Part 20
-
MicroPython

Version: 2021-01-25

Flashing MicroPython Firmware with esptool on ESP32

Instaling ESPTool

Unlike other boards, MicroPython isn't flashed onto the ESP32 by default. That's the first thing you need to do to start programming your boards with MicroPython

To work with `esptool`, you'll need Python 3.4 or a newer Python installation on your system. We recommend using Python 3.7.X or higher, so, if not installed yet, go to Python's website and install Python 3.7.X or higher on your computer.
With Python 3 installed, open a Terminal window and install the latest stable `esptool` release usibg `pip`:

```
pip3 install esptool
```

Note: with some Python installations that command may not work and you'll receive an error. If that's the case, try to install `esptool` with:

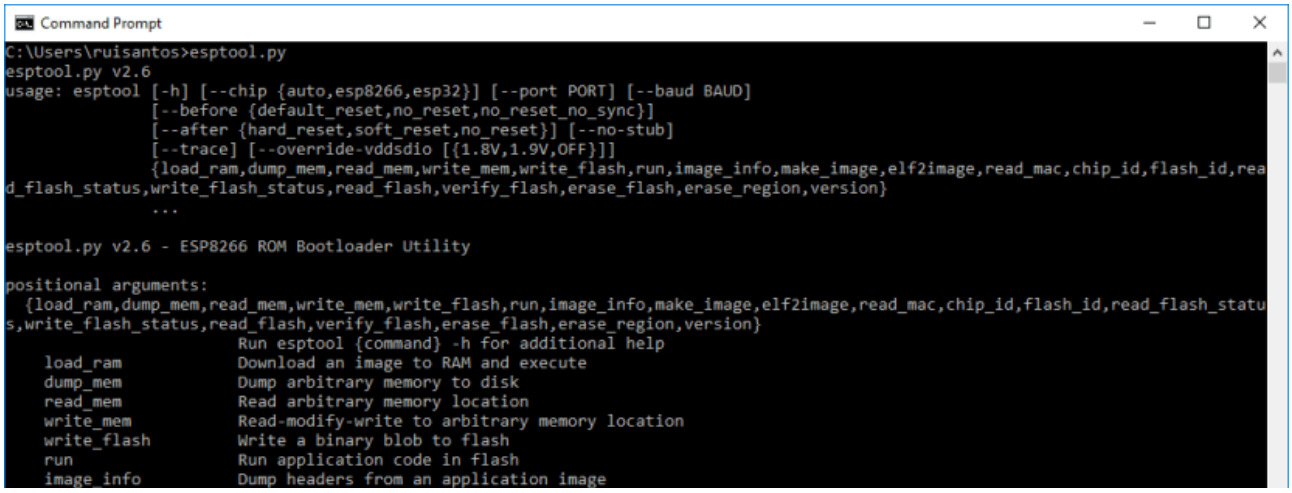
```
python -m pip install esptool
```

After installing, you will have `esptool.py` installed into the default Python executables directory and you should be able to run it with the command *`esptool.py`*. In your Terminal window, run the following command

```
esptool.py
```

Note: on Windwos you will find also `esptool.exe`. If found, use this one instead of `esptool.py`

