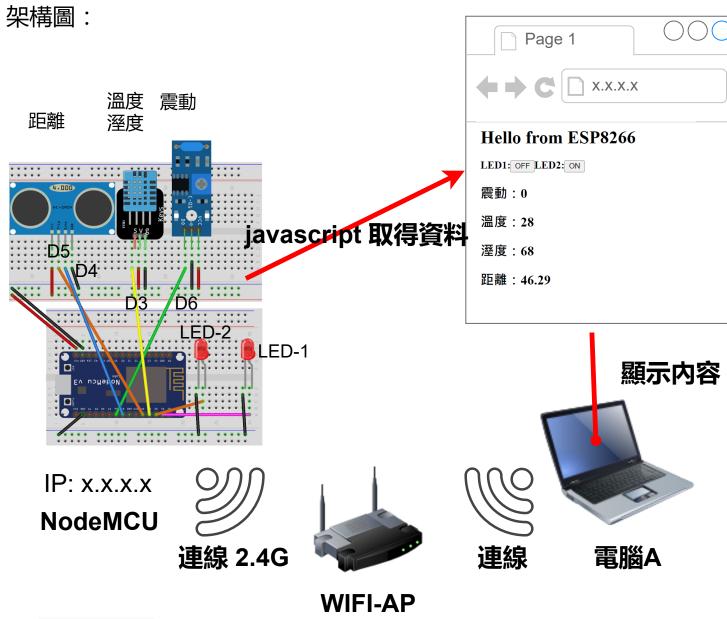
## **IOT HTTP Server v2**

此章節解說如何架設一個 IOT HTTP Server結合溫溼度、震動、距離感測器。

如果對於建置環境不了解,先參考「NodeMCU\_HelloWorld」章節。 如果對於溫溼度感測器不了解,先參考「DHT11」章節。 如果對於震動感測器不了解,先參考「SW420」章節。 如果對於距離感測器不了解,先參考「HC-SR04」章節。 如果對於HTTP server不了解,先參考「HTTPServer」章節。



#### 架構介紹:

- 1.電腦A連線上 WIFI-AP
- 2.NodeMCU連上 WIFI-AP(僅支援 2.4G)
- 3.電腦A打開網頁並輸入NodeMCU的IP
- 4.進入網頁看到震動、溫度、溼度、距離資訊及控制LED的開關

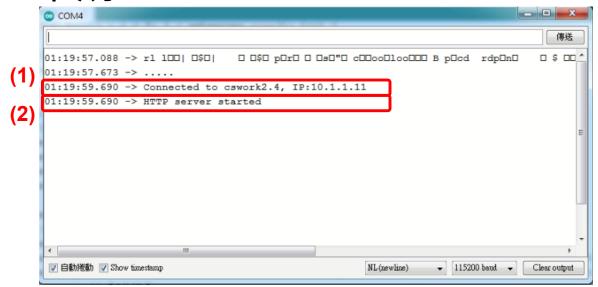
### 1. 編寫草稿碼 -> 上傳至 NodeMCU 開發板

```
IOT_HTTPServer_v2 Arduino 1.8.14
(4)
       IOT_HTTPServer_v2
       1 #include <ESP8266WiFi.h>
       2 #include <WiFiClient.h>
       3 #include <ESP8266WebServer.h>
       5 #include <DHT.h>
       7 #define STASSID "cswork2.4"
(2)
        8 #define STAPSK "22224444"
      10 #define LED1_PIN D1
(3)
      11 #define LED2_PIN D2
      13 #define TEMPERATURE_SENSOR_PIN D3
      14 #define SONA TRIG SENSOR PIN D4
      15 #define SONA_ECHO_SENSOR_PIN D5
      16 #define VIBRATION_SENSOR_PIN D6
      18 const char* ssid = STASSID;
      19 const char* password = STAPSK;
      21 DHT dht (TEMPERATURE_SENSOR_PIN, DHT11);
      22 ESP8266WebServer webserver(80);
      23
      24
      25 void handleNotFound() {
      26 String message = "File Not Found\n\n";
          message += "URI: ";
          message += webserver.uri();
      28
          message += "\nMethod: ";
          message += (webserver.method() == HTTP GET) ? "GET" : "POST";
          message += "\nArguments: ";
      31
          message += webserver.args();
           message += "\n";
           for (uint8_t i = 0; i < webserver.args(); i++) {
      35
             message += " " + webserver.argName(i) + ": " + webserver.arg(i) + "\n";
      36
      37
           webserver.send(404, "text/plain", message);
      38 }
      40 void handleHome() {
(5)
```

