

De ESP8266 programmeren

In een eerdere handleiding heb ik laten zien hoe je MicroPython kunt installeren op de ESP8266.

In deze handleiding laat ik zien hoe je op de ESP8266 commando's kunt laten uitvoeren (in MicroPython natuurlijk).

We beginnen hierbij met PuTTY, een terminal emulator. Een terminal werd voor de introductie van de pc's gebruikt om een computer of afstand te beheren. Een terminal emulator zoals PuTTY is een software programma dat zo'n terminal simuleert.

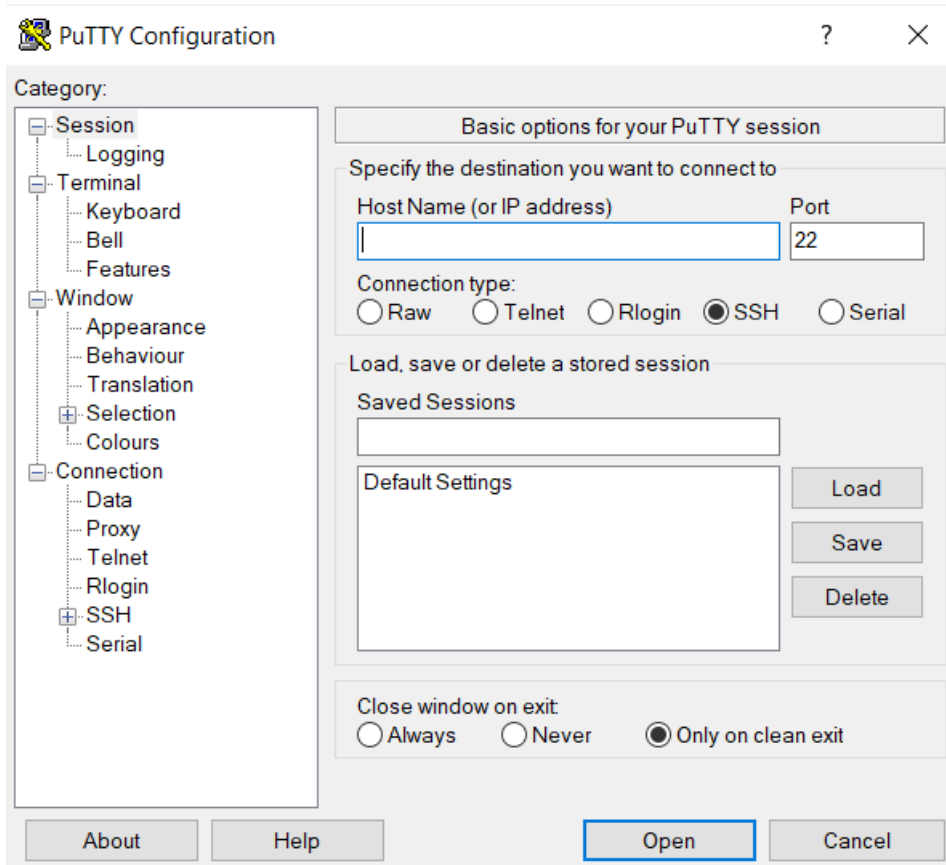
PuTTY is te downloaden van de site <https://www.putty.org/>

Alvorens PuTTY op te starten moeten we de ESP8266 verbinden met de computer via een usb-kabel. In mijn geval wordt de ESP8266 aangesloten via seriële poort COM3, maar dat kan voor iedereen verschillend zijn.

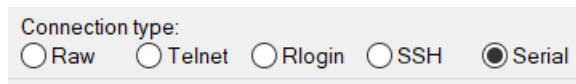


Bij het instellen van PuTTY maak ik gebruik van COM3, maar jullie dienen dus jullie eigen poortnummer te gebruiken.

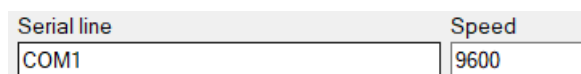
Starten we PuTTY op, dan krijgen we het volgende scherm:



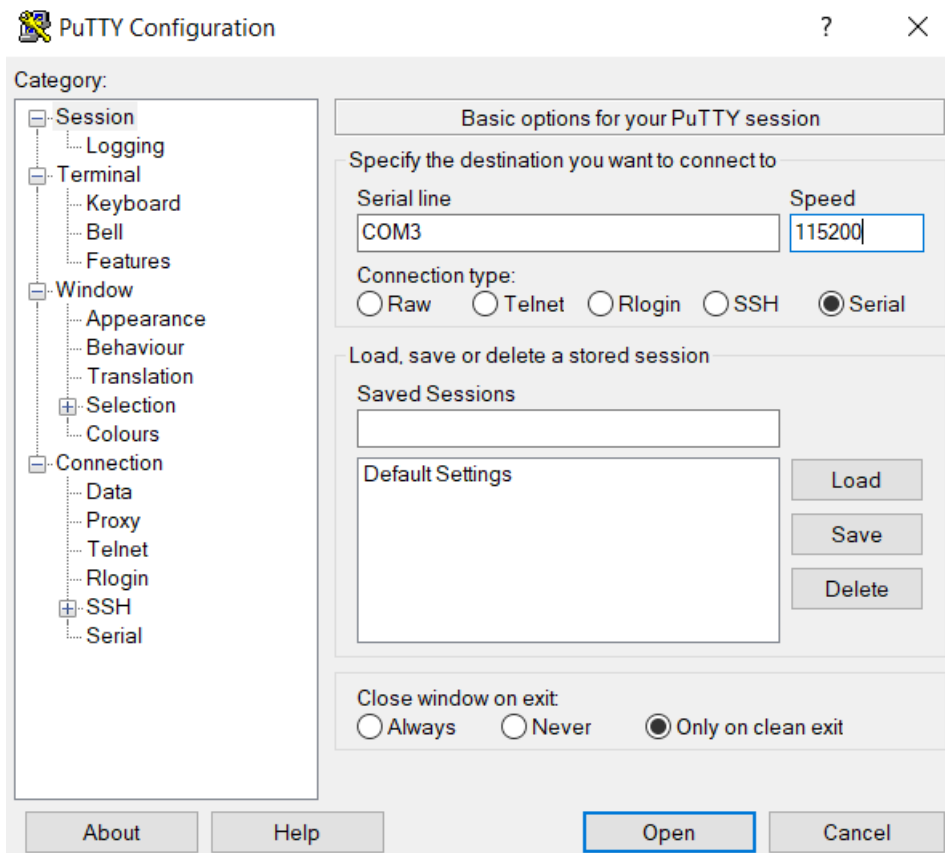
Om verbinding te maken met de ESP8266 moeten we bij het type verbinding kiezen voor **Serial**.



Standaard staan de instellingen daarna op poort COM1 en op een snelheid van 9600 baud.

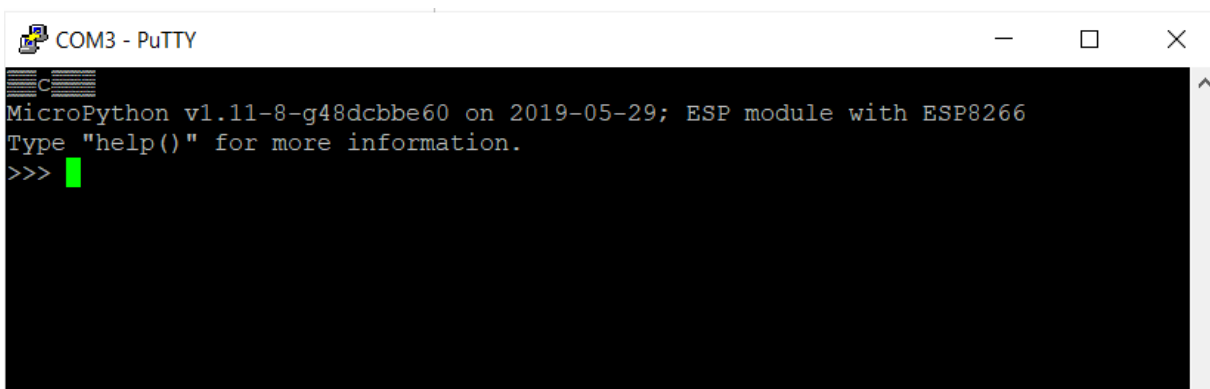


COM1 moeten we hierbij veranderen in de poort waarmee de ESP8266 is aangesloten en de snelheid moeten we wijzigen in 115200.



Druk daarna op de knop **Open** en de verbinding zou tot stand moeten komen.