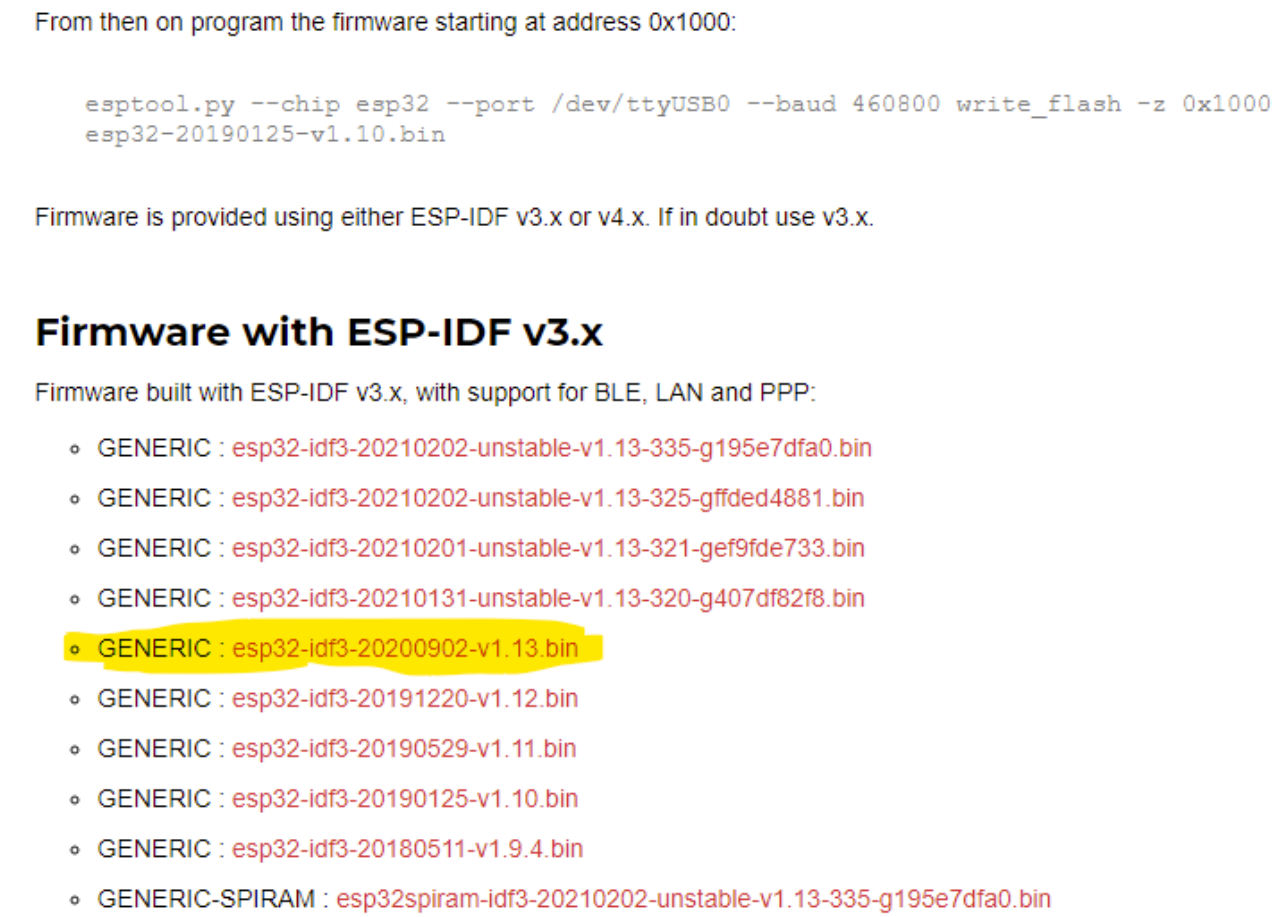
If it was installed properly, it should display a similar message (regardless of your operating system)



With `esptool` installed in your computer, you can easily flash your ESP32 boards with the MicroPython firmware.

Note: after installing MicroPython firmware on your ESP32, you can go back and use Arduino IDE again. You just need to upload code using Arduino IDE. Then, if you want to use MicroPython again, you need to flash MicroPython firmware.

Downloading and Flashing the MicroPython Firmware on ESP32
To download the latest version of MicroPython firmware for the ESP32, go to the MicroPython Downloads page and scroll all the way down to the ESP32 section.
You should see a similar web page (see figure below) with the latest link to download the ESP32 *.bin* file – for example: *esp32-20181007-v1.9.4-631-g338635ccc.bin*.



From then on program the firmware starting at address 0x1000:

```
esptool.py --chip esp32 --port /dev/ttyUSB0 --baud 460800 write_flash -z 0x1000
esp32-20190125-v1.10.bin
```

Firmware is provided using either ESP-IDF v3.x or v4.x. If in doubt use v3.x.

Firmware with ESP-IDF v3.x

Firmware built with ESP-IDF v3.x, with support for BLE, LAN and PPP:

- GENERIC : esp32-idf3-20210202-unstable-v1.13-335-g195e7dfa0.bin
- GENERIC : esp32-idf3-20210202-unstable-v1.13-325-gffded4881.bin
- GENERIC : esp32-idf3-20210201-unstable-v1.13-321-gef9fde733.bin
- GENERIC : esp32-idf3-20210131-unstable-v1.13-320-g407df82f8.bin
- GENERIC : esp32-idf3-20200902-v1.13.bin
- GENERIC : esp32-idf3-20191220-v1.12.bin
- GENERIC : esp32-idf3-20190529-v1.11.bin
- GENERIC : esp32-idf3-20190125-v1.10.bin
- GENERIC : esp32-idf3-20180511-v1.9.4.bin
- GENERIC-SPIRAM : esp32spiram-idf3-20210202-unstable-v1.13-335-g195e7dfa0.bin

Note: if you're using a different board (like a PyBoard, WiPy, or other), go to MicroPython Downloads page and download the right firmware for your board.

