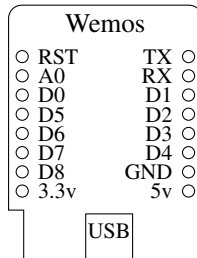


IoT et ESP8266

ESP8266 & MicroPython



Le Wemos est une carte de développement combinant :

- ☐ un composant USB ↔ série ;
- ☐ un adaptateur 5v (depuis l'USB) vers 3,3v (vers l'ESP8266) ;

On peut vérifier qu'il est connecté et que son chip série est reconnu :

```
xterm
pef@cube:~$ lsusb
...
Bus 001 Device 003: ID 10c4:ea60 Silicon Labs CP210x UART Bridge
```

Installation de µpython

La procédure est décrite à la page suivante :

<https://docs.micropython.org/en/latest/esp8266/tutorial/intro.html>

```
xterm
pef@cube:~$ wget
https://micropython.org/resources/firmware/esp8266-20210902-v1.17.bin
```

Pour l'installation de l'outil de « flashage » :

```
xterm
pef@cube:~$ python3 -m pip install esptool
```

Pour le flashage :

```
xterm
pef@cube:~$ esptool.py --port /dev/ttyUSB0 erase_flash
pef@cube:~$ esptool.py --port /dev/ttyUSB0 --baud 460800 write_flash
--flash_size=4MB 0 esp8266-20210902-v1.17.bin
```

Pour se connecter au micro python par le port série :

```
xterm
$ sudo screen /dev/ttyUSB0 115200
MicroPython v1.17 on 2021-09-02; ESP module with ESP8266
Type "help()" for more information.
>>>
>>> import os
>>> os.uname()
(sysname='esp8266', nodename='esp8266', release='2.0.0(5a875ba)', version='v1.17 on
2021-09-02', machine='ESP module with ESP8266')
>>> import port_diag
```

Pour configurer l'ESP8266 en point d'accès :

https://docs.micropython.org/en/latest/esp8266/tutorial/network_basics.html :

```
xterm
>>> import network
>>> ap_if = network.WLAN(network.AP_IF)
>>> ap_if.ifconfig()
('192.168.4.1', '255.255.255.0', '192.168.4.1', '208.67.222.222')
>>> ap_if.active(True)
>>> ap_if.config(essid='toto', channel=13, authmode=network.AUTH_OPEN)
```

Pour configurer l'ESP8266 en client :

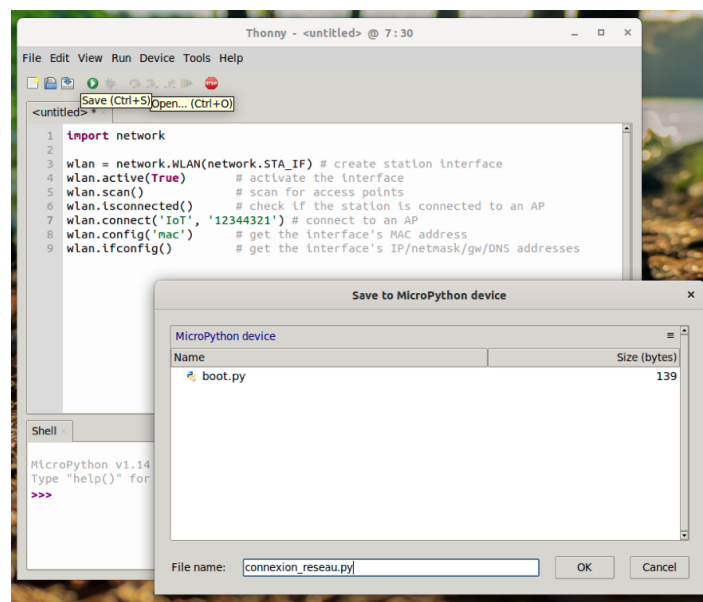
```
xterm
>>> import network
>>> sta_if = network.WLAN(network.STA_IF)
>>> sta_if.ifconfig()
('192.168.4.1', '255.255.255.0', '192.168.4.1', '208.67.222.222')
>>> sta_if.active(True)
>>> sta_if.connect('<your ESSID>', '<your password>')
>>> sta_if.active()
True
>>> sta_if.ifconfig()
('192.168.3.13', '255.255.255.0', '192.168.3.1', '192.168.3.1')
>>>
```

Pour quitter il faut taper sur 'CTRL-a' puis k.

Pour utiliser MicroPython avec des programmes :

```
xterm
$ sudo -H pip install adafruit-ampy
```

Ou avec Thonny:



Essayez les programmes de la page :

https://docs.micropython.org/en/latest/esp8266/tutorial/network_tcp.html

■ ■ ■ Exemple de programmes

Code source du programme « pins.py »:

```
import machine
import time

def clignotement(p):
    while 1:
        p.off()
        time.sleep(1)
        p.on()
        time.sleep(1)

LED = 2
p=machine.Pin(LED,machine.Pin.OUT)
clignotement(p)

xterm
$ export AMPY_PORT=/dev/ttyUSB0
$ ampy run pins.py
```

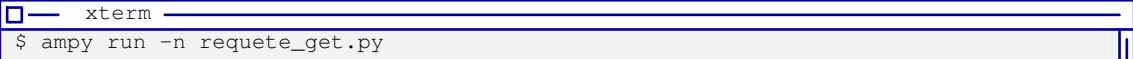
La commande `ampy run -n pins.py` permet de revenir au shell directement.

Après avoir configuré votre ESP8266 en mode client WiFi, essayez le programme suivant :

Code source du programme « requete_get.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()

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



Après avoir configuré votre ESP8266 en point d'accès essayez le programme suivant :

Code source du programme « simple_http.py » :

```
import machine
pins = [machine.Pin(i, machine.Pin.IN) for i in (0, 2, 4, 5, 12, 13, 14, 15)]

html = """<!DOCTYPE html>
<html>
  <head> <title>ESP8266 Pins</title> </head>
  <body> <h1>ESP8266 Pins</h1>
    <table border="1"> <tr><th>Pin</th><th>Value</th></tr> %s </table>
  </body>
</html>
"""

import socket
addr = socket.getaddrinfo('0.0.0.0', 80)[0][-1]

s = socket.socket()
s.bind(addr)
s.listen(1)

print('listening on', addr)

while True:
    cl, addr = s.accept()
    print('client connected from', addr)
    cl_file = cl.makefile('rwb', 0)
    while True:
        line = cl_file.readline()
        if not line or line == b'\r\n':
            break
    rows = ['<tr><td>%s</td><td>%d</td></tr>' % (str(p), p.value()) for p in pins]
    response = html % '\n'.join(rows)
    cl.send('HTTP/1.0 200 OK\r\nContent-type: text/html\r\n\r\n')
    cl.send(response)
    cl.close()
```