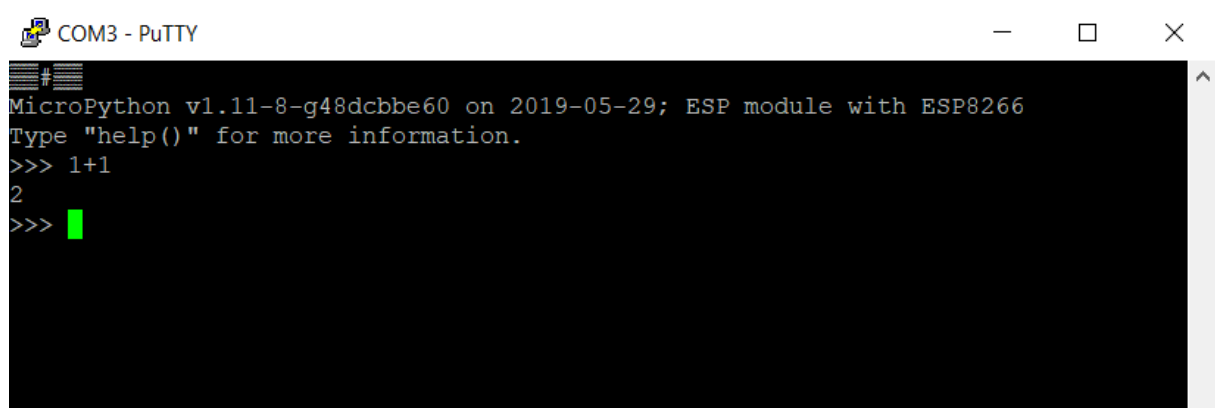
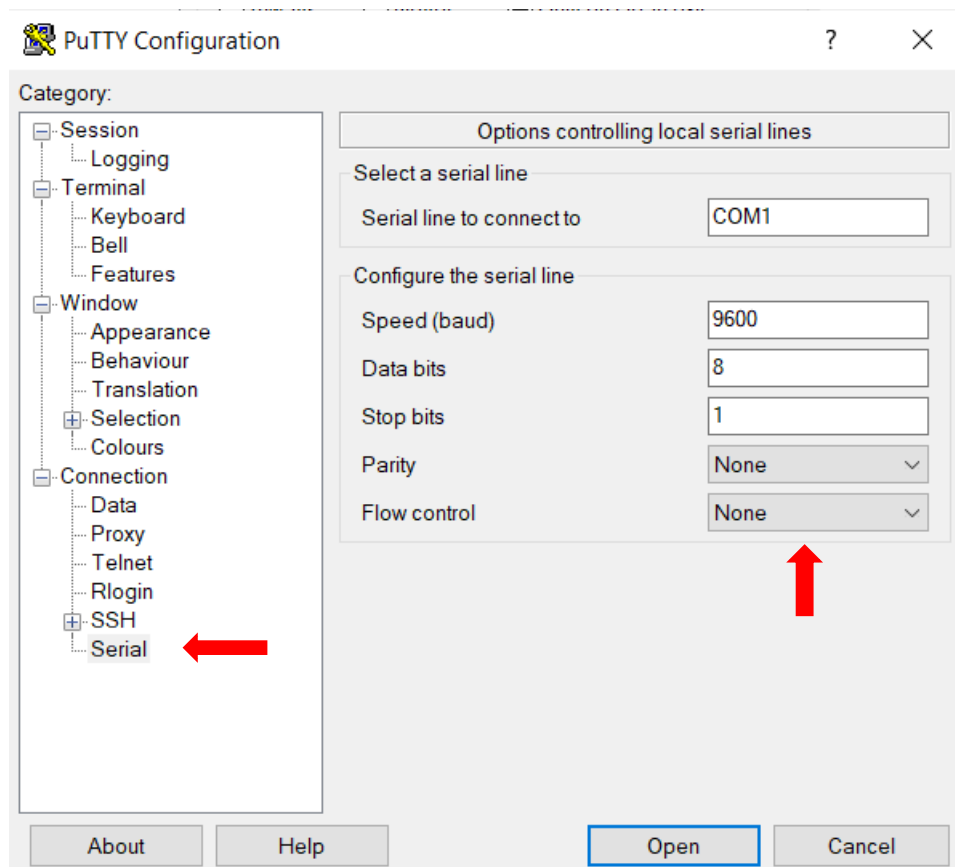
In eerste instantie gebeurt er niets maar dat verandert als je even op de **RST-knop** drukt van de ESP8266.



Je ziet een opdracht prompt (>>>) waarachter je de MicroPython commando's kunt invullen.

Tenminste, zo zou het moeten werken. Jammer genoeg kon ik geen tekens invoeren in het scherm. Het probleem zat 'm in de instellingen van PuTTY.

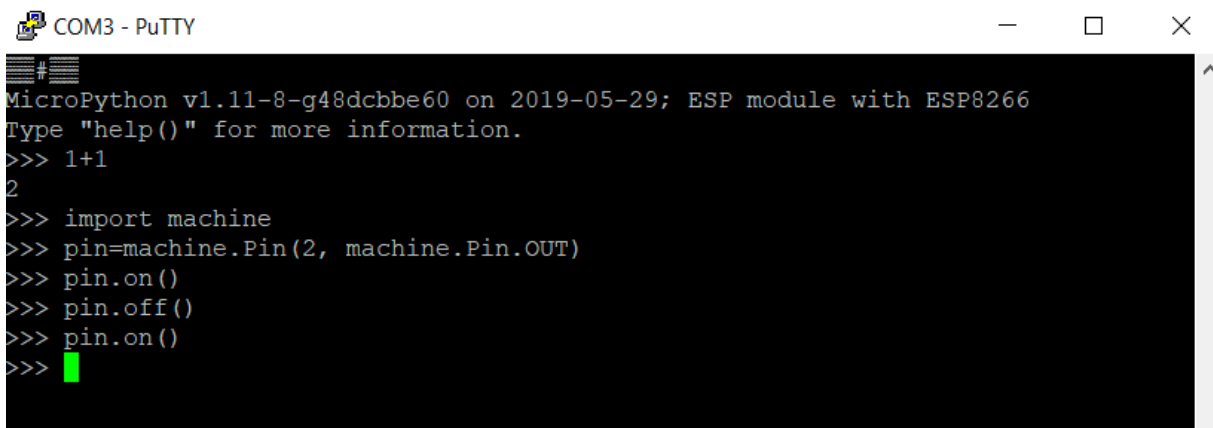
Nadat ik bij de instellingen voor de seriële verbinding gekozen had voor geen “Flow control” werkte het prima.



