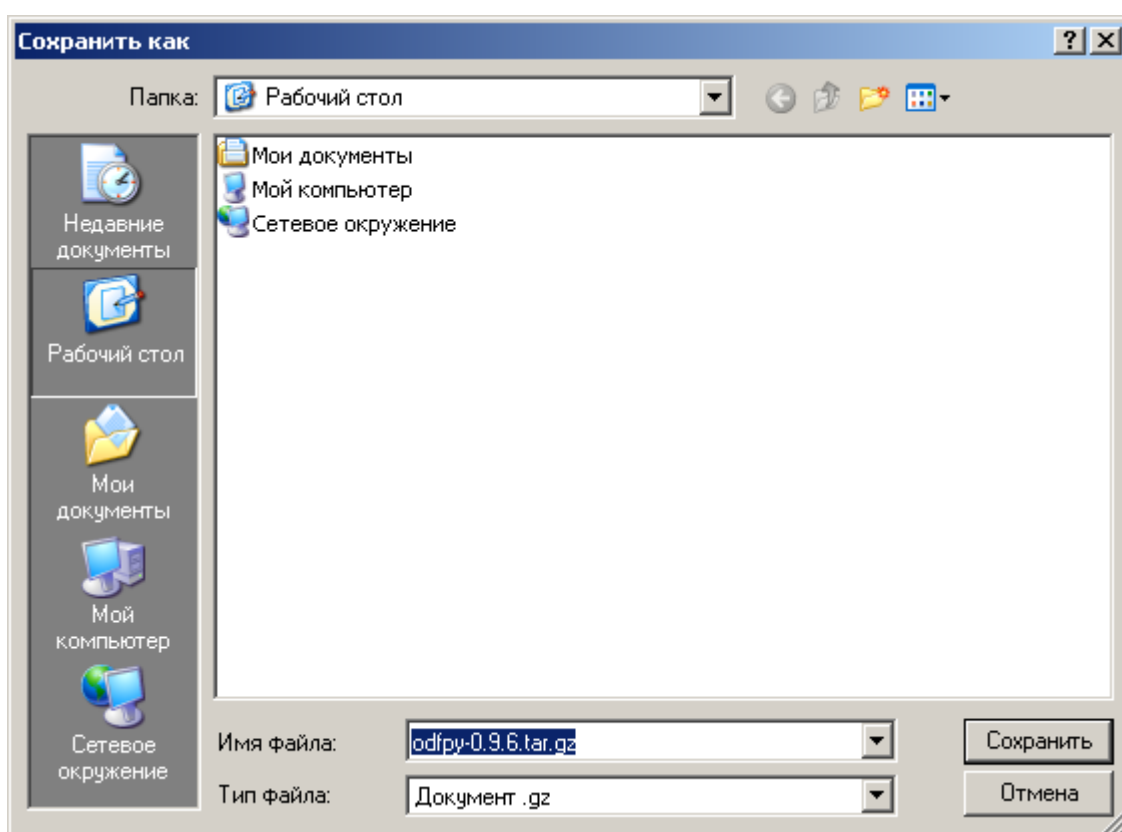


Рисунок 17. Диалог сохранения загрузки

- для этого нажимаем на файле «setup.py» правой кнопкой мыши (далее ПКМ) и не отпуская ее перетаскиваем файл на свободное место в папке, затем отпускаем ПКМ. Появится контекстное меню как на рис. 18. Выбираем пункт «Создать ярлыки» - ярлык создан.
- чтобы не запутаться, сразу же переименовываем его в «build» (имя можно задать любое).
- далее открываем свойства ярлыка. Для этого нажимаем ПКМ на ярлыке «build», в результате чего появится контекстное меню. Выбираем в самом низу пункт «Свойства» - откроется окно.

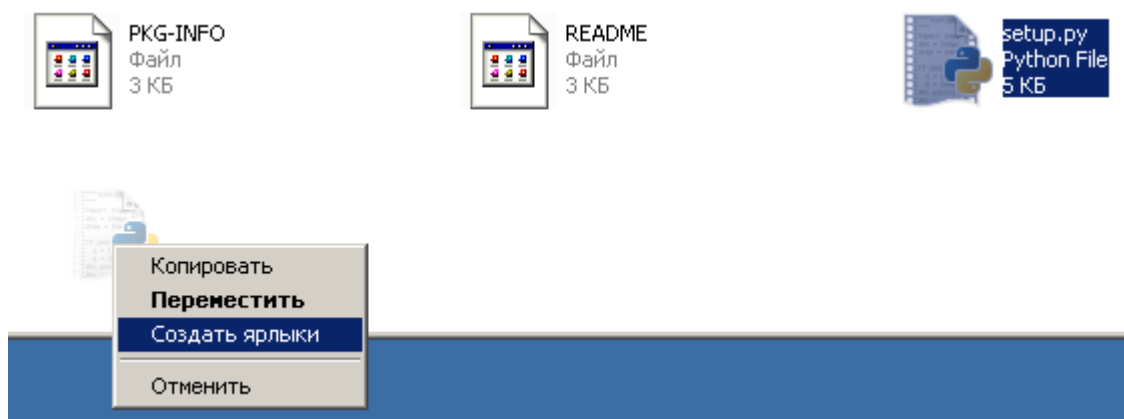


Рисунок 18. Создание ярлыка

- в окне свойств ярлыка (рис. 19) имеется некоторая информация о нем и несколько полей для редактирования. Нам нужно добавить в конец свойства «Объект» нужную команду, в данном случае «build» (через пробел).

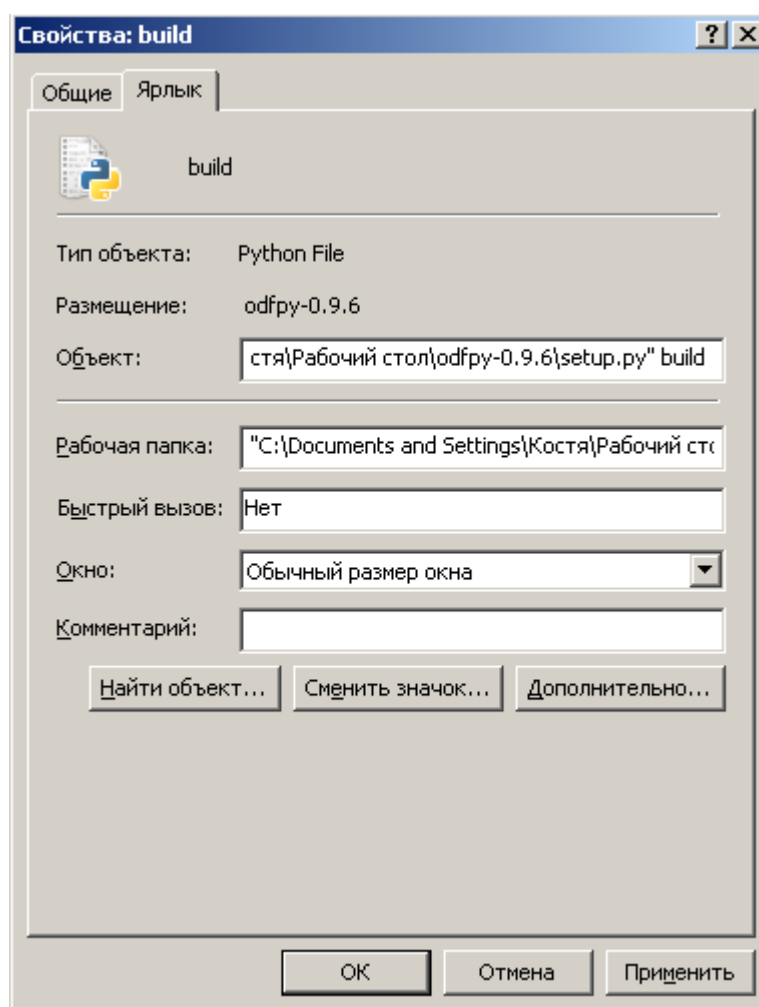
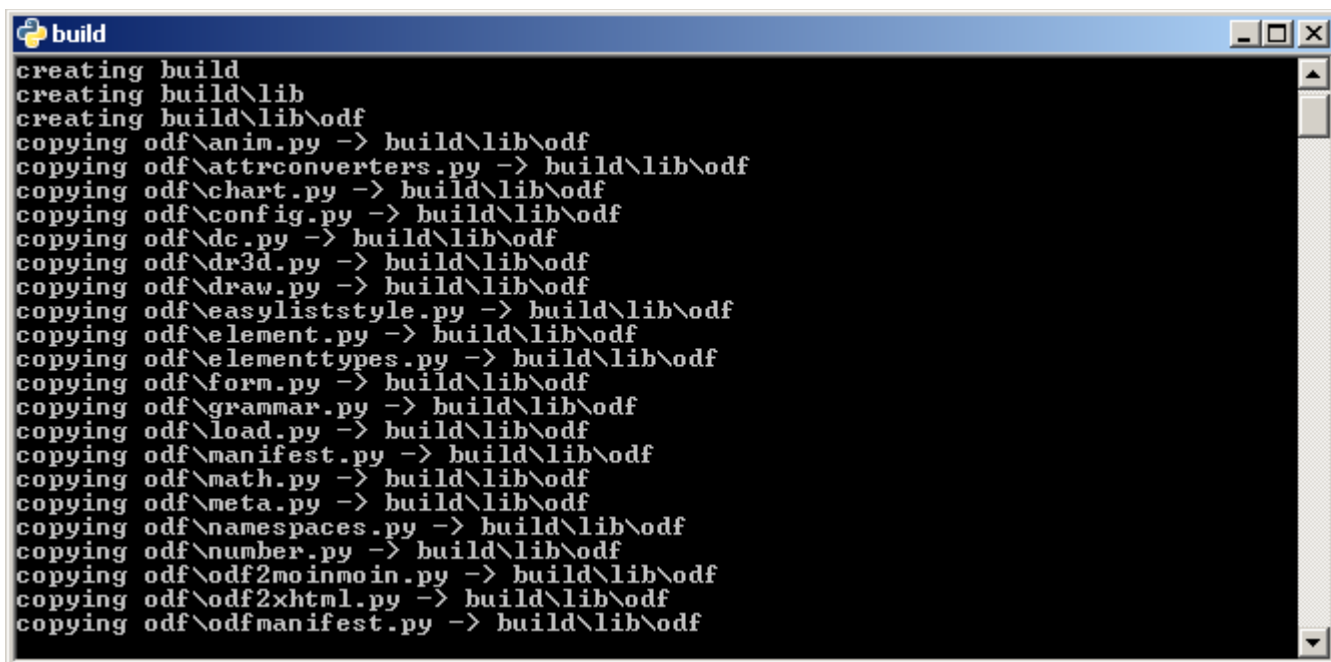


