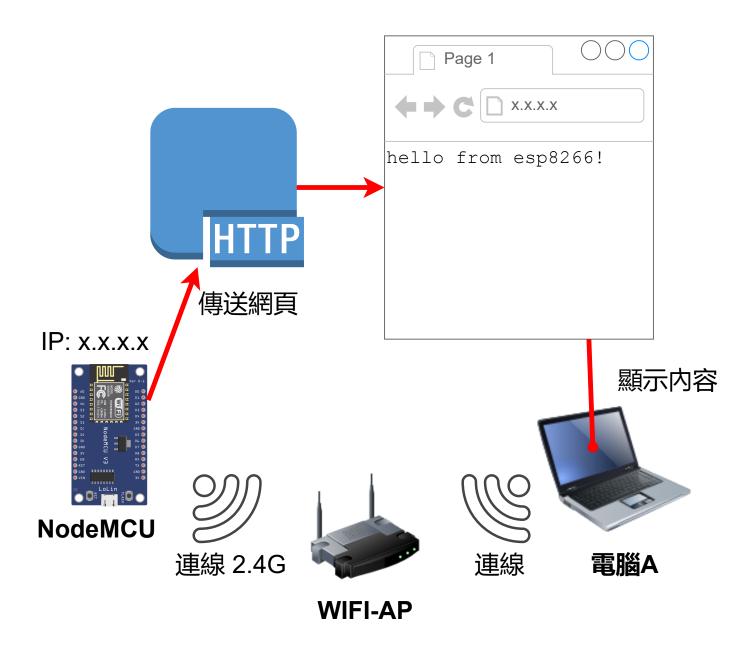
HTTP Server

此章節解說如何架設一個 HTTP Server。如果對於建置環境不了解,先參考「NodeMCU_HelloWorld」章節

架構圖:



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.NodeMCU傳送網頁給電腦A

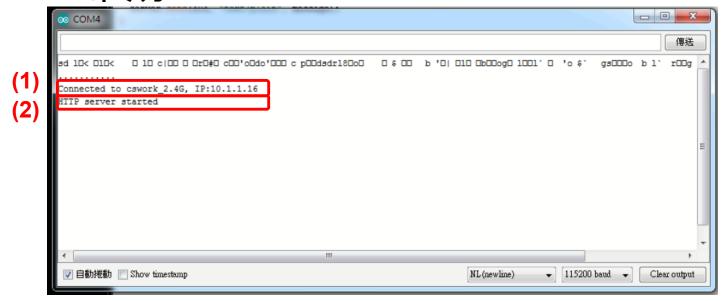
1. 編寫草稿碼 -> 上傳至 NodeMCU 開發板

```
_ D X
      HTTPServer | Arduino 1.8.14
      福案 編輯 草稿碼 工具 說明
(3)
        HTTPServer
        1 #include <ESP8266WiFi.h>
        2 #include <WiFiClient.h>
        3 #include <ESP8266WebServer.h>
(2)
       5 #define STASSID "cswork_2.4G"
        6 #define STAPSK "888888888"
       8 const char* ssid = STASSID;
       9 const char* password = STAPSK;
      10 ESP8266WebServer server(80);
      11
      12 void handleRoot() {
      13 server.send(200, "text/plain", "hello from esp8266!");
      14 }
      15
      16 void handleNotFound() {
      17
      18 String message = "File Not Found\n\n";
      19 message += "URI: ";
       20 message += server.uri();
       21 message += "\nMethod: ";
       22 message += (server.method() == HTTP_GET) ? "GET" : "POST";
       23 message += "\nArguments: ";
       24 message += server.args();
       25 message += "\n";
       26
          for (uint8 t i = 0; i < server.args(); i++) {
            message += " " + server.argName(i) + ": " + server.arg(i) + "\n";
       27
      28
           server.send(404, "text/plain", message);
       29
       30 }
       31
      32 void setup (void)
      33 {
           //console baud rate
          Serial.begin(115200);
```

- (1):Arduino IDE打開HTTPServer.ino
- (2):輸入WiFi SSID與密碼
- (3):點擊上傳,此時等待編譯與上傳

2. 觀看結果

2.1 命令列



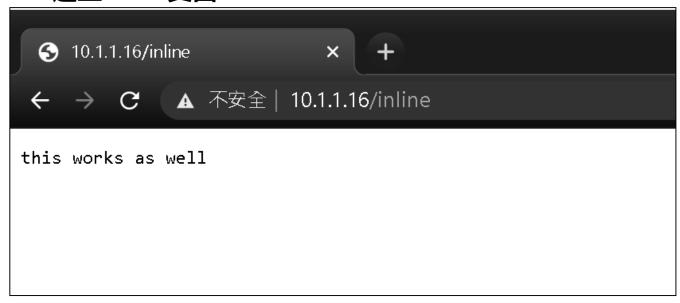
- (1):NodeMCU連上 WiFi-AP 取得 IP(10.1.1.16)
- (2):啟動HTTP Server

2.2 連上首頁



顯示 hello 字樣

2.2 連上inline頁面



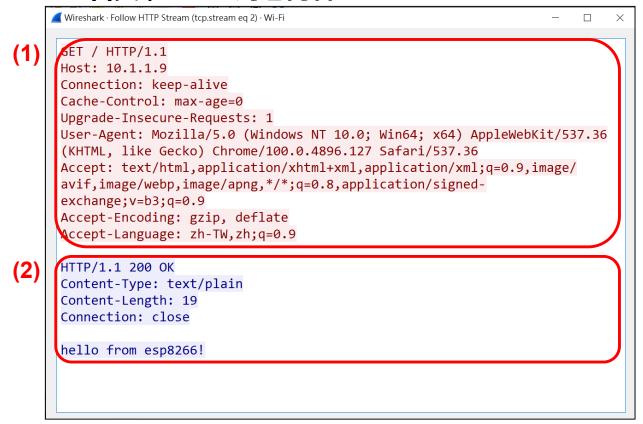
顯示文字

2.3 進入未定義的網頁



連上HTTP 路由找不到的頁面,回傳 Not Found

2.4 首頁, HTTP 封包内容



- (1):紅色字體代表瀏覽器發送HTTP請求的内容
- (2):藍色字體代表NodeMCU發送HTTP回應的内容

3. 網頁架構

