sphinxcontrib-eagle Documentation

Release 0.0.5

ponty

January 26, 2012

CONTENTS

sphinxcontrib-eagle

Date January 26, 2012

PDF sphinxcontrib-eagle.pdf

CHAPTER

ONE

ABOUT

This Sphinx 1.0 extension exports eagle partlist or image (2D/3D) of schematic or board during the build step and includes them into the documentation.

Links:

- home: https://github.com/ponty/sphinxcontrib-eagle
- documentation: http://ponty.github.com/sphinxcontrib-eagle

Features:

- eagexp is used for processing

CHAPTER

TWO

BASIC USAGE

```
.. eagle-image:: singlesided.sch
     :resolution: 100
     :scale: 30 %

.. eagle-image3d:: singlesided.brd

.. eagle-partlist:: singlesided.sch
       :header: part, value
```

CHAPTER

THREE

HOW IT WORKS

1. export image or text by eagle using eagexp
2. include image or text into documentation

CHAPTER

FOUR

INSTALLATION

4.1 General

- install eagle
- install povray (optional for 3D)
- install pip
- install pyvirtualdisplay , xvfb , xephyr (optional for background processing)
- install the program:

```
# as root
pip install sphinxcontrib-eagle
```

4.2 Ubuntu

```
sudo apt-get install eagle
sudo apt-get install povray
sudo apt-get install python-pip

# optional for background processing
sudo apt-get install xvfb xserver-xephyr

sudo pip install sphinxcontrib-eagle
```

4.3 Uninstall

```
# as root
pip uninstall sphinxcontrib-eagle
```

CHAPTER

FIVE

USAGE

5.1 Configuration

Add `sphinxcontrib.eagle` to `extensions` list in `conf.py`:

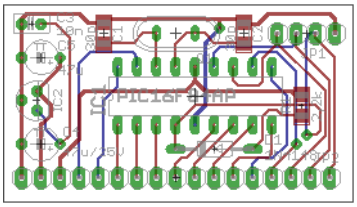
```
extensions = [
              'sphinxcontrib.eagle',
              ]
```

5.2 Directives

There are 3 directives, they accept a single string as argument, which is the path to the eagle .sch or .brd file. 3D is available only for board:

```
.. eagle-image:: ~/.eagle/projects/examples/tutorial/demo2.brd

.. eagle-image3d:: ~/.eagle/projects/examples/tutorial/demo2.brd

.. eagle-partlist:: ~/.eagle/projects/examples/tutorial/demo2.brd
```

The above snippet would render like this:



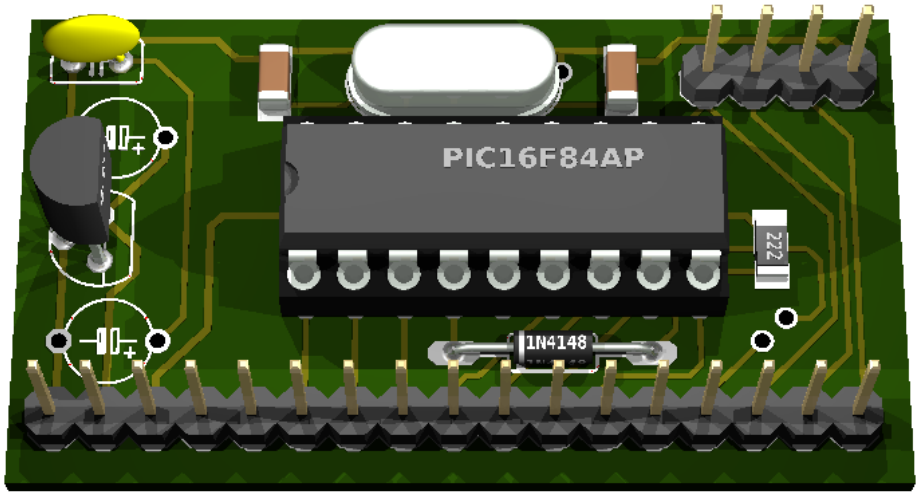


Table 5.1:

part	value	package	library	position	orientation
C1	30p	C1206	rcl	(0.5 0.85)	R270
C2	30p	C1206	rcl	(1.225 0.85)	R270
C3	10n	C025-025X050	rcl	(0.125 0.9)	R180
C4	47u/25V	TAP5-45	rcl	(0.175 0.275)	R180
C5	47u	TAP5-45	rcl	(0.175 0.725)	R180
D1	1N4148	DO35-10	diode	(1.075 0.25)	R0
IC1	PIC16F84AP	DIL18	microchip	(0.975 0.525)	R0
IC2	78L05Z	TO92	linear	(0.15 0.5)	R90
JP1	PROG	1X04	PINHEAD	(1.55 0.85)	R0
JP2	APPL	1X17	PINHEAD	(0.875 0.1)	R180
Q1		QS	special	(0.875 0.85)	R180
R1	2,2k	R1206	rcl	(1.525 0.475)	R270

The same for schematic without 3D:

```
.. eagle-image:: ~/.eagle/projects/examples/tutorial/demo2.sch
      :scale: 30%

.. eagle-partlist:: ~/.eagle/projects/examples/tutorial/demo2.sch
```

The above snippet would render like this:

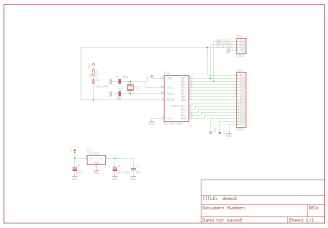


Table 5.2:

part	value	device	package	library	sheet
C1	30p	C-EUC1206	C1206	rcl	1
C2	30p	C-EUC1206	C1206	rcl	1
C3	10n	C-EU025-025X050	C025-025X050	rcl	1
C4	47u/25V	CPOL-EUTAP5-45	TAP5-45	rcl	1
C5	47u	CPOL-EUTAP5-45	TAP5-45	rcl	1
D1	1N4148	1N4148	DO35-10	diode	1
IC1	PIC16F84AP	PIC16F84AP	DIL18	microchip	1
IC2	78L05Z	78L05Z	TO92	linear	1
JP1	PROG	PINHD-1X4	1X04	pinhead	1
JP2	APPL	PINHD-1X17	1X17	pinhead	1
Q1		XTAL/S	QS	special	1
R1	2,2k	R-EU_R1206	R1206	rcl	1

5.3 Common options

5.3.1 timeout

Using the option `timeout` you can set the timeout (default 20) in seconds for processing. Eagle can block the export by displaying a messagebox. If this happens the export is aborted after timeout:

```
.. eagle-image:: ~/.eagle/projects/examples/tutorial/demo2.brd
   :timeout: 60
```

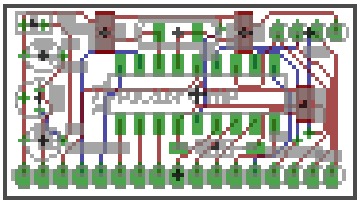
5.4 Image options

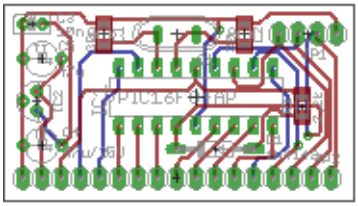
5.4.1 resolution

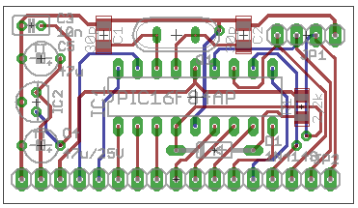
Using the option `resolution` you can set the resolution in dpi, valid range: 50..2400, default is 150:

```
.. eagle-image:: ~/.eagle/projects/examples/tutorial/demo2.brd
   :resolution: 50

.. eagle-image:: ~/.eagle/projects/examples/tutorial/demo2.brd
   :resolution: 100

.. eagle-image:: ~/.eagle/projects/examples/tutorial/demo2.brd
   :resolution: 200
```

The above snippet would render like this:







5.4.2 palette

Using the option `palette` you can set the background color.

