

Basics: Connecting and flashing the ESP8266

From Let's Control It

Contents

- 1 Basics: How to connect and flash the ESP8266
- 2 What hardware do i need?
 - 2.1 A word about power supplies
 - 2.2 Recommended UART (USB to TTL converter)
- 3 Software
 - 3.1 Firmware
- 4 Some basics about flashing
- 5 Connecting to the computer for flashing
 - 5.1 Get your computer ready
 - 5.2 Connecting UART to ESP
 - 5.3 Flashing the nodeMCU / WeMOS D1 mini
 - 5.4 Flashing other ESP devices
- 6 Where to go from here
- 7 Troubleshooting

Basics: How to connect and flash the ESP8266

WARNING: Some devices as the "Sonoff basic" have their own power supply on board.

DO NOT USE THIS POWER SUPPLY FOR FLASHING!

Mains Voltage is dangerous for you and your computer! Always use a separate power supply.

What hardware do i need?

You obviously need a device with ESP8266. This can be a device with built-in ESP like the "Sonoff basic" switch, or a ESP8266 on a breadboard.

You may also use a nodeMCU or a WeMOS D1 mini.

The nodeMCU and the WeMOS D1 mini are recommended for newcomers. They are the most easy to handle version of ESP8266

If you use a nodeMCU or a WeMOS, you just need an USB cable, that's all

For other devices you need a power supply with 3.3V or a power supply with higher voltage and a voltage regulator down to 3.3V.

DO NOT USE MORE THEN 3.3V AT THE ESP!

It will kill the chip.

Further you need a USB to TTL converter (UART) that supports 3.3V.

Again: Do NOT use 5V devices!

Last but not least you need some cables with plugs ("DuPont cables").

Depending on your ESP you may need different cables such as male/male, male/female or female/female.

It's recommended to get an assorted bunch of such cables as you may need them to connect sensors, actuators and such for testing. Buy cables with many different colours. It helps....

A word about power supplies

Do yourself a big favour: Avoid cheap power supplies!

We have seen some strange behaviour from cheap power supplies. If you put a voltmeter on these it shows the correct voltage.

Everything looks nice.

If you use an oscilloscope you may get some nasty surprises. Oscillating voltage, ripples.....

Use a high quality power supply like MeanWell or similiar. It should have at least a current of 1 ampere.

It is not recommended to use the 3.3V supplied by the UART board.

The voltage regulators on these boards usually are far too limited to feed the ESP.

Recommended UART (USB to TTL converter)



Recommended types are those with a CH340G or a CP2xxx chip (CP2102 pictured above).

If you get an UART with settable 5V/3.3V set it to 3.3V. With some UART cables a wire has to be soldered accordingly.

Do not use cheap converters with a FTDI232 chip!

The FTDI232 might be one of the most faked chips at all. The genuine manufacturer took measurements against these fakes.

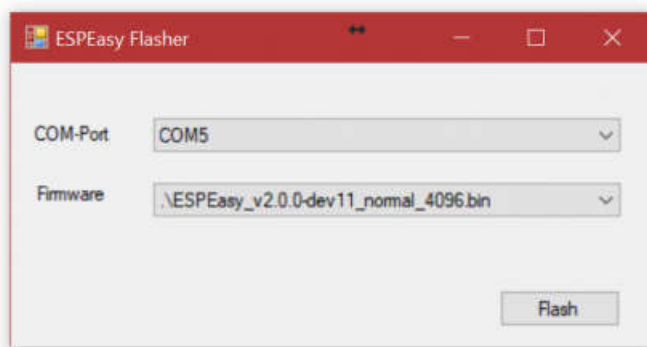
They will not work correctly at least with Windows. Everything looks fine, apparently flashing worked well.

But the code you flashed into the ESP is corrupted!

A genuine FTDI232 UART will cost around 15€ and up! Avoid these cheap 2€ FTDI's you may find on eBay or AliExpress!

Always remember to first install the drivers for the UART before plugging it to your computer!

Software



You can use several different softwares for flashing the ESP. It's recommended to use the flasher that is delivered with the ESP Easy software (FlashESP8266.exe).

It needs no settings and works just out of the box.

During testing it's useful to see what the ESP is doing.

ESP Easy is programmed to give information via the serial port, the same port you use for flashing.

For monitoring you may use the Arduino IDE, a simple terminal software like putty or a specialized program like ESPlorer (www.esplorer.ru).

