

การใช้งาน WiFi และ HTTP

การทำงานกับ WiFi มีอยู่ด้วยกัน 2 โหมด คือ AP (Access Point) และ STA (Station Accession Point) ในโหมด AP คือ การติดต่อโดยตรง ระหว่างกัน ในขณะที่ STA เป็นการติดต่อกับ ตัวปล่อยสัญญาณ สำหรับลูกค้า อย่าง เราเตอร์ (Router) สำหรับ ESP32 หรือ M5Stack สามารถติดต่อได้ทั้ง AP, STA, และ AP พร้อมๆ กับ STA

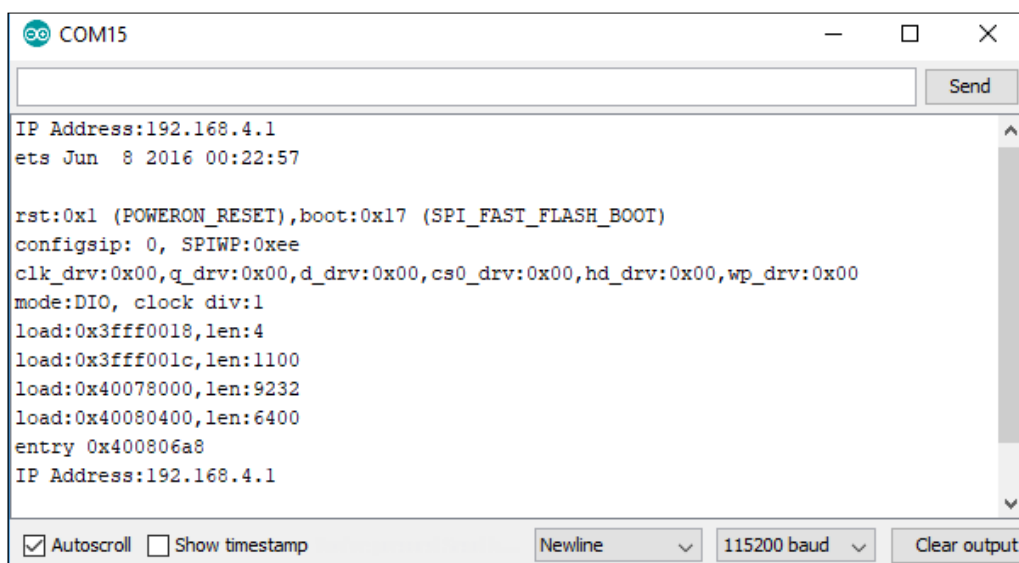
WiFi

ไลบรารี WiFi เป็นไฟล์สำคัญ ที่ใช้ในการติดต่อแบบ WiFi มีฟังก์ชันที่สำคัญคือ: mode(WIFI_AP) ใช้เพื่อเลือกโหมด ใน สำหรับ AP ใช้ WIFI_AP และ softAP(Name, Password) ใช้สำหรับติดต่อในโหมด AP โดยใช้ ชื่อ และรหัส

ตัวอย่างต่อไปนี้ ให้ M5Stack ต่อเชื่อมกับ อุปกรณ์อีกตัวหนึ่ง โดยใช้ โทรศัพท์เคลื่อนที่ โดยให้ ชื่อ M5Stack แทนด้วยตัวแปร WIFI_AP_NAME และ ให้เครื่องอื่น (โทรศัพท์เคลื่อนที่) ติดต่อผ่านรหัส ในตัวแปร WIFI_AP_PASS

ตัวอย่าง 1 การต่อเชื่อม WiFi ในโหมด AP

```
#include <WiFi.h>
#include <WiFiAP.h>
const char* WIFI_AP_NAME = "ESP32 AP";
const char* WIFI_AP_PASS = "12345678";
void setup() {
  Serial.begin(115200);
  WiFi.mode(WIFI_AP);
  WiFi.softAP(WIFI_AP_NAME, WIFI_AP_PASS);
  Serial.print("IP Address:");
  Serial.println(WiFi.softAPIP());
}
void loop() {
  // put your main code here, to run repeatedly:
}
```



รูป 1 ผลการต่อเชื่อม และเมื่อกดปุ่ม Power (ปุ่มสีแดง ด้านข้างของ M5Stack)

สำหรับโหมด STA มีการติดต่อผ่านการใช้ฟังก์ชัน begin(name, password) เมื่อต่อเชื่อมได้จะมีสถานะ โดยใช้ฟังก์ชัน status() โดยเทียบกับ สถานะ WL_CONNECTED ดังตัวอย่างต่อไปนี้

ตัวอย่าง 2 การต่อเชื่อม WiFi ในโหมด STA

```
#include <WiFi.h>

const char* WIFI_STA_NAME = "OPPO A3s";
const char* WIFI_STA_PASS = "12345678";
void setup() {
    Serial.begin(115200);
    WiFi.mode(WIFI_STA);
    WiFi.begin(WIFI_STA_NAME, WIFI_STA_PASS);
    while(WiFi.status() != WL_CONNECTED){
        delay(500);
        Serial.print(".");
    }
    Serial.println();
    Serial.println("WiFi Connected!");
    Serial.print("IP:");
    Serial.println(WiFi.localIP()); //Router gives IP.
}
void loop() {
    // put your main code here, to run repeatedly:
}
```



