

Einrichtung einer Umgebung für Micropython auf dem ESP8266/32

Hier Beispielhaft für den ESP8266. Das Vorgehen ist beim ESP32 entsprechend.

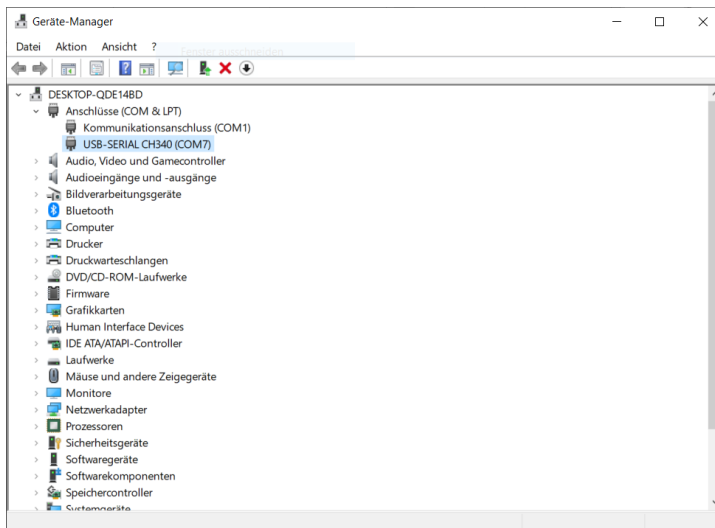
Vorbereitung des ESP

ESP wird mittels entsprechendem USB-Kabel an einen USB-Port des Computers angeschlossen. Dieser sollte sofort erkannt werden.

Anschließend muss ein Konsolenfenster (PowerShell/cmd) geöffnet werden. Via pip (pip installs package) wird das esptool heruntergeladen

```
> pip install esptool
```

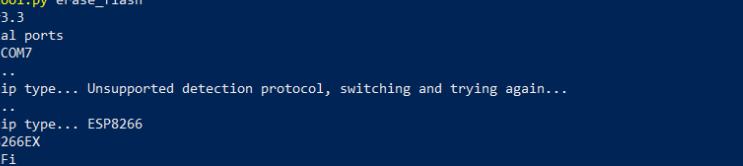
Mit des esptool wird der Flash des ESP geleert. Der Port kann über den Gerätemanager herausgefunden werden. (Hier „**COM7**“)



Mit dieser Information wird das esptool aufgerufen

```
> esptool.py --port COM7 erase_flash
```

```
> esptool.py erase_flash
```



```
Windows PowerShell
PS C:\> esptool.py erase_flash
esptool.py v3.3
Found 2 serial ports
Serial port COM7
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: a8:48:fa:c1:16:c4
Uploading stub...
Running stub...
Stub running...
Erasing flash (this may take a while)...
Chip erase completed successfully in 1.0s
Hard resetting via RTS pin...
PS C:\>
```

MicroPython downloads

 ESP266 with 1MiB flash Espressif	 ESP266 with 2MiB+ flash Espressif	 ESP266 with 512KiB flash Espressif
---	--	---

v1.18 (2022-01-17) .bin [elf] [map] [Release notes] (latest)
v1.17 (2021-09-02) .bin [elf] [map] [Release notes]
v1.16 (2021-06-18) .bin [elf] [map] [Release notes]
v1.15 (2021-04-18) .bin [elf] [map] [Release notes]
v1.14 (2021-02-02) .bin [elf] [map] [Release notes]
v1.13 (2020-09-11) .bin [elf] [map] [Release notes]
v1.12 (2019-12-20) .bin [elf] [map] [Release notes]
v1.11 (2019-05-29) .bin [elf] [map] [Release notes]
v1.10 (2019-01-25) .bin [elf] [map] [Release notes]
v1.9.4 (2018-05-11) .bin [elf] [map] [Release notes]
v1.9.3 (2017-11-01) .bin [elf] [map] [Release notes]
v1.9.2 (2017-08-23) .bin [elf] [map] [Release notes]
v1.9.1 (2017-06-12) .bin [elf] [map] [Release notes]
v1.9 (2017-05-26) .bin [elf] [map] [Release notes]
v1.8.7 (2017-01-08) .bin [elf] [map] [Release notes]

Der Speicherort ist wichtig und sollte sich gemerkt werden.

Nun wird mittels des esptools die Micropython Firmware aufgeflasht.

