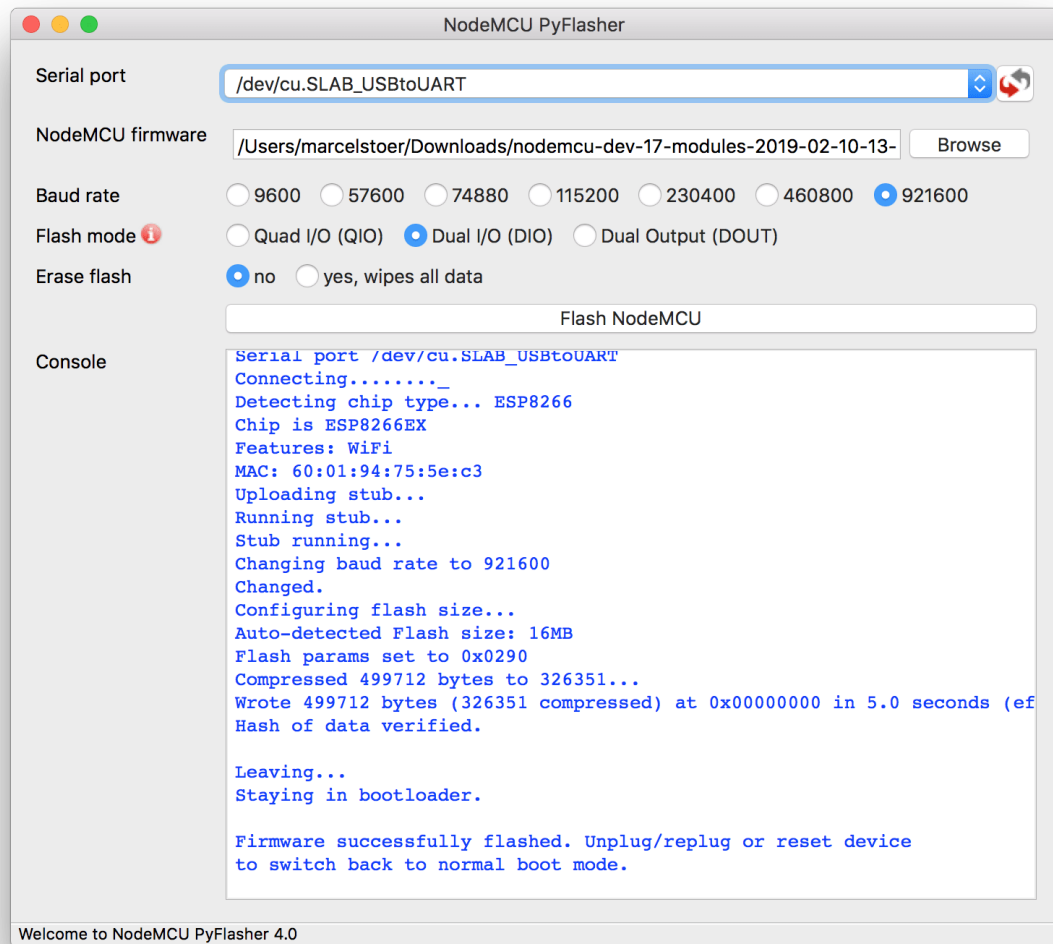

NodeMCU PyFlasher

license MIT downloads 425k donate PayPal Tweet Share

Self-contained NodeMCU flasher with GUI based on esptool.py and wxPython.



Installation

NodeMCU PyFlasher doesn't have to be installed, just double-click it and it'll start. Check the releases section for downloads for your platform. For every release there's at least a .exe file for Windows. Starting from 3.0 there's also a .dmg for macOS.

Status

Scan the list of open issues for bugs and pending features.

Note

This is my first Python project. If you have constructive feedback as for how to improve the code please do reach out to me.

Getting help

In the unlikely event that you're stuck with this simple tool the best way to get help is to turn to the "Tools and IDE" subforum on esp8266.com.

Donationware

All open-source development by the author is donationware. Show your love and support for open-source development by donating to the good cause through PayPal.



Build it yourself

If you want to build this application yourself you need to:

- Install Python 3.x and Pip (it comes with Python if installed from python.org).
- Create a virtual environment with `python -m venv venv`
- Activate the virtual environment with `. venv/bin/activate` (`. venv/Scripts/activate` if you are on Windows with Cygwin or Mingw)
- Run `pip install -r requirements.txt`

A note on Linux: As described on the downloads section of [wxPython](#), wheels for Linux are complicated and may require you to run something like this to install [wxPython](#) correctly:

```
1 # Assuming you are running it on Ubuntu 18.04 LTS with GTK3
2 pip install -U \
```

```
3 -f https://extras.wxpython.org/wxPython4/extras/linux/gtk3/ubuntu
   -18.04 \
4 wxPython
```

Why this project exists

Motivation

This addresses an issue the NodeMCU community touched on several times in the past, most recently at #1500 (comment).

I stated that based on my experience doing NodeMCU user support it should be a lot simpler to flash NodeMCU for Windows users.

- A number of flashing tools are available but only two are actively maintained: esptool-ck and esptool.py. Only one is endorsed by Espressif: esptool.py (they hired the developer(s)).
- 70% of the users of my nodemcu-build.com service are on Windows.
- BUT Windows doesn't come with Python installed - which is required for esptool.py.
- BUT Windows users in general are more reluctant to use the CLI than Linux/Mac users - which is required for esptool.py.

To conclude: this is not a comfortable situation for NodeMCU's largest user group.

The plan

For quite a while I planned to write a self-contained GUI tool which would use esptool.py in the background. It should primarily target Windows users but since I'm on Mac it should be cross-platform. Even though I had never used Python before I felt confident to pull this off.

Implementation

- Uses the cross-platform wxPython GUI framework. I also tried PyForms/PyQt4 but settled for wxPython.
- Requires absolutely minimal user input.
- The esptool.py "console" output is redirected to text control on the GUI.
- Uses PyInstaller to create self-contained executable for Windows and Mac. The packaged app can run standalone i.e. without installing itself, a Python interpreter or any modules.

License

MIT © Marcel Stör