

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

Руководство пользователя

Версия для операционных систем Linux

Содержание

Введение.....	3
1. Установка необходимых программ и компонентов.....	4
1.1 python.....	4
1.2 odfpy.....	4
1.3 opengostfont.....	6
1.4 LibreOffice.....	7
2. Работа с программой kicadbom2spec.....	8
2.1 Режим графического пользовательского интерфейса.....	8
2.2 Режим командной строки.....	10
2.2.1 Параметр -b, --bom.....	14
2.2.2 Параметр -s, --schematic.....	14
2.2.3 Параметр -o, --output.....	14
2.2.4 Параметр -c, --console.....	14
2.2.5 Параметр -v, --verbose.....	14
2.2.6 Параметр -h, --help.....	15
3. Генерация перечня элементов (BOM), пригодного для создания спецификации.....	16
3.1 Настройка редактора схем EESchema.....	16
3.2 Создание перечня элементов (BOM).....	17

Введение

kicadbom2spec — программа, написанная на языке python и предназначена для создания файлов спецификации в виде таблиц *.ods в формате Open Document Format, оформленных согласно требованиям ЕСКД, используя файлы перечней элементов (BOM), созданных с помощью программы EEschema из пакета программ САПР KiCAD.

Имеется два режима работы программы. Первый — режим графического пользовательского интерфейса, в котором отображается окно с текстовыми полями и кнопками. С их помощью можно задать названия необходимых файлов вручную или используя диалоги открытия или сохранения файлов. Второй — режим командной строки, в котором все необходимые данные передаются с помощью параметров.

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.

Файл спецификации 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 спецификации был создан с помощью свободного офисного пакета *LibreOffice* с применением открытого чертежного шрифта *opengostfont*. Наличие данных элементов не обязательно, но крайне желательно, дабы получить спецификацию с корректным форматированием.

О том как установить нужные элементы для корректной работы *kicadbom2spec* будет рассказано далее на примере операционной системы Ubuntu 13.04.

1.1 python

Практически во всех современных Linux дистрибутивах предварительно уже установлен интерпретатор языка python и готов к работе. Чтобы убедиться в этом можно запустить программу *kicadbom2spec* (двойным щелчком по файлу *kicadbom2spec.pyw*), появится окно, будут доступны поля для ввода и диалоги для выбора нужных файлов, но создать файл спецификации не выйдет, так как отсутствует необходимый модуль *odfpy*.

1.2 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.

» [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 1. Загрузка odfpy

Далее будет подробно рассмотрен процесс построения и установки модуля *odfpy*.

Итак, для того, чтобы установить *odfpy* 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. 1).
- после нажатия на эту кнопку появится диалоговое окно загрузки файлов (рис. 2) в котором отмечаем пункт «Открыть в» и подтверждаем начало загрузки нажатием на кнопку «ОК».

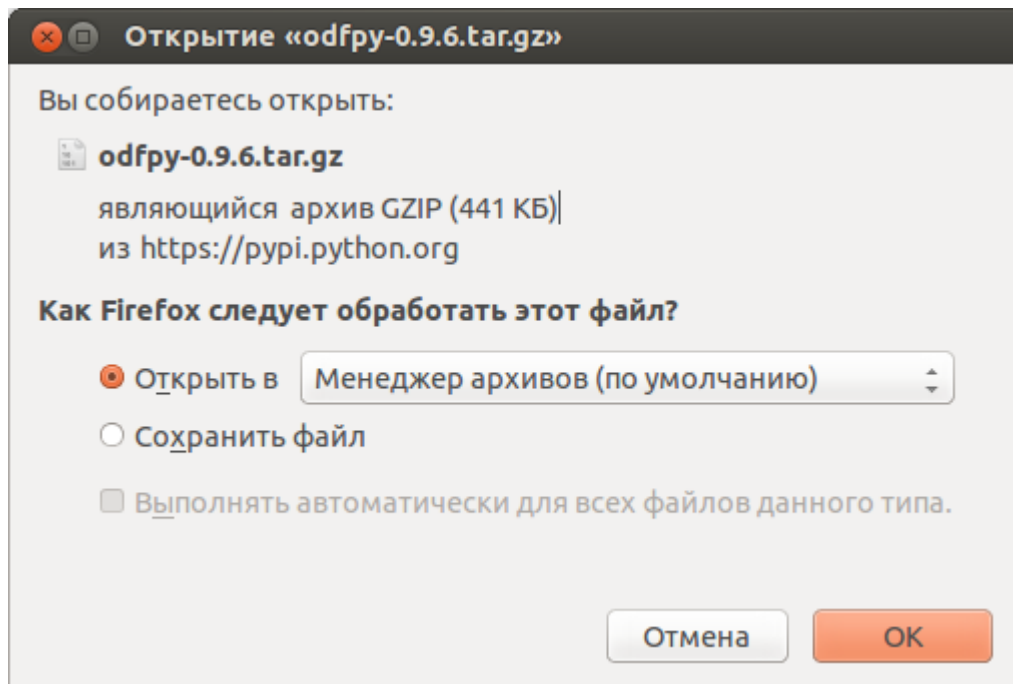


Рисунок 2. Открытие загрузки

- после загрузки файла откроется менеджер архивов в котором будет отображено содержимое загруженного пакета, а именно, папка с файлами исходных кодов «odfpy-0.9.6». Ее нужно извлечь в Домашнюю папку.
- если открыть папку «odfpy-0.9.6», можно увидеть, что 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 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 запустить скрипт с параметром «install».
- для дальнейшей работы понадобится Терминал, запускаем его из меню приложений.
- по умолчанию рабочей директорией является Домашняя папка пользователя. Нужно сделать текущей папку с файлами исходных кодов модуля *odfpy*, которая ранее была загружена и распакована в Домашнюю папку. Для этого выполняем в Терминале следующую команду:

```
$cd odfpy-0.9.6
```

или

```
$cd ~/odfpy-0.9.6
```

- для того, чтобы далее можно было работать с файлом скрипта «setup.py» как с программой, нужно изменить его свойства. Делаем это с помощью следующей команды:

```
$chmod +x setup.py
```