Zoals je ziet heb ik achter de prompt de opdracht **1+1** ingetypt en deze afgesloten met **ENTER**.

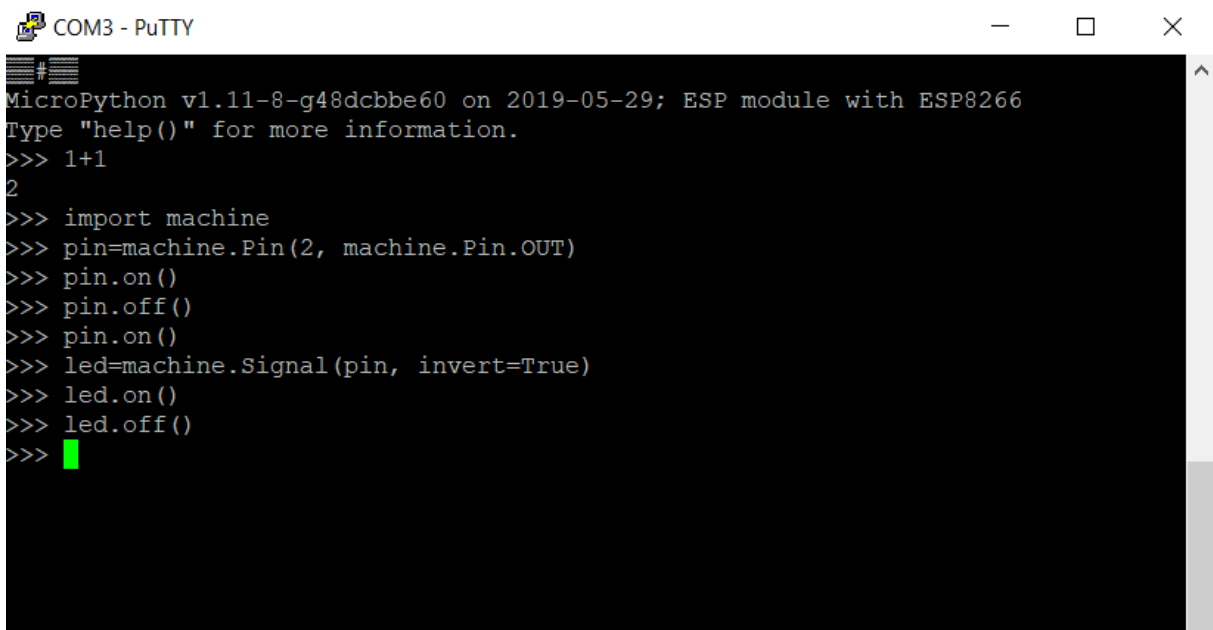
Mooi dat de MicroPython interpreter met het juiste antwoord komt.

In de MicroPython handleiding geven ze aan hoe je de ingebouwde led kunt aansturen.



```
COM3 - PuTTY
MicroPython v1.11-8-g48dcbb60 on 2019-05-29; ESP module with ESP8266
Type "help()" for more information.
>>> 1+1
2
>>> import machine
>>> pin=machine.Pin(2, machine.Pin.OUT)
>>> pin.on()
>>> pin.off()
>>> pin.on()
>>> █
```

Ik heb de commando's ingetypt en inderdaad kon ik de led op de ESP8266 aansturen. Jammer genoeg ging de led uit met het commando **pin.on()** en juist aan met het commando **pin.off()**. Volgens de handleiding heeft dit te maken met de wiring van de EP8266 en is het te verhelpen met de machine.Signal klasse.



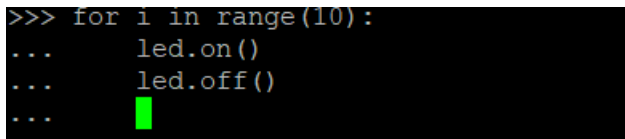
```
COM3 - PuTTY
MicroPython v1.11-8-g48dcbb60 on 2019-05-29; ESP module with ESP8266
Type "help()" for more information.
>>> 1+1
2
>>> import machine
>>> pin=machine.Pin(2, machine.Pin.OUT)
>>> pin.on()
>>> pin.off()
>>> pin.on()
>>> led=machine.Signal(pin, invert=True)
>>> led.on()
>>> led.off()
>>> █
```

Inderdaad, door het gebruik van het commando **led=machine.Signal(pin, invert=True)** kon ik de led aansturen met **led.on()** en **led.off()**.

