

Le serveur Web peut piloter un élément.

## 1. Piloter un élément

L'exemple suivant va permettre de piloter la LED (GPIO2) de l'ESP32 depuis son serveur Web.

Le début du programme doit permettre la déclaration de la LED (GPIO2).

La transmission de la requête va se faire avec un formulaire (<form>).

- A chaque requête du client (appui sur un bouton du formulaire ici : bouton ON ou bouton OFF), les deux informations **nom** et **valeur** sont renvoyés.
- Ces paramètres sont ajoutés à l'URL avec le signe « ? » et ils indiquent au serveur quelles ressources sont demandées. Le **nom** et sa **valeur** sont toujours séparés avec le symbole « = ». La syntaxe retenue est la suivante : ?nom=valeur.

```
from machine import Pin
```

```
led = Pin(2, Pin.OUT) #GPIO2 en sortie
```

```
# page WEB
```

```
def web_page():
```

```
    html = """<!DOCTYPE HTML>
```

```
    <html>
```

```
        <head>
```

```
            <meta name="viewport" content="width=device-width, initial-scale=1">
```

```
            <style>
```

```
            body{
```

```
                background-color: silver;
```

```
                text-align: center;
```

```
            }
```

```
            h1{
```

```
                color: green;
```

```
                background-color: black;
```

```
            }
```

```
            p{
```

```
                font-size: 20px;
```

```
            }
```

```
            button{
```

```
                margin: 20px;
```

```
                font-size: 30px;
```

```
            }
```

```
        </style>
```

```
    </head>
```

```
    <body>
```

```
        <h1>SERVEUR ESP32</h1>
```

```
        <p><br/><br/></p>
```

```
        <p>Bienvenue sur le serveur de l'ESP32</p>
```

```
        <p><br/><br/></p>
```

```
        <p>Commande de la led de l'ESP32</p>
```

```
        <form>
```

```
            <button name="LED" type="submit" value="1"> ON </button>
```

```
            <button name="LED" type="submit" value="0"> OFF </button>
```

```
        </form>
```

```
    </body>
```

```
</html>"""
```

```
    return html
```

Formulaire

nom

valeur

✍ Mettre en œuvre un serveur Web sur l'ESP32 en point d'accès afin de commander la LED de l'ESP32 avec le code de la page précédente.

✍ Vérifier que les paramètres nom et valeur (ici **LED=1** ou **LED=0**) s'affichent dans l'URL à chaque appui sur un bouton.

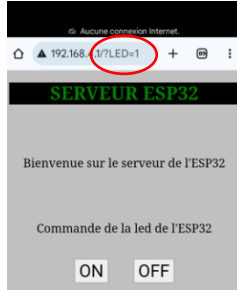


Fig 1 : affichage de LED=1

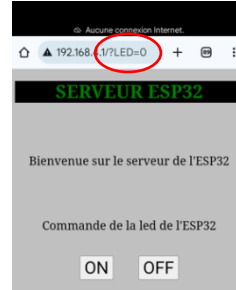


Fig 2 : affichage de LED=0

En décodant la requête, le serveur saura s'il faut allumer ou non la LED grâce à la recherche de **LED=0** ou **LED=1**.

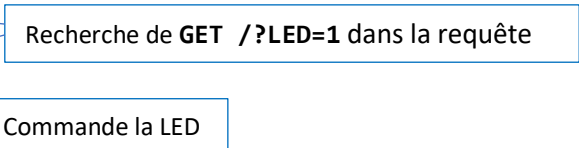
```
Requête du client = b'GET /?LED=1 HTTP/1.1\r\nHost: 192.168.4.1\r\nConnection: keep-alive\r\nUpgrade-Insecure-Requests: 1\r\nUser-Agent: Mozilla/5.0 (Linux; Android 10; K) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/131.0.0.0 Mobile Safari/537.36\r\nAccept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7\r\nReferer: http://192.168.4.1/?LED=0\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'
```

```
Requête du client = b'GET /?LED=0 HTTP/1.1\r\nHost: 192.168.4.1\r\nConnection: keep-alive\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/html,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\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'
```

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

# analyse de la requête, recherche de LED=1 et LED=0
if "GET /?LED=1" in requete:
    print("LED ON")
    led.value(1)
if "GET /?LED=0" in requete:
    print("LED OFF")
    led.value(0)

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



✍ Modifier le programme afin de commander la LED à partir du serveur à partir du code ci-dessus.

✍ Modifier le programme afin que le serveur affiche aussi l'état de la LED comme ci-contre.

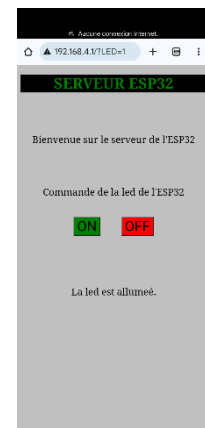


Fig 3 : affichage de l'état de la LED