

ESPflasher: GUI for esptool from Espressif

1. Abstract

Usually flashing of an ESP8266 module will be done by [Arduino IDE](#) when you test or final flash a sketch. But what if you have downloaded a compiled binary file? There are some tutorials in the web, but often not easy to use.

ESPflasher will provide a GUI to flash binary files to esp modules.

Learn more about ESP8266 modules:

<https://www.esp8266.com/wiki/doku.php?id=esp8266-module-family>

<http://stefanfrings.de/esp8266>

2. Preparations

2.1 Download tools

Download following tools dependent on your OS:

Espressif esptool: <https://github.com/espressif/esptool/releases>

ESPflasher: <https://github.com/h-elsner/ESPflasher>

ESPflasher is a simple GUI for the Espressif esptool offering only some important functions of the esptool but all we need here.

Note:

Supported OS	esptool	ESPflasher
Raspberry Pi (32bit)	esptool-v4.6.2-arm.zip	ESPflasher_arm.zip
Ubuntu LINUX	esptool-v4.6.2-linux-amd64.zip	ESPflasher_LINUX.zip
Windows 10	esptool-v4.6.2-win64.zip	ESPflasher_win.zip

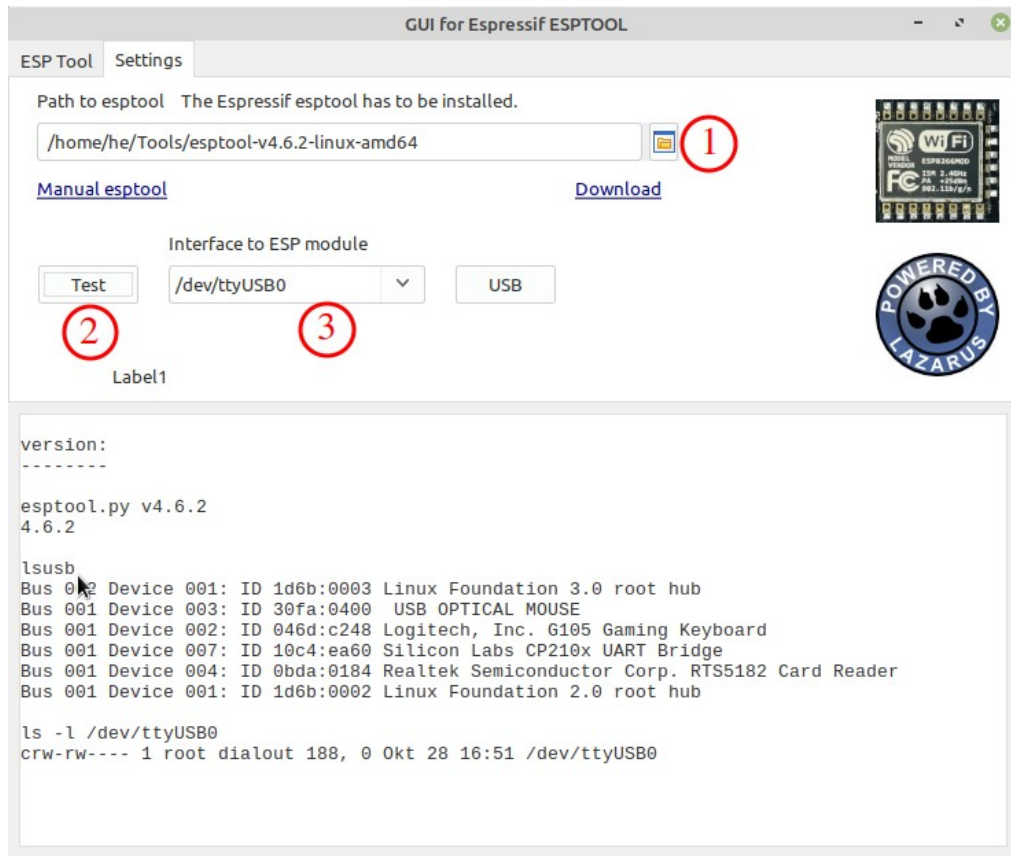
2.2 Installation

Create a folder where you want to have the flash tools. Create a subfolder /bin there. Move downloaded zip files to this folder and unzip all, move binary firmware files to subfolder /bin. This subfolder can be used for all binary firmware files that you want to flash on an ESP8266, no matter for what purpose.

Note:

- For Windows you may have to install HW dependent drivers for the device you connect to USB. At ESP-01 this is a USB-Serial converter, at NodeMCU boards the onboard USB-Serial chip.
- For UNIX-like OS you may have to make esptool executable: `chmod +x esptool`

Open ESPflasher > go to **Settings** page > enter the path to esptool **(1)**.



Connect your device you want to flash to USB port (no need to bring it into flash mode at this point). Click on **Test** in ESPflasher **(2)**.