Valid settings:

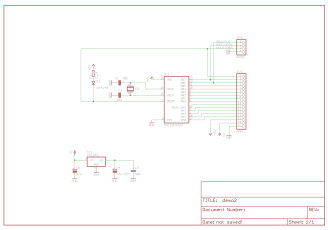
- white
- black
- colored

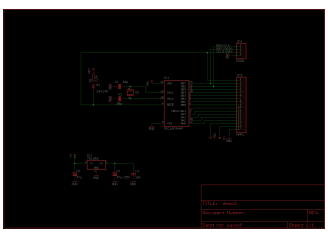
Default:white

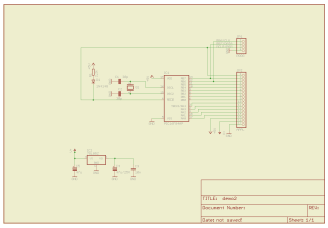
Example:

```
.. eagle-image:: ~/.eagle/projects/examples/tutorial/demo2.sch
   :palette:   white
   :scale: 30 %

.. eagle-image:: ~/.eagle/projects/examples/tutorial/demo2.sch
   :palette:   black
   :scale: 30 %

.. eagle-image:: ~/.eagle/projects/examples/tutorial/demo2.sch
   :palette:   colored
   :scale: 30 %
```

The above snippet would render like this:







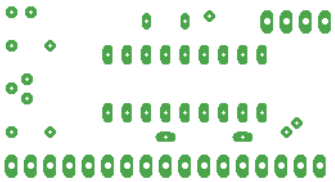
5.4.3 layers

Using the option `layers` you can display or hide layers. Check eagle documentation for valid settings.

Example:

```
.. eagle-image:: ~/.eagle/projects/examples/tutorial/demo2.brd
   :layers:   via,pads
```

The above snippet would render like this:



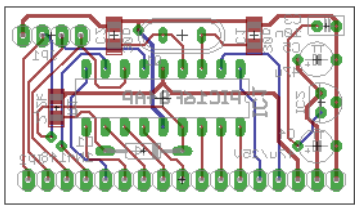
5.4.4 mirror

Using the option `mirror` you can mirror the image.

Example:

```
.. eagle-image:: ~/.eagle/projects/examples/tutorial/demo2.brd
   :mirror:
```

The above snippet would render like this:



5.4.5 command

Using the option `command` you can apply eagle commands.

Example:

```
.. eagle-image:: ~/.eagle/projects/examples/tutorial/demo2.brd
   :command:   display none dimension
```

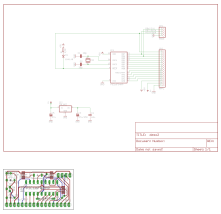
The above snippet would render like this:

5.4.6 scale, alt

Example:

```
.. eagle-image:: ~/.eagle/projects/examples/tutorial/demo2.sch
   :scale: 20 %
   :alt: alternate text

.. eagle-image:: ~/.eagle/projects/examples/tutorial/demo2.brd
   :scale: 20 %
   :alt: alternate text
```

The above snippet would render like this:

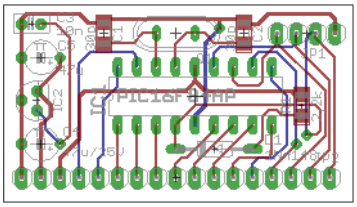


5.4.7 height, width

Example:

```
.. eagle-image:: ~/.eagle/projects/examples/tutorial/demo2.brd
     :height: 100px
     :width:  100 px
```

The above snippet would render like this:

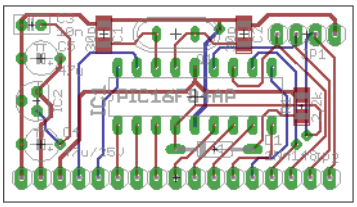


5.4.8 align

Example:

```
.. eagle-image:: ~/.eagle/projects/examples/tutorial/demo2.brd
     :height: 100px
     :width:  100 px
     :align: right
```

The above snippet would render like this:



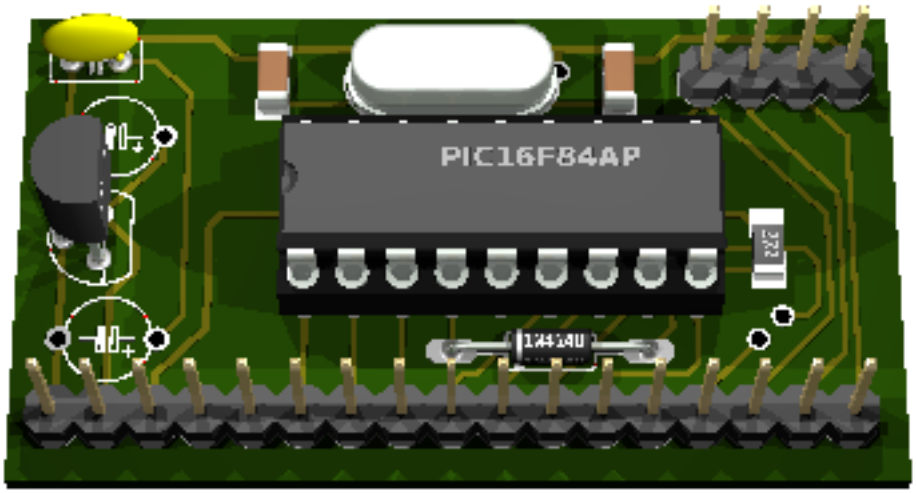
5.5 Image3D options

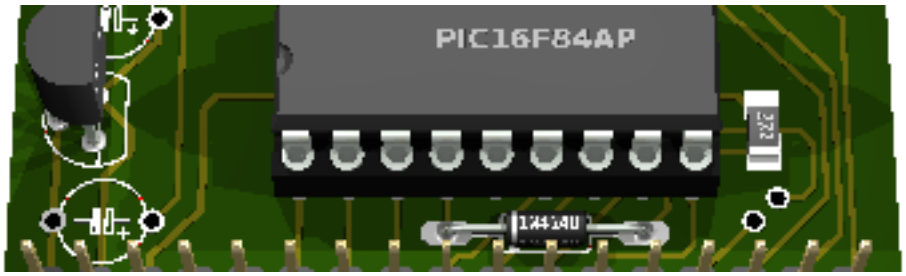
5.5.1 size

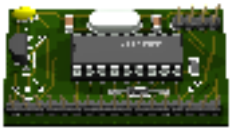
Size of image, width x height:

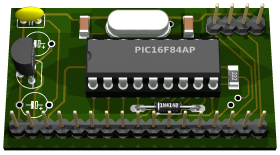
```
.. eagle-image3d:: ~/.eagle/projects/examples/tutorial/demo2.brd
   :size: 80x60

.. eagle-image3d:: ~/.eagle/projects/examples/tutorial/demo2.brd
   :size: 400x300

.. eagle-image3d:: ~/.eagle/projects/examples/tutorial/demo2.brd
   :size: 400x100

.. eagle-image3d:: ~/.eagle/projects/examples/tutorial/demo2.brd
   :size: 100x400

.. eagle-image3d:: ~/.eagle/projects/examples/tutorial/demo2.brd
   :size: 1600x1200
   :scale: 30%
```

The above snippet would render like this:


PIC16F84AP


PIC16F84AP





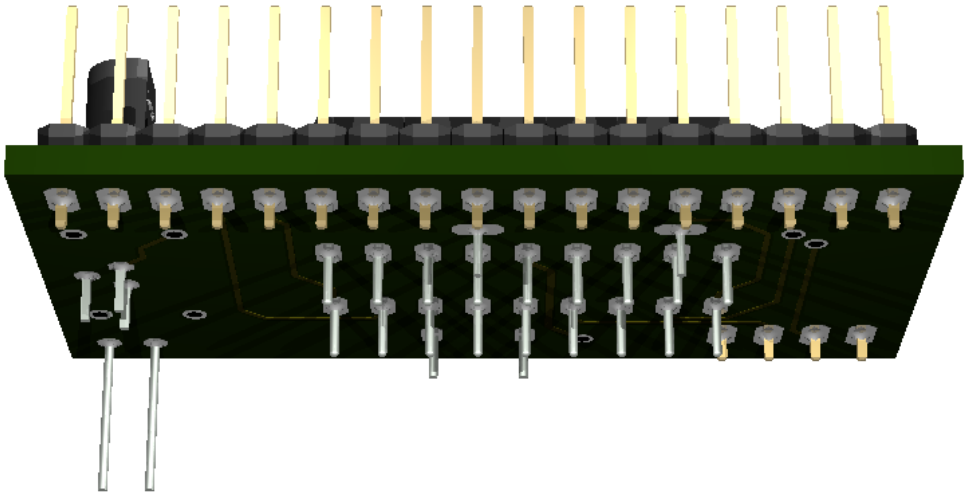
5.5.2 pcbrotate

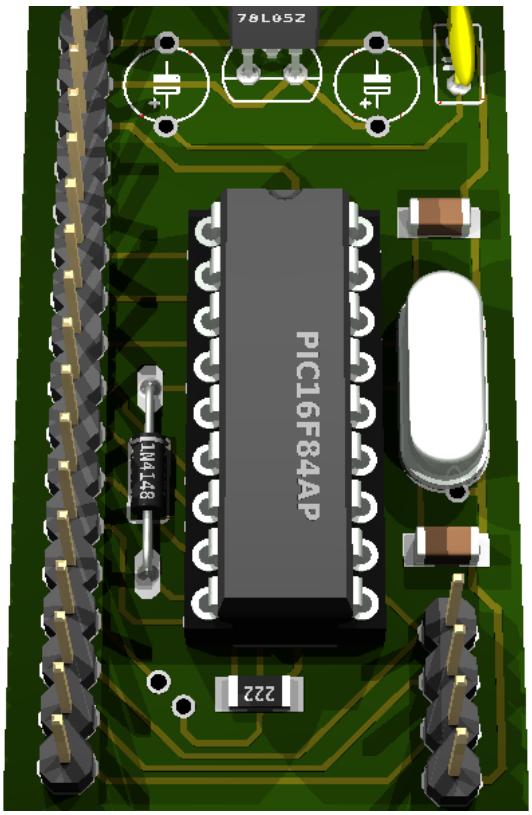
Rotate PCB around x,y,z:

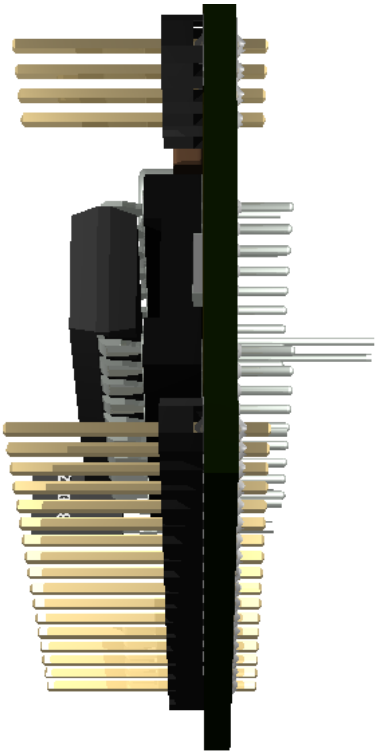
```
.. eagle-image3d:: ~/.eagle/projects/examples/tutorial/demo2.brd
   :pcbrotate: 90,0,0

.. eagle-image3d:: ~/.eagle/projects/examples/tutorial/demo2.brd
   :pcbrotate: 0,90,0

.. eagle-image3d:: ~/.eagle/projects/examples/tutorial/demo2.brd
   :pcbrotate: 0,0,90

.. eagle-image3d:: ~/.eagle/projects/examples/tutorial/demo2.brd
   :pcbrotate: 180,0,0
```