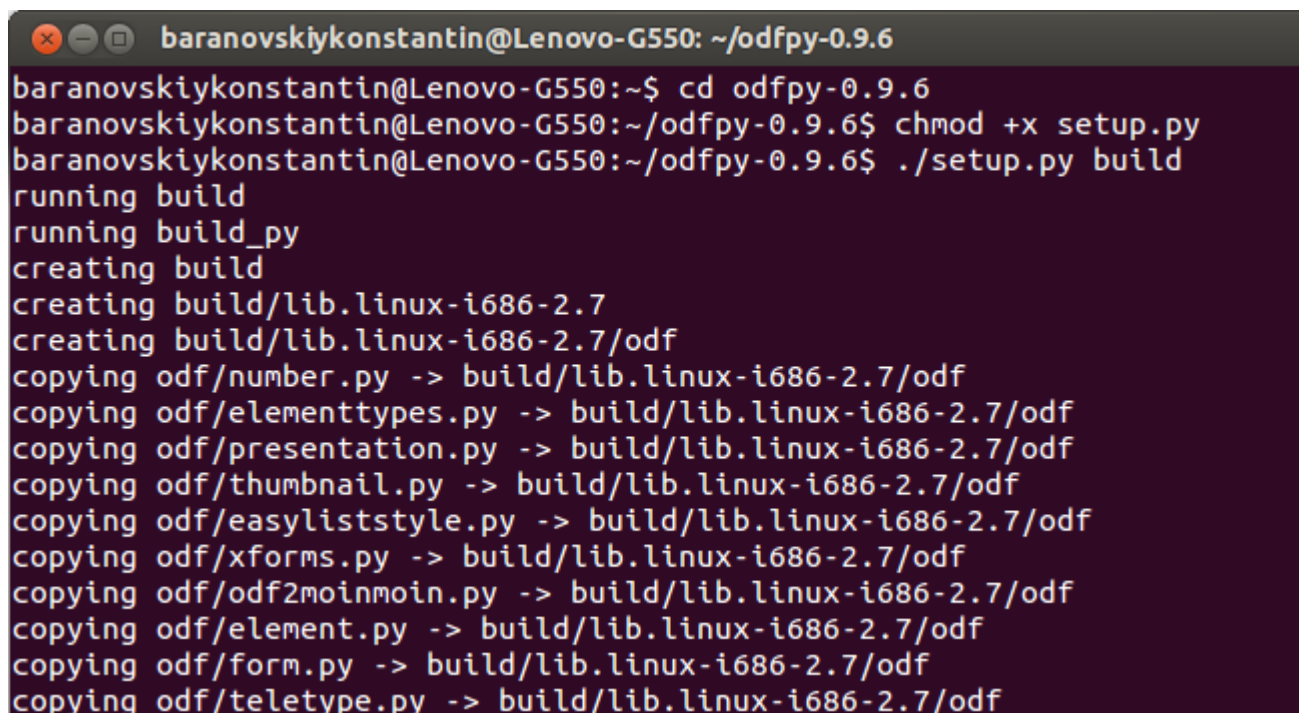
- теперь можно собрать модуль *odfpy*, выполнив следующую команду:

```
$./setup.py build
```

или

```
$python setup.py build
```

- в процессе построения в Терминале будут появляться строки с информацией о текущей операции (рис. 3). После завершения в папке с файлами исходных кодов модуля появится новая папка «build», содержимое которой подлежит дальнейшей установке.



```

baranovskiykonstantin@Lenovo-G550: ~/odfpy-0.9.6
baranovskiykonstantin@Lenovo-G550:~$ cd odfpy-0.9.6
baranovskiykonstantin@Lenovo-G550:~/odfpy-0.9.6$ chmod +x setup.py
baranovskiykonstantin@Lenovo-G550:~/odfpy-0.9.6$ ./setup.py build
running build
running build_py
creating build
creating build/lib.linux-i686-2.7
creating build/lib.linux-i686-2.7/odf
copying odf/number.py -> build/lib.linux-i686-2.7/odf
copying odf/elementtypes.py -> build/lib.linux-i686-2.7/odf
copying odf/presentation.py -> build/lib.linux-i686-2.7/odf
copying odf/thumbnail.py -> build/lib.linux-i686-2.7/odf
copying odf/easyliststyle.py -> build/lib.linux-i686-2.7/odf
copying odf/xforms.py -> build/lib.linux-i686-2.7/odf
copying odf/odf2moinmoin.py -> build/lib.linux-i686-2.7/odf
copying odf/element.py -> build/lib.linux-i686-2.7/odf
copying odf/form.py -> build/lib.linux-i686-2.7/odf
copying odf/teletype.py -> build/lib.linux-i686-2.7/odf

```

Рисунок 3. Процесс сборки модуля *odfpy*

- чтобы установить собранный модуль нужно выполнить в Терминале следующую команду:

```
$./setup.py install
```

или

```
$python setup.py install
```

- ход установки будет выглядеть подобно ходу построения. Появление приглашения командной строки свидетельствует о завершении установки.
- папка «*odfpy-0.9.6*» больше не нужна, ее можно удалить с помощью файлового менеджера или выполнив следующую команду в Терминале:

```
$rm -rf ~/odfpy-0.9.6
```
- готово, модуль *odfpy* собран и установлен, можно приступать к созданию спецификации.

1.3 opengostfont

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».

Загруженные файлы шрифтов нужно распаковать в папку «.fonts» (это скрытая папка и по умолчанию не отображается в файловом менеджере, нужно включить отображение скрытых папок и файлов), которая расположена в Домашней папке. Возможно в Вашей Домашней папке папка «.fonts» отсутствует, тогда ее нужно создать самостоятельно.

1.4 LibreOffice

Во всех современных дистрибутивах Linux уже имеется предварительно установленный офисный пакет. Это может быть LibreOffice, OpenOffice или другой. Как правило открытые офисные пакеты в полной мере поддерживают Open Document Format и их можно использовать для просмотра и редактирования спецификации.

Если же в Вашей системе отсутствует офисный пакет или не корректно отображается спецификация можно загрузить файл установки LibreOffice с официального сайта <http://www.libreoffice.org> и установить его. Процесс установки LibreOffice не отличается от установки других программ и не должен вызвать сложности.

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. 4).