These programs support a lot of devices so you have to set some parameters:

```
ESP Easy uses a transfer speed of 115200 bit/s.  
You have to set this speed in the program, otherwise you get "letter garbage" on the screen.
```

```
NOTE:  
A common mistake: Do not start the flashing program with the serial monitor running!  
The program started first will grab the COM-port! If your monitor program is running the flash program will not find the COM port and
```

```
Note:  
Please remember:  
The flasher is not able to really "read back" the flash rom.  
A "success" after flashing is done just says that data arrived  
at the header of the ESP-board.  
It does not say that data arrived in the flash chip itself!
```

Firmware

Of course you need the ESP Easy firmware. You will have to download the .zip file from GitHub (<https://github.com/letscontrolit/ESPEasy/releases>). Unpack the .zip to a folder of your choice. Also unpack the two EXE files

Those who want to compile their own version of ESP Easy should take a look at the wiki to find HowTo's.

Some basics about flashing

Those using the nodeMCU or WeMOS D1 mini may skip this paragraph. Everything is ready to use on these boards.

The ESP usually starts up from the internal flash that contains the firmware. This is called "run mode", it's the all days mode for using ESPEasy and the ESP chip.

For flashing the ESP it must be set into "flash mode". Said simply this starts another program at bootup, reading the serial port and writing the data into flash memory.

The ESP has four inputs that manage the startup behaviour. These settings are necessary to flash the chip!

The GPIO15 pin should always be set to "low". Use a 10KOhm resistor between GPIO15 and ground.

GPIO2 should be pulled to "high". Use a 10KOhm resistor between GPIO2 and 3.3V (Vcc).

The GPIO0 setting now decides between run mode and flash mode. Connect a 10 KOhm resistor between 3.3V (Vcc) and GPIO0. This sets the ESP8266 into run mode.

Now you may connect the GPIO0 directly to ground for flashing.

The fourth is the "reset" pin. It should be on "high", so put a 10 KOhm resistor between Vcc and reset pin.

For all ESP's besides nodeMCU and WeMOS there is a "CH-PG" pin. It's an "enable" pin you may use to switch off the whole ESP-Chip. Put a 10KOhm resistor between CH_PG and Vcc to keep the ESP running at all.

Devices with an embedded ESP-Chip (Sonoff basic or 4ch for example) will have these resistors on board. Just GPIO0 has to be set to ground with a jumper bridge or via a pushbutton.

Connecting to the computer for flashing

Get your computer ready

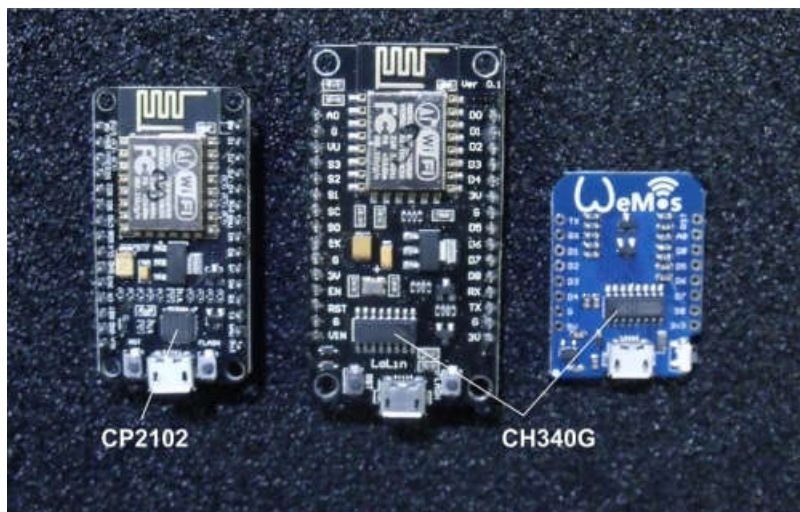
Install the serial monitoring program of your choice. Install the drivers for your UART (USB->TTL) converter.

Plug in the UART and open the device manager. Under the "Ports (COM & LPT)" you will notice an additional COM-Port, usually

COM3 or higher.

Note the number, you may need it.

The nodeMCU and the WeMos use same UART chips and drivers as the UART cables and boards.
Depending on your board you have to install the CH340G drivers or CP2102 drivers.
Identify by reading the marking on the chip near to the USB connector.