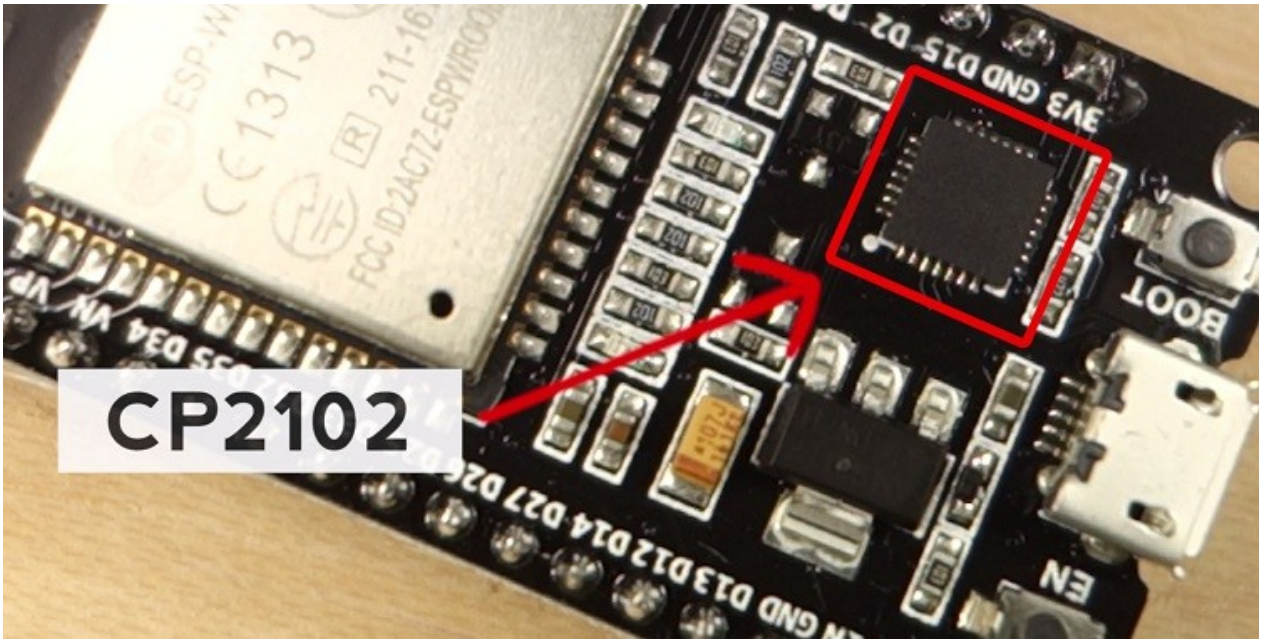
Finding the Serial Port Name
It's a bit different to find the Serial port name in each operating system, so for simplicity reasons we recommend finding your ESP serial port name through the Arduino IDE. Follow these steps:

1. Connect your board to your computer
2. Open the Arduino IDE
3. Go to Tools > Port
4. find your ESP32 serial port (in my case it's COM3)
5. Close your Arduino IDE software

