

## Das ESP-Modul als Access Point im WLAN/WiFi

Ein **Wireless Access Point** (englisch für drahtloser Zugangspunkt), auch **Access Point (AP)** oder **Basisstation** genannt, ist ein elektronisches Gerät, das als **Schnittstelle** für kabellose Kommunikationsgeräte fungiert. (Quelle: [Wikipedia](#))

Die **Schnittstelle** (englisch Interface) ist der Teil eines Systems, das der **Kommunikation** dient. (Quelle: [Wikipedia](#))

**WLAN** ist die Abkürzung für **Wireless Local Area Network** und bedeutet kabelloses, lokales Netzwerk.

**WiFi** steht für **Wireless Fidelity** und ist eine der gebräuchlichsten WLAN-Technologien.

Die Konsultation der offiziellen **MicroPython Dokumentation** unter <https://docs.micropython.org/en/latest/#> liefert zum Thema WLAN

### 1.6. WiFi

After a fresh install and boot the device configures itself as a WiFi access point (AP) that you can connect to. The ESSID is of the form MicroPython-xxxxxx where the x's are replaced with part of the MAC address of your device (so will be the same everytime, and most likely different for all ESP8266 chips). The password for the WiFi is micropythoN (note the upper-case N). Its IP address will be 192.168.4.1 once you connect to its network. WiFi configuration will be discussed in more detail later in the tutorial.

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

1. ESSID (Extended Service Set Identifier) ist **MicroPython-xxxxxx**.
2. MAC adress: z.B.: **46:17:93:0d:ab:32**  
Die MAC-Adresse eines ESP-Moduls unterscheidet sich von der MAC-Adresse anderer Module.
3. Passwort ist **MicroPythoN**.
4. IP-Adresse ist **192.168.4.1** .

Ein ESP-Modul mit der IP-Adresse **46:17:93:0d:ab:31** hat die ESSID **MicroPython-0dab31**

### Untersuchungen mit dem ESP-Modul

Mit der WLAN-Suche eines Computer oder Handy suchen wir nach vorhandenen WLAN-Verbindungen.

Ein neues **ESP-Modul (ohne MicroPython)** wird an die Stromversorgung per USB-Kabel angeschlossen.

Erneute WLAN-Suche mit meinem Computers findet kein neues WLAN-Gerät.

Auf dem ESP-Modul wurde nun die MicroPython- Firmware neu installiert.

Dies geschieht mit uPyDraft, da es mit Thonny nicht klappt.

Ein Test der neuen Firmware in der Kommandozeile von Thonny funktioniert.

```
Kommandozeile x
MicroPython v1.9-6-g821dc27e-dirty on 2017-07-21; ESP module with ESP8266
Type "help()" for more information.
>>> 3+4
7
```

Das folgende Programm konfiguriert das auf em ESP-Modul vorhandene WLAN unter dem Namen „ap“ als Access Point und gibt den Aktivitätszustand und die Konfiguration des WLAN aus.

```
[ AccessPoint-00.py ] x
1 import network
2 ap = network.WLAN(network.AP_IF)
3 print( ap.active() )
4 print( ap.ifconfig() )
5

Kommandozeile x
>>> %Run -c $EDITOR_CONTENT
True
('192.168.4.1', '255.255.255.0', '192.168.4.1', '208.67.222.222')
>>>
```

In der WLAN-Suche be finden sich ein neuer Eintrag mit einen Namen „MicroPython-XXXXXX“, wobei „XXXXXX“ eine sechsstellige Folge der Ziffern von 0 bis 9 und der Buchstaben von a bis f sind.

Das untersuchte ESP-Modul wird mit dem Namen **MicroPython-0dab32**-angezeigt.

Nach dem Start des folgenden Programmes ist der Access Point deaktiviert.

```
[ AccessPoint-00.py ] x
1 import network
2 ap = network.WLAN(network.AP_IF)
3 print( ap.active() )
4 print( ap.ifconfig() )
5 ap.active(False)
6 print( ap.ifconfig() )
7

Kommandozeile x
>>> %Run -c $EDITOR_CONTENT
False
('0.0.0.0', '0.0.0.0', '0.0.0.0', '208.67.222.222')
('0.0.0.0', '0.0.0.0', '0.0.0.0', '208.67.222.222')
>>>
```

Mit dem Aufruf des folgenden Programms wird der Access Point je nach Ausgangszustand deaktiviert bzw. aktiviert.

```
Access_Point-00.py x
1 import network
2
3 ap = network.WLAN(network.AP_IF)
4
5 if ap.active()==True:
6     print("AP wird deaktiviert.")
7     ap.active(False)
8     print( ap.ifconfig() )
9 else:
10    print("AP wird aktiviert.")
11    ap.active(True)
12    print( ap.ifconfig() )
13
```

```
Kommandozeile x
>>> %Run -c $EDITOR_CONTENT
AP wird deaktiviert.
bcn 0
del if1
us1
mode : null
('0.0.0.0', '0.0.0.0', '0.0.0.0', '208.67.222.222')
>>> %Run -c $EDITOR_CONTENT
AP wird aktiviert.
mode : softAP(ae:0b:fb:d0:08:c2)
add if1
#20 ets_task(4020edb8, 29, 3fff9510, 10)
dhcp server start:(ip:192.168.4.1,mask:255.255.255.0,gw:192.168.4.1)
bcn 100
('192.168.4.1', '255.255.255.0', '192.168.4.1', '208.67.222.222')
>>>
```

Diese Aktionen lassen sich auch in der Kommandozeile von Thonny ausführen.

```
Kommandozeile x
>>> import network
>>> ap = network.WLAN(network.AP_IF)
>>> ap.active()
False
>>> ap.ifconfig()
('0.0.0.0', '0.0.0.0', '0.0.0.0', '208.67.222.222')
>>> ap.active(True)
mode : softAP(46:17:93:0d:ab:31)
add if1
#8 ets_task(4020edb8, 29, 3fff9510, 10)
dhcp server start:(ip:192.168.4.1,mask:255.255.255.0,gw:192.168.4.1)
bcn 100
>>> ap.ifconfig()
('192.168.4.1', '255.255.255.0', '192.168.4.1', '208.67.222.222')
>>> |
```

Der Name des WLAN, welches durch das hier verwendete ESP-Modul bereitgestellt wurde, lautet „MicroPython-0dab31“. Diesen sechstelligen Code am Ende des Namens finden wir im letzten Bild.

Es ist ein Teil der MAC-Adresse... 46:17:93:0d:ab:32