We kunnen de led ook laten knipperen. Probeer de volgende code maar eens in te voeren:

```
for i in range(10):  
    led.on()  
    led.off()
```

Als je dit invoert in PuTTY komt het er als volgt uit te zien:



```
>>> for i in range(10):  
...     led.on()  
...     led.off()  
...     █
```

De console begrijpt dat je na een for-lus wilt inspringen en doet dat automatisch voor je. Wil het je inspringen afsluiten dan moet je de spaties op de laatste regel weghalen en op ENTER drukken.

En werkte het?

In eerste instantie denk je waarschijnlijk dat je iets fout hebt gedaan maar het programma werkt prima. Het knipperen van de led gaat alleen zo snel dat je er niets van merkt. We moeten dan ook tussen elke instructie een korte pauze inlassen. Probeer dit eens:

```
import time  
for i in range(10):  
    led.on()  
    time.sleep(0.5)  
    led.off()  
    time.sleep(0.5)
```

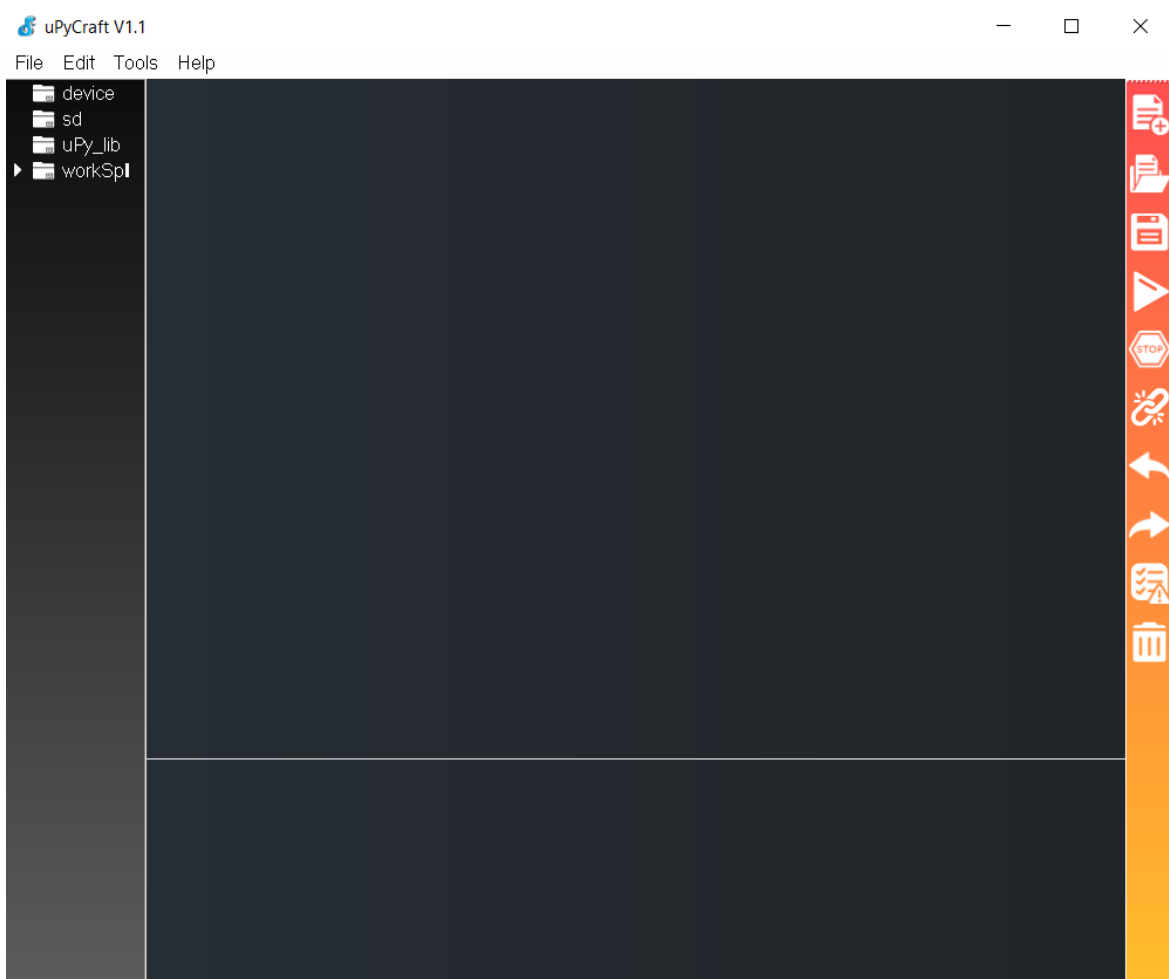
Nu zou je de led moeten zien knipperen.

PuTTY is een prima hulpmiddel om snel wat instructies op de ESP8266 uit te voeren maar voor het echte programmeerwerk kun je beter gebruik maken van een IDE. Een IDE is een software applicatie die een aantal faciliteiten levert aan computerprogrammeurs voor software ontwikkeling. Een IDE bestaat gewoonlijk uit een broncode editor, geautomatiseerde hulpmiddelen om bijvoorbeeld bestanden te beheren en een debugger (helpt je met het oplossen van fouten in je code).

