

gEEProg v2.0 Software User Manual

© Mark Chilenski, Matthew D'Asaro 2015
<http://www.dasarodesigns.com/>

January 17, 2015

1 Overview

gEEProg is a Python module and GUI program for controlling the 2801Prog, 2006Prog and other D'Asaro Designs EEPROM programmers. gEEProg is distributed under the terms of the GNU Public License (GPL), a copy of which should have been included with this software.¹ For the technical details of the programmer itself, refer to the accompanying user and service manual.

2 Installing the Software

2.1 Windows

An executable (.EXE file) is available at <https://github.com/markchil/gEEProg/releases>. Simply download and unzip this file to a convenient location. The software is launched by double-clicking the program icon, like any other program.

2.2 Mac OS X

An application (.app file) is available at <https://github.com/markchil/gEEProg/releases>. Simply download this file, unzip it and drag gEEProg.app to your Applications folder (or another convenient location). The software is launched by double-clicking the program icon, like any other program.

2.3 Linux/Install From Source

To install from source, you must have the following prerequisites (many of which are already present in most Linux installs):

Python version 2.7 Other versions may work but have not been tested. Either use your favorite package manager, or install from <https://www.python.org/downloads/>.

¹<http://www.gnu.org/licenses/>

Tkinter Either use your favorite package manager, or follow the directions here http://tkinter.unpythonic.net/wiki/How_to_install_Tkinter or here <http://www.tkdocs.com/tutorial/install.html>

pip Either use your favorite package manager, or follow the directions here <https://pip.pypa.io/en/latest/installing.html>

Once you have these prerequisites, installing gEEProg is as simple as running the command “`sudo pip install gEEProg`” (without the quotes). This will install/upgrade gEEProg’s dependencies as needed, then install the gEEProg software itself. The gEEProg Python module will be placed in your site-packages directory, and can then simply be accessed for use in your own Python scripts with “`import gEEProg`”. The GUI is called “gEEProgGUI” and its install location is determined by pip and how your Python installation is structured. The path to this script will be printed as part of pip’s installation process. Assuming that the directory containing the gEEProgGUI script is part of your shell PATH, you can now launch the software simply by typing “gEEProgGUI” from the shell (without the quotes). If Python 2.7 is not the version you get when typing “python” at the shell, then you need to modify the shebang statement at the start of the gEEProgGUI script. An example of checking for the prerequisites, installing and launching gEEProg on Ubuntu is given in figure 1.

```
markchilenski@ubuntu:~$ python
Python 2.7.3 (default, Apr 10 2013, 06:20:15) Python version is 2.7.x ✓
[GCC 4.6.3] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> import Tkinter Tkinter imports successfully ✓
>>> exit()
markchilenski@ubuntu:~$ sudo pip install gEEProg Install gEEProg using pip
[sudo] password for markchilenski:
Downloading/unpacking gEEProg
Running setup.py egg_info for package gEEProg
Downloading/unpacking pyserial>=2.7 (from gEEProg)
Running setup.py egg_info for package pyserial
Installing collected packages: gEEProg, pyserial
Running setup.py install for gEEProg
changing mode of /usr/local/bin/gEEProgGUI to 775
Found existing installation: pyserial 2.5
Uninstalling pyserial:
Successfully uninstalled pyserial
Running setup.py install for pyserial
changing mode of build/scripts-2.7/miniterm.py from 644 to 755
changing mode of /usr/local/bin/miniterm.py to 755
Successfully installed gEEProg pyserial
Cleaning up...
markchilenski@ubuntu:~$ gEEProgGUI gEEProg is now ready to use
```

Figure 1: How to check for the prerequisites and install gEEProg from source.

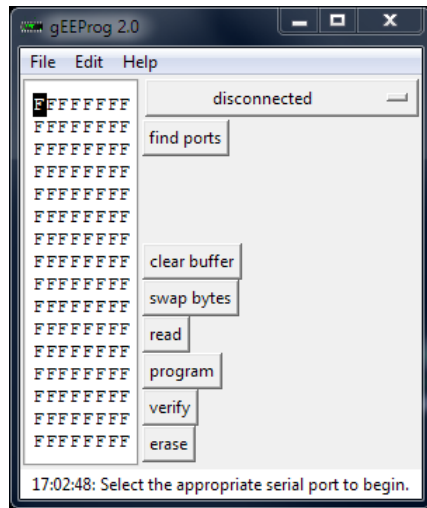


Figure 2: Starting state of gEESProg.

3 Connecting to your programmer

First, connect your programmer to the computer using a serial cable and/or USB-to-serial adapter. Then, connect the power to your programmer. Finally, start gEESProg as described above.

When gEESProg first launches, the window will look like figure 2. If you plugged your USB-to-serial adapter in after starting gEESProg, press the “find ports” button to refresh the list of available ports. Then, click the pull down menu (which should say “disconnected”) and select your programmer’s serial port from the list of available options.

Once gEESProg connects to the programmer, the bottom status bar will turn green and some explanatory text will appear – see figure 3.

