

# Informe de la Pràctica 3: WiFi i Bluetooth

## 1. Introducció

L'objectiu principal d'aquesta pràctica és comprendre el funcionament de les comunicacions sense fils mitjançant els protocols WiFi i Bluetooth en un microcontrolador ESP32. Aquesta pràctica es divideix en dues parts:

- **Generació d'un servidor web:** Configuració d'una ESP32 per actuar com un servidor web i mostrar una pàgina HTML a qualsevol dispositiu connectat a la xarxa.
- **Comunicació sèrie mitjançant Bluetooth:** Configuració d'una comunicació sèrie bidireccional entre l'ESP32 i un dispositiu mòbil mitjançant Bluetooth.

Aquestes funcionalitats permeten ampliar les capacitats del microcontrolador i integrar-lo en sistemes de comunicació IoT.

## 2. Fonaments Teòrics

### 2.1 WiFi

WiFi ("Wireless Fidelity") és una tecnologia de xarxa inalàmbrica basada en els protocols IEEE 802.11. Per al desenvolupament d'aquesta pràctica, és important comprendre els conceptes bàsics dels protocols:

- **TCP/IP i UDP:** Protocols de comunicació bàsics per a la transmissió de dades en xarxes.
- **API REST:** Un patró d'arquitectura per a la comunicació entre dispositius a través de peticions HTTP.
- **MQTT:** Protocol de missatgeria lleuger utilitzat en aplicacions IoT per a la comunicació entre dispositius.

## 2.2 Bluetooth

Bluetooth és un protocol de comunicació sense fils d'abast curt que permet la interconnexió entre dispositius. L'ESP32 suporta:

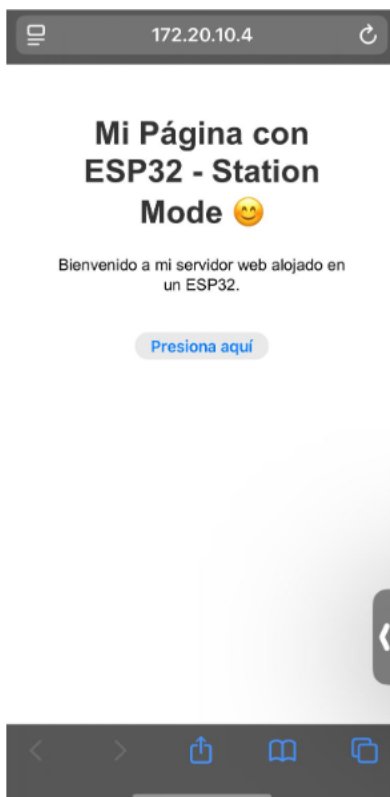
- **Bluetooth Classic:** Utilitzat per a la transmissió de dades en streaming.
- **Bluetooth Low Energy (BLE):** Optimitzat per al consum energètic reduït.

## 3. Desenvolupament de la Pràctica

### 3.1 Generació d'un Servidor Web

Per implementar un servidor web amb ESP32, s'ha configurat el microcontrolador per connectar-se a una xarxa WiFi i gestionar peticions HTTP. En aquesta part de la pràctica, s'ha dissenyat una interfície web accessible des de qualsevol dispositiu connectat a la xarxa.

#### Resultats



Sortida per el terminal:

**WiFi connected successfully**

**Got IP: 172.20.10.4**

**HTTP server started**

Per tant, al buscar la IP que ens proporciona el monitor sèrie, podem accedir al servidor web que hem creat

- L'ESP32 es connecta correctament a la xarxa i mostra la seva adreça IP al monitor sèrie.
- En accedir a l'IP des del navegador, es visualitza la pàgina web generada.

