

# TECNICATURA SUPERIOR EN Telecomunicaciones

## Electrónica Microcontrolada



### Optimización 2: Comunicación y Preprocesamiento



Dirección General de  
EDUCACIÓN TÉCNICA Y  
FORMACIÓN PROFESIONAL

Ministerio de  
EDUCACIÓN



**Profesor: Gonzalo Vera**

**Alumno: Raúl Jara**

#### Conexión básica ESP32 a una red WiFi y Monitoreo de Estado

##### Explicación Paso a Paso:

###### 1. Incluir la librería WiFi.h:

```
#include <WiFi.h>
```

- Se incluye la librería WiFi.h para manejar la conectividad Wi-Fi en el ESP32. Esta librería proporciona todas las funciones necesarias para conectar, monitorear y reconectar a una red Wi-Fi.

###### 2. Definir las credenciales de la red Wi-Fi (SSID y Contraseña):

```
const char* ssid = "tu_SSID";
```

```
const char* password = "tu_contraseña";
```

- Aquí defines las variables ssid y password que contienen el nombre de la red Wi-Fi y la contraseña que usará el ESP32 para conectarse.

3. **Función setup():** La función setup() es donde se configura el ESP32 al inicio. Se ejecuta una sola vez cuando el dispositivo se enciende o se reinicia.

#### 3.1 Inicializar el Monitor Serie:

```
Serial.begin(115200);
```

```
delay(1000);
```

- Se inicia la comunicación serie a 115200 baudios para poder imprimir información en el **Monitor Serie**.
- delay(1000) agrega una pequeña pausa de 1 segundo para asegurarse de que el Monitor Serie esté listo.

#### 3.2 Conectar a la red Wi-Fi:

```
WiFi.begin(ssid, password);
```

- El ESP32 intentará conectarse a la red Wi-Fi utilizando las credenciales ssid y password definidas anteriormente. Esta función inicia el proceso de conexión en modo estación (STA).

#### 3.3 Esperar a que la conexión Wi-Fi se establezca:

```
while (WiFi.status() != WL_CONNECTED) {
```

```
    delay(1000);
```

```
    Serial.println("Conectando...");
```

```
}
```

- Este bucle while se ejecuta repetidamente hasta que el ESP32 se conecte a la red Wi-Fi. Si WiFi.status() no devuelve WL\_CONNECTED, significa que aún no está conectado, por lo que se espera 1 segundo (delay(1000)) antes de intentarlo de nuevo.

#### 3.4 Imprimir la dirección IP cuando se establece la conexión:

```
Serial.println("Conexión establecida!");
```

```
Serial.print("Dirección IP: ");
```

```
Serial.println(WiFi.localIP());
```

- Una vez conectado, el ESP32 imprime un mensaje de confirmación en el Monitor Serie y muestra la **dirección IP** que se le ha asignado en la red.

4. **Función loop():** La función loop() se ejecuta repetidamente mientras el ESP32 esté encendido.

#### 4.1 Monitorizar el estado de la conexión Wi-Fi:

```
if (WiFi.status() != WL_CONNECTED) {
```

```
Serial.println("Conexión Wi-Fi perdida, intentando reconectar...");
```

```
...
```

```
}
```

- En cada ciclo del `loop()`, se verifica si el estado actual de la conexión Wi-Fi es `WL_CONNECTED`. Si no lo es, significa que la conexión se ha perdido.

#### 4.2 Intentar reconectar si se ha perdido la conexión:

```
while (WiFi.status() != WL_CONNECTED) {
```

```
    WiFi.reconnect();
```

```
    delay(5000);
```

```
    Serial.println("Intentando reconectar...");
```

```
}
```

- Si se pierde la conexión, el ESP32 intenta reconectarse automáticamente usando `WiFi.reconnect()`. El bucle `while` continúa intentándolo hasta que la conexión se restablezca. Entre cada intento, se espera 5 segundos (`delay(5000)`) para evitar múltiples reconexiones rápidas.

#### 4.3 Mensaje de reconexión exitosa:

```
Serial.println("Reconexión exitosa!");
```

- Una vez que el ESP32 se reconecta con éxito, se imprime este mensaje en el Monitor Serie.

#### 5. Pausa entre los ciclos de monitoreo:

```
delay(10000);
```

- Se agrega una pausa de 10 segundos en el bucle `loop()` para evitar que el ESP32 imprima demasiados mensajes en el Monitor Serie de forma continua, haciendo el monitoreo más manejable.

#### Resumen:

- **setup():** Inicializa el ESP32, intenta conectarse a la red Wi-Fi y, cuando se conecta, imprime la dirección IP.
- **loop():** Monitorea el estado de la conexión Wi-Fi. Si se pierde, intenta reconectar automáticamente. Si lo consigue, imprime un mensaje de éxito.

Este código es la base para una conexión Wi-Fi robusta con el ESP32, manejando la reconexión automática en caso de pérdida de red.