Troubleshooting

If gEESProg cannot connect to the programmer, the bottom status bar will turn red. First, make sure you have selected the correct port. If you cannot connect on any of the ports, try pulling the power cable from the programmer for a few seconds, then plugging it back in, then trying to connect to the programmer again as described above.

Potential issues on Linux

If you cannot connect to your serial port on Linux, it may be because you do not have sufficient user permissions. You can try running gEESProg with `sudo` to circumvent this, though the best solution is to add your user account

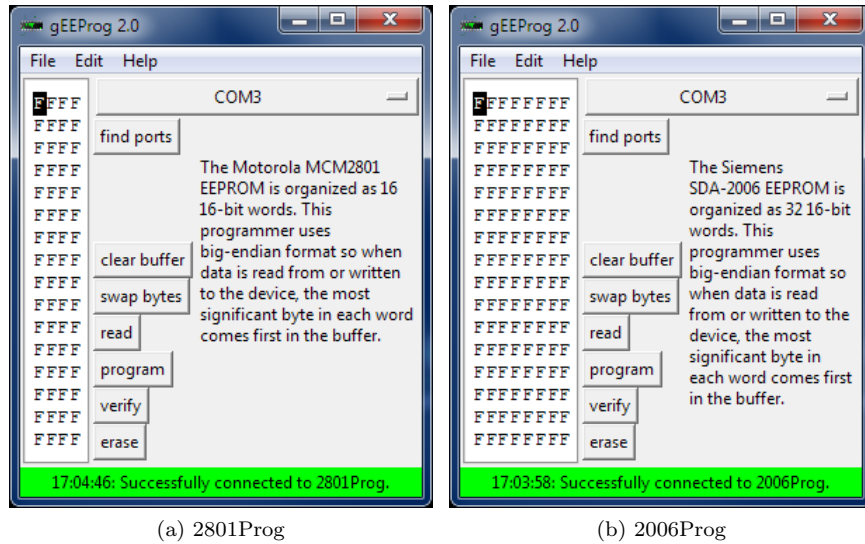


Figure 3: gEEProg after successfully connecting to a programmer.

to the “dialout” group with the command “`sudo gpasswd --add [username] dialout`”.

4 Interacting with the programmer

4.1 The buffer

The buffer on the left side of the window allows you to edit the contents of the memory in hexadecimal. Use the arrow keys to navigate, or click on a specific nibble to bring the cursor there. The byte ordering for your particular programmer is described in the text to the right. To clear the contents of the buffer (i.e., to set each byte to “FF”) click the “**clear buffer**” button. To swap the byte order click the “**swap bytes**” button.

4.2 Reading an EEPROM

To read the data from the EEPROM chip inserted into the programmer press the “**read**” button. If the chip is read successfully the bottom status bar will turn green and the contents of the chip will be placed into the buffer for viewing/editing. If the read fails the status bar will turn red.

4.3 Programming an EEPROM

To program the EEPROM chip inserted into the programmer with the contents of the buffer press the “**program**” button. If the chip is programmed successfully the bottom status bar will turn green. If the chip fails to program the status bar will turn red.

4.4 Verifying the contents of an EEPROM

To verify that the contents of the EEPROM chip inserted into the programmer match the contents of the buffer press the “**verify**” button. If the contents match the bottom status bar will turn green. If the contents do not match the status bar will turn red.

4.5 Erasing an EEPROM

To erase the EEPROM chip inserted into the programmer press the “**erase**” button. If the chip’s contents have been successfully erased the bottom status bar will turn green. If the contents were not successfully erased the status bar will turn red.

5 Advanced features

5.1 Saving and opening files with the “File” menu

The “**File**” menu lets you save and open files. If the file extension is “.txt” the file is an ASCII representation of the hexadecimal contents of the buffer, otherwise it is treated as a binary file. When reading an ASCII file, all non-valid characters are stripped from the input. This includes spaces and newlines, so you may format your ASCII files to be human-readable.

5.2 Copying and pasting with the “Edit” menu

The “**Edit**” menu allows you to copy and paste block of hexadecimal data in the buffer. The “**Paste**” function will paste whatever text you copied last into the buffer starting at the cursor’s location. Non-valid characters will be stripped, so you can copy from a human-readable representation of the desired data. The “**Copy Buffer**” function will copy the contents of the entire buffer so that you can paste them into another program.

5.3 Getting help with the “Help” menu

The “**gEEProg help...**” option under the “**Help**” menu will bring up the dialog shown in figure 4, which contains links to the websites for both your programmer and the gEEProg software.



Figure 4: Help window.

6 Using the Python API

gEEProg includes the “gEEProg” Python module which gives access to all features of your programmer from within Python. The main general-purpose functions are:

`enter_automation_mode` Puts the programmer into automation mode such that programs can talk to it properly.

`exit_automation_mode` Exits automation mode to allow a user to interact with the programmer directly using a program like minicom.

`read_chip` Reads the contents of the chip.

`verify_chip` Verifies that the contents of the chip match the desired data.

`program_chip` Programs the chip with the desired data.

`erase_chip` Erases the chip.

Consult the documentation for these functions in the file `gEEProg.py`, which should have been installed in your `site-packages` directory.