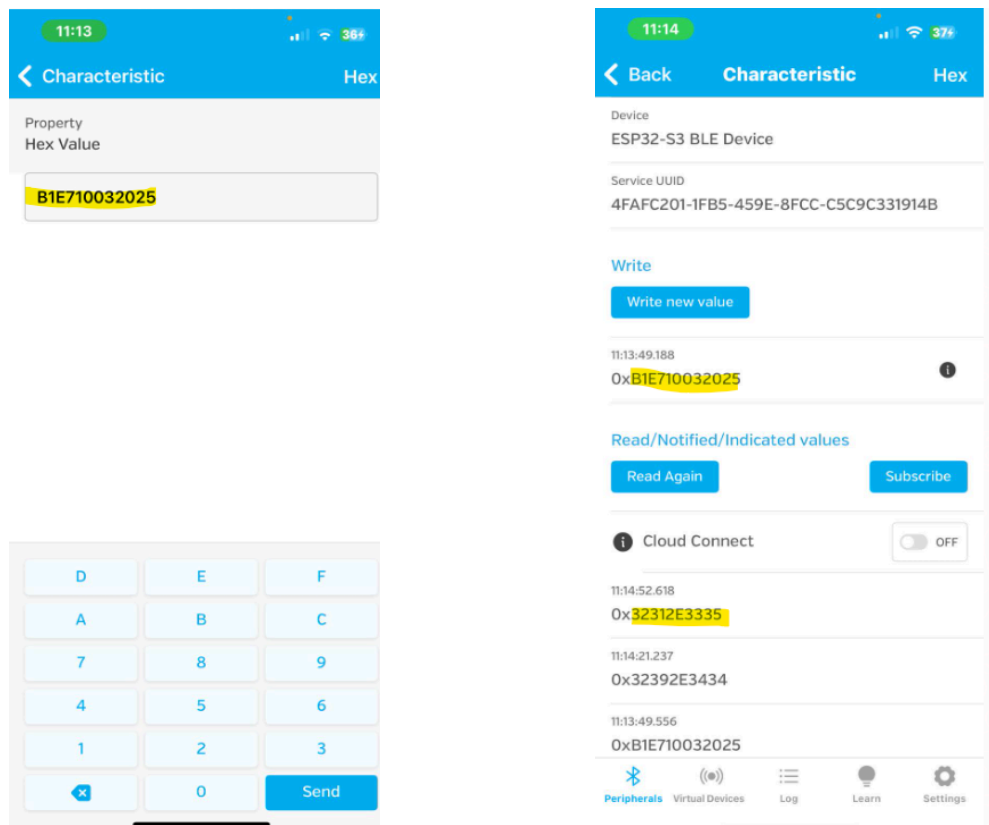
### 3.2 Comunicació Bluetooth

S'ha configurat l'ESP32 per establir una comunicació sèrie bidireccional mitjançant Bluetooth. Això permet enviar i rebre dades entre el microcontrolador i un dispositiu mòbil.

Per fer-ho, es crea un servidor BLE amb un identificador UUID específic. Quan un dispositiu mòbil s'hi connecta, pot llegir i escriure informació mitjançant les característiques definides en el servidor BLE. El dispositiu mòbil envia dades a l'ESP32, que respon amb valors simulats de temperatura.

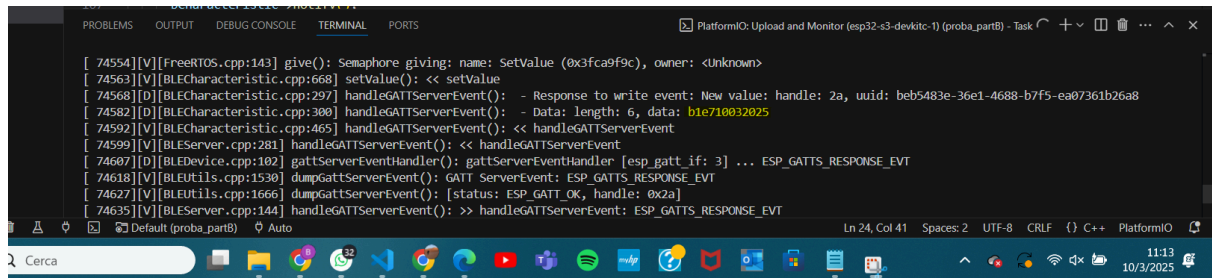
### 3.3 Resultats Obtinguts

Al dispositiu mòbil, es localitza la direcció associada a la temperatura, i es pot escriure el missatge que es vol enviar.



