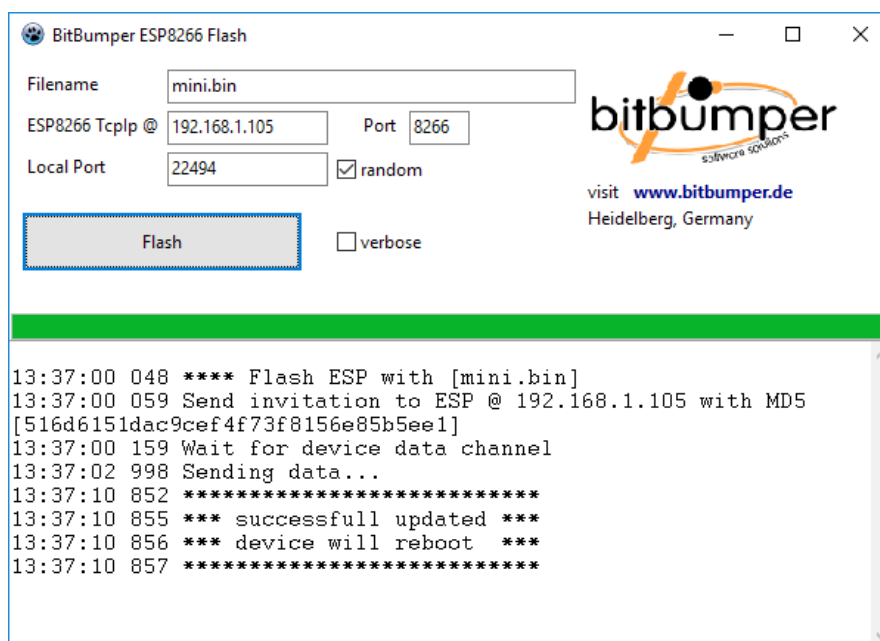


OTA FlashTool for the Esp8266

Developers Documentation



**BitBumper Ingenieurbüro Keil
Heidelberg, Germany**

Content

| | |
|----------------------------|---|
| 1. Introduction..... | 2 |
| 2. Reurements..... | 2 |
| 2.1. Operating System..... | 2 |
| 2.2. IoT Device..... | 2 |
| 2.3. Firmware file..... | 3 |
| 3. Licence..... | 3 |

1. Introduction

This document describes the flash tool "FlashEsp8266.exe".

The tool is to update a ESP8266 microcontroller in OTA mode (Over-The-Air, means using the WLAN/Wireless capabilities of the device) without usage of any other tool. It is "standalone", and therefore useable to flash a Esp device "on the field" or "by the customer".

There are some options to flash a Esp8266 ("Esp").

Most of them require some additional stuff - e.g. a certain requirement on the update machine or at least a python installed with some extra libs and moduls.

espFlasher is a quite good standalone solution, but only support the serial update mode.

Comparison to other options to flash a Esp8266

| Name | Source | Serial | OTA | Bin-Only | Dependencies |
|---------------------------|---|--------|-----|----------|---|
| espFlasher | github.com/nodemcu/nodemcu-flasher | Yes | No | Yes | Windows |
| PlatformIO | https://platformio.org/platformio-ide | | | | VisualStudioCode, Atom, Python, C Runtime |
| PIO Remote | http://docs.platformio.org/en/latest/plus/pio-remote.html | No | Yes | Indirect | Cloud |
| espTool.py | https://github.com/espressif/esptool | Yes | No | Yes | Python |
| espot.py ota_server.py | https://gist.github.com/igrr/d35ab8446922179dc58c | No | Yes | Yes | Python |
| FlashEsp8266 | here | No | Yes | Yes | Windows |

2. Requirements

Things you need to use the FlashEsp8266 tool.

2.1. Operating System

The program runs on any Windows installation (Windows XP, Windows 7, Windows 10). The program is written in Pascal with Lazarus and can be compiled also to run on other platforms. If you need other binaries just contact us.

2.2. IoT Device

As soon as we go over the air (OTA), the receiver device need to have implemented the same mechnism and protocolls than we use on the client side.

So if we speak about OTA we use the mechnism provided by PIO (PlatformIO) as it is implemented in the ESP code package

`\\.platformio\\packages\\framework-arduinoespressif8266\\libraries\\ArduinoOTA\\`

The IoT Device you plan to update must have an firmware on it which support this mechanism.

Normally this means you've developed the software for this device using the PlatformIO ecosystem - which is a good decision anyway.

Within your code you should have a line like

```
ArduinoOTA.begin();
```

This enables the OTA flash.

See for full doc about this at
<https://github.com/esp8266/Arduino>
and on other sources in the net.

Note:

You can use this tool to flash something else to your device JUST ONCE.

After you flash a firmware which do not support OTA, your only chance to change the firmware again ist to wire the serial connection and use a serial flash tool.

2.3. Firmware file

The firmware to uplaod is a file usually called "firmware.bin".

Unless you are a complete nerd use PlatformIO to create the firmware.bin file.

Steps to you firmware file:

- Write some nice code.
- Press [Strg+Alt-B] to build your code.
- Take generated file
[project-dir]\.pioenvs\esp12e\firmware.bin

3. Licence

This software is freeware.

DISCLAIMER OF WARRANTY: "THIS SOFTWARE IS SUPPLIED AS IS. THE AUTHOR DISCLAIMS ALL WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, THE WARRANTIES OF MERCHANTABILITY AND OF FITNESS FOR ANY PURPOSE. THE AUTHOR ASSUMES NO LIABILITY FOR DAMAGES, DIRECT OR CONSEQUENTIAL, WHICH MAY RESULT FROM THE USE OF THIS SOFTWARE."

ALL BRAND AND PRODUCT NAMES ARE MARKS OR REGISTERED MARKS OF THEIR RESPECTIVE COMPANIES.