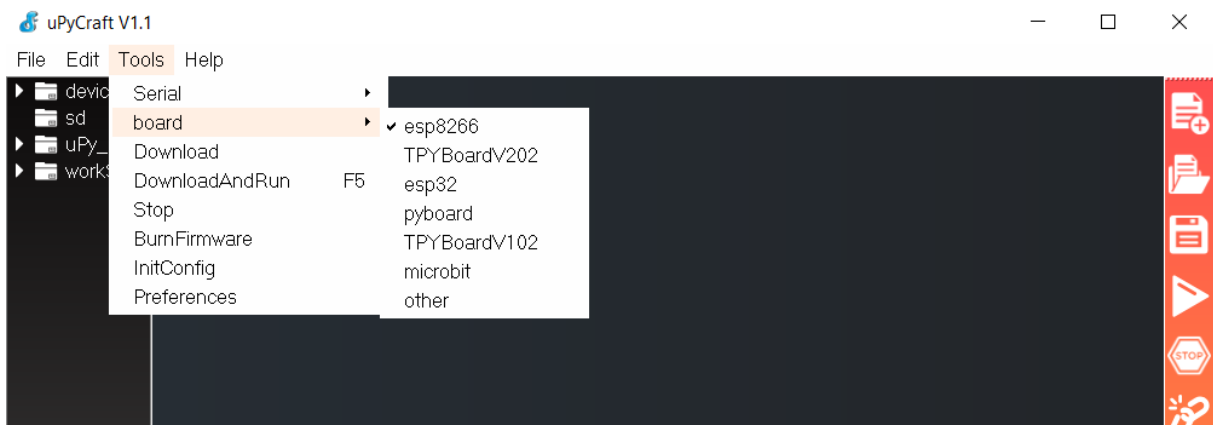
Voor lezers die ervaring hebben met de Arduino zou de *Arduino IDE* uitkomst kunnen bieden. Anderen zou ik willen aanraden om *uPyCraft* te gebruiken.

De *uPYCraft* installatie link is [hier](#) te vinden. Het is een windows exe file dus downloaden en opstarten is voldoende. Tijdens de installatie wordt gevraagd of je het SourceCode font wilt installeren. Volgens mij is het niet perse noodzakelijk maar met het font zie het er waarschijnlijk wel wat mooier uit. Een handleiding om het font onder Windows te installeren is [hier](#) te vinden.

Nadat je een eventuele update hebt geïnstalleerd krijg je het volgende scherm te zien.