For **UNIX-like OS** you should get an output like that:

```
version:
-----
esptool.py v4.6.2
4.6.2

lsusb
Bus 002 Device 001: ID 1d6b:0003 Linux Foundation 3.0 root hub
Bus 001 Device 003: ID 30fa:0400 USB OPTICAL MOUSE
Bus 001 Device 002: ID 046d:c248 Logitech, Inc. G105 Gaming Keyboard
Bus 001 Device 005: ID 10c4:ea60 Silicon Labs CP210x UART Bridge
Bus 001 Device 004: ID 0bda:0184 Realtek Semiconductor Corp. RTS5182 Card Reader
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub

ls -l /dev/ttyUSB0
crw-rw---- 1 root dialout 188, 0 Oct 28 08:24 /dev/ttyUSB0
```

If installation of esptool is OK and path to esptool proper set then you will get the **version number** of the esptool.

In the second section you will see the **devices** that are connected to USB. Check if your device is in the list. In the third section you will get the **port the device is assigned to**. Usually it is **/dev/ttyUSB0**. If your USB device is not there, unplug and plug it again USB connection. Often this helps.

Select the listed port as interface in ESPflasher (3).

For **Windows** you should get an output like that:

```
version:
-----
esptool.py v4.6.2
4.6.2

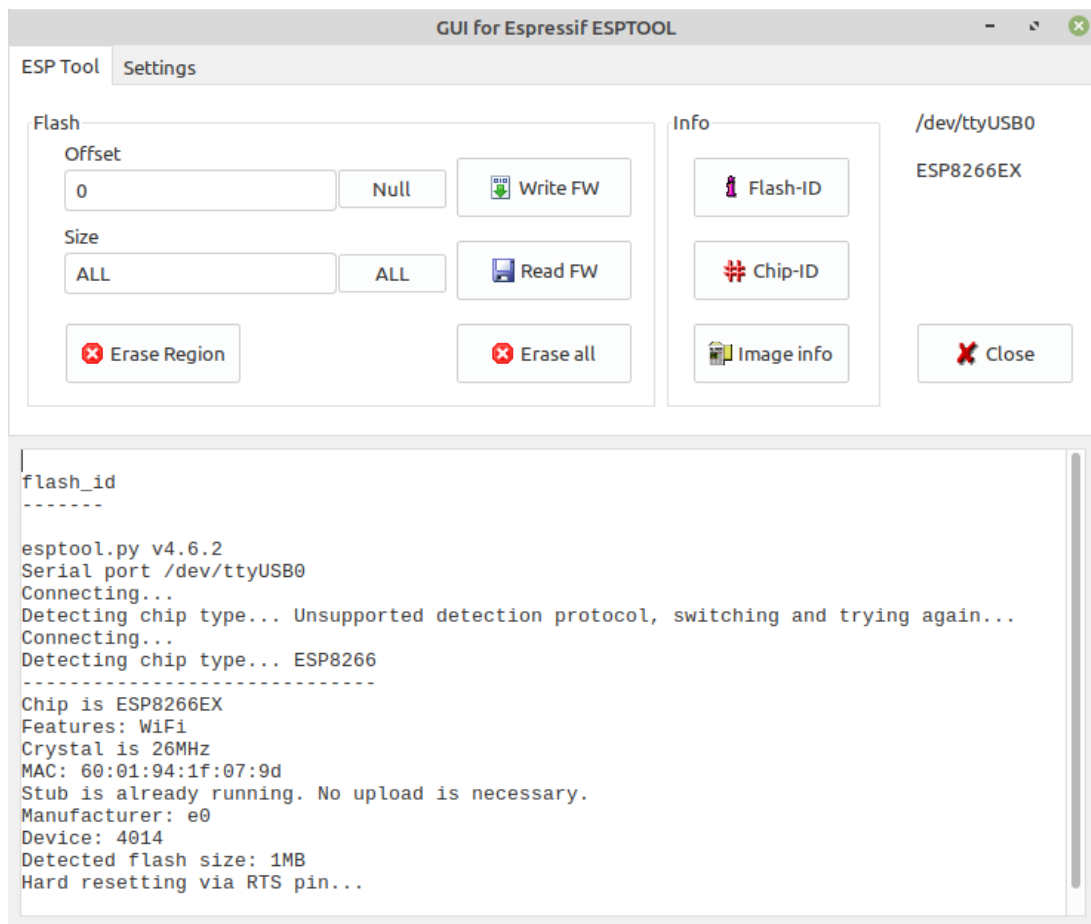
Ports:
-----
COM1
COM4
COM6
```

If installation of esptool is OK and path to esptool proper set then you will get the **version number** of the esptool.

In the second section you will see the **available COM ports**. If no new COM port appeared, unplug and plug again USB connection. Often this helps and it helps to identify the correct COM port, usually the one with the highest number.

