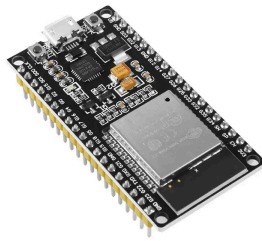


PRACTICA 3 : WIFI

MATERIAL

Para esta practica necesitaremos únicamente el microcontrolador ESP32



OBJETIVO Y FUNCIONALIDAD DE LA PRACTICA

El objetivo de la practica es comprender el funcionamiento de WIFI.

Para lo cual realizaremos una practica donde generaremos un web server desde nuestra ESP32.

Practica A generación de una pagina web

(No se requiere montaje)

Este código configura un ESP32 como un servidor web que muestra una página HTML simple cuando se accede a su dirección IP desde un navegador.

- El ESP32 intenta conectarse a la red Wi-Fi usando el SSID y la contraseña proporcionados. Crea un servidor web en el puerto asignado en el código, define una página HTML y configura el servidor para que responda con esta página cuando un usuario acceda a la dirección raíz.
- Cuando la conexión es exitosa, muestra en la consola la dirección IP asignada por el router.
- Para acceder a la página web es necesario estar conectado a ese wifi, y ingresar la IP en tu buscador de internet.

Código comentado

```

#include <WiFi.h> // Librería para manejar la conexión Wi-Fi en el ESP32
#include <WebServer.h> // Librería para crear un servidor web en el ESP32

// Credenciales de la red Wi-Fi a la que se conectará el ESP32
const char* ssid = "Nautilus"; // Nombre de la red Wi-Fi (SSID)
const char* password = "20000Leguas"; // Contraseña de la red Wi-Fi

// Creación de un objeto servidor web en el puerto 80 (HTTP por defecto)
WebServer server(80);

// Contenido HTML que se mostrará en la página web
String HTML = "<!DOCTYPE html>\n // Definición del documento HTML
<html>\n
<body>\n
<h1>My Primera Pagina con ESP32 - Station Mode &#128522; :)</h1>\n
</body>\n
</html>";

// Función para manejar la ruta raíz ("/") del servidor web
void handle_root() {
    // Envía el código HTML con un código de estado 200 (OK)
    server.send(200, "text/html", HTML);
}

void setup() {
    Serial.begin(115200); // Inicia la comunicación serie a 115200 baudios
    Serial.println("Try Connecting to ");
    Serial.println(ssid); // Muestra el nombre de la red

    // Inicia la conexión Wi-Fi con las credenciales especificadas
    WiFi.begin(ssid, password);

    // Espera hasta que el ESP32 se conecte a la red Wi-Fi
    while (WiFi.status() != WL_CONNECTED) {
        delay(1000);
        Serial.print("."); // Muestra puntos mientras intenta conectarse
    }

    // Mensajes de confirmación de conexión exitosa

```

```

Serial.println("");
Serial.println("WiFi connected successfully");
Serial.print("Got IP: ");
Serial.println(WiFi.localIP()); // Muestra la dirección IP asignada al ESP32.

// Asocia la función `handle_root` para manejar solicitudes a la URL raíz ("/")
server.on("/", handle_root);

// Inicia el servidor web
server.begin();
Serial.println("HTTP server started");
}

void loop() {
  // Atiende las peticiones de los clientes que acceden al servidor web
  server.handleClient();
}

```

Salida en el monitor serial

- Cuando arranca el sistema aparece un mensaje de inicialización.

Try Connecting to Nautilus (Nombre_WIFI)

- Mientras el micro ESP32 intenta conectarse, se imprimen puntos cada segundo

.....

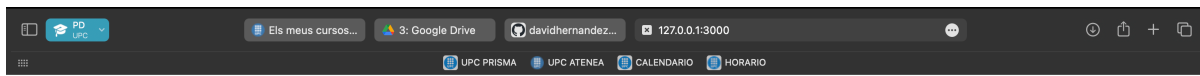
- Una vez conectado, aparece un mensaje de confirmación y su IP correspondiente. Esto significa que ya puedes entrar en la página web.

```

WiFi connected successfully
Got IP: 192.168.X.Y
HTTP server started

```

- Para la visualización de la conexión a la página web tienes que ingresar la IP que aparece en el monitor en tu navegador, en mi caso aparece la siguiente página web (*visualizar mejor en el pdf*).