Рисунок 19. Свойства ярлыка

- подтверждаем изменения нажатием кнопки «ОК».



```

build
creating build
creating build\lib
creating build\lib\odf
copying odf\anim.py -> build\lib\odf
copying odf\attrconverters.py -> build\lib\odf
copying odf\chart.py -> build\lib\odf
copying odf\config.py -> build\lib\odf
copying odf\dc.py -> build\lib\odf
copying odf\dr3d.py -> build\lib\odf
copying odf\draw.py -> build\lib\odf
copying odf\easyliststyle.py -> build\lib\odf
copying odf\element.py -> build\lib\odf
copying odf\elementtypes.py -> build\lib\odf
copying odf\form.py -> build\lib\odf
copying odf\grammar.py -> build\lib\odf
copying odf\load.py -> build\lib\odf
copying odf\manifest.py -> build\lib\odf
copying odf\math.py -> build\lib\odf
copying odf\meta.py -> build\lib\odf
copying odf\namespaces.py -> build\lib\odf
copying odf\number.py -> build\lib\odf
copying odf\odf2moinmoin.py -> build\lib\odf
copying odf\odf2xhtml.py -> build\lib\odf
copying odf\odfmanifest.py -> build\lib\odf

```

Рисунок 20. Ход построения модуля

- после этого можно запустить построение двойным щелчком по ярлыку «build». На экране ненадолго появится окно командной строки с бегущими строками (рис. 20), что свидетельствует о работе скрипта «setup.py».
- после сборки в папке с исходными кодами появится новая папка «build».
- далее нужно установить собранный модуль. Для этого нужно создать ярлык для скрипта «setup.py», изменить имя на «install», открыть свойства этого ярлыка и добавить к свойству «Объект» через пробел команду «install», сохранить изменения в свойствах и запустить на выполнение. То есть, выполнить все точно так же как и для сборки модуля, но с командой «install».
- на время установки, также как и при сборке, кратковременно появится консоль.
- готово — модуль «odfpy» собран и установлен. Загруженный архив и распакованная папка уже не нужны и их можно удалить.

1.4 opengostfont

Устанавливается данный шрифт как и любой другой, путем копирования файлов шрифтов в системную папку `C:\WINDOWS\Fonts`.

Загрузить файлы шрифта можно на странице www.bitbucket.org/fat_angel/opengostfont/downloads, выбрав последнюю версию. Нужно заметить, что архив содержащий в названии «src» содержит исходные файлы шрифта, в нашем случае нужно выбрать один из архивов содержащий в названии «ttf» или «otf» (разница между ними не значительна, выглядят они одинаково).

1.5 LibreOffice

Офисный пакет устанавливается как обычное приложение, подобно тому как был установлен интерпретатор языка *python*, без каких либо особенностей, поэтому подробно на этом останавливаться не будем. Загрузить файл установки можно на официальном сайте <http://www.libreoffice.org>.

2. Работа с программой kicadbom2spec

Скачать архив с программой и документацией можно на странице проекта launchpad.net/kicadbom2spec, с правой стороны которой расположена кнопка загрузки (рис. 21).

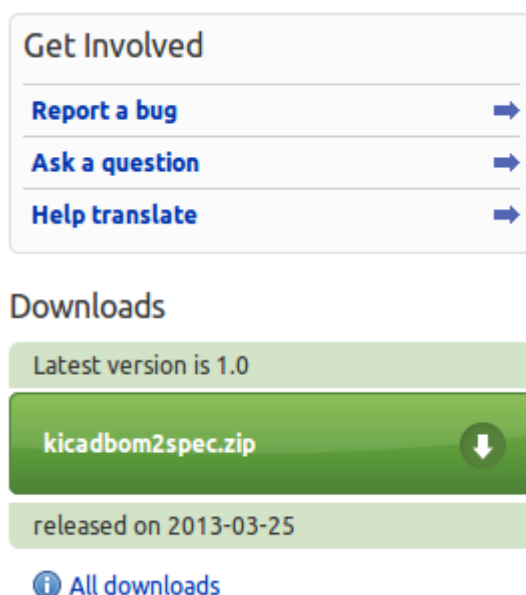


Рисунок 21. Загрузка kicadbom2spec

На этой странице также можно оставить отчет об ошибке или задать вопрос, касающийся работы программы.

kicadbom2spec не требует установки, для его использования нужно просто загрузить архив и распаковать содержимое в удобное место. Чтобы упростить работу с программой можно создать ссылку для «kicadbom2spec.ру» и поместить ее, скажем, на рабочий стол или в папку с проектом *KiCAD*.

2.1 Графический пользовательский интерфейс

Внешний вид графического пользовательского интерфейса показан на рис.23. Он имеет следующие основные элементы:

- **меню;**
- **панель инструментов;**
- **таблицу полей элементов схемы.**

Рассмотрим дальше каждый из этих элементов отдельно.

2.1.1 Меню

Меню программы *kicadbom2spec* (рис. 22) состоит из четырех элементов:

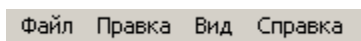


Рисунок 22. Меню программы

- **Файл** — в этом меню собраны команды, которые относятся к работе с файлами;
- **Правка** — здесь собраны команды для редактирования полей элементов;
- **Вид** — это меню содержит элементы настройки внешнего вида программы;
- **Справка** — в этом меню доступны команды для получения справки.