Select the used COM port from the list as interface in ESPflasher (3).

Now we have a complete environment to flash ESP8266 modules with any binary firmware files.

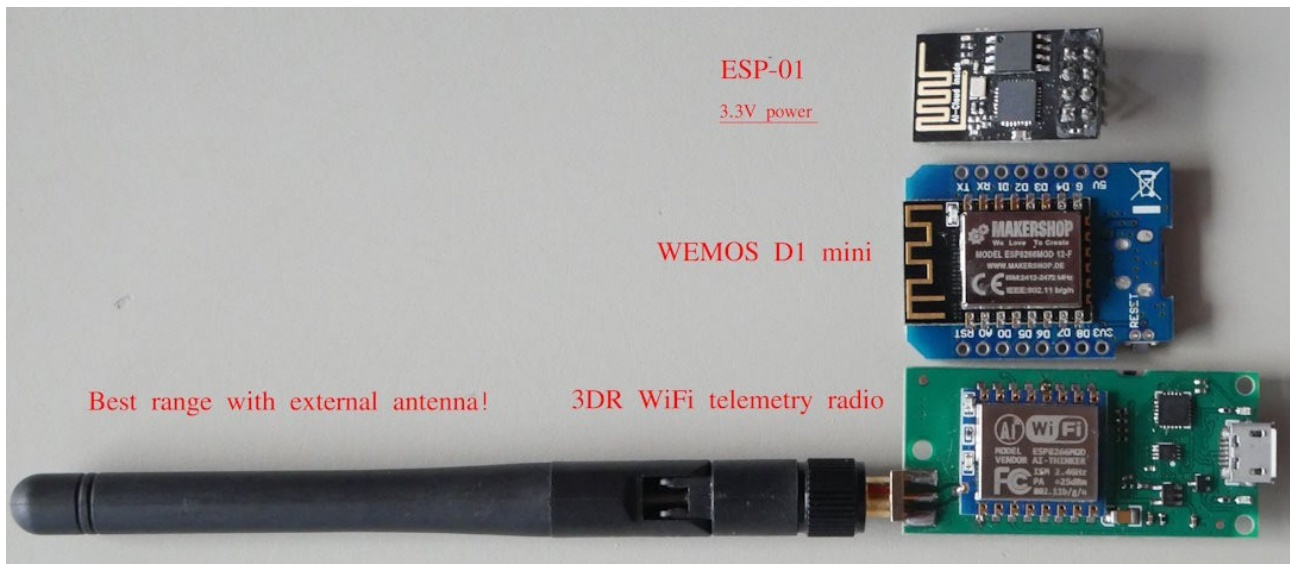


3. Flash ESP8266 module

Check **documentation of the ESP8266 modul** you want to use. Important items are

- pin layout
- power supply (3.3V or 5V or whatever you module can handle)
- signal voltage of the Tx and Rx pins (usually 3.3V)
- how to bring it in flash mode (GPIO0 to ground, automatically or by button).

I have tested this procedure with following modules:



Bring ESP module into flash mode. In ESPflasher, page **ESP tool** click on **Flash-ID**. The ESP8266 should respond with some data about it. If so, we can go ahead flashing the firmware file.

Example for ESP-01 module:

```
flash_id
-----

esptool.py v4.6.2
Serial port /dev/ttyUSB0
Connecting....
Detecting chip type... Unsupported detection protocol, switching and trying again...
Connecting...
Detecting chip type... ESP8266
-----
Chip is ESP8266EX
Features: WiFi
Crystal is 26MHz
MAC: 60:01:94:1f:07:9d
Uploading stub...
Running stub...
Stub running...
Manufacturer: e0
Device: 4014
Detected flash size: 1MB
Hard resetting via RTS pin...
```

To flash a binary firmware click on **Write FW**. You will be asked for a firmware file. Select the file from /bin folder. Once selected the flash procedure starts. Wait until flashing is completed. Output shall look like this

```
write_flash: MAVlink_firmware-1.2.2.bin
-----

esptool.py v4.6.2
Serial port /dev/ttyUSB0
Connecting...
Detecting chip type... Unsupported detection protocol, switching and trying again...
Connecting...
Detecting chip type... ESP8266
-----
Chip is ESP8266EX
Features: WiFi
Crystal is 26MHz
MAC: 60:01:94:1f:07:9d
Stub is already running. No upload is necessary.
Configuring flash size...
Flash will be erased from 0x00000000 to 0x00051fff...
Compressed 332768 bytes to 232984...
Wrote 332768 bytes (232984 compressed) at 0x00000000 in 20.7 seconds (effective 128.7 kbit/s)...
Hash of data verified.

Leaving...
Hard resetting via RTS pin...
```

Flashing finished. Remove ESP8266 module from USB port. Done!