This is the documentation for the latest development branch of MicroPython and may refer to features that are not available in released versions.

If you are looking for the documentation for a specific release, use the drop-down menu on the left and select the desired version.

class WLAN – control built-in WiFi interfaces

This class provides a driver for WiFi network processors. Example usage:

```
import network
# enable station interface and connect to WiFi access point
nic = network.WLAN(network.STA_IF)
nic.active(True)
nic.connect('your-ssid', 'your-key')
# now use sockets as usual
```

Constructors

class **network.WLAN**(*interface_id*)

Create a WLAN network interface object. Supported interfaces are `network.STA_IF` (station aka client, connects to upstream WiFi access points) and `network.AP_IF` (access point, allows other WiFi clients to connect). Availability of the methods below depends on interface type. For example, only STA interface may `WLAN.connect()` to an access point.

Methods

WLAN.active([*is_active*])

Activate (“up”) or deactivate (“down”) network interface, if boolean argument is passed. Otherwise, query current state if no argument is provided. Most other methods require active interface.

WLAN.connect(*ssid=None, key=None, *, bssid=None*)

Connect to the specified wireless network, using the specified key. If *bssid* is given then the connection will be restricted to the access-point with that MAC address (the *ssid* must

also be specified in this case).

`WLAN.disconnect()`

Disconnect from the currently connected wireless network.

`WLAN.scan()`

Scan for the available wireless networks. Hidden networks – where the SSID is not broadcast – will also be scanned if the WLAN interface allows it.

Scanning is only possible on STA interface. Returns list of tuples with the information about WiFi access points:

(ssid, bssid, channel, RSSI, security, hidden)

bssid is hardware address of an access point, in binary form, returned as bytes object. You can use `binascii.hexlify()` to convert it to ASCII form.

There are five values for security:

- 0 – open
- 1 – WEP
- 2 – WPA-PSK
- 3 – WPA2-PSK
- 4 – WPA/WPA2-PSK

and two for hidden:

- 0 – visible
- 1 – hidden

`WLAN.status( [ param ] )`

Return the current status of the wireless connection.

When called with no argument the return value describes the network link status. The possible statuses are defined as constants:

- `STAT_IDLE` – no connection and no activity,
- `STAT_CONNECTING` – connecting in progress,
- `STAT_WRONG_PASSWORD` – failed due to incorrect password,
- `STAT_NO_AP_FOUND` – failed because no access point replied,
- `STAT_CONNECT_FAIL` – failed due to other problems,
- `STAT_GOT_IP` – connection successful.

When called with one argument *param* should be a string naming the status parameter to retrieve. Supported parameters in WiFI STA mode are: `'rssi'`.

`WLAN.isconnected()`

In case of STA mode, returns `True` if connected to a WiFi access point and has a valid IP address. In AP mode returns `True` when a station is connected. Returns `False` otherwise.

`WLAN.ifconfig( [ (ip, subnet, gateway, dns) ] )`

Get/set IP-level network interface parameters: IP address, subnet mask, gateway and DNS server. When called with no arguments, this method returns a 4-tuple with the above information. To set the above values, pass a 4-tuple with the required information. For example:

```
nic.ifconfig(('192.168.0.4', '255.255.255.0', '192.168.0.1', '8.8.8.8'))
```

`WLAN.config('param')`

`WLAN.config(param=value, ...)`

Get or set general network interface parameters. These methods allow to work with additional parameters beyond standard IP configuration (as dealt with by `WLAN.ifconfig()`). These include network-specific and hardware-specific parameters. For setting parameters, keyword argument syntax should be used, multiple parameters can be set at once. For querying, parameters name should be quoted as a string, and only one parameter can be queries at time:

```
# Set WiFi access point name (formally known as SSID) and WiFi channel
ap.config(ssid='My AP', channel=11)
# Query params one by one
print(ap.config('ssid'))
print(ap.config('channel'))
```

Following are commonly supported parameters (availability of a specific parameter depends on network technology type, driver, and MicroPython port).

Parameter	Description
mac	MAC address (bytes)
ssid	WiFi access point name (string)
channel	WiFi channel (integer)
hidden	Whether SSID is hidden (boolean)
security	Security protocol supported (enumeration, see module constants)
key	Access key (string)
hostname	The hostname that will be sent to DHCP (STA interfaces) and mDNS (if supported, both STA and AP)

Parameter	Description
reconnects	Number of reconnect attempts to make (integer, 0=none, -1=unlimited)
txpower	Maximum transmit power in dBm (integer or float)