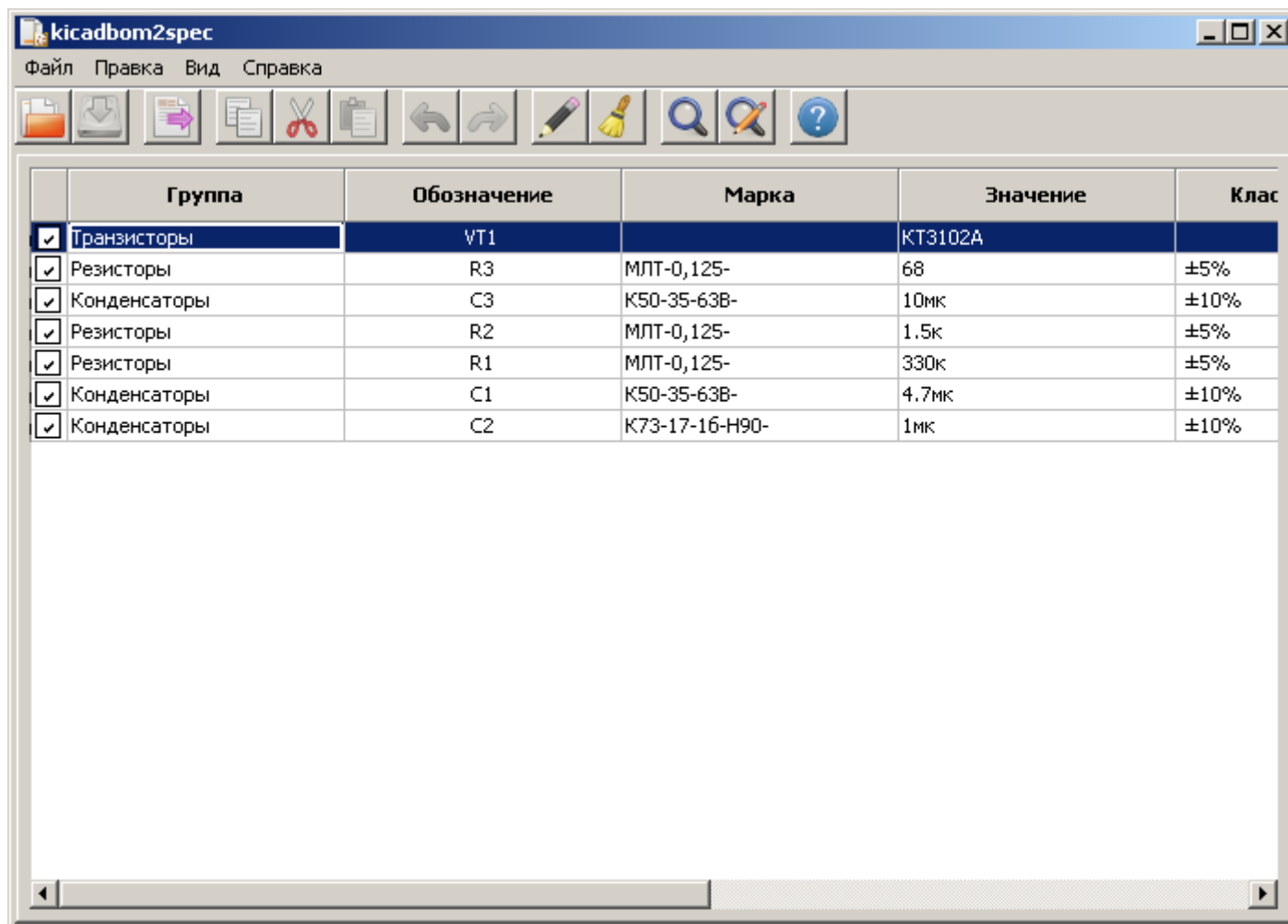


Рисунок 23. Внешний вид

2.1.1.1 Файл

В этом меню содержатся команды для управления файлами:

- **Открыть схему...** — открывает стандартный диалог выбора файла схемы;
- **Сохранить схему** — сохраняет изменения полей элементов в файл схемы;
- **Сохранить схему как...** — открывает стандартный диалог сохранения для записи имеющихся данных в новый файл схемы;
- **Открыть библиотеку...** — открывает стандартный диалог выбора файла библиотеки элементов схемы;
- **Сохранить библиотеку** — сохраняет изменения полей элементов в файл библиотеки;

- **Сохранить библиотеку как...** — открывает стандартный диалог сохранения для записи имеющихся данных в новый файл библиотеки;
- **Создать перечень элементов...** — открывает диалог создания перечня элементов (рис. 24), в котором нужно указать файл, в который будет записан перечень элементов, и задать некоторые параметры;

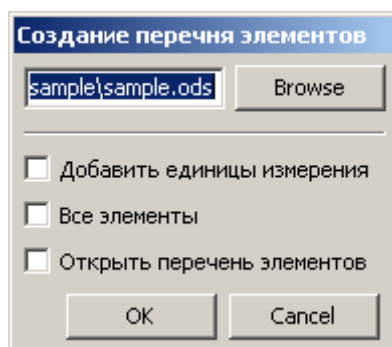


Рисунок 24. Диалог создания перечня элементов

- **Выход** — завершит выполнение программы, если имеются не сохраненные изменения — будет показано предупреждающее сообщение.

2.1.1.2 Правка

В этом меню содержатся команды редактирования полей элементов:

- **Отменить** — отменяет последнее изменение поля/полей элементов;
- **Повторить** — повторяет отмененное ранее изменение;

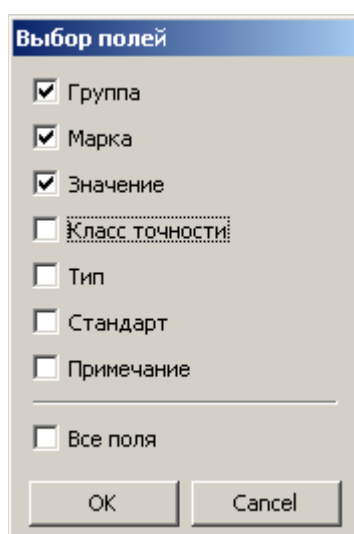


Рисунок 25. Диалог выбора полей

Все изменения сохраняются в оперативной памяти и количество сохраняемых изменений не ограничивается. При длительной работе с большим количеством элементов может использоваться значительный объем памяти и если это критично, то для ее освобождения нужно сохранить

схему/библиотеку и открыть ее вновь.

- **Копировать** — копирует значения всех пользовательских полей выбранного элемента в буфер обмена;
- **Вырезать...** — вырезает значения полей элемента, указанных в диалоге выбора полей (рис. 25), в буфер обмена. Значения полей, которые не были указаны, просто копируются в буфер обмена.

Команды *Копировать* и *Вырезать...* применяются только к одному элементу. Если при выполнении одной из этих команд будет выделено несколько элементов — отобразится предупреждающее сообщение.

- **Вставить...** — вставляет в выделенный(е) элемент(ы) значения из буфера обмена. Поля, значения которых нужно заменить, выбираются с помощью диалога выбора полей.
- **Найти...** — открывает диалог поиска текста (рис. 26) в полях элементов;



Рисунок 26. Диалог поиска текста

- **Заменить...** — открывает диалог замены текста (рис. 27) в полях элементов;

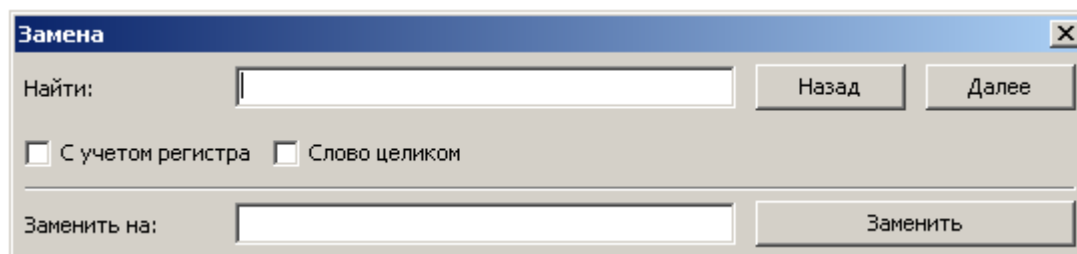


Рисунок 27. Диалог замены текста

- **Редактировать поля...** — открывает редактор полей (рис. 28) для выделенного(ых) элемента(ов);
- **Очистить поля...** — удаляет содержимое полей, указанных в диалоге выбора полей, во всех выбранных элементах.

2.1.1.3 Вид

В этом меню доступна одна команда — **Панель инструментов**. С помощью этой команды можно скрыть или отобразить панель инструментов.

2.1.1.4 Справка

Из меню справки можно выполнить два действия:

- **Руководство пользователя** — открывает данное справочное руководство пользователя;
- **О программе...** — открывает диалоговое окно с краткими сведениями о программе kicadbom2spec.

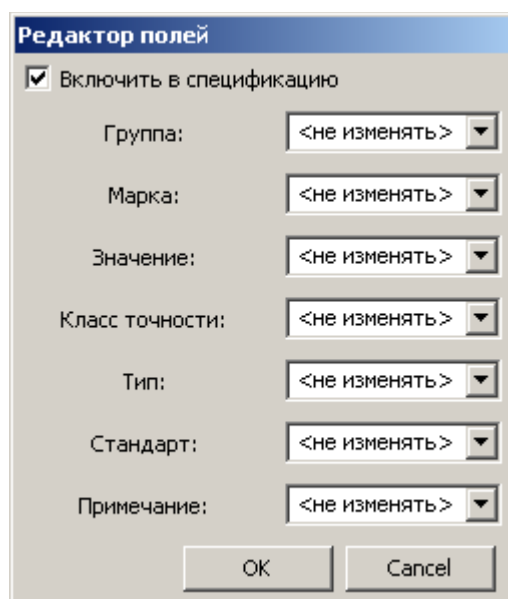


Рисунок 28. Редактор полей

2.1.2 Панель инструментов

Панель инструментов расположена в верхней части окна и повторяет часто используемые команды меню, что повышает скорость и облегчает работу с программой.

Внешний вид панели инструментов показан на рис. 29.



Рисунок 29. Панель инструментов

Доступны следующие команды (в порядке расположения слева направо):

- *Открыть схему;*
- *Сохранить схему;*
- *Создать перечень элементов;*
- *Копировать;*
- *Вырезать;*
- *Вставить;*
- *Отменить;*

- *Повторить;*
- *Редактировать поля;*
- *Очистить поля;*
- *Найти;*
- *Заменить;*
- *Руководство пользователя.*

В случае необходимости панель инструментов можно скрыть выбрав команду меню *Вид-Панель инструментов*.