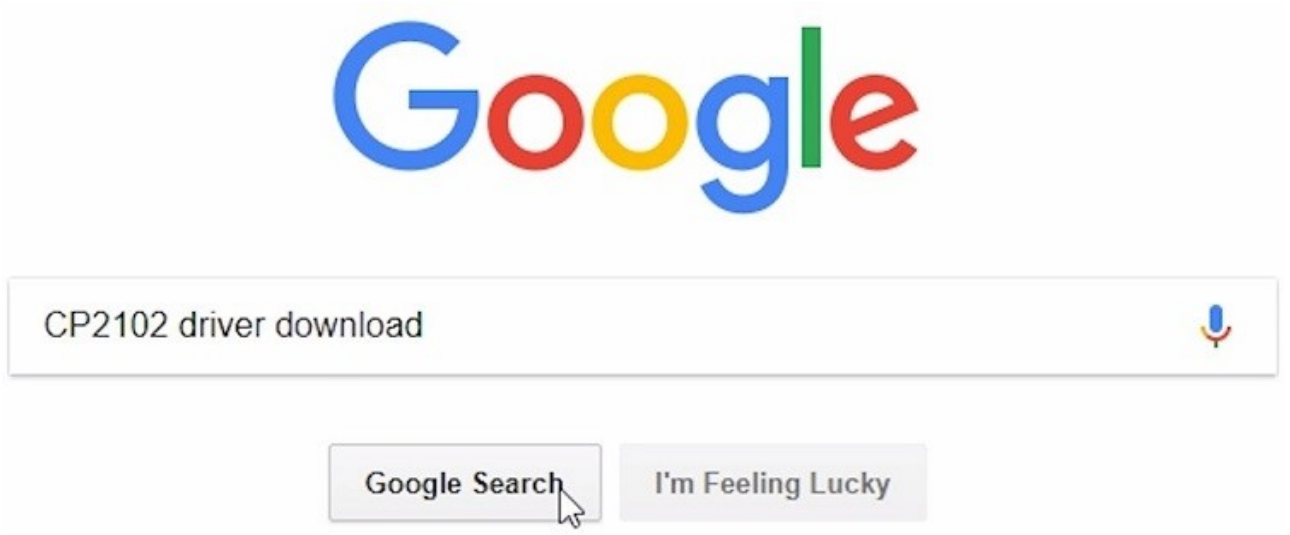
Important: if you plug your ESP32 board to your computer, but you can't find the ESP32 Port available in your Arduino IDE, it might be one of these two problems

1. USB drivers missing
2. USB cable without data wires.

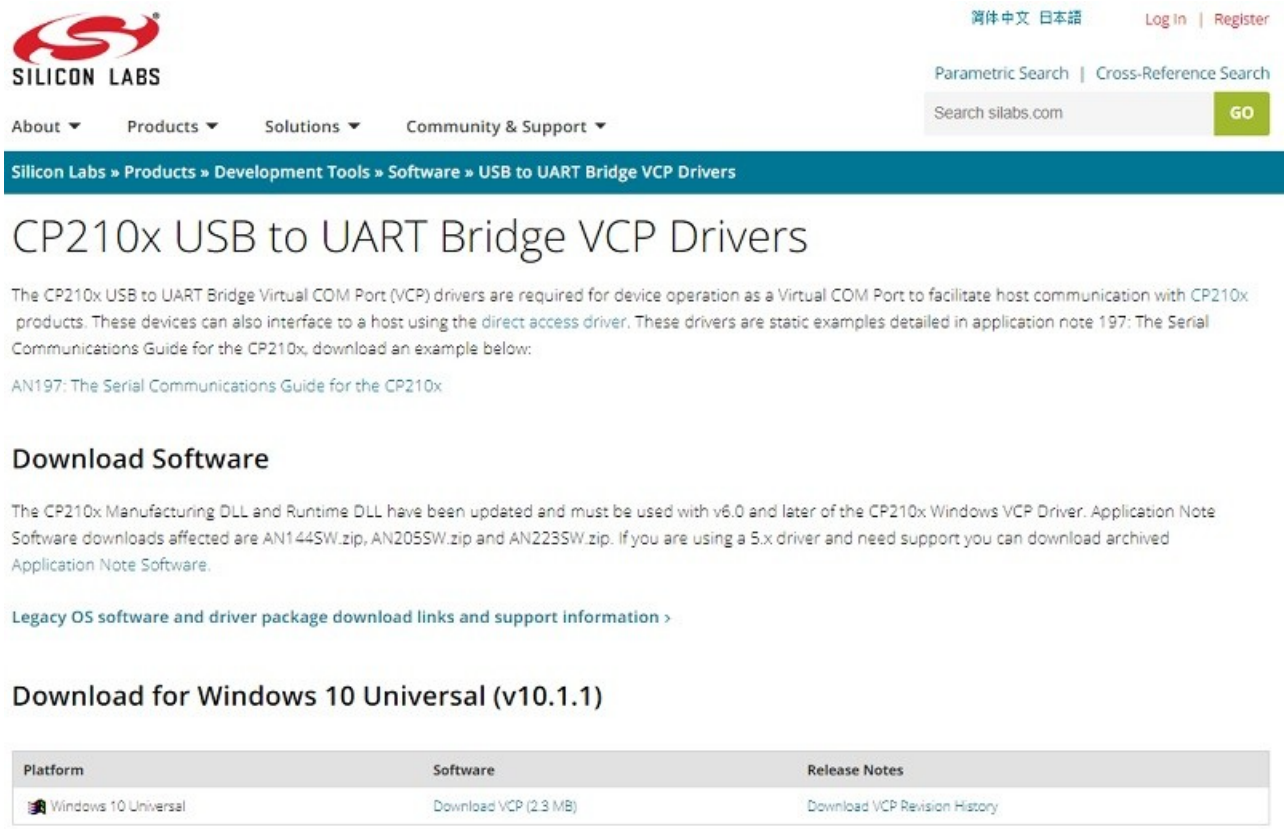
1. If you don't see your ESP's COM port available, this often means you don't have the USB drivers installed. Take a closer look at the chip next to the voltage regulator on board and check its name. The ESP32 DEVKIT V1 DOIT board uses the CP2102 chip.



