MicroPython op de ESP8266

Vorig jaar stond er in het Noordhollands dagblad een artikel over het zelf fabriceren van een fijnstofmeter. In de krant werd verwezen naar het Duitse burgermeetproject Luftdaten waar verder informatie alsmede de bouwinstructies voor de fijnstofmeter te vinden zijn.
Ik heb de fijnstofmeter volgens de instructies in elkaar gezet en dat was prima te doen. Het werkte prima alleen had ik een beetje moeite met de stroomvoorziening. Hij hing bij mij buiten op het balkon en omdat ik daar geen elektriciteit heb gebruikte ik batterijen en die waren sneller leeg dan ik verwachtte.

De onderdelen van de fijnstofmeter kocht ik bij Ali Expres en dat was voor mij wel een eye-opener. Ongelofelijk dat het zo goedkoop kan zijn en dan sturen ze het ook nog eens gratis op. Het duurt meestal wel een paar weken voor je de waar binnen hebt maar het is het wachten wel waard.

Een van de onderdelen was de ESP8266, een microcontroller met een wifi-module. Deze zorgde er voor dat de meetgegevens via wifi naar een online database gestuurd konden worden.

In eerdere handleidingen heb je kunnen lezen dat ik ook met de Micro:bit heb lopen experimenteren. Omdat de Micro:bit geen wifi-module heeft leek het mij aardig om te kijken of ik de ESP8266 kon koppelen aan de Micro:bit om zo de gegevens van de Micro:bit ook in de cloud te kunnen opslaan.

Die probeersels zijn voor een volgende keer, in deze handleiding laat ik zien hoe de ESP8266 werkt en hoe je daar MicroPython op kunt gebruiken.

Standaard wordt dit type ESP8266 (de NodeMCU) ondersteunt met LUA een nieuwe programmeertaal, maar omdat ik geen nieuwe programmeertaal wil leren voor alleen dit project heb ik gebruik gemaakt van MicroPython.

Op de home-site van MicroPython staat beschreven hoe je de ESP8266 kan voorzien van MicroPython. De eerste stap is het downloaden van de laatste firmware, in mijn geval was dat versie 1.11

Firmware for ESP8266 boards

The following files are stable firmware for the ESP8266. Program your board using the esptool.py program as described in the tutorial.

- esp8266-20190529-v1.11.bin (elf, map) (latest)
- esp8266-20190125-v1.10.bin (elf, map)
- esp8266-20180511-v1.9.4.bin (elf, map)
- esp8266-20171101-v1.9.3.bin (elf, map)

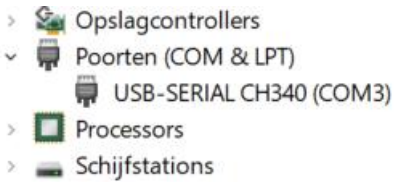
Daarna moest ik **esptool** installeren en omdat ik al Python op mijn laptop had staan kon ik volstaan met het commando:

```
pip install esptool
```

Om de bestaande firmware te verwijderen moest ik het commando

```
esptool.py --port /dev/ttyUSB0 erase_flash
```

Uitvoeren. Daarvoor moest ik wel weten via welke poort de ESP8266 aan mijn laptop was verbonden. Deze gegevens zijn te vinden via Apparaatbeheer. In mijn geval werd de ESP8266 aan mijn laptop verbonden via port COM3, voor een ieder kan dat anders zijn.



Dit betekende dat ik onderstaand commando moest uitvoeren:

```
esptool.py --port COM3 erase_flash
```

```
PS C:\Users\gebruiker> esptool.py --port COM3 erase_flash
esptool.py v2.6
Serial port COM3
Connecting....
Detecting chip type... ESP8266
Chip is ESP8266EX
Features: WiFi
MAC: 5c:cf:7f:89:10:ff
Uploading stub...
Running stub...
Stub running...
Erasing flash (this may take a while)...
Chip erase completed successfully in 14.9s
Hard resetting via RTS pin...
PS C:\Users\gebruiker>
```

Daarna de nieuwe firmware installeren via

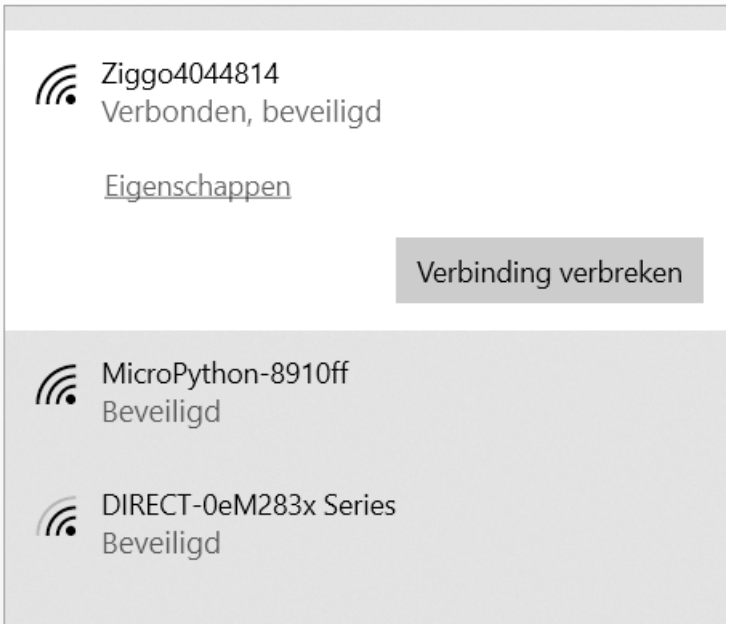
```
esptool.py --port COM3 --baud 460800 write_flash --flash_size=detect 0
esp8266-20190529-v1.11.bin
```

```
PS C:\Users\gebruiker> esptool.py --port COM3 --baud 460800 write_flash --flash_size=detect 0 esp8266-20190529-v1.11.bin
esptool.py v2.6
Serial port COM3
Connecting....
Detecting chip type... ESP8266
Chip is ESP8266EX
Features: WiFi
MAC: 5c:cf:7f:89:10:ff
Uploading stub...
Running stub...
Stub running...
Changing baud rate to 460800
Changed.
Configuring flash size...
Auto-detected Flash size: 4MB
Flash params set to 0x0040
Compressed 617880 bytes to 402086...
Wrote 617880 bytes (402086 compressed) at 0x00000000 in 9.3 seconds (effective 529.8 kbit/s)...
Hash of data verified.

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

Hou er met de laatste instructie wel rekening mee dat het firmware-bestand in de juiste map staat (in mijn geval C:\Users\gebruiker).

Na de installatie van de firmware verscheen de ESP8266 in mijn lijst van wifi verbindingen.



Zou ik vanaf mijn laptop via wifi met de ESP8266 willen communiceren, dan heb ik twee mogelijkheden. Ik kan met mijn laptop inloggen op het door de ESP8266 opgezette netwerk (in mijn geval MicroPython-8910ff) met het wachtwoord “micropythoN” of ik zou de ESP8266 zo kunnen programmeren dat het apparaat inlogt op het netwerk waarop mijn laptop is verbonden (in mijn geval Ziggo4044814). Dit laatste heeft de voorkeur omdat ik dan met mijn computer verbonden blijf aan het Internet, iets wat niet mogelijk is bij de eerste optie.

In de eerder genoemde handleiding laten ze het netwerk nog even voor wat het is en beginnen ze met de REPL, een interactieve prompt vergelijkbaar met de command prompt van Windows. In een volgende handleiding zal ik daar aandacht aan besteden.