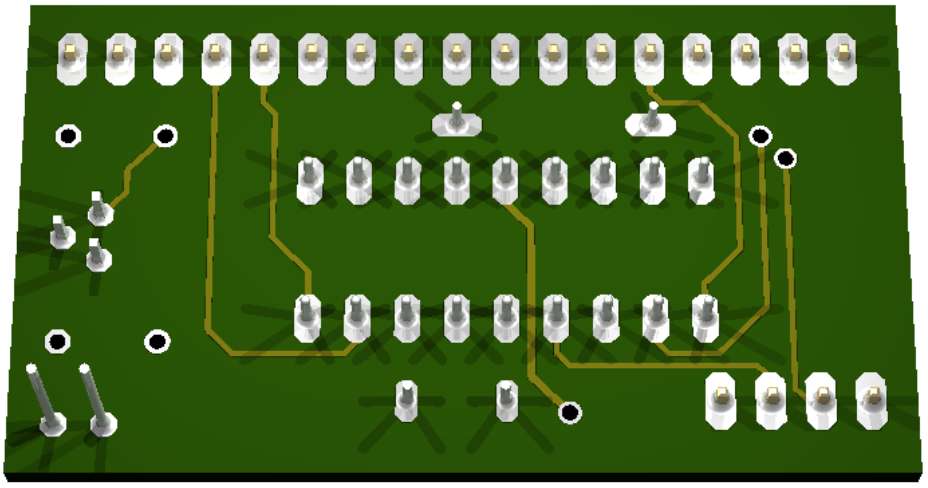
```
.. eagle-image3d:: ~/.eagle/projects/examples/tutorial/demo2.brd
   :pcbrotate: 0,180,0

.. eagle-image3d:: ~/.eagle/projects/examples/tutorial/demo2.brd
   :pcbrotate: 0,0,180
```

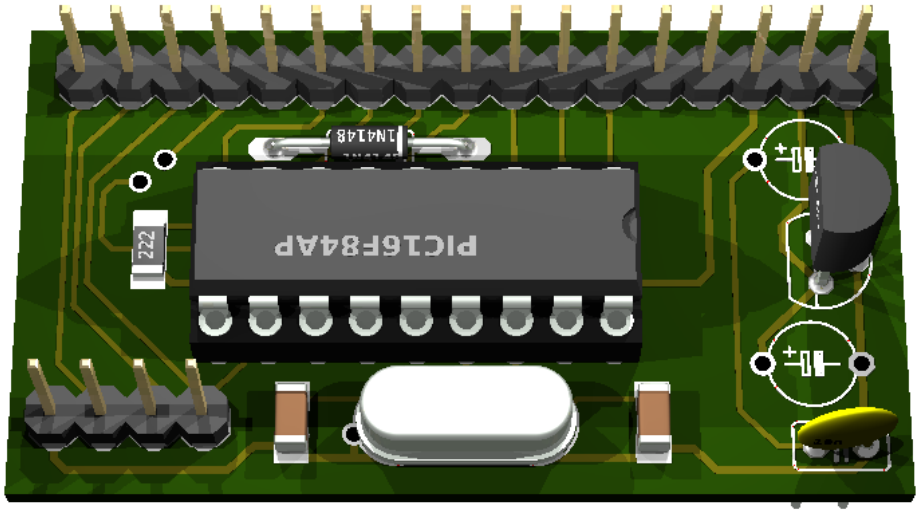
The above snippet would render like this:

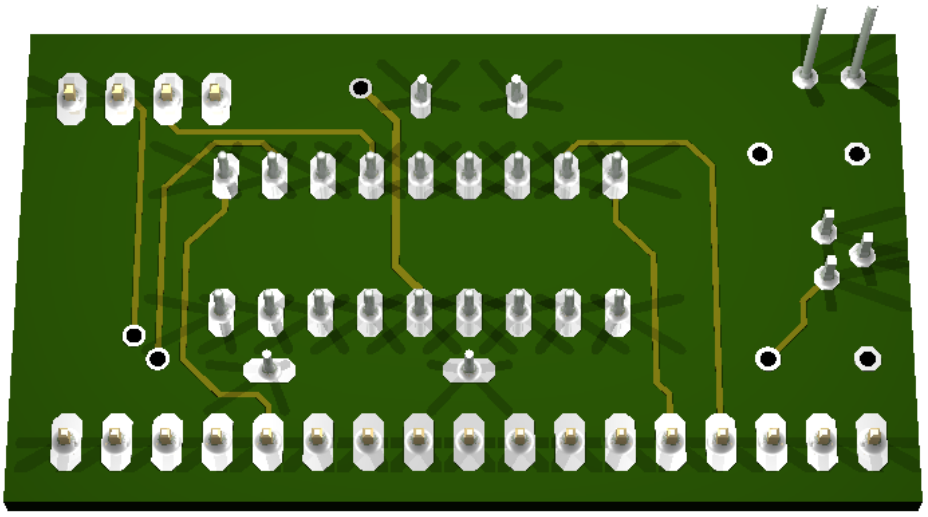









1N4148
PIC16F84AP
222



5.6 Partlist options

5.6.1 raw

Eagle partlist export is included as literal text:

```
.. eagle-partlist:: ~/.eagle/projects/examples/tutorial/demo2.sch
     :raw:
```