Go to Google and search for your specific chip to find the drivers and install them in your operating system.



You can download the CP2102 drivers on the Silicon Labs website.

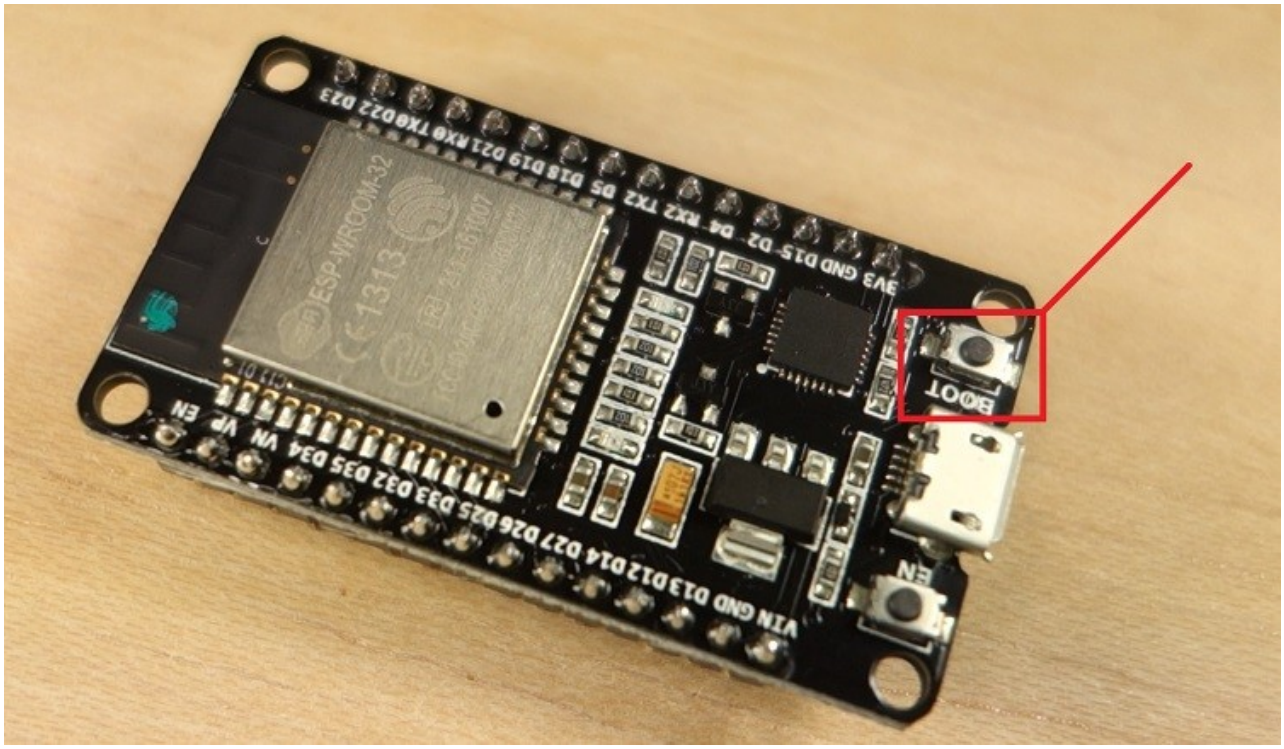


After they are installed, restart the Arduino IDE and you should see the serial port in the Tools > Port menu.

