

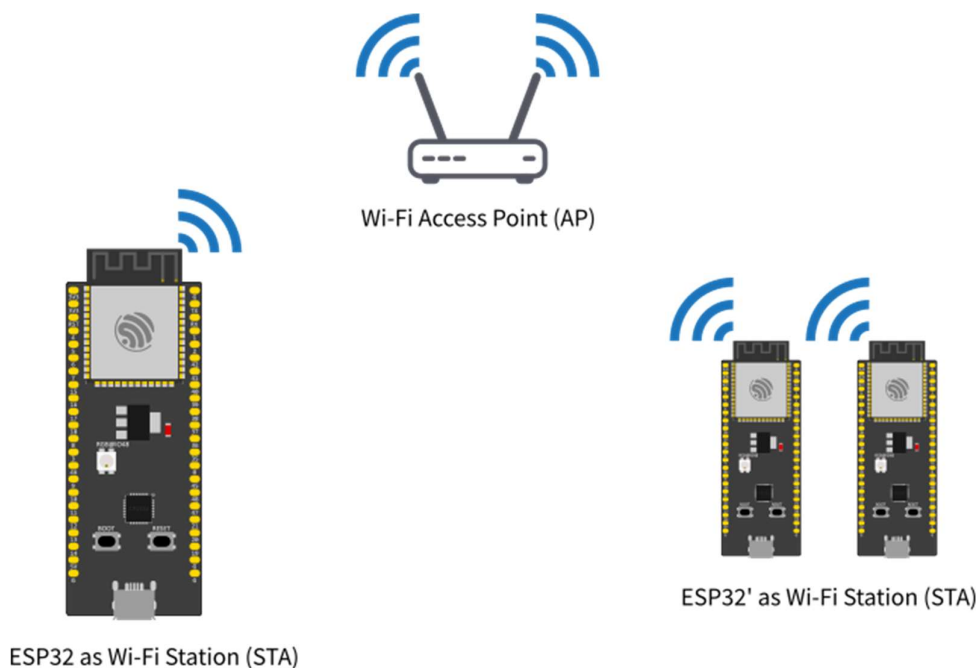
Práctica 09 - Conexión del Microcontrolador ESP32 a una Red Wifi

El microcontrolador ESP32 tiene integrado un transceptor Wifi HT40 802.11b/g/n, lo que nos permite conectarlo a una red Wifi e interactuar con la Internet. También podemos usar al microcontrolador ESP32 para crear su propia red Wifi permitiendo que otros dispositivos se conecten con él.

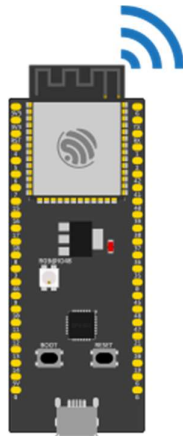
Modos de Operación del Microcontrolador ESP32 en una Red Wifi

La API de Wifi del microcontrolador ESP32 nos permite configurarlo para que opere de los siguientes modos:

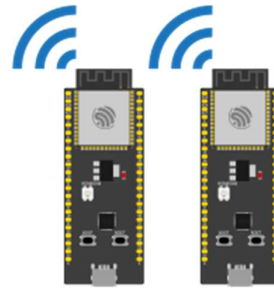
- Modo Estación o modo cliente: En este caso, el microcontrolador ESP32 se conecta a una red Wifi provista por un punto de acceso.



- Modo Punto de Acceso: En este caso, el microcontrolador ESP32 crea una red Wifi y es capaz de recibir conexiones de otros dispositivos (estaciones).



ESP32 as Wi-Fi Access Point (AP)



ESP32' as Wi-Fi Station (STA)

En estas prácticas solo se usará al microcontrolador ESP32 en el modo estación.

Para tener acceso a la API Wifi del microcontrolador ESP32 debemos incluir en nuestros programas la biblioteca WiFi. Al compilar el programa, se creará una instancia de la clase WiFi que estará disponible para nuestro código como una variable global.

Algunas de las funciones de la clase WiFi

```
int8_t scanNetworks(bool async = false, bool show_hidden = false);
```

Inicia la búsqueda de las redes Wifi disponibles

Regresa el número de redes Wifi encontradas

```
String SSID(uint8_t networkItem);
```

Regresa el SSID de la red dada por el parámetro `networkItem`, obtenida por el método `scanNetworks()`.

```
wifi_auth_mode_t encryptionType(uint8_t networkItem);
```

Regresa el tipo de encriptado de la red dada por el parámetro `networkItem`, obtenida por el método `scanNetworks()`.

```
int32_t RSSI(uint8_t networkItem);
```

Regresa el valor de la RSSI (indicador de fuerza de la señal recibida) de la red dada por el parámetro `networkItem`, obtenida por el método `scanNetworks()`.

```
wl_status_t begin(char* ssid, char *passphrase = NULL);
```

Inicia la conexión del microcontrolador ESP32 a la red Wifi con el SSID y contraseña dados por los parámetros.