2.1.3 Таблица полей элементов

Таблица полей элементов занимает основную часть окна приложения и имеет следующую структуру. Каждая строка представляет собой элемент схемы или библиотеки, а столбцы — поля элементов. Всего столбцов девять:

- **Первый** столбец не имеет названия и содержит во всех ячейках флажки. С помощью этих флажков можно выбирать элементы, которые нужно использовать при создании перечня элементов. По умолчанию все элементы отмечены флажками. При редактировании файла библиотеки элементов этот столбец хоть и отображается, но не используется.
- **Второй** столбец *Группа* содержит текст одноименного поля элемента.
- **Третий** столбец *Обозначение* содержит текст одноименного поля элемента. Это поле используется как уникальный идентификатор элемента и доступно только для чтения. Нормально заполненная схема не должна иметь элементов с одинаковыми *обозначениями*. Если в схеме есть не пронумерованные элементы (оканчивающиеся вопросительным знаком, например, *R?*, *C?* и т.д.), то они не будут загружены в таблицу.
- **Четвертый — девятый** столбцы отображают содержимое полей *Марка*, *Значение*, *Класс точности*, *Тип*, *Стандарт*, *Примечание*, соответственно.

При редактировании библиотеки элементов, ячейки столбца *Значение* доступны только для чтения, изменять значение данного поля элемента библиотеки нельзя.

2.1.3.1 Сортировка элементов

Для представления элементов в более удобном виде предусмотрена возможность сортировки содержимого таблицы. Выполняется она путем нажатия на заголовке столбца по содержимому которого должна производиться сортировка. Например, если нужно отобразить элементы собранные по группам и в алфавитном порядке — нужно нажать на заголовок столбца *Группа*. Если нажать еще один раз — содержимое отсортируется в обратном порядке.

2.1.3.2 Выделение элементов

Чтобы выделить один элемент нужно нажать ЛКМ на одном из его полей.

Если необходимо выделить несколько элементов подряд — нужно выделить первый нужный элемент, затем нажать клавишу «Shift» и не отпуская ее нажать ЛКМ на последнем нужном элементе.

Эту же операцию можно выполнить другим способом - нажать ЛКМ на первом нужном элементе и не отпуская ее тянуть курсор к последнему нужному элементу и после этого отпустить ЛКМ.

Чтобы выделить несколько не последовательных элементов нужно выделить первый элемент, а последующие выделять при нажатой клавише «Ctrl».

2.1.3.3 Редактирование полей

Для редактирования содержимого полей предусмотрено несколько способов. Любое изменение в таблице можно отменить с помощью команды меню *Правка-Отменить*, а отмененное действие повторить с помощью команды *Правка-Повторить*. Если после отмены было произведено новое изменение то команда повтора становится недоступной.

Прежде всего каждое доступное для редактирования поле можно править щелкнув по нему двойным щелчком ЛКМ. В случае с флажками — однократный щелчок ЛКМ изменит состояние флажка на противоположное. Эти же операции можно выполнять с клавиатуры. С помощью стрелок нужно переместиться к ячейке с содержимым поля, которое необходимо отредактировать и просто начать вводить текст. Для изменения состояния флажка нужно нажать клавишу «пробел».

Но редактирование каждого поля для каждого элемента в отдельности может быть очень утомительным. Для редактирования нескольких элементов одновременно необходимо воспользоваться редактором полей (рис. 28). Итак чтобы изменить значения полей одновременно для нескольких элементов необходимо их выделить и затем выполнить команду *Правка-Редактировать поля...*, откроется редактор полей, который содержит флажок для указания того, нужно ли помещать выделенные элементы в перечень элементов или нет (при редактировании библиотеки — отсутствует) и текстовые поля для редактирования полей элементов с выпадающим меню.

По умолчанию во всех текстовых полях установлено значение *<не изменять>*, что указывает на то, что значение полей не будет изменено. Это позволяет редактировать только нужные поля, а остальные оставить без изменений. Чтобы изменить значение поля всех выделенных элементов нужно заменить надпись *<не изменять>* на нужное значение и нажать кнопку *Ок*.

Для ускорения процесса редактирования выполнена следующая функция. Для каждого поля выделенных элементов собираются разные значения и сохраняются в виде выпадающего меню в редакторе полей. Рассмотрим данную функцию на примере. Допустим, выделено два элемента с значением полей *Значение - 10кОм* и *3.3кОм*. Так вот, при открытии редактора полей и нажатии на кнопке справа от текстового поля *Значение*, появится выпадающее меню со следующими вариантами выбора: *<не изменять>*, *10кОм* и *3.3кОм*. Выбранный вариант помещается в текстовое поле.

Значения полей одного или нескольких элементов можно очищать. Для этого существует команда меню *Правка-Очистить поля...*, после выбора которой отобразится диалог выбора полей (рис. 25). Чтобы очистить нужные поля их необходимо отметить и нажать кнопку *Ок*.

При редактировании доступны операции для работы с буфером обмена *Правка-Копировать/Вырезать.../Вставить...* Команды *Копировать* и *Вырезать...* применимы только к одному элементу. Команду *Вставить...* можно применять сразу к нескольким выделенным элементам.

При копировании, в буфер обмена сохраняются значения полей выделенного элемента. При вырезании выполняется копирование значений всех полей с последующей очисткой только тех полей, которые были указаны в диалоге выбора полей. При вставке будут заменены значения только тех полей, которые были отмечены в диалоге выбора полей.

2.1.3.4 Поиск/замена текста

Чтобы найти текст в полях элементов нужно воспользоваться командой *Правка-Найти...* После выбора данной команды будет открыт диалог поиска текста (рис. 26) в котором нужно ввести фрагмент текста в текстовое поле и для поиска ниже выделенного элемента, нажать кнопку *Далее* или нажать

клавишу *Enter*. Для поиска текста в обратном направлении нужно нажать кнопку *Назад*.

Чтобы уточнить результаты поиска можно воспользоваться опциями:

- **С учетом регистра** – поиск будет выполнен учитывая регистр введенных букв;
- **Слово целиком** – будут найдены только те слова, которые полностью соответствуют введенному поисковому запросу. Нужно заметить, если запрос состоит из нескольких слов, с данной опцией они никогда не будут найдены. Т.е. если нужно найти слово, то и запрос должен быть словом.

Для замены текста используется команда *Правка-Заменить...* При ее выборе появится расширенное диалоговое поиска — диалог замены текста (рис. 27). В начале выполняется поиск фрагмента текста, который нужно заменить подобно тому как было описано ранее. Далее вводится текст на который нужно заменить найденный текст в поле *Заменить на* и нажатием кнопки *Заменить* выполняется замена.

2.2 Параметры командной строки

При запуске программы *kicadbom2spec*, ей можно передавать один или два позиционных аргумента. Это дает возможность подключать *kicadbom2spec* к редактору схем *EESchematic* в качестве плагина.

Формат вызова программы из командной строки:

```
python kicadbom2spec.pyw [-h] [schematic] [spec]
```

2.2.1 Первый позиционный аргумент *schematic*

С помощью этого параметра в программу передается имя файла схемы в формате KiCAD Schematic.

Примечание. Так как при запуске *kicadbom2spec* в качестве плагина *EESchema*, ей передается в первом параметре имя файла списка цепей (*.xml), то для обеспечения правильной работы применяется следующий процесс обработки параметра:

- считывается имя файла;
- удаляется расширение;
- добавляется расширение **.sch*;
- проверяется наличие файла с полученным именем:
 - если файл существует — его содержимое загружается для дальнейшего редактирования;
 - если файл не существует — открывается диалог выбора файла схемы.

2.2.2 Второй позиционный аргумент *spec*

С помощью этого параметра в программу передается имя файла в который будет сохранен перечень элементов. Расширение файла автоматически заменяется на **.ods*.

2.2.3 Параметр *-h, --help*

С этим параметром программа выводит справочное сообщение с краткой информацией о программе и доступных параметрах и закрывается без выполнения каких-либо вычислений.

3. Подключение kicadbom2spec к EESchema в качестве плагина

Возможности программы *EESchema* позволяют использовать сторонние инструменты для создания перечней элементов. Как использовать в данном случае *kicadbom2spec* будет рассказано далее.

3.1 Настройка редактора схем EESchema

Прежде всего можно определить несколько пользовательских полей для элементов схемы. Для этого нужно открыть диалог настроек, выбрав в меню *Настройки-Параметры* и во вкладке *Шаблон имен полей* заполнить поля как показано на рис. 30. Порядок полей не имеет значения.

| Имя поля: | Значение по умолчанию: | Видимый |
|----------------|------------------------|---------|
| Группа | | Скрытые |
| Марка | | Скрытые |
| Класс точности | | Скрытые |
| Тип | | Скрытые |
| Стандарт | | Скрытые |
| Примечание | | Скрытые |

Настройки поля

Имя
Группа

Значение по умолчанию:

☐ Видимый

Добавить

Удалить

Ok Отмена

Рисунок 30. Настройка пользовательских полей

Благодаря данным шаблонам, при открытии редактора свойств элемента, в нем уже будут присутствовать указанные выше поля. Иначе, пришлось бы добавлять поля каждому элементу отдельно.

