1. Station et point d'accès

1.1. Station

En mode station, le routeur agit comme un point d'accès et la carte ESP32 est définie comme une station. Ainsi, il faut être connecté au routeur (réseau local) pour contrôler l'ESP32.

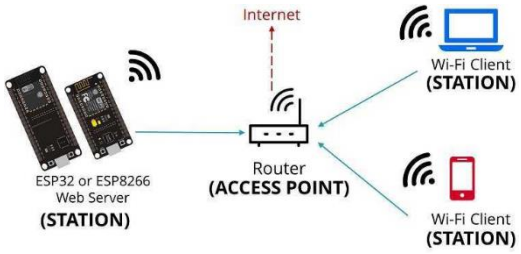


Fig 1 : ESP32 en mode station

1.2. Point d'accès

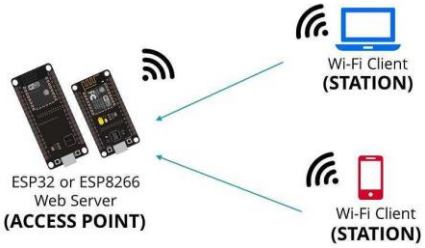


Fig 2 : ESP32 en point d'accès

En définissant la carte ESP32 comme point d'accès (hotspot), un réseau Wifi propre à l'ESP32 est créé et n'importe quel appareil peut s'y connecter.

2. Le protocole HTTP

La création du serveur va nécessiter :

- La création et configuration du point d'accès ;
- La création d'un canal de communication entre le client et le serveur (socket) ;
- La création de la page qui sera envoyée lors de la réponse à la requête ;
- La communication entre le client et le serveur défini par la figure 3.

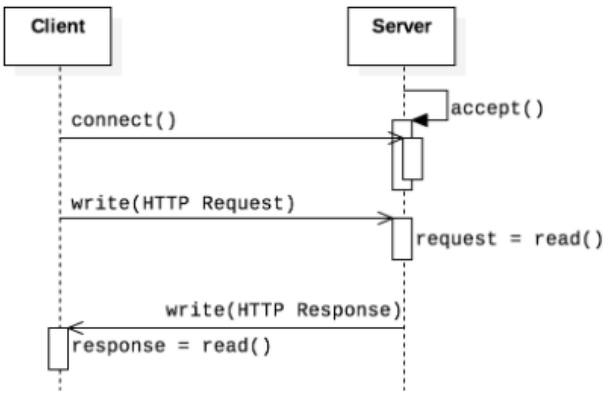
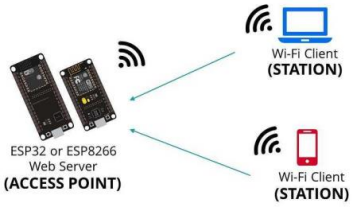


Fig 3 : communication entre client et serveur

3. Création un point d'accès (hotspot)

Le code ci-dessous permet de créer un point d'accès.




```
# configuration du point d'accès
import network

ssid = 'CHEREL'
password = '0000'

ap = network.WLAN(network.AP_IF)                # création du point d'accès
ap.active(True)                                 # activation du point d'accès
ap.config(essid=ssid, password=password)        # configuration avec le ssid et le password
while not ap.active():                          # tant que c'est pas activé, on attend
    pass
print(f'configuration réseau{ap.ifconfig()}')   # IP du point d'accès, masque, passerelle, DNS
```

✍ Créer votre point d'accès en personnalisant le **ssid** avec votre nom.

✍ Vérifier la configuration obtenue.

```
MPY: soft reboot
configuration réseau ('192.168.4.1', '255.255.255.0', '192.168.4.1', '0.0.0.0')
```

✍ Tester le point d'accès avec votre smartphone (se connecter au wifi de l'ESP32).

4. Création d'un socket

La création d'un socket permet d'ouvrir un canal de communication entre le client (smartphone) et le serveur (ESP32).

```
# configuration du socket
import socket

# AF_INET : IPV4 ; SOCK_STREAM : TCP
serveursocket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
serveursocket.setsockopt(socket.SOL_SOCKET, socket.SO_REUSEADDR, 1)
serveursocket.bind(('', 80))                # port 80 pour HTTP
serveursocket.listen(5)                     # ecoute de 5 requêtes
```

5. Création d'un page WEB dans le langage Python

Lors de la connexion, une page Web doit être envoyée.

```
# page WEB
def web_page():
    html = """<!DOCTYPE HTML>
    <html>
        <head>
            <meta name="viewport" content="width=device-width, initial-scale=1">
        </head>
        <body>
            <h1>Hello, World!</h1>
        </body>
    </html>"""
    return html
```