Regresa el estado de la conexión. Uno de los siguientes valores definidos para el tipo enumerado `wl_status_t`:

```
typedef enum {
    WL_NO_SHIELD          = 255,    // for compatibility with WiFi
                                // Shield library
    WL_IDLE_STATUS        = 0,
    WL_NO_SSID_AVAIL      = 1,
    WL_SCAN_COMPLETED     = 2,
    WL_CONNECTED          = 3,
    WL_CONNECT_FAILED     = 4,
    WL_CONNECTION_LOST    = 5,
    WL_DISCONNECTED       = 6
} wl_status_t;
```

<code>IPAddress localIP();</code>
Obtiene la dirección IP asignada al microcontrolador ESP32
<code>String WiFiSTAClass::macAddress(void)</code>
Obtiene la dirección MAC del microcontrolador ESP32
<code>const char * getHostname();</code>
Obtiene el nombre asignado al microcontrolador ESP32
<code>wl_status_t status();</code>
Regresa el estado de la conexión
<code>String SSID() const;</code>
Regresa el SSID de la red a la que está conectado el microcontrolador ESP32.
<code>int8_t RSSI();</code>
Regresa el valor de la RSSI (indicador de fuerza de la señal recibida) de la red a la que está conectado el microcontrolador ESP32.

Objetivo

Determinar las redes Wifi disponibles y conectar un Microcontrolador ESP32 a una red Wifi.

Equipo y Materiales

1 Microcontrolador ESP32 DEVKIT DOIT de 30 pines
 1 Base para el Microcontrolador ESP32 DEVKIT DOIT de 30 pines
 1 Cable USB a micro USB

Procedimiento

Programa que lista las redes Wifi detectadas por un Microcontrolador ESP32

Este programa lista para cada una de las redes detectadas por un microcontrolador Wifi, su SSID, la intensidad de la señal Wifi de la red y si está protegida mediante un mecanismo de encriptamiento.

1. Usando el IDE 2.0 de Arduino, edite el siguiente programa. Guárdelo con el nombre de **blinkSerial** y compílelo.

scanRedes.ino

```

/*
 * scanRedes.ino
 *
 * Este programa permite determinar que redes Wifi están
 * disponibles y su nivel de intensidad.
 */

#include <WiFi.h>

// Velocidad de transmisión del puerto serie
const unsigned int BAUD_RATE = 115200;

void setup() {
  // Inicializa la comunicación serial
  Serial.begin(BAUD_RATE);
  delay(100);
}

```

```

void loop() {
  Serial.println("Inicia la busqueda de las redes Wifi");
  // Se busca las redes
  int n = WiFi.scanNetworks();
  Serial.println("Finalizo la busqueda");

  // Si no se encontraron redes Wifi
  if (n == 0) {
    Serial.println("No se encontraron redes");
  }
  else {
    // Si se encontraron redes Wifi
    Serial.print("Se encontraron ");
    Serial.print(n);
    Serial.println(" redes:");

    // Para cada red encontrada
    for (int i = 0; i < n; ++i) {
      // Despliega su SSID
      Serial.print(i + 1);
      Serial.print(": SSID: ");
      Serial.print(WiFi.SSID(i));
      // Despliega su RSSI ()
      Serial.print(" RSSI: ");
      Serial.print(WiFi.RSSI(i));
      Serial.print(" Protegida: ");
      Serial.println((WiFi.encryptionType(i) == WIFI_AUTH_OPEN)
                     ? "No": "Si");

      delay(10);
    }
  }
  Serial.println("");

  // Pausa antes de la siguiente busqueda
  delay(5000);
}

```

2. Conecte el Microcontrolador ESP32 a la computadora mediante el cable USB a micro USB.
3. Cargue el programa al Microcontrolador ESP32.
4. Abra el monitor serie.
5. Deberá ver en el monitor serie, cada 5 segundos, la lista de redes Wifi detectadas por el microcontrolador ESP32: su SSID, la intensidad de la señal Wifi de la red y si está protegida mediante un mecanismo de encriptamiento.
6. Verifique que el programa funciona correctamente.
7. Cierre el monitor serie.
8. Desconecte el Microcontrolador ESP32 de la computadora.

Programa que conecta un Microcontrolador ESP32 con una red Wifi

El siguiente programa conecta un microcontrolador ESP32 a una red Wifi. Se requiere la SSID y contraseña de la red a conectarse. El programa desplegará mediante el monitor serial la dirección IP que le asigna la red.

1. Usando el IDE 2.0 de Arduino, edite el siguiente programa. Guárdelo con el nombre de **conexionWifi_ESP32** y compílelo.