Из пользовательских полей 2...5 и поля элемента *Значение* формируется наименование элемента. Элементы имеющие одинаковые значения поля *Группа* будут объединены в одну группу с указанным названием. Значение поля *Примечание* помещается в одноименную графу перечня.

Рассмотрим пример образования одной строки перечня элементов:

| Поз. обозначение | Наименование | Кол. | Примечание |
|------------------|---------------------------------|------|------------|
| | | | |
| | Резисторы | | |
| | | | |
| R1 | МЛТ-0,25-1,8кОм±5%-В ОЖ0.467.18 | 1 | 1...2,8кОм |
| | | | |

| ЗНАЧЕНИЕ В ПЕРЕЧНЕ: | ПОЛЕ ЭЛЕМЕНТА: |
|---------------------|-----------------------|
| Резисторы | Группа |
| R1 | Обозначение |
| МЛТ-0,25- | Марка |
| 1,8к | Значение ¹ |
| ±5% | Класс точности |
| -В | Тип |
| ОЖ0.467.18 | Стандарт ² |
| 1...2,8кОм | Примечание |

Примечания:

1. Единицы измерения (в данном случае "Ом") могут добавляться автоматически для резисторов, конденсаторов и индуктивностей если при создании перечня элементов была установлена отметка возле параметра *Добавлять единицы измерения*.

2. Стандарт указан с пробелом в начале.

При образовании наименования разделители не используются, поэтому их нужно указывать в значениях полей.

Если какое-нибудь поле не нужно - его оставляют пустым. Количество элементов подсчитывается автоматически. Если имеется несколько одинаковых элементов идущих подряд, они сводятся в одну строку с указанием первого и последнего элемента и их общего количества.

3.2 Создание и настройка плагина

Следующим этапом является создание нового инструмента генерации перечня элементов (плагина). Для этого в *EESchema* нужно выполнить команду *Инструменты-Сформировать перечень элементов* или нажать соответствующую кнопку на панели инструментов, после чего откроется диалог (рис. 31).

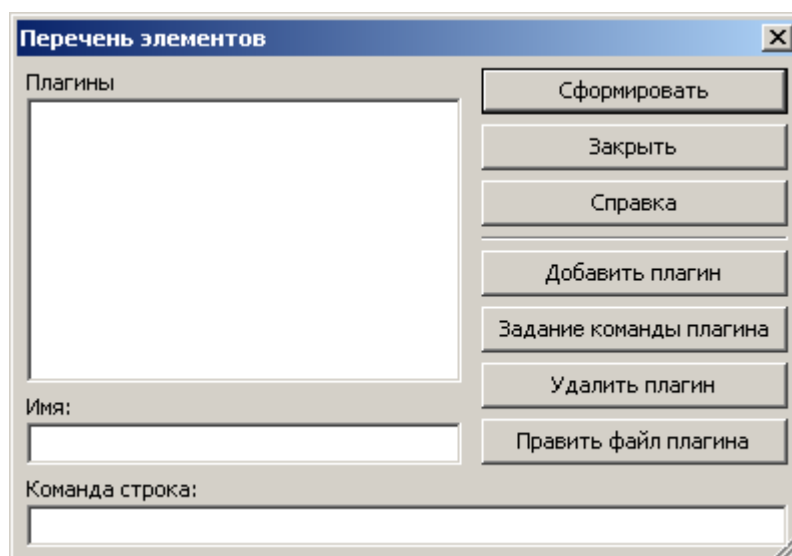


Рисунок 31. Диалог создания перечня элементов в EESchema

Чтобы добавить новый инструмент генерации перечня элементов нужно нажать кнопку *Добавить плагин* и в появившемся окне (рис. 32) ввести его имя (например, «kicadbom2spec») и подтвердить создание нажатием кнопки *Ок*.

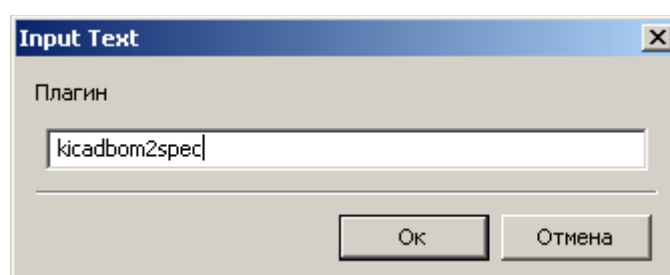


Рисунок 32. Запрос имени плагина

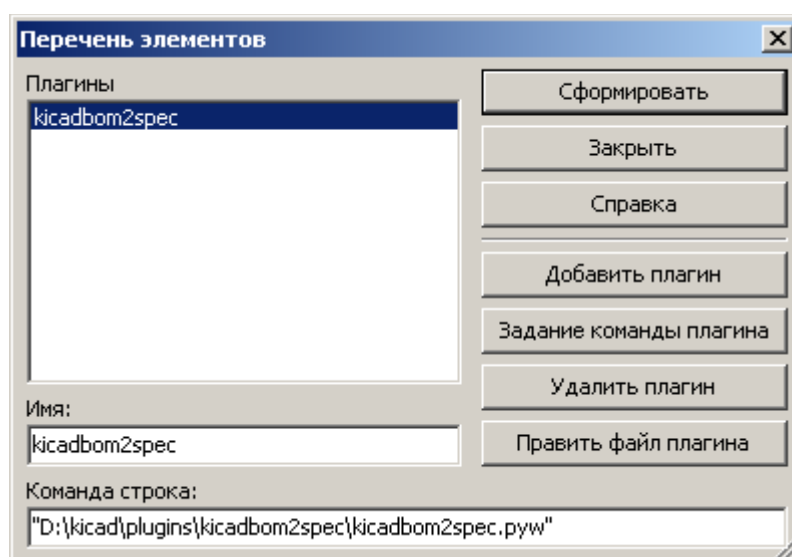


Рисунок 33. Диалог создания перечня элементов с указанным файлом плагина

Теперь, когда плагин создан, его нужно настроить. Для этого нужно нажать кнопку *Задание команды плагина*. Появится стандартный диалог выбора файлов *Файлы плагинов*, в котором нужно найти и выбрать файл *kicadbom2spec.pyw*. После этого в нижнем поле ввода появится полное имя выбранного файла в кавычках (рис. 33). Но этого не достаточно для работы, нужно немного изменить строку запуска плагина следующим образом. В начале строки нужно добавить *pythonw*, а в конце - *"%I" "%O"* (рис. 34). После этих изменений плагин готов к работе. Его можно запустить нажав кнопку *Сформировать*.

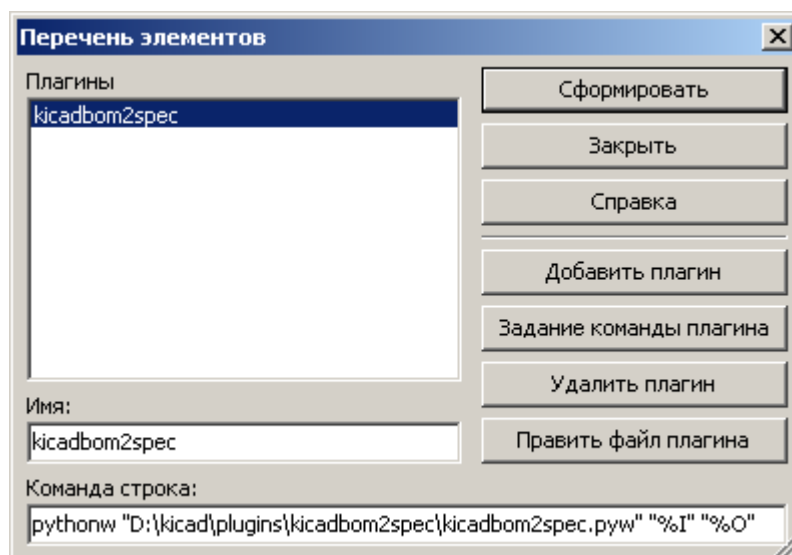


Рисунок 34. Диалог создания перечня элементов с правильно настроенным плагином

Примечание. Между элементами командной строки обязательно должны быть пробелы:

`pythonw[пробел]"D:\kicad\...\kicadbom2spec.pyw"[пробел]"%I"[пробел]"%O"`

Если будет упущен хотя бы один из пробелов, команда будет интерпретирована не верно, что приведет к появлению ошибки.

4. Редактирование настроек программы

Редактор настроек вызывается из меню с помощью команды *Правка-Параметры...* и выглядит как показано на рис. 35. Все настройки разбиты на группы, каждая из которых расположена на отдельной вкладке.

