

Wifi op de esp8266

Wanneer de esp8266 wordt aangezet start deze automatisch op in access point mode. Dit betekent dat de esp8266 zelf een wifi netwerk heeft opgezet waarmee andere apparatuur zich kan verbinden.



Naast deze access point mode is er ook nog een station mode. In de station mode kan de esp8266 zelf verbonden worden met een wifi netwerk.

Om te laten zien dat de esp8266 standaard in access point mode opstart zou je de volgende instructies kunnen uitvoeren.

```
>>> import network
>>> sta_if=network.WLAN(network.STA_IF)
>>> ap_if=network.WLAN(network.AP_IF)
>>> sta_if.active()
True
>>> ap_if.active()
True
>>>
```

Zoals je ziet bevindt de esp8266 zich in access point mode.

Willen we dat de esp8266 zich verbindt aan een andere wifi netwerk, dan moeten we eerst de station mode activeren. Dit doe je met het commando `sta_if.active(True)`

Daarna kun je de verbinding maken met het commando

```
sta_if.connect('<jouw netwerk-ID>', '<jouw wachtwoord>')
```

Om te controleren of de verbinding is gemaakt kun je gebruik maken van het commando

```
sta_if.isconnected()
```

En daarna kun je de access point mode deactiveren met

```
ap_if.active(False)
```

```
>>> sta_if.active(True)
>>> sta_if.connect('Ziggo4044814', 'XXXXXXXXXX')
>>> sta_if.isconnected()
True
>>> ap_if.active(False)
```

Meestal wil je bij het opstarten van de esp8266 direct verbinding maken met het netwerk. Alle code in een bestand *main.py* wordt na het opstarten van de esp8266 automatisch uitgevoerd. Je zou bovenstaande code dus kunnen opnemen in het bestand *main.py* maar gebruikelijk is het om *main.py* niet helemaal vol te stoppen met allerlei code maar alleen functieaanroepen te gebruiken. De code voor de functie wordt dan in een apart bestand *functions.py* geplaatst.

Maak dus een bestand *main.py* met de volgende code:

```
import functions
```

```
functions.do_connect()
```

En een bestand *functions.py* met deze code:

```
def do_connect():
    import network
    sta_if = network.WLAN(network.STA_IF)
    if not sta_if.isconnected():
        print('verbinding maken...')
        sta_if.active(True)
        sta_if.connect('Ziggo4044814', 'XXXXXXXXXX')
        while not sta_if.isconnected():
            pass
    print('netwerk configuratie:', sta_if.ifconfig())
```

En plaats deze bestanden vervolgens op de esp8266.

```
uPyCraft V1.1
File Edit Tools Help

device
├── boot.py
├── knipperen.py
├── functions.py
├── main.py
├── sd
├── uPy_lib
└── workSpace

functions.py × main.py ×
1 import functions
2
3 functions.do_connect()
4

...
download ok
exec(open('functions.py').read(),globals())
>>>
>>>

>>>

Ready to download this file,please wait!
download ok
exec(open('main.py').read(),globals())
netwerk configuratie: ('192.168.178.16', '255.255.255.0', '192.168.178.1',
'84.116.46.20')
>>>
```

Zoals je ziet bevinden zich op de esp8266 nu vier bestanden. Het bestand *boot.py* is gegenereerd bij het aanmaken van de firmware. Het bestand *knipper.py* komt uit een eerdere handleiding en de bestanden *main.py* en *functions.py* heb je zojuist zelf aangemaakt.

Wil je nog meer code standaard uitvoeren bij het opstarten of resetten van de esp8266 kun je deze code opnemen in een functie binnen *functions.py* en deze aanroepen in *main.py*.

Als voorbeeld gaan we een pagina van het internet ophalen.

Voeg onderstaande code toe aan *functions.py*.

```
def http_get(url):
    import socket
    _, _, host, path = url.split('/', 3)
    addr = socket.getaddrinfo(host, 80)[0][-1]
    s = socket.socket()
    s.connect(addr)
    s.send(bytes('GET /%s HTTP/1.0\r\nHost: %s\r\n\r\n' % (path, host),
'utf8'))
    while True:
        data = s.recv(100)
        if data:
```

```
        print(str(data, 'utf8'), end='')
    else:
        break
s.close()
```

En deze aanroep in *main.py*

```
functions.http_get('http://micropython.org/ks/test.html')
```

Plaats de aangepaste bestanden op de esp8266 en als het goed is verschijnt in het console de webpagina.

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <title>Test</title>
  </head>
  <body>
    <h1>Test</h1>
    It's working if you can read this!
  </body>
</html>

MicroPython v1.11-8-g48dcbbe60 on 2019-05-29; ESP module with ESP8266
Type "help()" for more information.
>>>
```