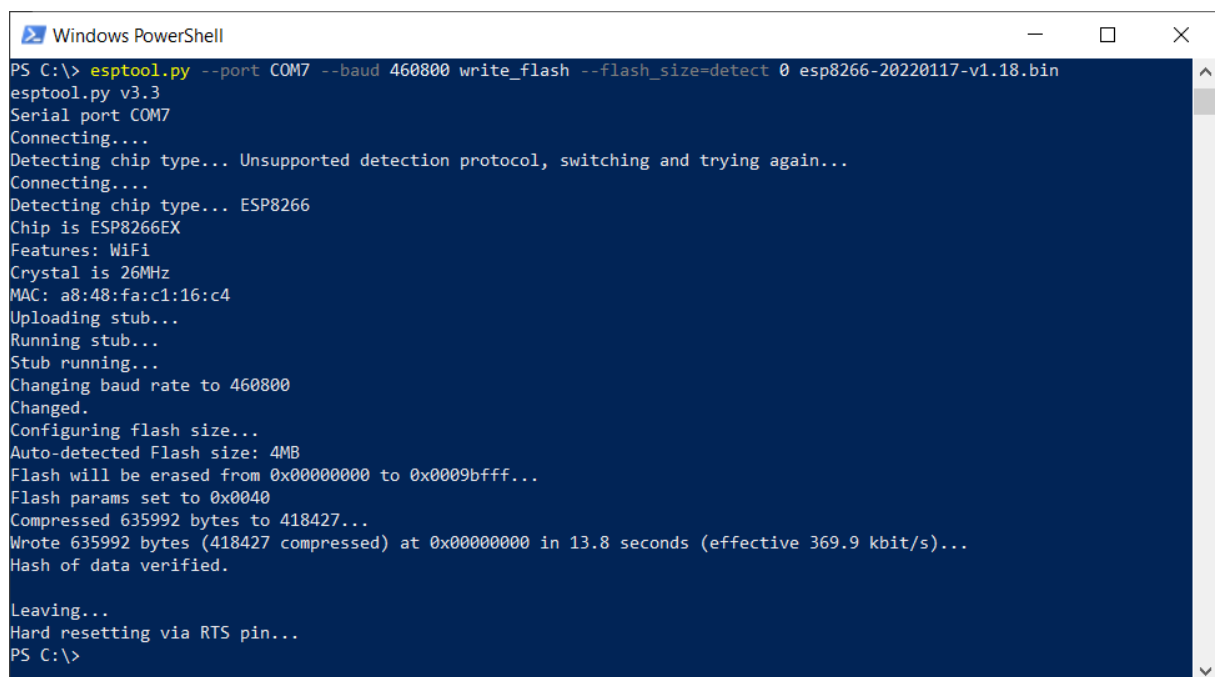
Für ESP8266

```
> esptool.py --port COM7 --baud 460800 write_flash  
--flash_size=detect 0 esp8266-20220117-v1.18.bin
```

Für ESP32

```
> esptool.py --chip esp32 --port COM7 write_flash -z 0x1000  
esp32-20180511-v1.9.4.bin
```

Port und Dateiname müssten entsprechend angepasst werden!

A screenshot of a Windows PowerShell window titled "Windows PowerShell". The command prompt shows the execution of the command: `PS C:\> esptool.py --port COM7 --baud 460800 write_flash --flash_size=detect 0 esp8266-20220117-v1.18.bin`. The output of the command is as follows:
`esptool.py v3.3
Serial port COM7
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: a8:48:fa:c1:16:c4
Uploading stub...
Running stub...
Stub running...
Changing baud rate to 460800
Changed.
Configuring flash size...
Auto-detected Flash size: 4MB
Flash will be erased from 0x00000000 to 0x0009bfff...
Flash params set to 0x0040
Compressed 635992 bytes to 418427...
Wrote 635992 bytes (418427 compressed) at 0x00000000 in 13.8 seconds (effective 369.9 kbit/s)...
Hash of data verified.

Leaving...
Hard resetting via RTS pin...
PS C:\>`

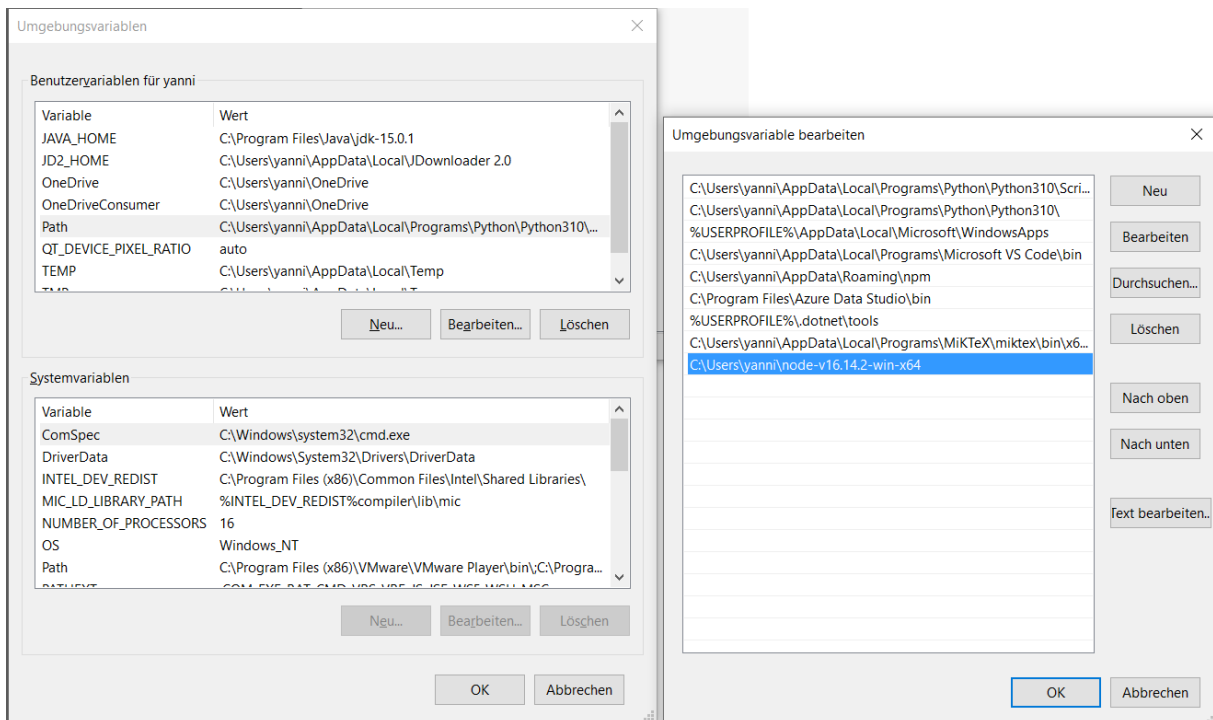
Der ESP ist jetzt vorbereitet.

