Toprammer - TOPxxxx OpenSource suite

Copyright (c) 2009-2024 Michael Büsch <m@bues.ch>

June 28, 2024

1 Dependencies

- Python 3.7 or later is required

  `https://www.python.org/`

- python3-usb module is required:

  `https://pypi.org/project/pyusb/`

- python3-pkg-resources module is required 'pkg_resources' is part of the Python 'setuptools' package.

  `https://pypi.org/project/setuptools/`

- PyQT6 is required for the graphical QT user interface (optional):

  `https://riverbankcomputing.com/software/pyqt/download`

2 Supported chips

Just execute the following command to get a list of supported chips:

```
toprammer --list
```

For a more verbose list, also pass the -V parameter with a verbosity level number:

```
toprammer --list -V99
```

3 Installation

Just execute

```
python ./setup.py install
```

as root inside of the toprammer distribution package's root directory. Note that "setuptools" have to be installed on your system. See dependencies.

4 Graphical user interface tool usage

The toprammer GUI tool is called “toprammer-gui”. To start it just run:

```
toprammer-gui
```

Note that, depending on your system config, you might need to run toprammer-gui as root to allow USB hardware access. If you get “Operation not permitted" errors, try to re-run toprammer-gui as root.

5 Commandline tool usage

Toprammer needs the identification string of the chip that is inserted into the ZIF socket. Additionally it needs an action to be performed on the chip. So a call to toprammer might look like this:

```
toprammer --chip-id atmega32dip40 --read-prog flash.img
```

That command selects an Atmel AtMega32 DIP40 chip and reads its flash contents into the flash.img file.

For a list of supported chips, see the

```
topgrammer --list
```

command. It will print a list of supported chip-IDs.

6 ZIF socket layout

The ZIF socket layout generator "toprammer-layout" will try to generate a chip-insert layout with the given parameters and print it as ASCII-art to the console.

For example, if you have a DIP28 packaged chip, that needs VCC on pin 7, VPP on pin 1 and GND on pin 8, you'd issue the following command:

```
toprammer-layout -d top2049 --package DIP28 --vcc 7 --vpp 1 --gnd 8
```

This will show you a layout of how to insert the chip into the programmer. It will also show which pins of the ZIF are powered. This will hopefully match your request. :) Alternatively, it will yield an error message, if it was unable to find a layout that fits the contraints. Note that the –vcc, –vpp and –gnd pins specified on the commandline are with respect to the package (_not_ the ZIF socket). The whole purpose of the tool is to map the package and its pin layout to the ZIF socket, while obeying the programmer constraints.

Alternatively you can specify one of the supported chip-IDs to toprammer-layout:

```
toprammer-layout -d top2049 --package atmega8dip28
```

This will show you the layout of the AtMega8 DIP28.