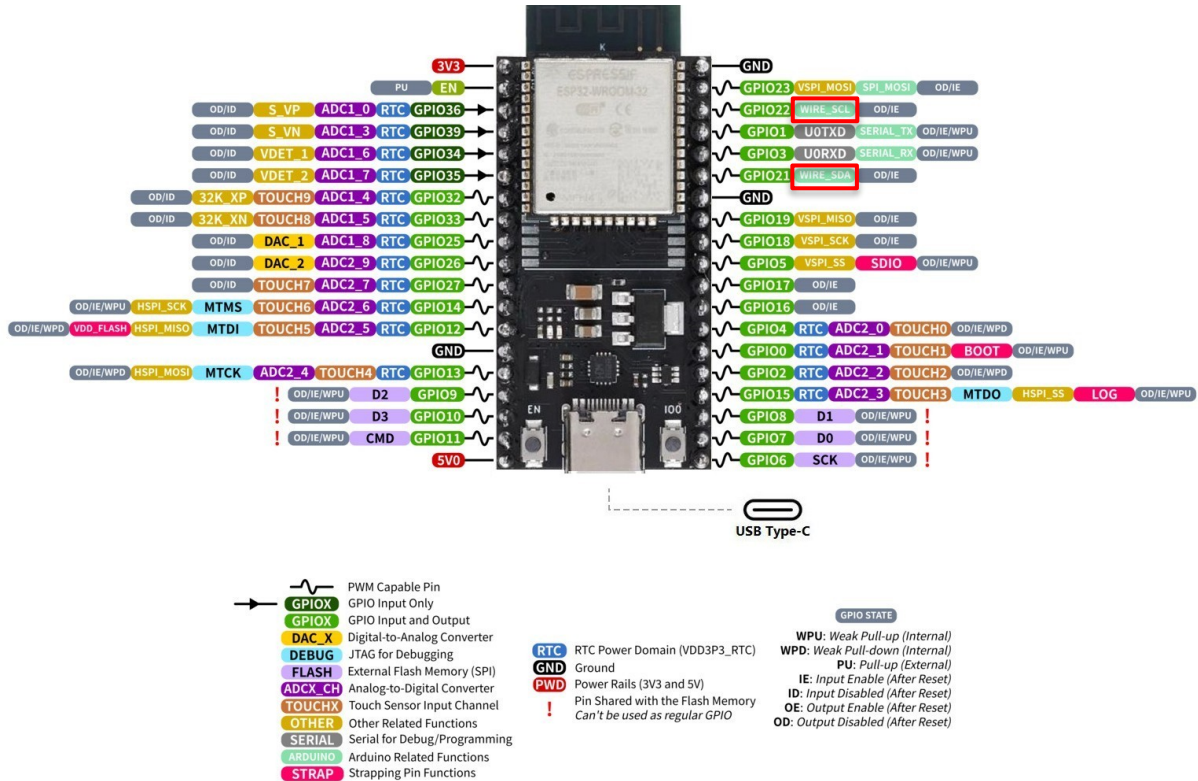


## 單元 7-1：ESP32 WIFI 設定

### 學習目標：

- ESP32 的連線模式
- 顯示 WiFi 名稱、IP 位置、連線強度

### ESP32 接腳圖：



### 材料需求：

ESP32 開發板	X1
電阻 220Ω	X2
LED	X2

### Wifi 模式

用 WiFi.h 程式庫會有 4 種 wifi 模式可以用:

語法	WIFI 模式	功能
WiFi.mode(WIFI_AP);	Access Point (AP)	ESP32可以讓其他設備透過wifi接入(就像家裡的wifi基地台, 可供手機連線)。
WiFi.mode(WIFI_STA);	Station(STA)	無線終端模式, 也就是讓ESP32可以連接上其他的熱點(就像手機一樣, 可以連上家裡wifi)。
WiFi.mode(WIFI_AP_STA);	AP+STA	將ESP32設置成兩個模式並存。
WiFi.mode(WIFI_OFF);	OFF	關閉wifi

## 程式內容(一)無線終端模式

```
#include<WiFi.h>

const char ssid[]="LTSHEE"; //修改 WiFi 網路名稱
const char pwd[]="123456789"; //修改 WiFi 密碼

void setup() {
  Serial.begin(115200);    //序列埠設定鮑率為 115200

  WiFi.mode(WIFI_STA); //設置 WiFi 模式
  WiFi.begin(ssid,pwd);

  Serial.print("WiFi connecting");

  //當 WiFi 連線時會回傳 WL_CONNECTED，因此跳出迴圈時代表已成功連線
  while(WiFi.status()!=WL_CONNECTED){
    Serial.print(".");
    delay(500);
  }
  Serial.println('\n');
  Serial.print("Connected to ");          // NodeMCU 將通過串口監視器輸出。
  Serial.println(WiFi.SSID()); //顯示 WIFI 名稱
  Serial.println("");
  Serial.print("IP 位址:");
  Serial.println(WiFi.localIP()); //讀取 IP 位址
  Serial.print("WiFi RSSI:");
  Serial.println(WiFi.RSSI()); //讀取 WiFi 強度
}

void loop() {
}
```