VS Code einrichten

Wichtig ist hierbei, dass eine node.js Instanz installiert ist. Dies kann hier heruntergeladen werden: <https://nodejs.org/en/download/>

Sofern die Berechtigung zur Installation von Software vorliegt, kann node.js einfach mittel des .msi Packages installiert werden.

Ansonsten muss die Binary .zip heruntergeladen und bspw. im Benutzerordner entpackt werden. Anschließend ist es nötig, eine Umgebungsvariable auf das entsprechende Verzeichnis zu setzen. Dazu im Startmenü nach „**Umgebungsvariablen für diese Konto bearbeiten**“ suchen, im oberen Bereich „**Path**“ auswählen und auf „**Bearbeiten...**“ klicken. Im neuen Fenster dann auf „**Neu**“ und dort den Pfad des entpackten node.js-Ordners angeben. (Der Ordner, in dem sich die „node.exe“ befindet).



Anschließend in allen Fenstern auf „**OK**“ klicken.

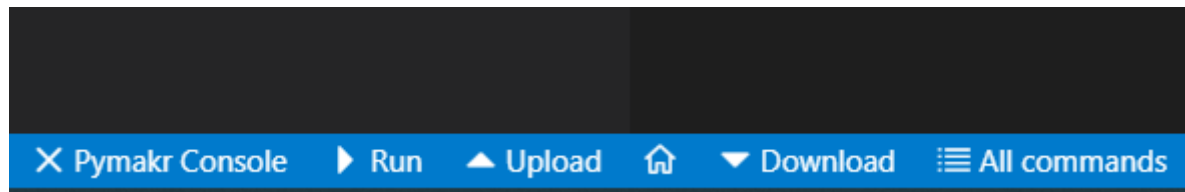
Nun wird VSCode geöffnet.

Hier wird über den Erweiterungsmanager die Erweiterung „**Pymakr**“ installiert.



Nun wird in einem beliebigen Ordner eine beliebige **.py** Datei erstellt.

Im unteren Bereich finden sich die Kontrollbuttons für Pymakr.



Sollte sich die **pymakr.json** nicht von selbst geöffnet haben, kann diese über „**All commands**“ → „**Pymakr > Global Setting**“ geöffnet werden.

A screenshot of a code editor showing the 'pymakr.json' file. The editor has a dark theme. The file path at the top is 'C: > Users > yanni > AppData > Roaming > Code > User > {} pymakr.json > ...'. The JSON content is as follows:

```
1 {  
2   "address": "COM7",  
3   "username": "",  
4   "password": "",  
5   "sync_folder": "",  
6   "open_on_start": true,  
7   "safe_boot_on_upload": false,  
8   "py_ignore": [  
9     "pymakr.conf",  
10    ".vscode",  
11    ".gitignore",  
12    ".git",  
13    "project.pymakr",  
14    "env",  
15    "venv"  
16  ],  
17   "fast_upload": false,  
18   "sync_file_types": "py,txt,log,json,xml,html,js,css,mpy",  
19   "ctrl_c_on_connect": false,  
20   "sync_all_file_types": false,  
21   "auto_connect": false,  
22   "autoconnect_comport_manufacturers": [  
23     "Pycom",  
24     "Pycom Ltd.",  
25     "FTDI",  
26     "Microsoft",  
27     "Microchip Technology, Inc.",  
28     "1a86"  
29   ]  
30 }
```

Hier wird für „**address**“ der COM-Port eingetragen und „**auto_connect**“ auf „**false**“ gesetzt.

Über „**All commands**“ → „**Pymakr > Connect**“ kann der ESP nun verbunden werden und anschließend die Python Skripte hochgeladen werden.