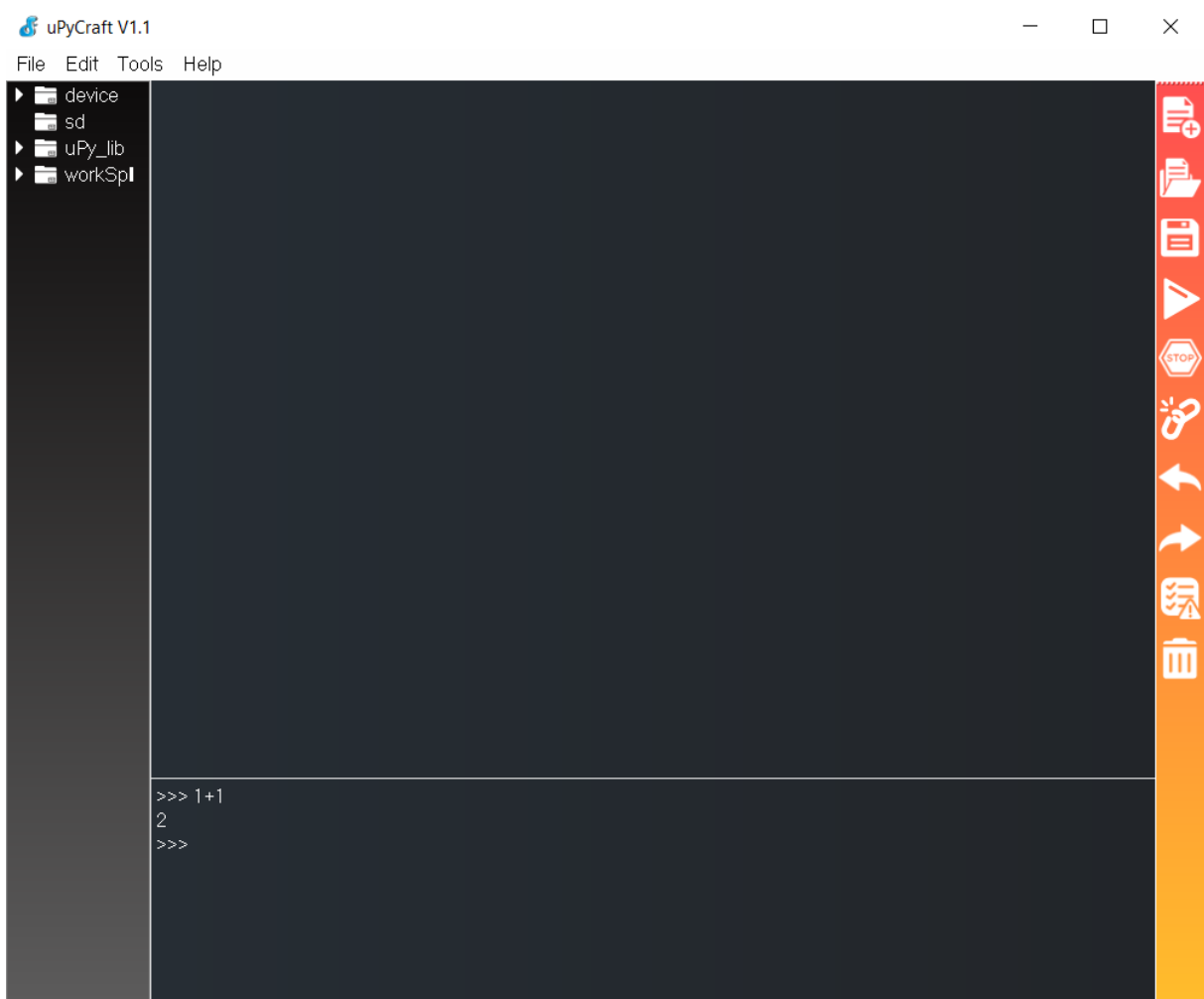


Om te beginnen zullen we de esp8266 moeten verbinden. Druk daartoe op de knop **Tools** en kies vervolgens voor het board esp8266.




Druk nogmaals op de Tools knop en kies de juiste seriële poort (bij mij is dat COM3).


Als je dit gedaan hebt verschijnt onder aan het scherm de console van de esp8266 net als bij PuTTY. Je kunt hier weer commando's intypen en deze vervolgens laten uitvoeren.

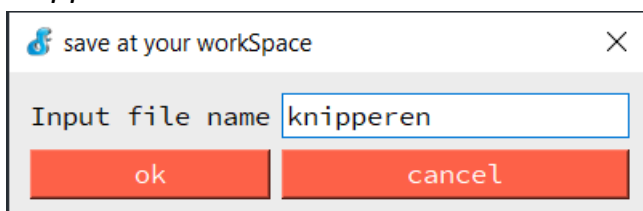


Laten we eens kijken of we de led weer kunnen laten knipperen

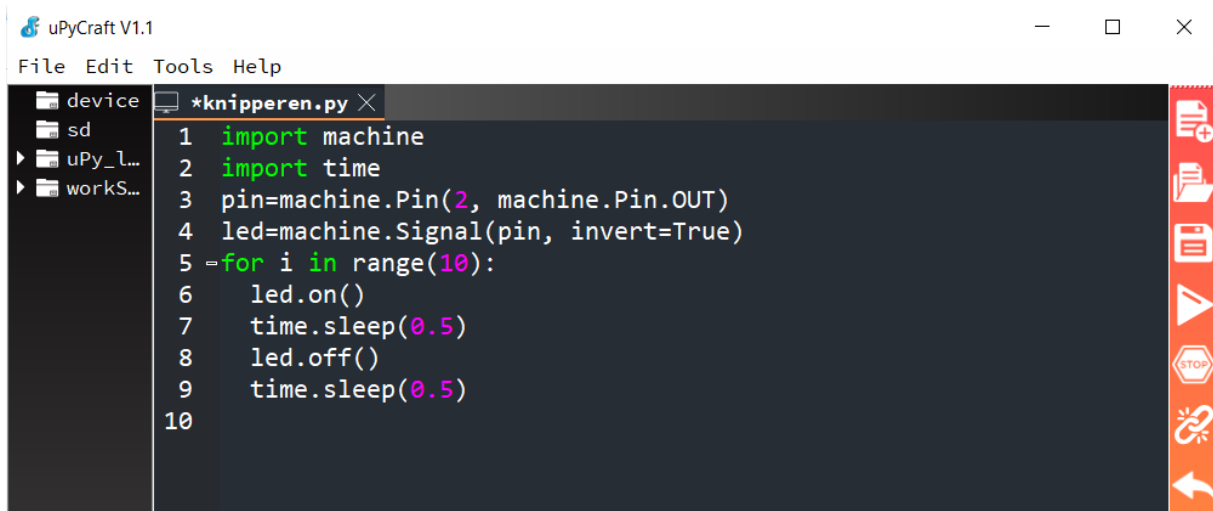
Druk op de  knop om een nieuw bestand te openen. Je krijgt het volgende scherm te zien




Er is automatisch een bestand *untitled* aangemaakt en geopend in de editor. Het beste kun je gelijk dit bestand even opslaan zodat je het een geschikte naam kan geven. Druk op de  knop en sla het bestand op onder de naam *knipperen*.