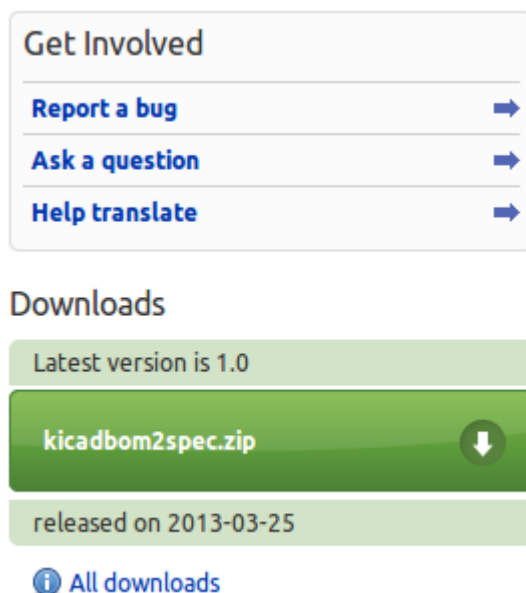


Рисунок 4. Загрузка kicadbom2spec

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.pyw» и поместить ее, скажем, на рабочий стол или в папку с проектом KiCAD.

2.1 Режим графического пользовательского интерфейса

Это основной режим. Для запуска программы нужно открыть файл «kicadbom2spec.pyw» двойным щелчком мыши, после чего появится окно, как показано на рис. 5

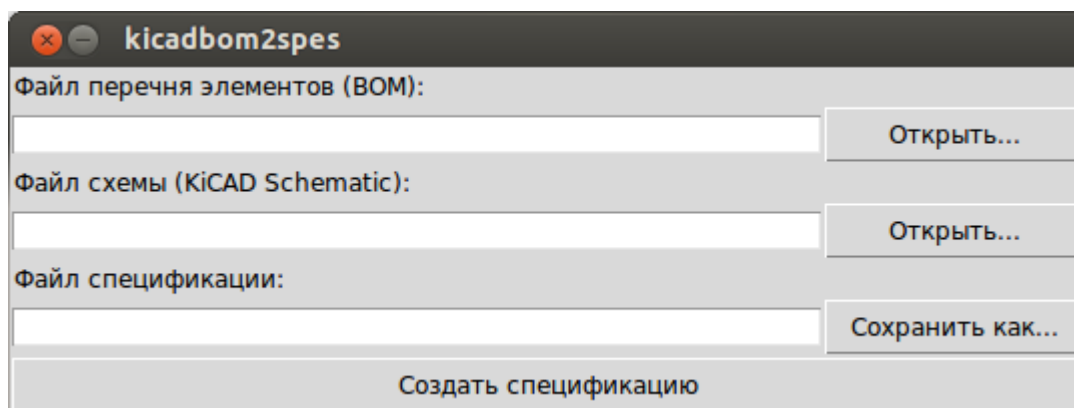


Рисунок 5. Окно программы kicadbom2spec

Окно состоит из трех полей ввода с подписями и четырех кнопок.

В первое текстовое поле нужно обязательно ввести полное имя файла перечня элементов (включая путь к файлу). Это можно сделать вручную, набрав текст с клавиатуры, либо с помощью диалога открытия файлов (рис. 6), нажав на кнопку «Открыть...», расположенную возле текстового поля.

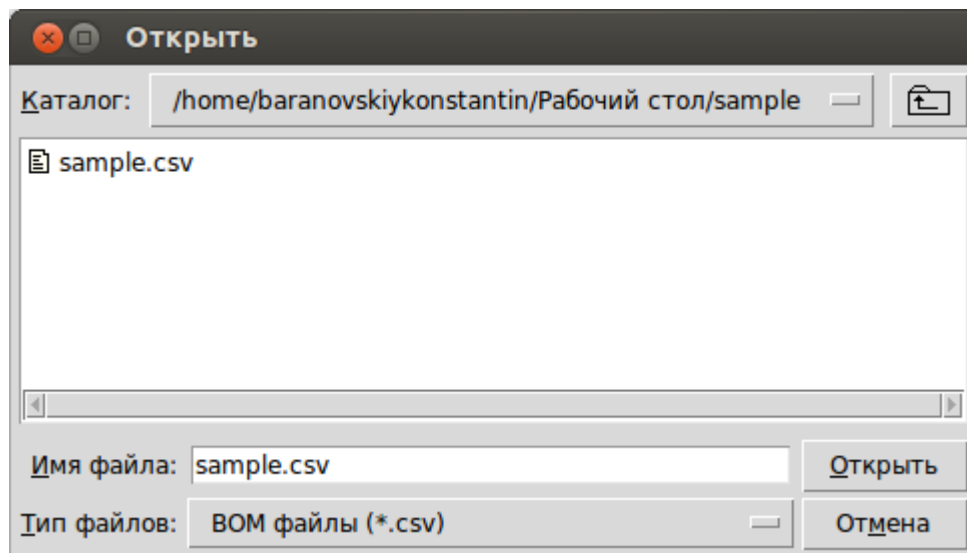


Рисунок 6. Диалог открытия файлов

Во втором поле указывается имя файла схемы в формате KiCAD Schematic аналогичным образом. Данное поле не является обязательным для заполнения. Если файл схемы указан, то из него будут считаны значения полей основной надписи и занесены в спецификацию. В противном случае в основную надпись спецификации будут занесены только номера страниц.

В третьем поле нужно обязательно указать полное имя файла спецификации. Задать его можно так же, как и в предыдущих случаях, за одним исключением, вместо диалога открытия файлов будет использован диалог сохранения файлов (рис. 7) при нажатии на кнопку «Сохранить как...».

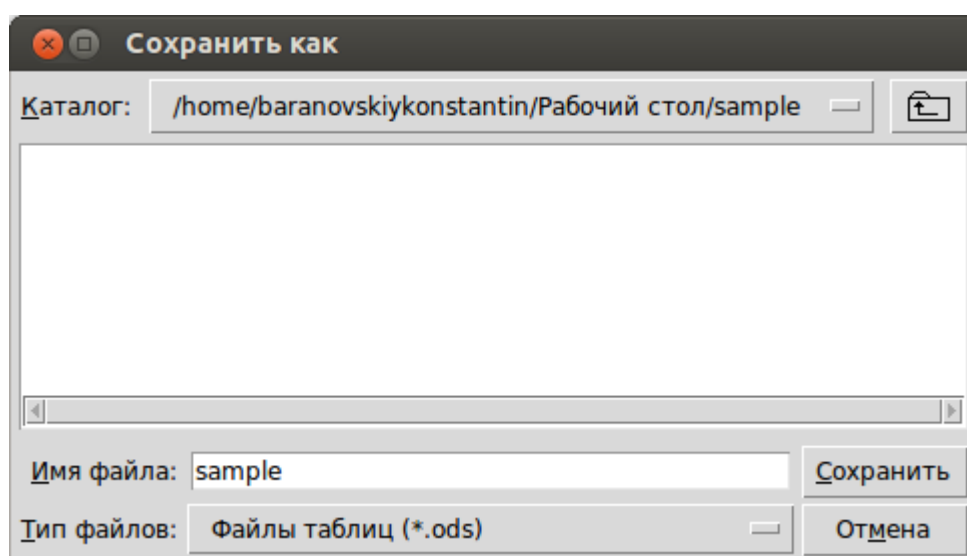


Рисунок 7. Диалог сохранения файлов

Как правило файлы проектов KiCAD (чертежи схем, печатных плат и пр.) хранятся в отдельной