My Primera Pagina con ESP32 - Station Mode 😊 :)

Generar fichero HTML

Para el siguiente apartado únicamente tendrás que modificar tu función `handle_root` para que, en vez de leer la variable HTML, lea tu archivo `.html`. De la siguiente manera:

```
// Función para leer y servir el archivo HTML
void handle_root() {
  // En esta linea introduces el nombre de tu archivo .html, en mi caso index
  File file = SPIFFS.open("/index.html", "r");
  if (!file) {
    server.send(500, "text/plain", "Error al cargar el archivo HTML");
    return;
  }
  String htmlContent = file.readString();
  file.close();
  server.send(200, "text/html", htmlContent);
}
```

(El archivo html hay que introducirlo dentro de la carpeta data de PlataformIO)

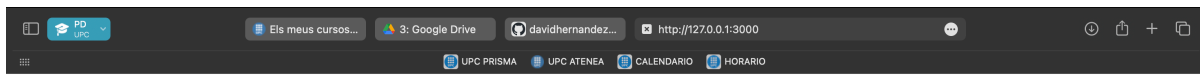
Después hay que realizar los mismos pasos que en el anterior apartado, subir el código a tu

placa. En el monitor serie aparecerán mensajes de confirmación si todo ha funcionado correctamente.

Ejemplo código html:

```
<!DOCTYPE html>
<html>
<head>
  <title>Página ESP32</title>
  <meta charset="UTF-8">
  <style>
    body { font-family: Arial, sans-serif; text-align: center; }
    h1 { color: #007bff; }
    p { font-size: 18px; }
  </style>
</head>
<body>
  <h1>Bienvenido a mi Servidor ESP32 🌐</h1>
  <p>Esta es una página creada directamente desde mi ESP32 usando PlatformIO.</p>
</body>
</html>
```

Aparecerá una página web como esta (*visualizar mejor en el pdf*):



Bienvenido a mi Servidor ESP32 🌐

Esta es una página creada directamente desde mi ESP32 usando PlatformIO.

Ejercicios opcionales

1. Realizar el mismo ejercicio Wifi pero en lugar de realizar la conexión STA hacer una conexión AP

En este caso el ESP32 crea su propia red WiFi, a la que otros dispositivos pueden conectarse. Para conseguir esto hay que utilizar la función:

```
WiFi.softAP(ssid_ap, password_ap);
```

Esto crea una red WiFi con el SSID, por ejemplo, "ESP32-AccessPoint" y la contraseña, por ejemplo, "12345678".

Código comentado:

```

#include <WiFi.h>
#include <WebServer.h>

// Definir el SSID y contraseña de la red WiFi AP

// Nombre de la red WiFi creada por el ESP32
const char* ssid_ap = "ESP32-AccessPoint";
// Contraseña de la red WiFi
const char* password_ap = "12345678";

WebServer server(80); // Crear el servidor web en el puerto 80

// Contenido HTML de la página web
String HTML = "<!DOCTYPE html>\
<html>\
<head>\
    <title>ESP32 AP Mode</title>\
    <meta charset='UTF-8'>\
    <style> body { font-family: Arial, sans-serif; text-align: center; } </style>\
</head>\
<body>\
    <h1>ESP32 en Modo AP 🚀</h1>\
    <p>Conéctate a esta red WiFi sin necesidad de un router.</p>\
</body>\
</html>";

// Función para manejar la solicitud a la página principal "/"
void handle_root() {
    server.send(200, "text/html", HTML);
}

void setup() {
    Serial.begin(115200);
    Serial.println("\nConfigurando Access Point...");

    // Configurar el ESP32 en modo AP
    WiFi.softAP(ssid_ap, password_ap);

    // Obtener la IP asignada en modo AP (normalmente 192.168.4.1)

```



```
IPAddress IP = WiFi.softAPIP();
Serial.print("Dirección IP del ESP32 AP: ");
Serial.println(IP);

// Configurar el servidor web
server.on("/", handle_root);
server.begin();
Serial.println("Servidor Web en modo AP iniciado");
}

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

Por último hay que realizar los mismos pasos que en los apartados anteriores, subir el código a tu placa ESP32 y en el monitor serie aparecerán mensajes de confirmación si todo ha funcionado bien.

Esto hará que todas las personas que se conecten a esta conexión WIFI podrán acceder a la página web.