RSSI 一般以 **dBm** (分貝毫瓦) 為單位，數值為**負數**，例如：-30 dBm、-70 dBm，數字越接近 0 表示訊號越強。

RSSI 值 (dBm)	訊號品質說明	使用狀況建議
-30 ~ -50	非常強	最佳連線品質
-51 ~ -60	強	很穩定，無線網路品質佳
-61 ~ -70	普通	可接受，可能略有延遲
-71 ~ -80	弱	容易掉線，建議靠近基地台
-81 以下	非常弱或無訊號	幾乎無法正常連線

透過 Wifi 連接控制 LED 燈開關

```
#include <WiFi.h>

const char* ssid = "你的WiFi名稱";
const char* password = "你的WiFi密碼";

WiFiServer server(80);
const int ledPin = 2; // 內建 LED (可改成其他 GPIO)

void setup() {
  Serial.begin(115200);
  pinMode(ledPin, OUTPUT);
  digitalWrite(ledPin, LOW);

  // 連接 WiFi
  Serial.print("Connecting to ");
  Serial.println(ssid);
  WiFi.begin(ssid, password);

  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }

  // 顯示 IP 位址
  Serial.println("");
  Serial.println("WiFi connected.");
  Serial.print("IP address: ");
  Serial.println(WiFi.localIP());

  server.begin();
}

void loop() {
  WiFiClient client = server.available();
  if (!client) return;

  Serial.println("New Client.");
  String request = client.readStringUntil('\r');
  Serial.println(request);
  client.flush();

  // 控制 LED
  if (request.indexOf("/LED=ON") != -1) {
    digitalWrite(ledPin, HIGH);
  } else if (request.indexOf("/LED=OFF") != -1) {
    digitalWrite(ledPin, LOW);
  }

  // HTML 回應
  client.println("HTTP/1.1 200 OK");
  client.println("Content-type:text/html");
  client.println();
  client.println("<html><body>");
  client.println("<h1>ESP32 控制 LED</h1>");
  client.println("<a href=\\\"/LED=ON\\\">開啟 LED</a><br>");
  client.println("<a href=\\\"/LED=OFF\\\">關閉 LED</a><br>");
}
```

```

    client.println("</body></html>");
    client.println();
    delay(1);
    Serial.println("Client disconnected.");
}

```

程式內容(三)

---

```

#include <WiFi.h>

const char* ssid = "LTSHEE";
const char* password = "123456789";

WiFiServer server(80);
const int ledPin = 2; // 內建 LED (可改成其他 GPIO)

void setup() {
    Serial.begin(115200);
    pinMode(ledPin, OUTPUT);
    digitalWrite(ledPin, LOW);

    // 連接 WiFi
    Serial.print("Connecting to ");
    Serial.println(ssid);
    WiFi.begin(ssid, password);

    while (WiFi.status() != WL_CONNECTED) {
        delay(500);
        Serial.print(".");
    }

    // 顯示 IP 位址
    Serial.println("");
    Serial.println("WiFi connected.");
    Serial.print("IP address: ");
    Serial.println(WiFi.localIP());

    server.begin(); // 伺服器啟動
}

void loop() {
    WiFiClient client = server.available(); // 伺服器連線成功

    if (client) {
        Serial.println("有用戶端連線"); // 若有連線序列埠顯示

        String request = "";
        while (client.connected()) {
            if (client.available()) {
                char c = client.read();
                request += c;

                if (c == '\n') {
                    // 處理 HTTP 請求
                    if (request.indexOf("GET /LED=ON") >= 0) {
                        digitalWrite(ledPin, HIGH);
                    }
                }
            }
        }
    }
}

```

```

    } else if (request.indexOf("GET /LED=OFF") >= 0) {
        digitalWrite(ledPin, LOW);
    }

    // 傳送 HTML 回應
    client.println("HTTP/1.1 200 OK");
    client.println("Content-type:text/html");
    client.println();

    client.println("<!DOCTYPE html>");
    client.println("<html lang='zh-Hant'>");
    client.println("<head><meta charset='UTF-8'><title>LED 控制
</title></head>");
    client.println("<body style='text-align:center;'>");
    client.println("<h1>ESP32 LED 控制介面</h1>");
    client.println("<p>請選擇下列按鈕來控制 LED 燈 : </p>");
    client.println("<a href='\"/LED=ON\"'><button style='font-size:20px;'>開啟
LED</button></a>");
    client.println("<a href='\"/LED=OFF\"'><button style='font-size:20px;'>關
閉 LED</button></a>");
    client.println("</body></html>");

    client.println();
    break;
}
}
}

delay(1);
client.stop();
Serial.println("客戶端已離線");
}
}

```

試著用 AI 寫出一個控制 2LED 的程式