папке и имеют одинаковые имена (за исключением расширений). Благодаря этому процесс загрузки файлов удалось несколько автоматизировать и он происходит следующим образом. В первую очередь нужно открыть файл перечня элементов нажав соответствующую кнопку «Открыть...». В появившемся диалоговом окне выбрать нужный файл перечня элементов. Далее проверяется, есть ли в той же папке файл схемы с таким же именем и если есть, то он сразу же загружается в соответствующее поле. На основе имени файла перечня элементов создается имя файла спецификации (путем замены расширения) и тоже заносится в соответствующее текстовое поле. Если же файл схемы или спецификации уже определен, то он остается без изменений.

После того как все необходимые файлы определены можно создать спецификацию, нажав на кнопку «Создать спецификацию». По завершению появится информационное окно (рис. 8) подтверждающее успешность выполнения операции. В случае возникновения ошибки будет выведено сообщение с ее описанием.

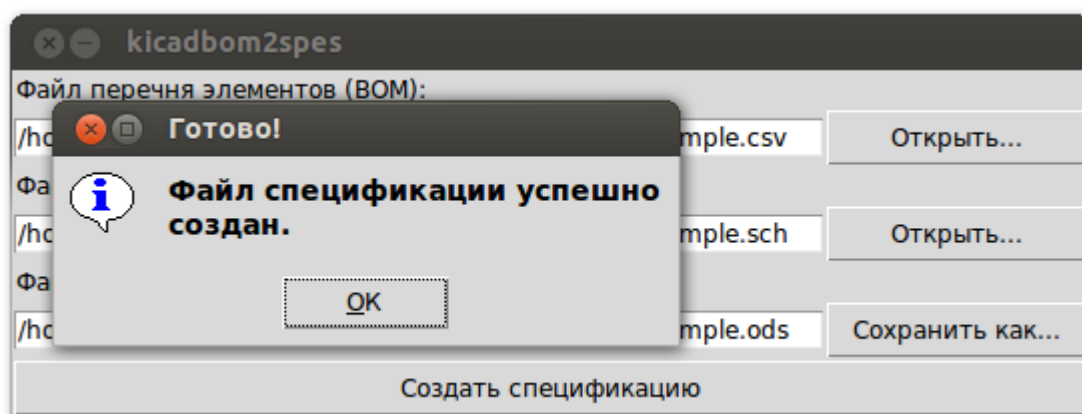


Рисунок 8. Сообщение о создании спецификации

Созданную спецификацию можно просмотреть и, если необходимо, отредактировать с помощью имеющегося офисного пакета.

2.2 Режим командной строки

Этот режим может быть использован при пакетной обработке файлов или для взаимодействия с другими приложениями/скриптами.

Для того чтобы получить краткую справку о работе с *kicadbom2spec* в режиме командной строки нужно ввести следующую команду:

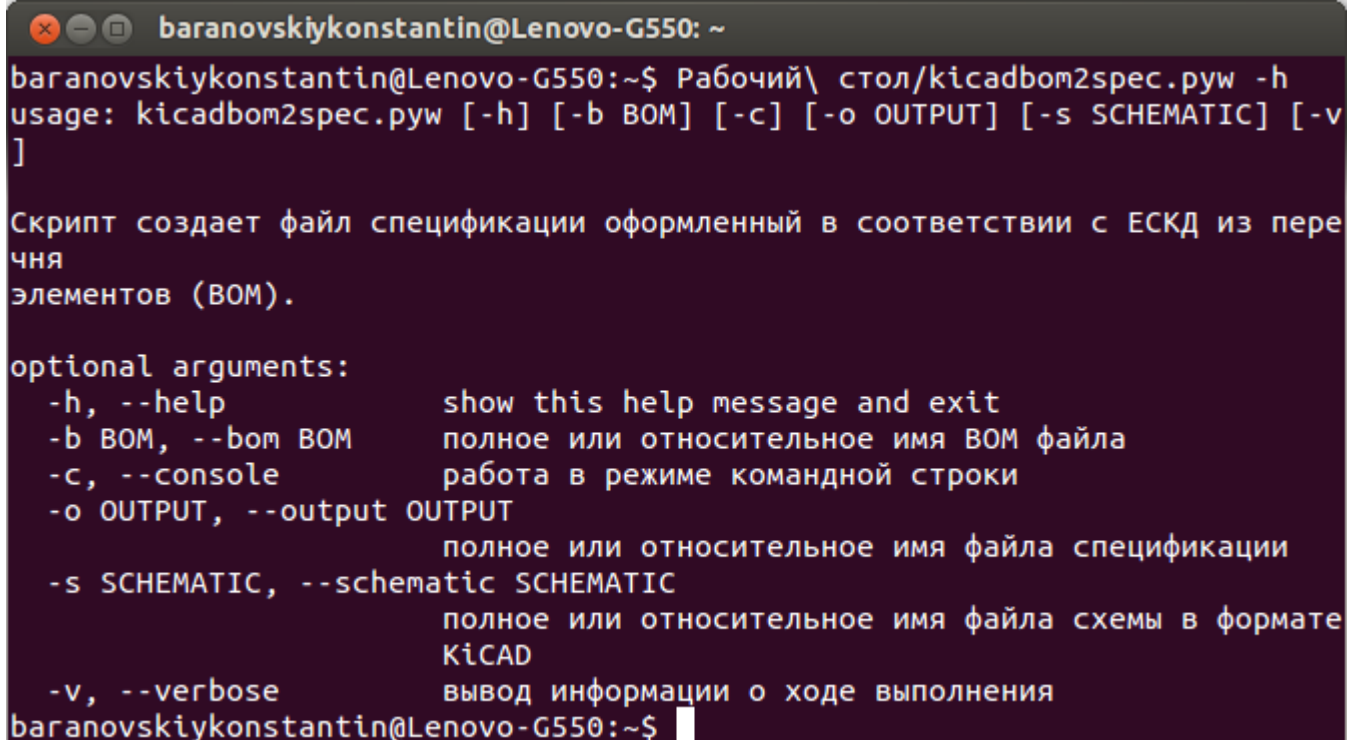
```
$python kicadbom2spec.pyw -h
```

или

```
$python kicadbom2spec.pyw --help
```

то есть запустить скрипт на выполнение с заданным параметром `-h` или `--help`, что эквивалентно. Результат выполнения описанной команды показан на рис. 9.

При работе с программой в режиме командной строки нужно обязательно задать лишь имя файла перечня элементов (BOM), остальные имена файлов будут найдены автоматически, подобно тому, как это делается в режиме графического интерфейса. Если же имя файла схемы или спецификации отличается от имени файла перечня элементов, то их можно задать с помощью соответствующих параметров.



```

baranovskiykonstantin@Lenovo-G550: ~
baranovskiykonstantin@Lenovo-G550:~$ Рабочий\ стол/kicadbom2spec.pyw -h
usage: kicadbom2spec.pyw [-h] [-b BOM] [-c] [-o OUTPUT] [-s SCHEMATIC] [-v]

Скрипт создает файл спецификации оформленный в соответствии с ЕСКД из пере-
чня
элементов (BOM).

optional arguments:
  -h, --help                show this help message and exit
  -b BOM, --bom BOM         полное или относительное имя BOM файла
  -c, --console             работа в режиме командной строки
  -o OUTPUT, --output OUTPUT
                           полное или относительное имя файла спецификации
  -s SCHEMATIC, --schematic SCHEMATIC
                           полное или относительное имя файла схемы в формате
                           KiCAD
  -v, --verbose             вывод информации о ходе выполнения
baranovskiykonstantin@Lenovo-G550:~$

```

Рисунок 9. Получение справки в командной строке

При создании спецификации проверяется расширение заданного имени файла и если оно отсутствует или не корректно, то будет добавлено расширение «.ods».

В случае возникновения иных ошибочных ситуаций, будут выводиться краткие сообщения об ошибке.

Рассмотрим пример создания спецификации с помощью командной строки. Для примера на рабочем столе создана папка «sample», в которой расположен проект KiCAD, состоящий из четырех файлов:

sample.pro – файл проекта KiCAD;

sample.sch – файл схемы в формате KiCAD Schematic;

sample.csv – файл перечня элементов (BOM);

kicadbom2spec.pyw – ссылка на программу *kicadbom2spec*.

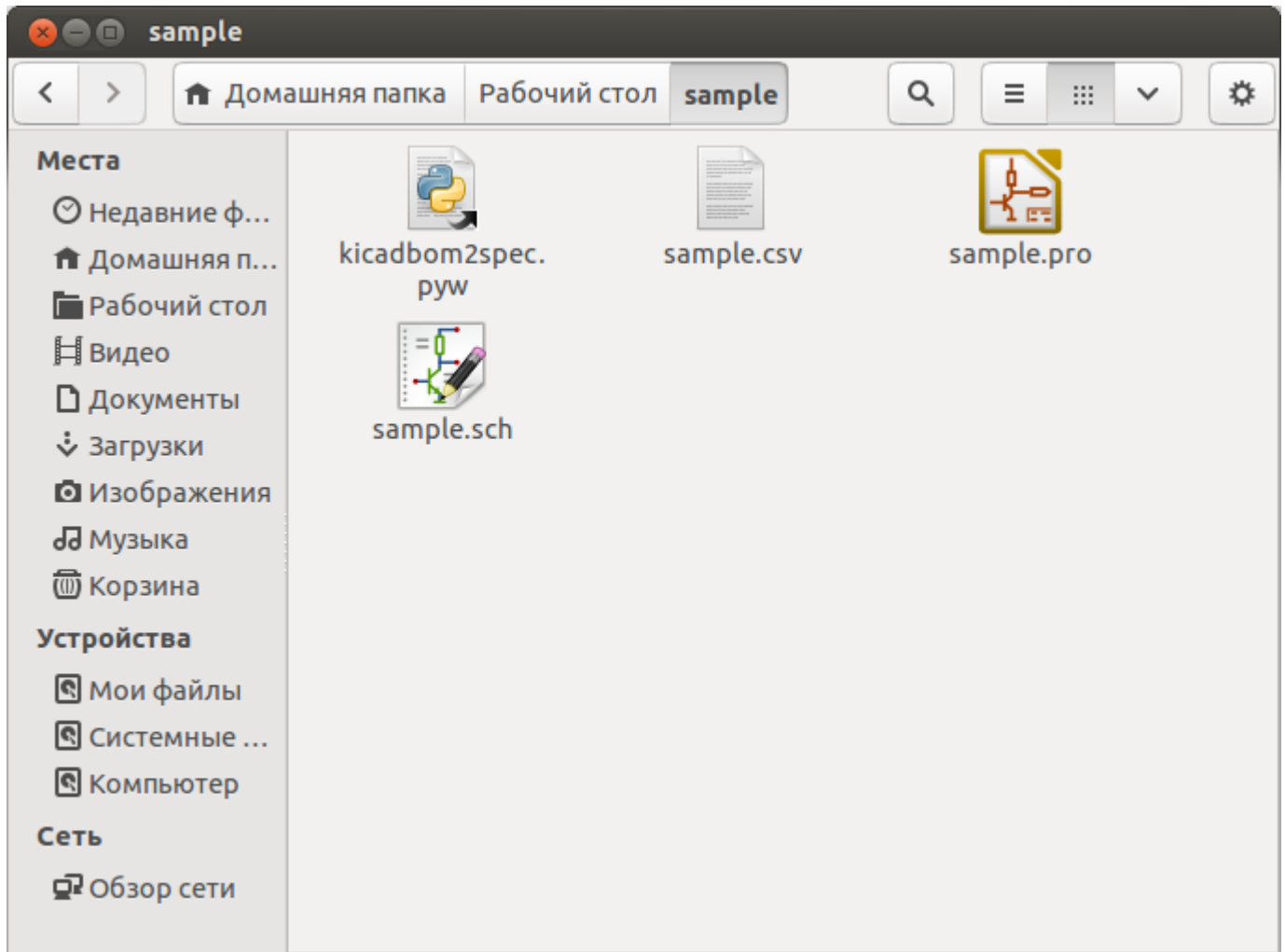


Рисунок 10. Содержимое папки sample

Для начала работы запускаем Терминал. По умолчанию рабочей директорией будет папка текущего пользователя (о чем свидетельствует приглашение командной строки). Нам нужно сделать директорией по умолчанию нашу папку с проектом, для этого нужно ввести следующую команду:

```
$cd Рабочий\ стол/sample/
```