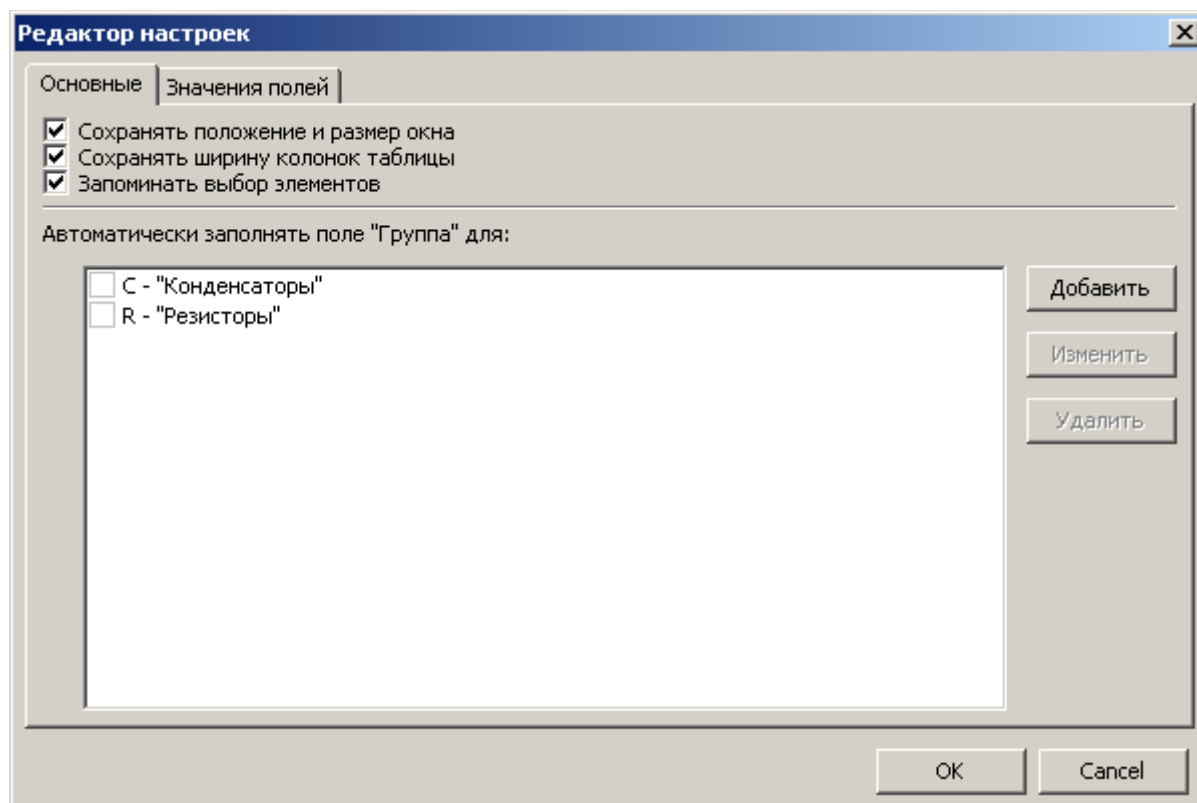


Рисунок 35. Внешний вид редактора настроек

4.1 Основные

Эти параметры относятся к программе в целом. Рассмотрим каждый из параметров в отдельности.

Сохранять положение и размер окна — если этот параметр установлен, то при завершении работы программы будет сохранено положение окна и его размер, а при следующем запуске эти значения будут восстановлены. Если же параметр не установлен, то программа будет каждый раз запускаться с размером и положением на экране заданным по умолчанию.

Сохранять ширину колонок таблицы — если этот параметр установлен, то в процессе работы и при завершении программы ширина столбцов таблицы элементов будет сохраняться и при следующем запуске они будут восстановлены.

Запоминать выбор элементов — если это параметр установлен, то при сохранении файла схемы для элементов, которые исключены из перечня элементов (отметка в таблице снята) будет добавлено новое поле с именем «Исключен из ПЭ». В этом поле можно указать комментарий с описанием причины, по которой элемент был исключен из перечня (только в *EESchema*). При открытии файла схемы у элементов которые содержат поле «Исключен из ПЭ» будет автоматически снята отметка и при создании перечня эти элементы в перечень не попадут.

В общей сложности этот параметр позволяет сохранять состояние выбора элементов в файле схемы.

Автоматически заполнять поле «Группа» — данный параметр позволяет несколько автоматизировать процесс оформления перечня элементов. А именно, для заданных типов элементов автоматически заполнять поле «Группа» при загрузке документа.

Примечание: автоматическое заполнение применяется только к элементам с пустым полем «Группа».

Для организации пар *тип элемента — значение поля «Группа»* предусмотрен специальный редактор, в котором доступны три действия *Добавить*, *Изменить* и *Удалить*. Существующие элементы отображаются в списке и могут включаться или отключаться с помощью отметок.

Чтобы добавить новый элемент нужно нажать на кнопку *Добавить* после чего отобразится диалоговое окно, как показано на рис. 36.

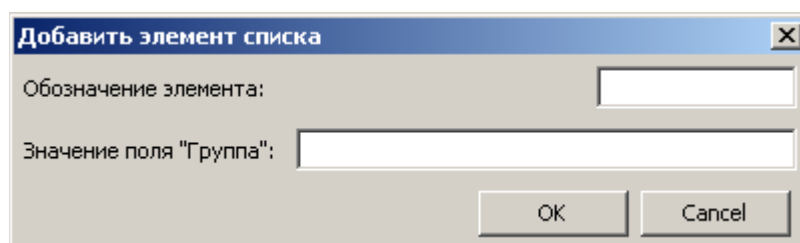


Рисунок 36. Диалог создания элемента списка

В этом диалоге нужно заполнить два поля.

Первое из них *Обозначение элемента* — это один или два символа, с которых начинается обозначение элемента, и по которым можно определить принадлежность элемента к группе. Например, «С» для конденсаторов, «R» для резисторов, «VT» для транзисторов, «D» для аналоговых и цифровых микросхем (если нужно конкретизировать можно указать «DA» для, скажем, стабилизаторов или операционных усилителей) и т. д.

Второе — *Значение поля «Группа»* — здесь нужно задать значение поля «Группа», которое будет автоматически присваиваться элементам, обозначение которых начинается с символов, указанных в предыдущем поле.

Если нужно отредактировать уже существующий элемент, необходимо его выделить и нажать кнопку *Изменить* или произвести двойной щелчок ЛКМ по элементу списка. В результате откроется диалоговое окно (рис. 37), подобное диалогу создания элемента.

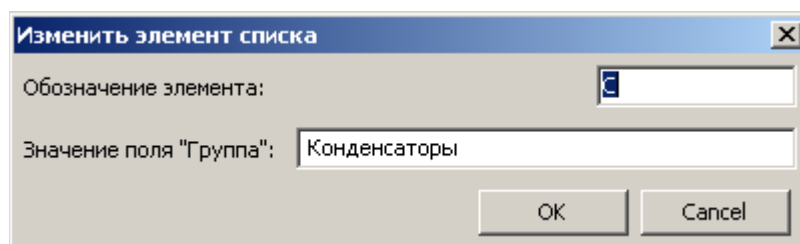


Рисунок 37. Диалог редактирования элемента списка

Для сохранения выполненных изменений нужно нажать кнопку *Ок*, для отмены — *Отмена*.

Чтобы удалить элемент из списка, нужно выделить его и нажать кнопку *Удалить*.

4.2 Значения полей

При оформлении перечня элементов, заполняя поля элементов часто приходится использовать стандартные значения, такие, например, как «Резисторы», «Конденсаторы», «Диоды» и т. д. для поля «Группа» или « $\pm 1\%$ », « $\pm 5\%$ », « $\pm 10\%$ » для поля «Класс точности».

Чтобы ускорить процесс заполнения перечня предусмотрена возможность задавать стандартные значения для определенных полей элементов. Делается это с помощью вкладки *Значения полей* редактора настроек (рис. 38).