2. If you have the drivers installed, but you can't see your device, double-check that you're using a USB cable with data wires. USB cables from powerbanks often don't have data wires (they are charge only). So, your computer will never establish a serial communication with your ESP32. Using a proper USB cable should solve your problem.

Finding your MicroPython .bin file
After downloading the ESP32 *.bin* file, it should be in your Downloads folder. So, with your Terminal window, you'll need to navigate to the Downloads folder
List all files in your downloads folder to ensure that's where the *esp32-idf3-20200902-v1.13.bin* file is located.

Erasing ESP32 Flash Memory
Before flashing the MicroPython firmware, you need to erase the ESP32 flash memory. So, with your ESP32 connected to your computer, hold-down the "BOOT/FLASH" button in your ESP32 board



While holding down the "BOOT/FLASH" button, run the following command to erase the ESP32 flash memory

```
esptool.py/.exe --chip esp32 erase_flash
```

or

```
esptool.exe --chip esp32 erase_flash
```

When the "Erasing" process begins, you can release the "BOOT/FLASH" button. After a few seconds, the ESP32 flash memory will be erased.

```
D:\Users\Marc\My Desktop>esptool.exe --chip esp32 erase_flash
esptool.py v2.8
Found 2 serial ports
Serial port COM3
Connecting.....
Chip is ESP32D0WDQ6 (revision 1)
Features: WiFi, BT, Dual Core, 240MHz, VRef calibration in efuse, Coding Scheme None
WARNING: Detected crystal freq 41.01MHz is quite different to normalized freq 40MHz. Unsupported crystal in use?
Crystal is 40MHz
MAC: 10:52:1c:68:0f:0c
Uploading stub...
Running stub...
Stub running...
Erasing flash (this may take a while)...
Chip erase completed successfully in 7.3s
Hard resetting via RTS pin...
```

Note: if after the "Connecting …" message you keep seeing new dots appearing, it means that your ESP32 is not in flashing mode. You need to repeat all the steps described earlier and hold the "BOOT/FLASH" button again to ensure that your ESP32 goes into flashing mode and completes the erasing process successfully.

Fashing MicroPython Firmware on ESP32 with esptool.py
With your ESP32 flash memory erased, you can finally flash the MicroPython firmware. You need your serial port name (COM3 in my case) and the ESP32 *.bin* file location. Replace the next command with your details

```
esptool.py/.exe --chip esp32 --port <serial_port> write_flash -z 0x1000
<esp32-X.bin>
```

In my case, the final command looks like this:

```
esptool.exe --chip esp32 --port COM3 write_flash -z 0x1000  esp32-idf3-
20200902-v1.13.bin
```