или

```
$cd ~/Рабочий\ стол/sample/
```

С помощью последней команды можно перейти к папке «sample» с любой текущей директории.

Теперь, когда есть все необходимые файлы и командная строка настроена можно создать спецификацию с помощью следующей команды:

```
$python kicadbom2spec.pyw -c -b sample.csv -v
```

Параметр `-v` включает вывод сообщений о текущей выполняемой операции.

После успешного создания спецификации содержимое окна командной строки примет вид, подобный рис. 11, а в папке «sample» появится файл спецификации «sample.ods» (рис. 12).

Далее будет рассмотрено назначение каждого из имеющихся параметров.

```
baranovskiykonstantin@Lenovo-G550: ~/Рабочий стол/sample
baranovskiykonstantin@Lenovo-G550:~$ cd Рабочий\ стол/sample/
baranovskiykonstantin@Lenovo-G550:~/Рабочий стол/sample$ python kicadbom2spec.pyw -c -b sample.csv -v
Файл перечня элементов загружен.
Найден файл схемы. Значения полей основной надписи будут скопированы в спецификацию.
Файл спецификации успешно создан.
baranovskiykonstantin@Lenovo-G550:~/Рабочий стол/sample$
```

Рисунок 11. Создание спецификации с помощью командной строки

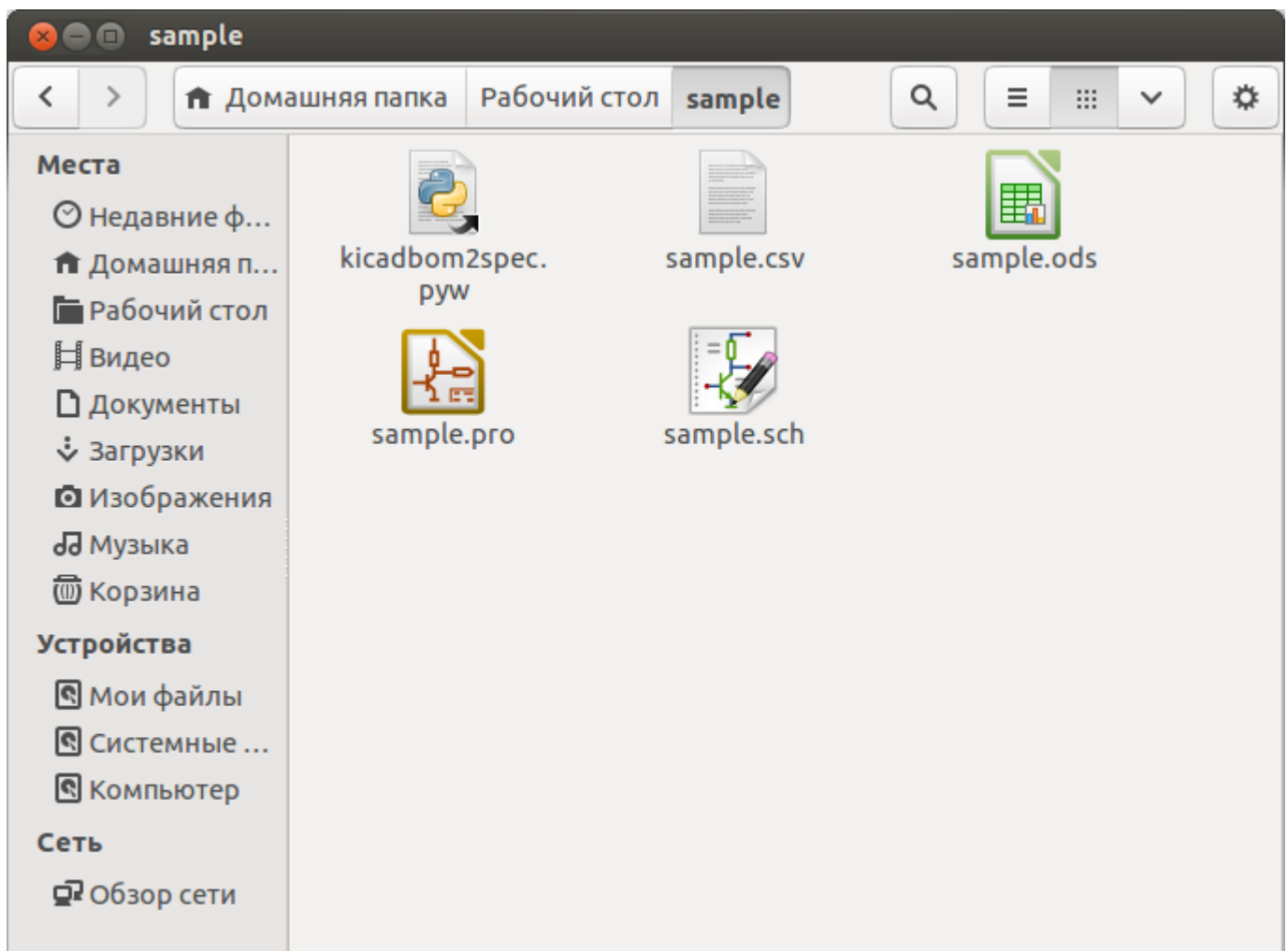


Рисунок 12. Созданный файл спецификации

2.2.1 Параметр -b, --bom

С помощью этого параметра в программу передается имя файла перечня элементов (BOM). Его можно применять как в режиме командной строки, так и в режиме графического пользовательского интерфейса. В режиме командной строки этот параметр является обязательным. Имя файла может быть полным или относительным (относительно текущей директории командной строки).

