

Práctica 3: WIFI y BLUETOOTH

Para llevar a cabo el proyecto se insertan dos librerías para conectar el Wifi y crear un servidor con la placa. Se usan funciones de esas librerías para establecer la conexión Wifi, ver el estado de esta conexión y ver el IP de la placa.

Una vez haya ejecutado el código y subido a la placa, se podrá verificar:

- 1) Establecimiento de conexión
- 2) IP, servidor inicializado.

Establecimiento de conexión

Se realiza gracias a unas funciones de la librería <Wifi.h>, estas son:

```
const char* ssid = "MIWIFI_zt7R"; // Enter your SSID here
const char* password = "HKxSFXJG"; //Enter your Password

WebServer server(80); // Object of WebServer(HTTP port, 80 is default)

// Handle root url (/)
void handle_root(void);

void setup(){
  WiFi.begin(ssid, password);

  // Check wi-fi is connected to wi-fi network
  while (WiFi.status() != WL_CONNECTED) {
    delay(1000);
    Serial.print(".");
  }

  Serial.println("");
  Serial.println("WiFi connected successfully");
  Serial.print("Got IP: ");
  Serial.println(WiFi.localIP()); //Show ESP32 IP on serial
}
```

Primeramente se intenta conectar al Wifi mediante las dos variables especificadas anteriormente, ssid y password. Se establece el puerto al que se conectará la ESP32 del router.

Se utiliza el `WiFi.status()` para informar al usuario de forma gráfica, en este caso con '.' si se ha podido conectar o está habiendo algún problema. Una vez establecida informa al usuario e imprime la IP de la ESP32 de este. Esta IP es necesaria porque se trata de un servidor local.

Finalmente, la placa está conectada a un Wifi y se ha creado un servidor para poder acceder mediante la IP. Para acceder a este servidor hay que estar conectado al mismo Wifi.

Creación del servidor

Se realiza mediante las funciones de la librería <webServer.h>

```
server.on("/", handle_root);
server.begin();
Serial.println("HTTP server started");
delay(100);

void loop() {
  server.handleClient();
}

// HTML & CSS contents which display on web server
String HTML = "<!DOCTYPE html>\n
<html>\n
<body>\n
<h1> Johny's Web </h1>\n
<h2> Esto es un código de HTML enviado con la placa ESP32. </h2>\n
</body>\n
</html>";
// Handle root url (/)
void handle_root() {
  server.send(200, "text/html", HTML);
}
```

El webserver sirve para recibir, enviar, procesar y guardar información. Además se puede enviar la información a una web. Para comunicarse se usa el protocolo *HTTP*. Cuando se envía algo al servidor, devuelve el siguiente código:

- 200 Si la conexión se ha establecido correctamente.
- 404 Si no se ha podido establecer.

En el caso de que sea 200, se inicializa el servidor. Dentro la función `loop()` se llama a `handle_Client` el cual he supuesto que establece la conexión cliente.

Fuera de esta podemos crear nuestro código html para la página web y con una función enviarla al servidor. El código en html creado lo ponemos dentro de una variable de tipo `String` que le llamamos `HTML`.