Connecting UART to ESP

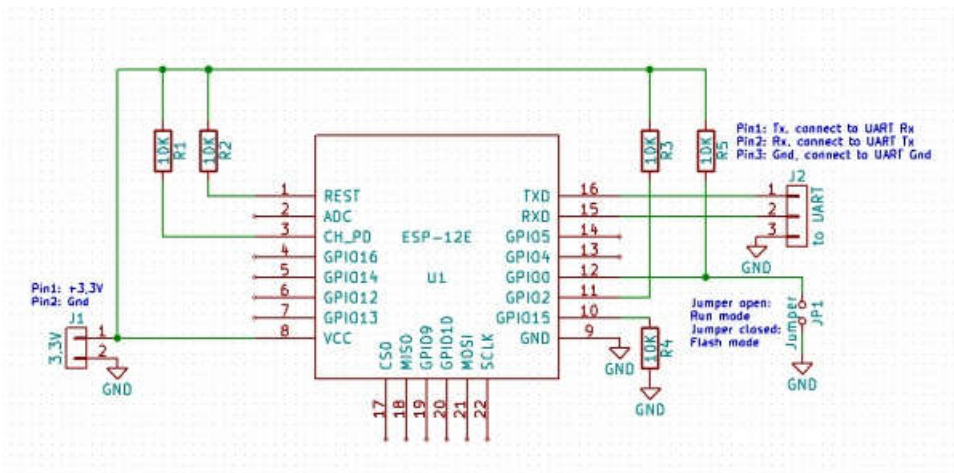
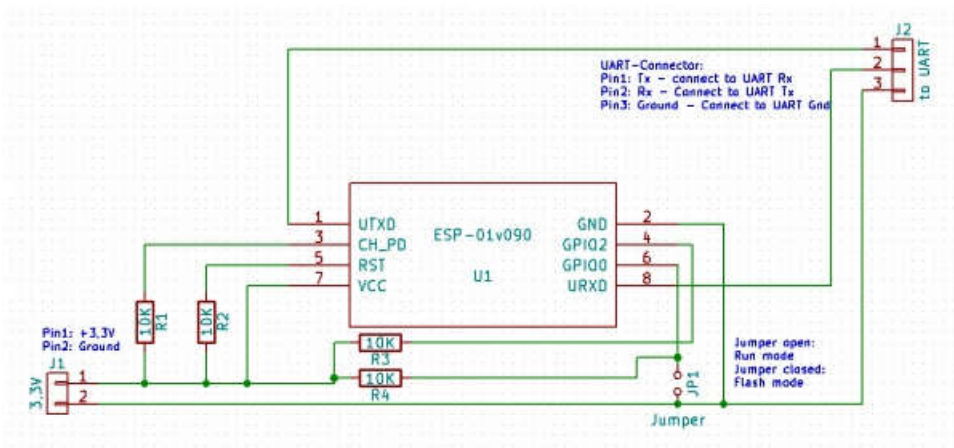
Again, those using the nodeMCU or WeMos D1 mini may skip this paragraph.

Remember the settings mentioned under some basics about flashing above. Now it depends a lot on what ESP you use.

The small ESP-01 has some connections already on board (GPIO15) You have to connect GPIO2 and Reset to Vcc and GPIO0 to ground as described above.

If you use the bigger ESP's (ESP07, 07s, 12, 12e, 12f) you have to take care of all connections being properly made.

Note: These ESP's are not very handy to use. If you want to use them an adapter board makes things a lot easier and breadboard friendly. They often have the necessary resistors on board too.



Now connect the UART:

| UART | | ESP |
|------|------|-----|
| Rx | <--> | Tx |
| Tx | <--> | Rx |
| Gnd | <--> | Gnd |

Do not switch on the power supply yet!

Connect the power supply to Vcc (3.3V) and Gnd. If using higher voltage power supplies, use a voltage regulator!

Watch out not to swap the leads accidentally - it will kill the ESP.

Double check your cabling! All connections made properly? Check against section above. Check it again.

Failures often result from cabling faults.

Flashing the nodeMCU / WeMOS D1 mini

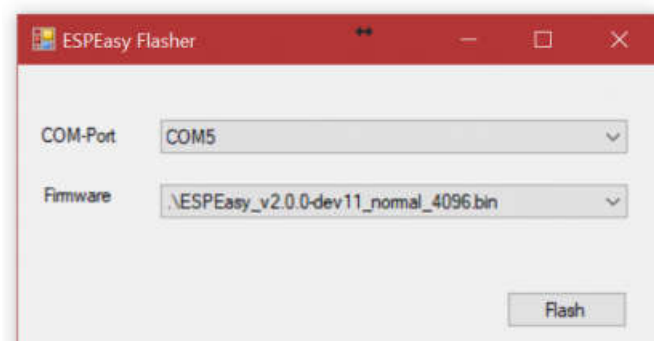
Just connect an USB cable to your board - that's all (since the units have onboard UART controllers). The flashing program will do the rest for you.

Start the flash program (FlashESP8266.exe) from the ESPEasy .zip file.