Пример использования:

```
$python kicadbom2spec.pyw -c -b ~/myproject/device.cvs
```

2.2.2 Параметр -s, --schematic

С помощью этого параметра в программу передается имя файла схемы в формате KiCAD Schematic. Его можно применять как в режиме командной строки, так и в режиме графического пользовательского интерфейса. Он используется в том случае, если имя файла схемы отличается от имени файла перечня элементов или расположен в другой папке. Имя файла может быть полным или относительным (относительно текущей директории командной строки).

Пример использования:

```
$python kicadbom2spec.pyw -b ../bom/bom.csv -s ../sch/schematic.sch
```

2.2.3 Параметр -o, --output

С помощью этого параметра в программу передается имя файла в который будет сохранена спецификация. Его можно применять как в режиме командной строки, так и в режиме графического пользовательского интерфейса. Он используется в том случае, если имя файла спецификации должно быть отличным от имени файла перечня элементов или расположен в другой папке. Имя файла может быть полным или относительным (относительно текущей директории командной строки). Расширение файла должно быть «.ods», если это не так, то в режиме графического пользовательского интерфейса будет показано сообщение об ошибке, а в режиме командной строки автоматически добавится расширение «.ods» без вывода каких-либо сообщений.

Пример использования:

```
$python /usr/local/share/kicad/addons/kicadbom2spec.pyw -c -b project  
s/clock/clock.csv -o documents/Спецификация -v
```

2.2.4 Параметр -c, --console

Этот параметр указывает программе работать в режиме командной строки без использования графического пользовательского интерфейса. Параметр не принимает никаких данных.

2.2.5 Параметр -v, --verbose

С помощью этого параметра можно включить вывод информации о ходе выполнения программы. В результате в командной строке будут появляться сообщения с описанием операции, которая выполняется в данный момент. Параметр не принимает никаких данных.

2.2.6 Параметр -h, --help

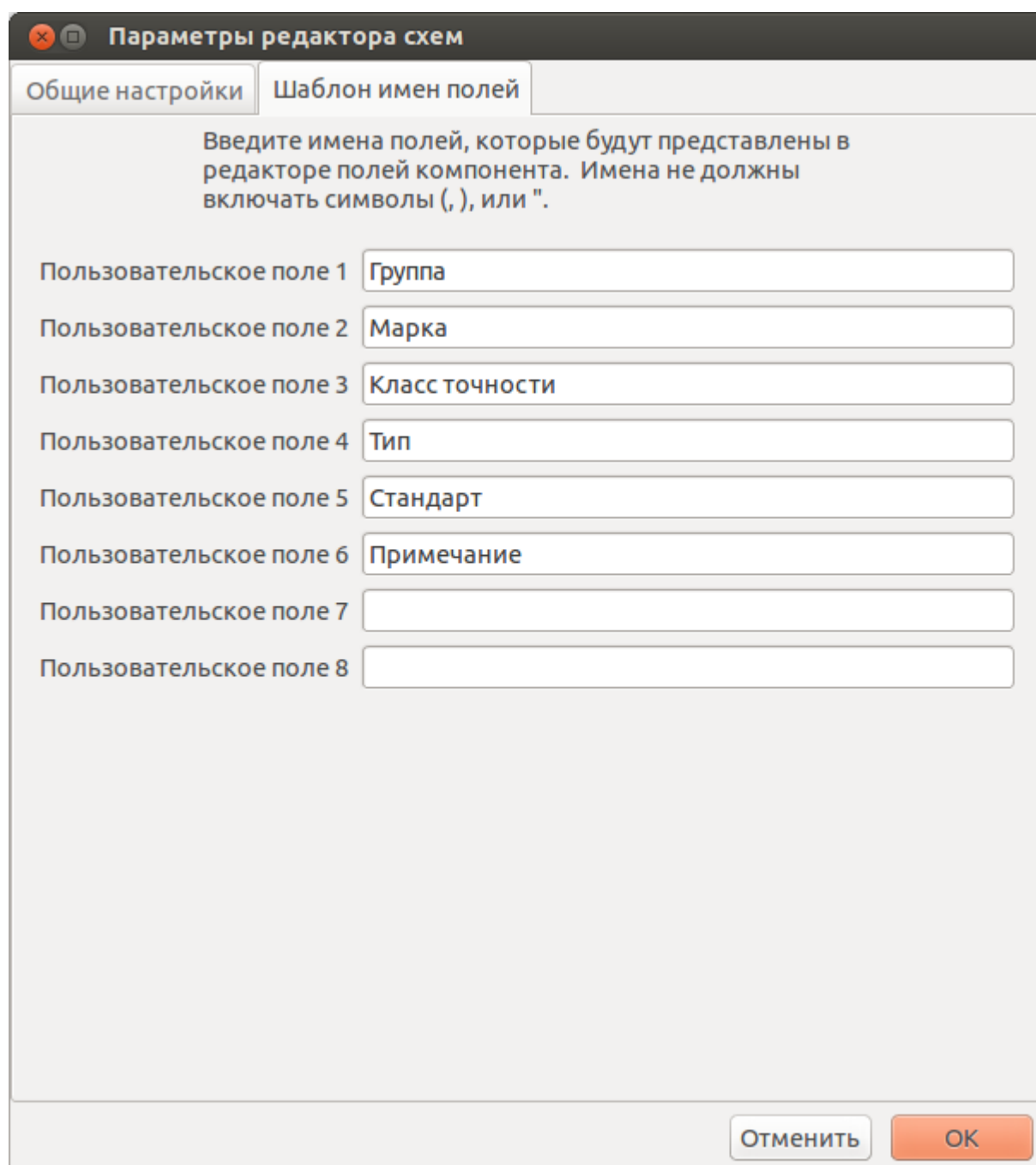
С этим параметром программа выводит справочное сообщение с краткой информацией о программе и имеющихся параметрах и закрывается без выполнения каких-либо вычислений.

3. Генерация перечня элементов (BOM), пригодного для создания спецификации

Для того, чтобы сгенерировать спецификацию должным образом необходимо правильно составить перечень элементов. Для этого сначала нужно настроить редактор схем KiCAD Eeschema.

3.1 Настройка редактора схем EEschema

Сперва нужно 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. 13.