The above snippet would render like this:

```
Partlist

Exported from demo2.sch at 1/26/12 10:48 PM

EAGLE Version 5.10.0 Copyright (c) 1988-2010 CadSoft

Part      Value          Device          Package      Library   Sheet

C1        30p            C-EUC1206       C1206        rcl       1
C2        30p            C-EUC1206       C1206        rcl       1
C3        10n            C-EU025-025X050 C025-025X050 rcl       1
C4        47u/25V        CPOL-EUTAP5-45  TAP5-45      rcl       1
C5        47u            CPOL-EUTAP5-45  TAP5-45      rcl       1
D1        1N4148         1N4148          DO35-10      diode     1
IC1       PIC16F84AP     PIC16F84AP      DIL18        microchip 1
IC2       78L05Z         78L05Z          TO92         linear    1
JP1       PROG           PINHD-1X4       1X04         pinhead   1
JP2       APPL           PINHD-1X17      1X17         pinhead   1
Q1                       XTAL/S          QS           special   1
R1        2,2k           R-EU_R1206      R1206        rcl       1
```

5.6.2 header

A comma-separated list of selected column names:

```
.. eagle-partlist:: ~/.eagle/projects/examples/tutorial/demo2.sch
     :header: part, value
```

The above snippet would render like this:

Table 5.3:

part	value
C1	30p
C2	30p
C3	10n
C4	47u/25V
C5	47u
D1	1N4148
IC1	PIC16F84AP
IC2	78L05Z
JP1	PROG
JP2	APPL
Q1	
R1	2,2k

5.6.3 widths

A comma- or space-separated list of relative column widths. The default is equal-width columns:

```
.. eagle-partlist:: ~/.eagle/projects/examples/tutorial/demo2.sch
     :header: part, value
     :widths:  2,8
```

The above snippet would render like this:

Table 5.4:

part	value
C1	30p
C2	30p
C3	10n
C4	47u/25V
C5	47u
D1	1N4148
IC1	PIC16F84AP
IC2	78L05Z
JP1	PROG
JP2	APPL
Q1	
R1	2,2k

CHAPTER

SIX

DEVELOPMENT

6.1 Tools

1. setuptools
2. Paver
3. nose
4. ghp-import
5. pyflakes
6. pychecker
7. paved fork
8. Sphinx
9. sphinxcontrib-programscreenshot
10. sphinxcontrib-paverutils
11. `autorun` from sphinx-contrib (there is no simple method, you have to download/unpack/setup)

6.2 Install on ubuntu

```
sudo apt-get install python-setuptools
sudo apt-get install python-paver
sudo apt-get install python-nose
sudo easy_install ghp-import
sudo apt-get install pyflakes
sudo apt-get install pychecker
sudo easy_install https://github.com/ponty/paved/zipball/master
sudo apt-get install scrot
sudo apt-get install xvfb
sudo apt-get install xserver-xephyr
sudo apt-get install python-imaging
sudo apt-get install python-sphinx
sudo easy_install sphinxcontrib-programscreenshot
sudo easy_install sphinxcontrib-programoutput
sudo easy_install sphinxcontrib-paverutils
```

6.3 Tasks

Paver is used for task management, settings are saved in pavement.py. Sphinx is used to generate documentation.

print paver settings:

```
paver printoptions
```

clean generated files:

```
paver clean
```

generate documentation under *docs/_build/html*:

```
paver cog pdf html
```

upload documentation to github:

```
paver ghpages
```

run unit tests:

```
paver nose
#or
nosetests --verbose
```

check python code:

```
paver pyflakes
paver pychecker
```

generate python distribution:

```
paver sdist
```

upload python distribution to PyPI:

```
paver upload
```