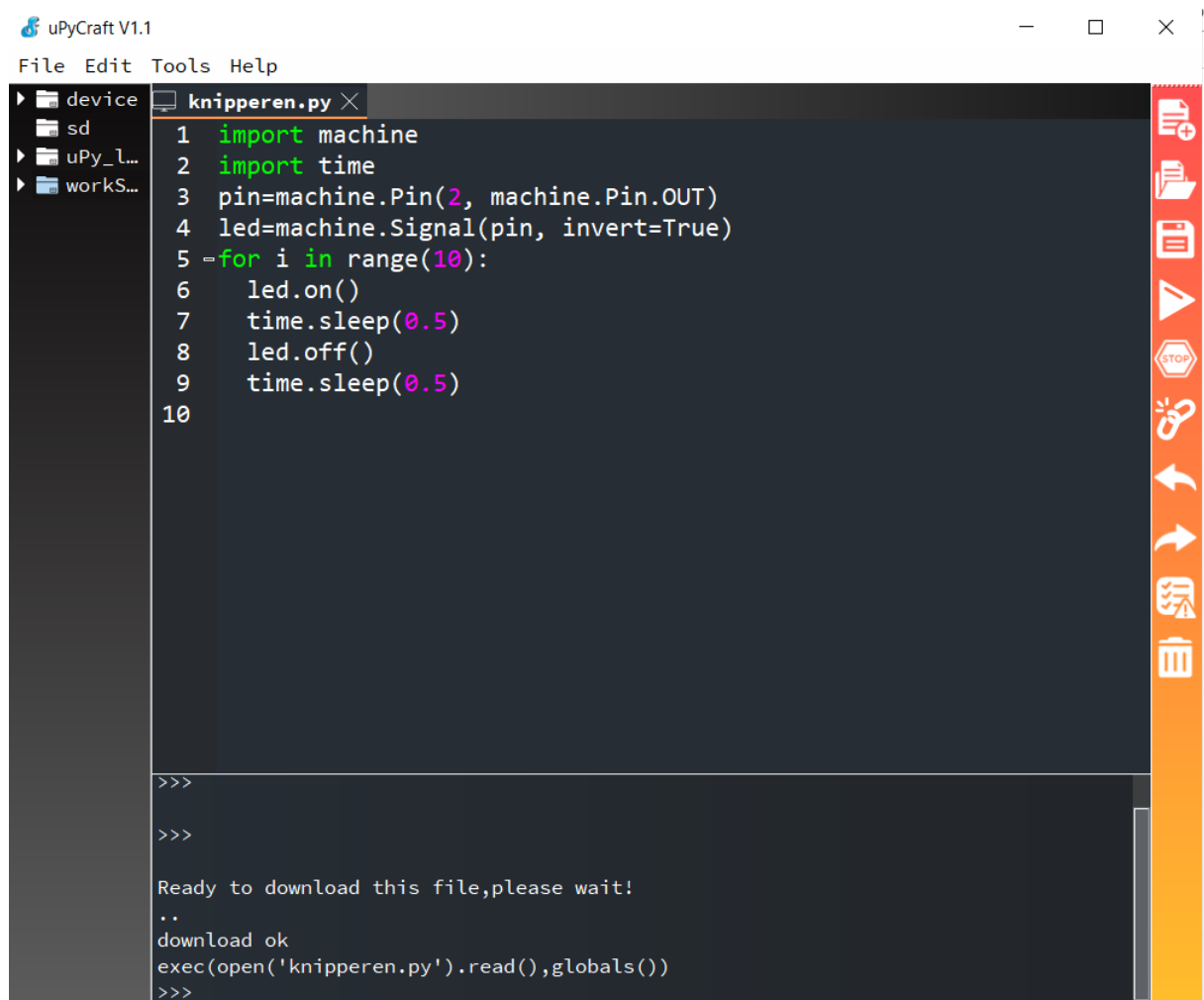
Type daarna de code om de led te knipperen in en sla het bestand vervolgens op.



Druk vervolgens op de  knop en het bestand wordt overgebracht naar de esp8266 waarna het zal worden uitgevoerd.

Als je alles goed hebt gedaan zal de led dus weer een tijdje gaan knipperen.

Onder in je scherm krijg je wat informatie over het uitgevoerde proces te zien.



```
uPyCraft V1.1
File Edit Tools Help
device knipperen.py x
sd
uPy_l...
workS...
1 import machine
2 import time
3 pin=machine.Pin(2, machine.Pin.OUT)
4 led=machine.Signal(pin, invert=True)
5 for i in range(10):
6     led.on()
7     time.sleep(0.5)
8     led.off()
9     time.sleep(0.5)
10

>>>
>>>
Ready to download this file,please wait!
..
download ok
exec(open('knipperen.py').read(),globals())
>>>
```

Je zou nu in staat moet zijn zelf wat programmacode te schrijven en het uit te proberen.

Succes.