6. Communication client-serveur

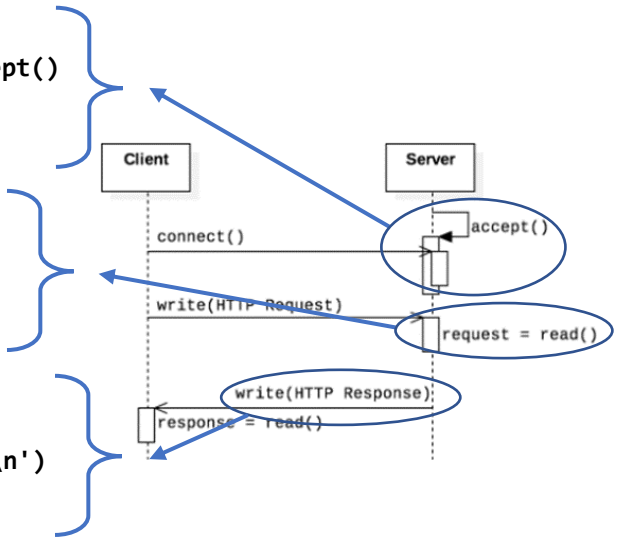
Lors de la connexion avec un client (smartphone), le serveur (ESP32) reçoit la requête et lui envoi la réponse, ici la page Web.

```
# boucle
while True:
    # serveursocket accept
    connexion_client, addresse = serveursocket.accept()
    connexion_client.settimeout(3.0)
    print(f'Connecté avec le client {addresse}')
    print("")

    # réception de la requête
    requete = connexion_client.recv(1024)
    connexion_client.settimeout(None)
    print(f'Requête du client = {requete}')
    print("")

    # envoi de la réponse
    connexion_client.send('HTTP/1.1 200 OK\n')
    connexion_client.send('Content-Type: text/html\n')
    connexion_client.send("Connection: close\n\n")
    reponse = web_page()
    connexion_client.sendall(reponse)

    # serveursocket fermeture
    connexion_client.close()
```



✍ Se connecter au point d'accès de l'ESP32 à l'aide du navigateur de votre smartphone (dans l'URL de votre navigateur, rentrer l'adresse IP du serveur (IP du point d'accès)).

Les informations ci-dessous doivent apparaitre dans le shell de l'IDE Python.

```
MPY: soft reboot
configuration réseau ('192.168.4.1', '255.255.255.0', '192.168.4.1', '0.0.0.0')

Connecté avec le client ('192.168.4.2', 41560)

Requête du client = b'GET / HTTP/1.1\r\nHost: 192.168.4.1\r\nConnection: keep-alive\r\nCache-Control: max-age=0\r\nUpgrade-Insecure-Requests: 1\
r\nUser-Agent: Mozilla/5.0 (Linux; Android 10; K) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/131.0.0.0 Mobile Safari/537.36\r\nAccept: text/h
tml,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7\r\nAccept-Enco
ding: gzip, deflate\r\nAccept-Language: fr-FR,fr;q=0.9,en-US;q=0.8,en;q=0.7\r\n\r\n'

Connecté avec le client ('192.168.4.2', 41568)

Requête du client = b'GET /favicon.ico HTTP/1.1\r\nHost: 192.168.4.1\r\nConnection: keep-alive\r\nUser-Agent: Mozilla/5.0 (Linux; Android 10; K)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/131.0.0.0 Mobile Safari/537.36\r\nAccept: image/avif,image/webp,image/apng,image/svg+xml,image/*,*
/*;q=0.8\r\nReferer: http://192.168.4.1/\r\nAccept-Encoding: gzip, deflate\r\nAccept-Language: fr-FR,fr;q=0.9,en-US;q=0.8,en;q=0.7\r\n\r\n'
```

✍ Vérifier le bon fonctionnement du serveur et de communication client-serveur en observant la page Web sur votre smartphone.

7. Le programme complet

```
import network
import socket

ssid = 'CHEREL'
password = '0000'

# configuration du point d'accès
ap = network.WLAN(network.AP_IF)           # création du point d'accès
ap.active(True)                            # activation du point d'accès
ap.config(essid=ssid, password=password)   # configuration avec le ssid et le password
while not ap.active():                     # tant que c'est pas activé, on attend
    pass
print(f'configuration réseau {ap.ifconfig()}')   # IP du point d'accès, masque, passerelle,
DNS
print("")

# configuration du socket
# AF_INET : IPV4    SOCK_STREAM : TCP
serveursocket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
serveursocket.setsockopt(socket.SOL_SOCKET, socket.SO_REUSEADDR, 1)
serveursocket.bind(('', 80)) # port 80 pour HTTP
serveursocket.listen(5) # ecoute de 5 requêtes

# page WEB
def web_page():
    html = """<!DOCTYPE HTML>
    <html>
        <head>
            <meta name="viewport" content="width=device-width, initial-scale=1">
        </head>
        <body>
            <h1>Hello, World!</h1>
        </body>
    </html>"""
    return html

# boucle
while True:
    # serveursocket accept
    connexion_client, addresse = serveursocket.accept()
    connexion_client.settimeout(3.0)
    print(f'Connecté avec le client {addresse}')
    print("")

    # réception de la requête
    requete = connexion_client.recv(1024)
    connexion_client.settimeout(None)
    print(f'Requête du client = {requete}')
    print("")

    # envoi de la réponse
    connexion_client.send('HTTP/1.1 200 OK\n')
    connexion_client.send('Content-Type: text/html\n')
    connexion_client.send("Connection: close\n\n")
    reponse = web_page()
    connexion_client.sendall(reponse)

    # serveursocket fermeture
    connexion_client.close()
```