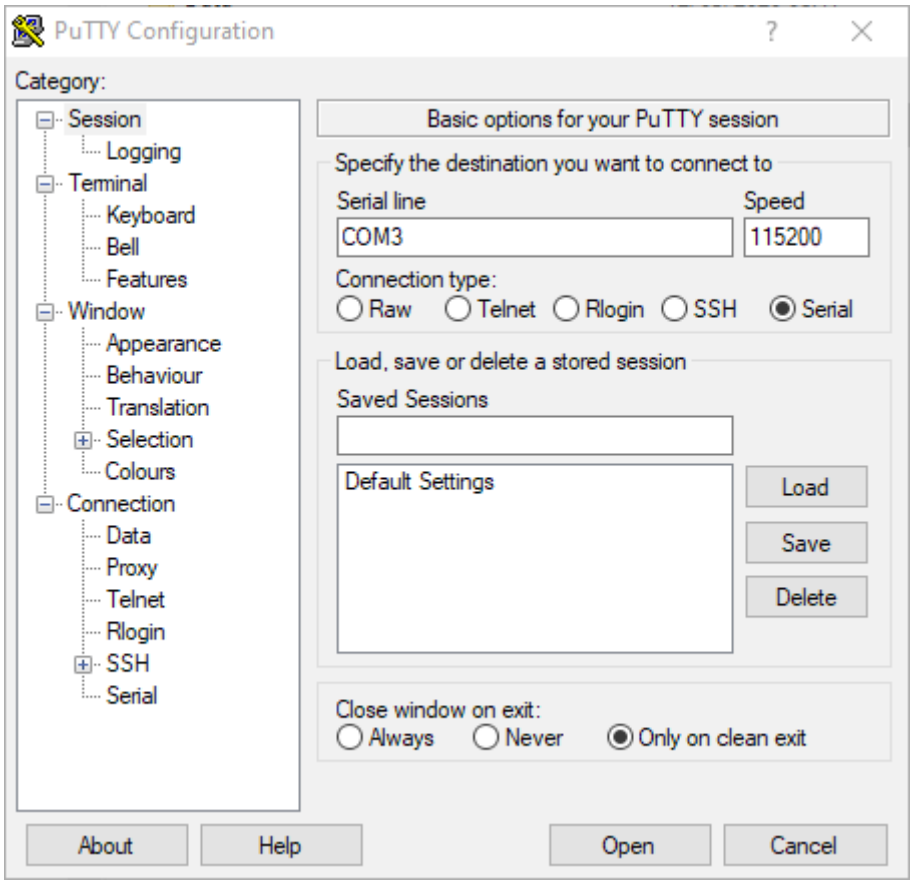
```
D:\Users\Marc\My Desktop>esptool.exe --chip esp32 --port COM3 write_flash -z 0x1000  esp32-idf3-20200902-v1.13.bin
esptool.py v2.8
Serial port COM3
Connecting....
Chip is ESP32D0WDQ6 (revision 1)
Features: WiFi, BT, Dual Core, 240MHz, VRef calibration in efuse, Coding Scheme None
Crystal is 40MHz
MAC: 10:52:1c:68:0f:0c
Uploading stub...
Running stub...
Stub running...
Configuring flash size...
Auto-detected Flash size: 4MB
Compressed 1448768 bytes to 926007...
Wrote 1448768 bytes (926007 compressed) at 0x00001000 in 82.5 seconds (effective 140.5 kbit/s)...
Hash of data verified.

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

Your ESP32 was successfully flashed with MicroPython firmware!

Programming your ESP32 with MircoPython – The simple way

You can use any serial terminal to access your ESP32 over is USB port. Eg. Putty.
Note: you van download Putty here https://www.putty.org/
Open Putty, select 'Serial'. Set the port to the port for your ESP32 on Windows and set Speed to 115200. Make sure no other serial program is accessing this COM port.



Click Open and you should see some like this

```
load:0x3fff0018,len:4
load:0x3fff001c,len:5008
ho 0 tail 12 room 4
load:0x40078000,len:10600
ho 0 tail 12 room 4
load:0x40080400,len:5684
entry 0x400806bc
I (539) cpu_start: Pro cpu up.
I (539) cpu_start: Application information:
I (540) cpu_start: Compile time:     Sep  2 2020 03:00:08
I (543) cpu_start: ELF file SHA256:  0000000000000000...
I (549) cpu_start: ESP-IDF:          v3.3.2
I (554) cpu_start: Starting app cpu, entry point is 0x40082f30
I (0) cpu_start: App cpu up.
I (564) heap_init: Initializing. RAM available for dynamic allocation:
I (571) heap_init: At 3FFAFF10 len 000000F0 (0 KiB): DRAM
I (577) heap_init: At 3FFB6388 len 00001C78 (7 KiB): DRAM
I (583) heap_init: At 3FFB9A20 len 00004108 (16 KiB): DRAM
I (589) heap_init: At 3FFBDB5C len 00000004 (0 KiB): DRAM
I (595) heap_init: At 3FFCA9E8 len 00015618 (85 KiB): DRAM
I (601) heap_init: At 3FFE0440 len 00003AE0 (14 KiB): D/IRAM
I (608) heap_init: At 3FFE4350 len 0001BCB0 (111 KiB): D/IRAM
I (614) heap_init: At 4009DE28 len 000021D8 (8 KiB): IRAM
I (620) cpu_start: Pro cpu start user code
I (303) cpu_start: Starting scheduler on PRO CPU.
I (0) cpu_start: Starting scheduler on APP CPU.
MicroPython v1.13 on 2020-09-02; ESP32 module with ESP32
Type "help()" for more information.
>>>
```

From here on you can start typing in Python code.