Source

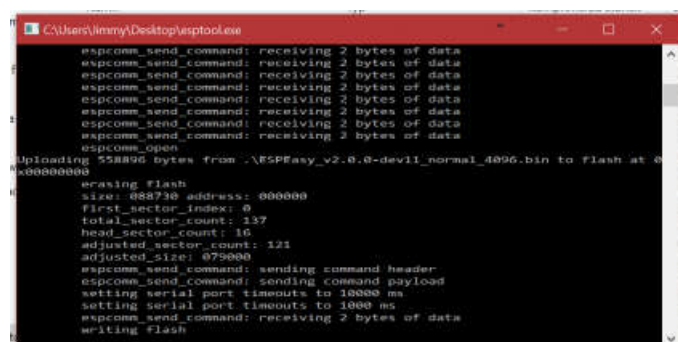
- ESPEasy_v2.0.0-dev11_dev_1024.bin
- ESPEasy_v2.0.0-dev11_dev_4096.bin
- ESPEasy_v2.0.0-dev11_dev_8285.bin
- ESPEasy_v2.0.0-dev11_normal_1024.bin
- ESPEasy_v2.0.0-dev11_normal_4096.bin
- ESPEasy_v2.0.0-dev11_normal_8285.bin
- ESPEasy_v2.0.0-dev11_test_1024.bin
- ESPEasy_v2.0.0-dev11_test_4096.bin
- ESPEasy_v2.0.0-dev11_test_8285.bin
- esptool.exe
- FlashESP8266.exe
- Plugin_sizes.txt
- Release_notes.txt

Select the COM-port you have noted above. In most cases it will show the correct port already.

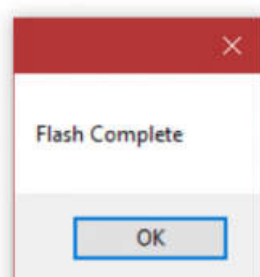


Remember, the file has to reside in the same folder as the flash program.

In case you are flashing a bare ESP click the flash button and power up immediately afterwards.



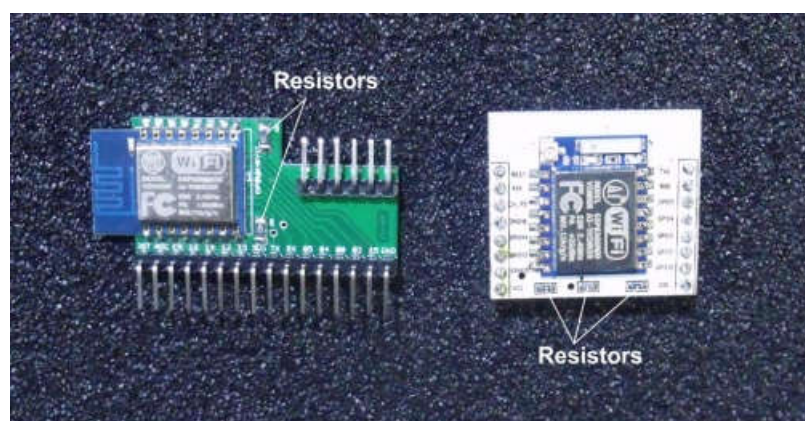
BE PATIENT! it takes some seconds until flashing finishes.



Wait for about two minutes: ESPEasy 2.x first formats it's internal file system, it takes a while to get this ready.

Now you're prepared to connect your ESP8266.

Flashing other ESP devices



You should have done all connections. **Do not power up the ESP now!**

... Follow the same flash procedure as above using the FlashESP8266.exe software. ...

Disconnect GPIO0 from ground, the resistor to Vcc should remain. Reset or power off / on your nodeMCU or WeMOS. It should be ready to use now.

Where to go from here

If everything has gone right you have a useable ESP Easy device now.

As no parameters are set it will go to "AP mode" for configuration.

Use your computer, tablet or smartphone and search a WiFi network named "ESP_Easy_0".

Connect to ESP_Easy_0 using the password "configesp" (without "").

Open your internet browser and type 192.168.4.1 as internet address into the browser. The WiFi setup of the ESP Easy opens.

You can choose your WiFi network now. Select the SSID and enter your passphrase, click connect.

It will take 20 seconds until a result is shown. If you typed everything correctly it will show a message that it is connected to the network and it shows the IP address.

Note the IP address!

Connect your computer or whatever back to your usual network. Open a browser and type the ip address into the browser.

You should see the config pages of ESPEasy now.

Troubleshooting

Flashing can go wrong by a lot of reasons. In many cases cabling faults are the reason.

If the flashing program says it can't find the ESP, try to swap the Tx and Rx cables at the ESP. Sadly some manufacturers are not too consequent with naming.....

Didn't help?

Check all of your cabling. All GPIO's set accordingly? Are all cables OK? If you got cheap connector cables they might have some contacting problems. A digital multimeter is a helpfull device for such cases. You might consider to get one for future use. Don't chose the cheapest one.

Retrieved from "http://www.esp8266.nu/wiki/index.php?title=Basics:_Connecting_and_flashing_the_ESP8266&oldid=4011"

■ This page was last edited on 16 February 2018, at 21:49.