- (1):Arduino IDE打開IOT HTTPServer v2.ino
- (2):輸入WiFi SSID與密碼
- (3):設定NodeMCU與LED-1、LED-2、震動感測器、溫溼度感測器、距離感測器的針腳位置
- (4):點擊上傳,此時等待編譯與上傳
- (5):上傳完成

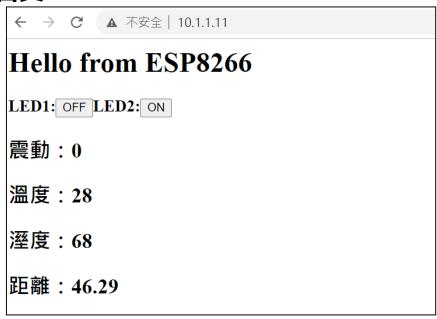
#### 2. 觀看結果

## 2.1 命令列



- (1):NodeMCU 連上 WiFi-AP 取得 IP(10.1.1.11)
- (2):啟動HTTP Server

#### 2.2 進入首頁



首頁顯示震動、溫度、溼度、距離資訊及控制LED的按鈕 note:如果出現亂碼的話,很可能是中文編碼在網頁顯示的問題,建議將中文換英文,或者在NodeMCU回傳網頁時,加入 <meta>並設定網頁為UTF8編碼。

```
void handleHome() {
  String message = "<html>";
  message += "<head>";
  message += "<meta http-equiv=\"Content-Type\" content=\"text/html; charset=utf-8\">";
  message += "</head>";
  message += "</head>";
  message += "<body>";
```

## 2.3 進入未定義的網頁



File Not Found

URI: /zzz

Method: GET Arguments: 0

連上HTTP 路由找不到的頁面,回傳 Not Found

# 2.4 進入API - getdata 網頁

← → C ▲ 不安全 | 10.1.1.11/api/getdata
{"vibration":"0","temp":"27","hum":"63","sona":"57.82"}

在HTTP 路由定義了API - getdata可直接取得NodeMCU向所有感測器收集到的資訊,欄位說明如下:

vibration: 震動感測器 ,布林值 ,0=無震動 ,1=有震動。

temp :溫溼度感測器,數字,攝氏溫度。 hum :溫溼度感測器,數字,20-90%。 snoa :距離感測器,數字,單位公分。

### 2.5 進入API - setdata 網頁



在HTTP 路由定義了API - setdata可直接設定NodeMCU的LED-1、LED-2,設定說明如下:

方式: GET

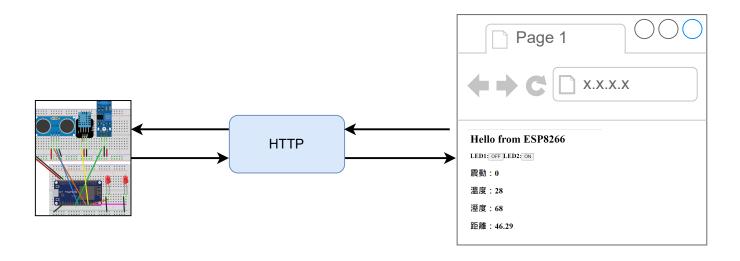
網址: /api/setdata 參數: led1, led2 值: 0=關, 1=開

範例: http://10.1.1.11/api/setdata?led1=1&led2=0

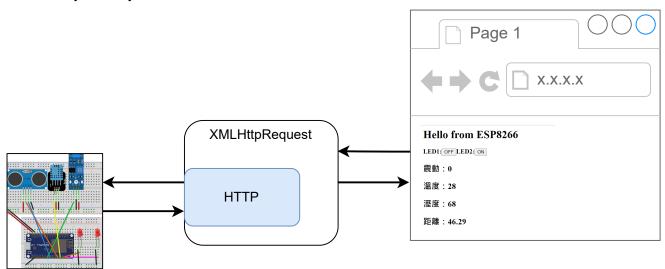
(將LED-1燈打開、LED-2燈關閉)

## 2.6 javascript - XMLHttpRequest

### 瀏覽器向NodeMCU的請求過程



# XMLHttpRequest向NodeMCU的請求過程



## 3. 網頁架構