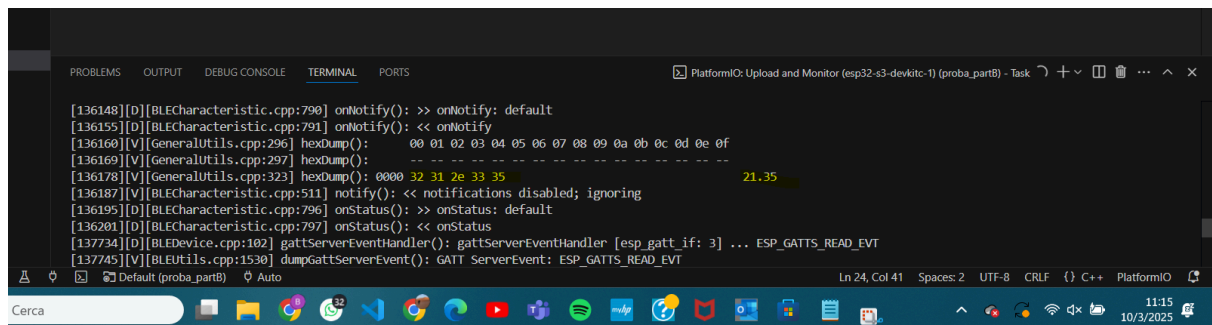
També es veuen els missatges rebuts per la sortida del terminal:

**Text:**



```
[ 74554][V][FreeRTOS.cpp:143] give(): Semaphore giving: name: SetValue (0x3fca9f9c), owner: <Unknown>
[ 74563][V][BLECharacteristic.cpp:668] setValue(): << setValue
[ 74568][D][BLECharacteristic.cpp:297] handleGATTServerEvent(): - Response to write event: New value: handle: 2a, uuid: beb5483e-36e1-4688-b7f5-ea07361b26a8
[ 74582][D][BLECharacteristic.cpp:300] handleGATTServerEvent(): - Data: length: 6, data: b1e710032025
[ 74592][V][BLECharacteristic.cpp:465] handleGATTServerEvent(): << handleGATTServerEvent
[ 74599][V][BLEServer.cpp:281] handleGATTServerEvent(): << handleGATTServerEvent
[ 74607][D][BLEDevice.cpp:102] gattServerEventHandler(): gattServerEventHandler [esp_gatt_if: 3] ... ESP_GATTS_RESPONSE_EVT
[ 74618][V][BLEUtils.cpp:1530] dumpGATTServerEvent(): GATT ServerEvent: ESP_GATTS_RESPONSE_EVT
[ 74627][V][BLEUtils.cpp:1666] dumpGATTServerEvent(): [status: ESP_GATT_OK, handle: 0x2a]
[ 74635][V][BLEServer.cpp:144] handleGATTServerEvent(): >> handleGATTServerEvent: ESP_GATTS_RESPONSE_EVT
```

**Temperatura(21.35):**



```
[136148][D][BLECharacteristic.cpp:790] onNotify(): >> onNotify: default
[136155][D][BLECharacteristic.cpp:791] onNotify(): << onNotify
[136160][V][GeneralUtils.cpp:296] hexDump(): 00 01 02 03 04 05 06 07 08 09 0a 0b 0c 0d 0e 0f
[136169][V][GeneralUtils.cpp:297] hexDump(): -- -- -- -- --
[136178][V][GeneralUtils.cpp:323] hexDump(): 0000 32 31 2e 33 35 21.35
[136187][V][BLECharacteristic.cpp:511] notify(): << notifications disabled; ignoring
[136195][D][BLECharacteristic.cpp:796] onStatus(): >> onStatus: default
[136201][D][BLECharacteristic.cpp:797] onStatus(): << onStatus
[137734][D][BLEDevice.cpp:102] gattServerEventHandler(): gattServerEventHandler [esp_gatt_if: 3] ... ESP_GATTS_READ_EVT
[137745][V][BLEUtils.cpp:1530] dumpGATTServerEvent(): GATT ServerEvent: ESP_GATTS_READ_EVT
```

## 4. Anàlisi de Resultats

Les dues aplicacions desenvolupades en aquesta pràctica permeten comprovar la capacitat de l'ESP32 per gestionar comunicacions sense fils:

- **Servidor Web:** Permet interactuar amb el microcontrolador a distància mitjançant una interfície web.
- **Bluetooth:** Facilita la transmissió de dades en temps real entre l'ESP32 i altres dispositius.