conexionWifi_ESP32.ino

```
/*
 * conexionWifi_ESP32.ino
 *
 * Este programa permite conectar un microcontrolador ESP32 a una
 * red Wifi.
 *
 * Despliega la dirección IP asignada al microcontrolador ESP32 al
 * conectarse a la red Wifi
 */

#include <WiFi.h>

// Substituya "SSID" por el nombre del SSID de la red Wifi a
// conectarse
const char* ssid = "SSID";

// Substituya "contraseña" por la contraseña de la red Wifi a
// conectarse
const char* password = "contraseña";

// Velocidad de transmisión del puerto serie
const unsigned int BAUD_RATE = 115200;

void conectaRedWifi(const char* ssid, const char* password);

void setup() {
    // Inicializa la comunicación serial
    Serial.begin(BAUD_RATE);
    delay(100);

    Serial.println();
    // Conecta a una red Wifi
    conectaRedWifi(ssid, password);
}

void loop() {
}

/*
```

```

* Conecta el microcontrolador ESP32 a una red WiFi
*/
void conectaRedWiFi(const char* ssid, const char* password) {
    // Conexion a la red
    WiFi.begin(ssid, password);
    Serial.print("Conectandose a la red ");
    Serial.print(ssid);
    Serial.println(" ...");

    // Mientras no se ha conectado a la red WiFi
    while (WiFi.status() != WL_CONNECTED) {
        delay(1000);
        Serial.print(".");
    }

    Serial.println('\n');
    Serial.println("Connexion establecida");

    // Obten la direccion IP del microcontrolador ESP32
    Serial.print("Direccion IP del servidor web: ");
    Serial.println(WiFi.localIP());
}

```

2. Conecte el Microcontrolador ESP32 a la computadora mediante el cable USB a micro USB.
3. Cargue el programa al Microcontrolador ESP32.
4. Abra el monitor serie.
5. Después de unos segundos, deberá aparecer la dirección IP asignada por la red al microcontrolador ESP32.
6. Verifique que el programa funciona correctamente.
7. Cierre el monitor serie.
8. Desconecte el Microcontrolador ESP32 de la computadora.

Programa que muestra la Dirección IP y nombre del Microcontrolador ESP32 y estadísticas de la red Wifi

El siguiente programa conecta un microcontrolador ESP32 a una red Wifi. Se requiere la SSID y contraseña de la red a conectarse. El programa desplegará mediante el monitor serial la dirección IP que le asigna la red, la dirección MAC y nombre del microcontrolador ESP32 y algunas estadísticas de la red Wifi.

1. Usando el IDE 2.0 de Arduino, edite el siguiente programa. Guárdelo con el nombre de **wifiStats_ESP32** y compílelo.

wifiStats_ESP32.ino

```
/*
 * wifiStats_ESP32.ino
 *
 * Este programa permite conectar un microcontrolador ESP32 a una
 * red Wifi.
 *
 * Despliega la dirección MAC del microcontrolador ESP32 y la
 * dirección IP asignada al microcontrolador ESP32 al conectarse a
 * la red Wifi
 *
 * También despliega algunas estadísticas de la red WiFi
 */

#include <WiFi.h>

// Nombre del SSID de la red Wi-Fi a conectarse
const char* ssid      = "INFINITUM0B90_2.4";

// Contraseña para la red Wi-Fi
const char* password = "McuYG6q332";

// Velocidad de transmisión del puerto serie
const unsigned int BAUD_RATE = 115200;

void conectaRedWifi(const char* ssid, const char* password);
void redWiFiStats(void);

void setup() {
    // Inicializa la comunicación serial
    Serial.begin(BAUD_RATE);
    delay(100);

    Serial.println('\n');
    // Conecta a una red Wifi
    conectaRedWifi(ssid, password);
    redWiFiStats();
}

void loop() {
}
```

```
/*
 * Conecta el microcontrolador ESP32 a una red WiFi
 */
void conectaRedWiFi(const char* ssid, const char* password) {
    // Conexion a la red
    WiFi.begin(ssid, password);
    Serial.print("Conectandose a la red ");
    Serial.print(ssid);
    Serial.println(" ...");

    // Mientras no se ha conectado a la red WiFi
    while (WiFi.status() != WL_CONNECTED) {
        delay(1000);
        Serial.print(".");
    }

    Serial.println('\n');
    Serial.println("Connexion establecida");

    // Obten la direccion IP del microcontrolador ESP32
    Serial.print("Direccion IP del servidor web: ");
    Serial.println(WiFi.localIP());
}

/**
 * Obtiene la dirección MAC y nombre del microcontrolador ESP32
 * y algunas estadísticas de la red Wifi
 */
void redWiFiStats(void) {
    Serial.println();
    Serial.print("Direccion MAC del microcontrolador ESP32: ");
    Serial.println(WiFi.macAddress());
    Serial.println();
    Serial.print("Nombre asignado al microcontrolador ESP32: ");
    Serial.println(WiFi.getHostname());
    Serial.println();
    Serial.print("SSID de la red Wifi: ");
    Serial.println(WiFi.SSID());
    Serial.println();
    Serial.print("RSSI de la red Wifi: ");
    Serial.println(WiFi.RSSI());
}
```


2. Conecte el Microcontrolador ESP32 a la computadora mediante el cable USB a micro USB.
3. Cargue el programa al Microcontrolador ESP32.
4. Abra el monitor serie.
5. Después de unos segundos, deberá aparecer la dirección IP asignada por la red al microcontrolador ESP32, la dirección MAC y nombre del microcontrolador ESP32 y algunas estadísticas de la red Wifi.
6. Verifique que el programa funciona correctamente.
7. Cierre el monitor serie.
8. Desconecte el Microcontrolador ESP32 de la computadora.