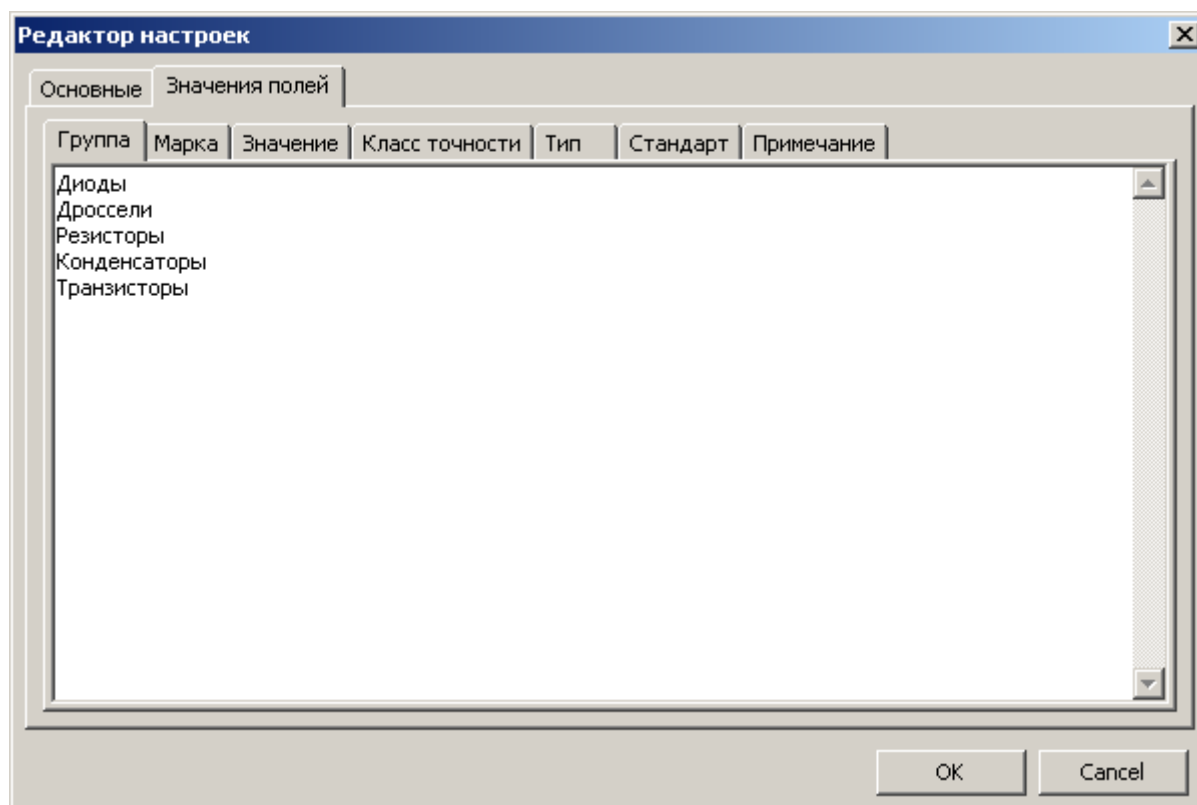


Рисунок 38. Редактор стандартных значений для полей

Для каждой группы можно задать несколько значений, каждое из которых расположено на отдельной строке.

Теперь при редактировании элементов с помощью редактора полей (рис. 28) все эти варианты значений будут доступны в выпадающем меню.

5. Пример создания перечня элементов

Для демонстрации работы программы *kicadbom2spec* в ее директории имеется папка *sample*, в которой находятся файлы небольшой схемы (рис. 39):

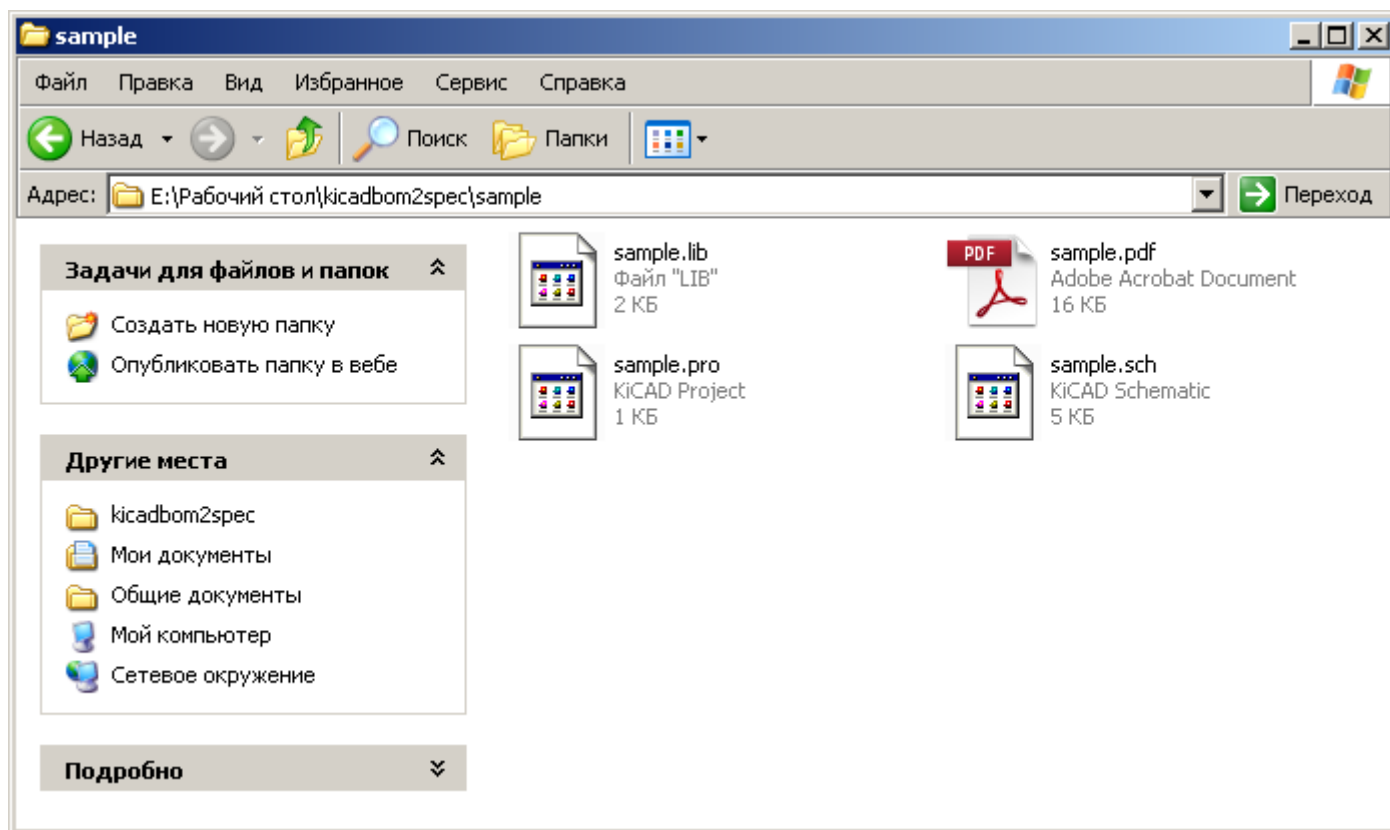


Рисунок 39. Содержимое папки *sample*

- *sample.lib* — библиотека элементов схемы;
- *sample.pdf* — образец готового перечня элементов;
- *sample.pro* — файл проекта KiCAD;
- *sample.sch* — файл схемы (поля элементов уже заполнены в соответствии с данным руководством).

Итак, чтобы создать перечень элементов для схемы *sample.sch* нужно запустить *kicadbom2spec*. Это можно сделать двумя способами:

- 1) непосредственно запустить программу двойным щелчком по файлу *kicadbom2spec.pyw*.

В этом случае откроется пустое окно программы, в котором нужно выполнить команду *Файл-Открыть схему...* или нажать соответствующую кнопку на панели инструментов. В появившемся диалоге выбрать файл схемы *sample.sch* и открыть его. Все элементы схемы будут загружены и отображены в таблице.

- 2) открыть схему *sample.sch* с помощью *EESchema* и выполнить команду *Инструменты-Сформировать перечень элементов* или нажать соответствующую кнопку на панели инструментов (*EESchema* должна быть настроена, как описано в главе 3). Откроется диалог создания перечня элементов (рис. 34) в котором нужно выбрать плагин *kicadbom2spec* и нажать

кнопку *Сформировать*. В результате будет открыта программа *kicadbom2spec* с загруженными элементами в таблице из файла *sample.sch*.

После того как в таблицу *kicadbom2spec* загружены элементы, становится доступной команда *Файл-Создать перечень элементов...* и кнопка на панели инструментов (рис. 40).