Aquesta funcionalitat és essencial en aplicacions IoT, on la connectivitat sense fils és un requisit fonamental.

## 5. Exercici adicional

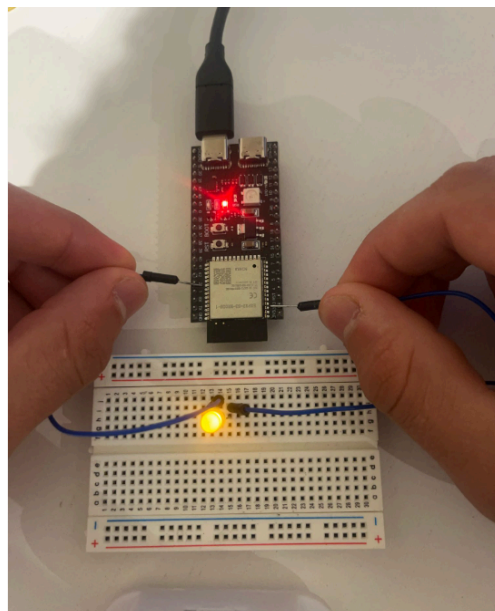
L'ESP32 s'ha configurat com un servidor BLE que permet la connexió amb un dispositiu extern per enviar dades simulades de temperatura i humitat, així com rebre ordres per controlar un LED mitjançant autenticació.

Un cop iniciat el servei, el sistema ha generat valors aleatoris de temperatura i humitat, enviant-los periòdicament. Per activar el LED, ha estat necessari enviar la clau "1234" per autenticar-se. Després de l'autenticació, els comandaments "ON" i "OFF" han funcionat correctament, engegant i apagant el LED.

Els resultats han mostrat que la comunicació BLE entre el dispositiu i l'ESP32 s'ha establert amb èxit, permetent la transmissió de dades i el control remot del LED.

### Resultats obtinguts

Les imatges mostren els resultats obtinguts que he explicat anteriorment:



```
load:0x403cc700,len:0x2a0c
entry 0x403c98d0
🚀 Iniciando aplicación BLE...
✅ Servicio BLE iniciado. Esperando conexión..
🌡️ Temperatura simulada: 23.17
💧 Humedad simulada: 48.29
🌡️ Temperatura simulada: 27.95
💧 Humedad simulada: 47.71
🌡️ Temperatura simulada: 20.48
💧 Humedad simulada: 50.78
📁 Default (proba_partB) 🔌 Auto
```

```
✉️ ----> onWrite llamado <----
📁 Datos recibidos: 1234
🔑 Autenticación exitosa
🌡️ Temperatura simulada: 22.50
💧 Humedad simulada: 54.45
🌡️ Temperatura simulada: 26.25
```

```
💧 Humedad simulada: 48.32
✉️ ----> onWrite llamado <----
📁 Datos recibidos: ON
💡 LED encendido
🌡️ Temperatura simulada: 23.68
💧 Humedad simulada: 48.40
```

```
💧 Humedad simulada: 59.43
✉️ ----> onWrite llamado <----
📁 Datos recibidos: OFF
💡 LED apagado
🌡️ Temperatura simulada: 22.00
💧 Humedad simulada: 43.48
```

## **6. Conclusions**

Aquesta pràctica ha permès comprendre el funcionament bàsic de les comunicacions sense fils mitjançant WiFi i Bluetooth en un ESP32. S'ha demostrat la seva utilitat per a aplicacions IoT, permetent la interacció remota amb el dispositiu mitjançant un navegador web i la transmissió de dades a través de Bluetooth. Aquestes tecnologies obren la porta a aplicacions en domòtica, monitoratge de sensors i control remot d'equips.