รูป 2 ผลการต่อเชื่อม ในโหมด STA

เว็บเซิร์ฟเวอร์

เมื่อต่อเชื่อมกับ เราเตอร์ได้ ก็ทำให้เข้าสู่โลกอินเทอร์เน็ตได้ หรือจะสร้างโลกอินเทอร์เน็ตส่วนตัวตัวเอง ผ่าน M5Stack ใช้งานโปรโตคอล HTTP ที่ให้มีไคลเอ็นท์ ติดต่ออ่านเว็บได้ รวมทั้งควบคุมอุปกรณ์ผ่าน M5Stack เช่น LED

ในการสร้างเว็บเซิร์ฟเวอร์ ของ M5Stack จะต้องใช้การสร้าง ผ่าน WiFiServer กับ พอร์ต (Port) ที่อ่านเว็บ เช่น 80 ให้ทดลองทำกับตัวอย่างต่อไปนี้ ที่นอกจากให้อ่านเว็บแล้ว ยังให้ผู้ใช้เว็บควบคุม LED ทางไกลผ่านเว็บได้ด้วย

ตัวอย่าง 3 สร้างเว็บเซิร์ฟเวอร์ ควบคุม LED

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

```

const char* WIFI_STA_NAME = "OPPO A3s";
const char* WIFI_STA_PASS = "354354354";

WiFiServer server(80);
int ledPin = 21;

void setup() {
  Serial.begin(115200);
  WiFi.mode(WIFI_STA);
  WiFi.begin(WIFI_STA_NAME, WIFI_STA_PASS);
  while(WiFi.status() != WL_CONNECTED){
    delay(500);
    Serial.print(".");
  }
  Serial.println();
  Serial.println("WiFi Connected!");

  server.begin();
  Serial.println("Server started");

  // Print the IP address
  Serial.print("Use this URL to connect: ");
  Serial.print("http://");
  Serial.print(WiFi.localIP());
  Serial.println("/");

  pinMode(ledPin, OUTPUT);
  digitalWrite(ledPin, LOW);
}

void loop() {
  // Check if a client has connected
  WiFiClient client = server.available();
  while(client.connected()){
    String request = client.readStringUntil('\r');
    Serial.println(request);
    int value = LOW;
    if (request.indexOf("/LED=ON") != -1) {
      digitalWrite(ledPin, HIGH);
      value = HIGH;
    }
    if (request.indexOf("/LED=OFF") != -1) {
      digitalWrite(ledPin, LOW);
      value = LOW;
    }
  }

  client.println("HTTP/1.1 200 OK");
  client.println("Content-type: text/html");
  client.println(); //do not forget this one
  client.println("<!DOCTYPE HTML>");
  client.println("<html>");
  client.println("<head>");

```

```

        client.println("<style>");
        client.println("h1{text-align:center}");
        client.println("button{padding:20px;margin:20 100px;text-align:center;
font-size:40px;}");
        client.println("</style>");
        client.println("</head>");
        client.println("<body>");
        if(value == HIGH) {
            client.print("<h1>Led pin is now: On</h1>");
            client.println("<a href=\""/LED=OFF\"/\"><button>Turn Off
</button></a>");
        }
        else {
            client.print("<h1>Led pin is now: Off</h1>");
            client.println("<a href=\""/LED=ON\"/\"><button>Turn On</button></a>");
        }

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

```



รูป 3 การควบคุมผ่านหน้าเว็บ

ฟังก์ชันที่สำคัญ ของ WiFiServer มีดังนี้

`available( )` ใช้ในการสร้างไคลเอ็นท์

ฟังก์ชันที่สำคัญ ของ WiFiClient มีดังนี้

`connected( )` ตรวจสอบการต่อเชื่อมกับ ไคลเอ็นท์ หรือเบราว์เซอร์ (Browser)

`readStringUntil('r')` อ่านการส่งคำขอ(อ่านเว็บ) จากไคลเอ็นท์ จะหมดการร้องขอบริการ ใช้ ขึ้นบรรทัดใหม่ (\n) หรือ (\r)

`println()`, `print()` ใช้พิมพ์ เนื้อหาของ HTML

`stop()` ใช้เพื่อหยุดการติดต่อกับไคลเอ็นท์

เตรียมพร้อมสู่งาน IOT เมื่อ WiFi พร้อม ก็ไม่ยากที่ดำเนินการต่อไป เข้าอินเทอร์เน็ตต่อไป งาน IoT ก็เป็นงานปลายทางอีกตัวหนึ่งของ M5Stack ซึ่งมีตัวอย่างมากมาย ให้เลือกใช้บริการ บริการของ IoT ตัวอย่างในเว็บที่ให้นี้ก็เป็นตัวอย่างที่จะศึกษาได้ตนเอง

<https://medium.com/@visitwnk/%E0%B9%83%E0%B8%AA%E0%B9%88%E0%B9%83%E0%B8%88-7-%E0%B9%80%E0%B8%95%E0%B8%A3%E0%B8%B5%E0%B8%A2%E0%B8%A1%E0%B8%84%E0%B8%A7%E0%B8%B2%E0%B8%A1%E0%B8%9E%E0%B8%A3%E0%B9%89%E0%B8%AD%E0%B8%A1%E0%B8%81%E0%B9%88%E0%B8%AD%E0%B8%99%E0%B8%81%E0%B8%B2%E0%B8%A3%E0%B9%83%E0%B8%8A%E0%B9%89-blynk-app-1ab60aa1b9e9>