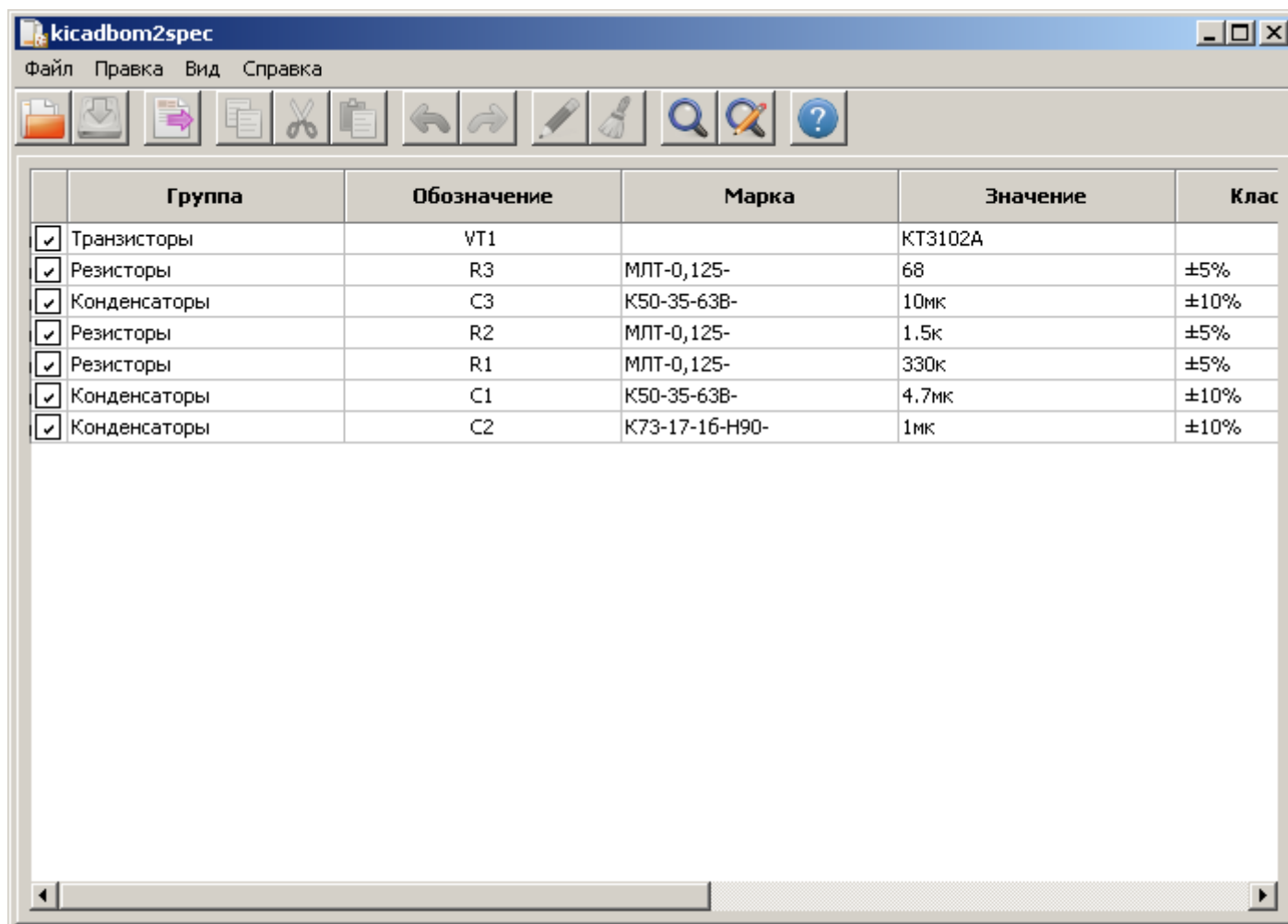


Рисунок 40. Окно программы *kicadbom2spec* с загруженными элементами из *sample.sch*

На данном этапе можно выполнить редактирование полей элементов и сохранить эти изменения в файл схемы. В нашем случае ничего редактировать не нужно, сразу переходим к созданию перечня элементов. Для этого нужно выполнить команду *Файл-Создать перечень элементов...*, откроется диалог создания перечня элементов (рис. 24). В этом диалоге нужно указать имя файла в который будет записан перечень. По умолчанию это имя образуется из имени файла схемы путем замены расширения на **.ods*. Если же данный вариант не устраивает, можно исправить имя в текстовом поле или с помощью диалога выбора файлов нажав кнопку справа от текстового поля.

Дополнительно доступны три параметра:

- *Добавить единицы измерения* — если этот параметр отмечен, то при формировании наименования элемента для резисторов/конденсаторов/индуктивностей (определяется по литерам обозначения *R*, *C*, *L*) в случае отсутствия единиц измерения в значении (например, 10к, 1мк, 2.2н) они будут добавлены автоматически.
- *Все элементы* — если этот параметр отмечен, то перечень будет составлен из всех элементов схемы, т. е. состояние флажка в таблице будет проигнорировано.

- *Открыть перечень элементов* — если этот параметр установлен, то по завершении будет открыт созданный перечень элементов в редакторе по умолчанию.

Для данного примера имя перечня нужно оставить без изменений и поставить отметку для параметра *Добавлять единицы измерения*, завершить создание перечня нажав кнопку *Ок*. После завершения генерации перечня элементов будет показано сообщение (рис. 41) и в папке с примером появится новый файл *sample.ods* (рис. 42). Если перед созданием перечня был установлен параметр *Открыть перечень элементов*, то сообщение о завершении показано не будет, вместо него будет открыт созданный перечень элементов.

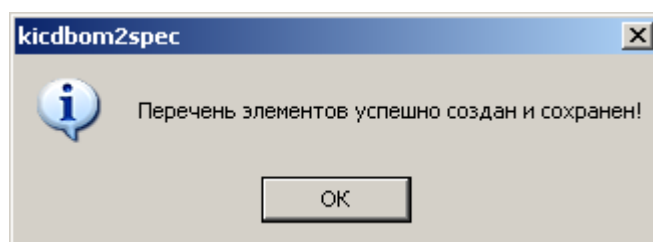


Рисунок 41. Сообщение о создании перечня элементов

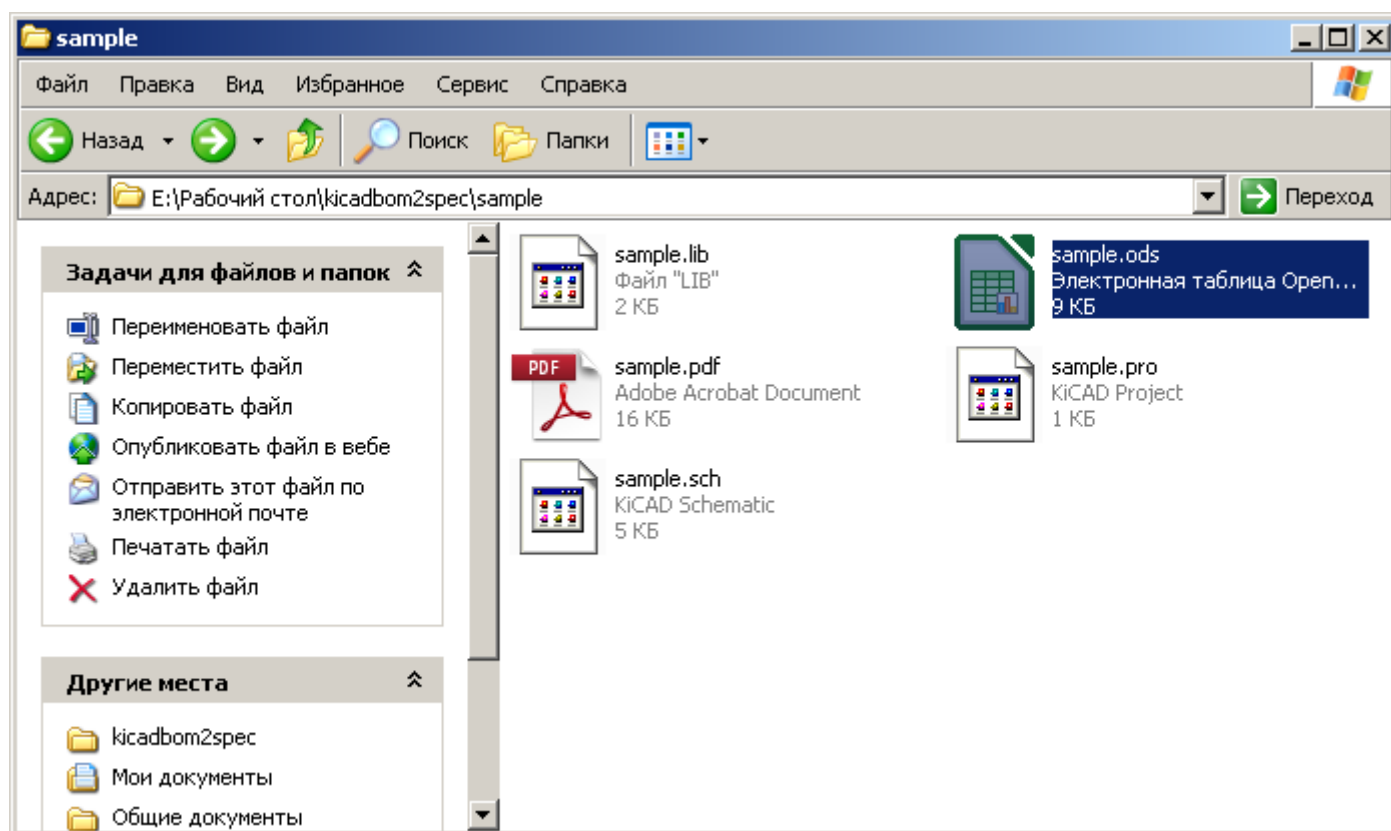


Рисунок 42. Созданный файл перечня элементов

Готово — перечень элементов создан. Чтобы убедиться что все выполнено верно, можно сравнить полученный перечень с образцом *sample.pdf*.

Примечание. При создании перечня элементов из *EESchema* в директории фала схемы автоматически создаются два новых файла с названием схемы и у одного файла расширение **.xml*, а второй без расширения. Они не используются программой *kicadbom2spec* и если в них нет необходимости, могут быть удалены.

kicadbom2spec

Версия 3.4

Руководство пользователя

Вариант для операционных систем MS Windows

Барановский Константин

baranovskiykonstantin@gmail.com

2014