The image shows a dialog box titled "Параметры редактора схем" (Parameters of the schematic editor). It has two tabs: "Общие настройки" (General settings) and "Шаблон имен полей" (Field name template). The "Шаблон имен полей" tab is selected. The dialog contains a text instruction: "Введите имена полей, которые будут представлены в редакторе полей компонента. Имена не должны включать символы (,), или \"." (Enter field names that will be presented in the component field editor. Names must not contain symbols (,), or "). Below this instruction are eight input fields labeled "Пользовательское поле 1" through "Пользовательское поле 8". The fields are filled with the following text: "Группа", "Марка", "Класс точности", "Тип", "Стандарт", "Примечание", and two empty fields. At the bottom right of the dialog are two buttons: "Отменить" (Cancel) and "ОК".

Пользовательское поле	Значение
Пользовательское поле 1	Группа
Пользовательское поле 2	Марка
Пользовательское поле 3	Класс точности
Пользовательское поле 4	Тип
Пользовательское поле 5	Стандарт
Пользовательское поле 6	Примечание
Пользовательское поле 7	
Пользовательское поле 8	

Рисунок 13. Настройка пользовательских полей

Пользовательские поля 7 и 8 могут быть использованы по усмотрению пользователя, при создании спецификации используются только первые шесть полей, остальные игнорируются.

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 спецификации.

Рассмотрим пример образования одной строки спецификации:

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

ЗНАЧЕНИЕ В СПЕЦИФИКАЦИИ:	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") добавляются автоматически для резисторов, конденсаторов и индуктивностей если они отсутствуют.

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 перечня элементов (BOM)

После заполнения полей элементов и завершения работы над схемой можно создать список элементов. Для этого необходимо нажать на панели инструментов кнопку "Сформировать перечень элементов и материалов" или выбрать в меню "Инструменты → Сформировать перечень элементов". В

появившемся диалоговом окне нужно задать следующие параметры (рис. 14):

Формат вывода: Импорт в эл.таблицу (по обозн.)

Разделитель полей для импорта в эл.таблицу: Tab

Добавить в список: Обозначение

Поля пользователя 1...6 (можно выбрать "Все поля пользователя")

Создать перечень элементов нажав кнопку "ОК" диалогового окна.

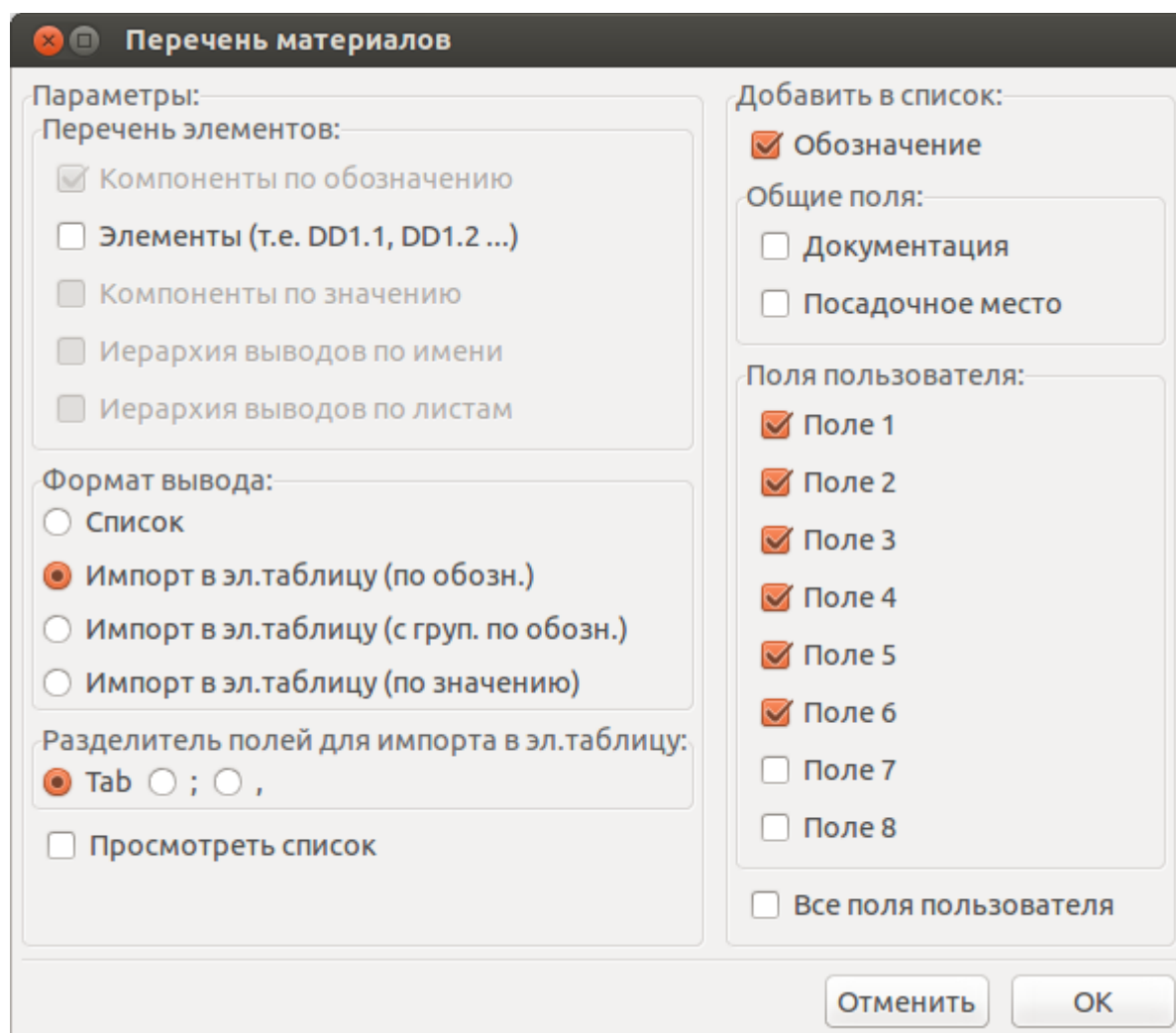


Рисунок 14. Настройки генератора BOM файлов

Созданный таким образом перечень элементов можно использовать для генерации спецификации.

kicadbom2spec
Руководство пользователя
Версия для операционных систем Linux
Барановский Константин
baranovskiykonstantin@gmail.com
2013