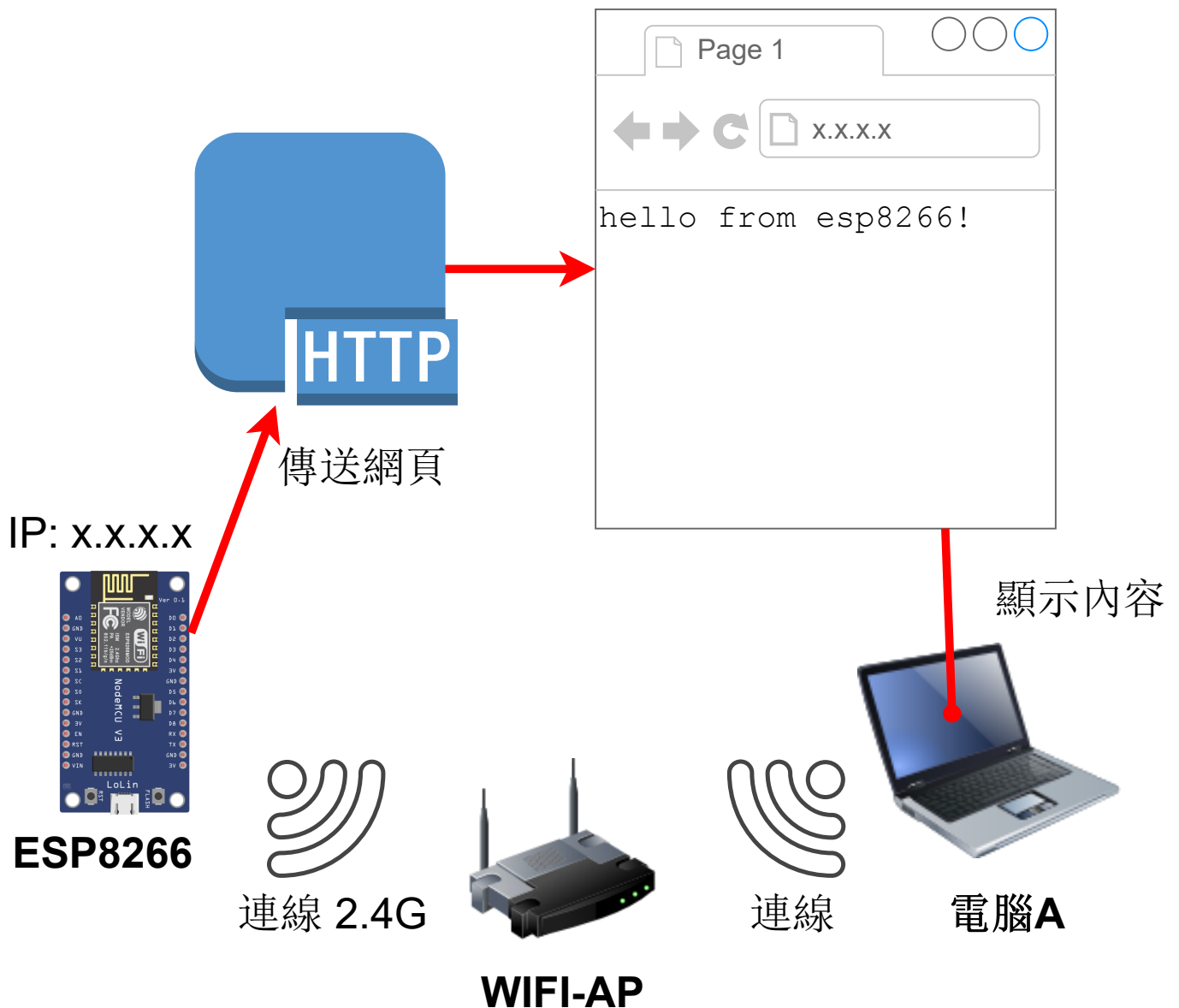


HTTP Server

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.ESP8266連上 WIFI-AP(僅支援 2.4G)
- 3.電腦A打開網頁並輸入ESP8266的IP
- 4.ESP8266 傳送網頁給電腦A

1. 編寫草稿碼 -> 上傳至 NodeMCU 開發板

```
#include <ESP8266WiFi.h>
#include <WiFiClient.h>
#include <ESP8266WebServer.h>

#define STASSID "cswork_2.4G"
#define STAPSK "88888888"

const char* ssid = STASSID;
const char* password = STAPSK;
ESP8266WebServer server(80);

void handleRoot() {
  server.send(200, "text/plain", "hello from esp8266!");
}

void handleNotFound() {

  String message = "File Not Found\n\n";
  message += "URI: ";
  message += server.uri();
  message += "\nMethod: ";
  message += (server.method() == HTTP_GET) ? "GET" : "POST";
  message += "\nArguments: ";
  message += server.args();
  message += "\n";
  for (uint8_t i = 0; i < server.args(); i++) {
    message += "  " + server.argName(i) + ": " + server.arg(i) + "\n";
  }
  server.send(404, "text/plain", message);
}

void setup(void) {
  //console baud rate
  Serial.begin(115200);
  WiFi.mode(WIFI_STA);
  WiFi.begin(ssid, password);
  printf("\n");

  // Wait for WiFi connection
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  printf("\n");
  printf("Connected to %s, IP:%s\n", ssid, WiFi.localIP().toString().c_str());

  //定義 HTTP Server 路由
  //首頁
  server.on("/", handleRoot);

  //inline 網頁
  server.on("/inline", []() {
    server.send(200, "text/plain", "this works as well");
  });

  //當 HTTP Server 路由找不到時，進入 Not Found
  server.onNotFound(handleNotFound);
  //HTTP Server 啟動
  server.begin();
  printf("HTTP server started\n");
}

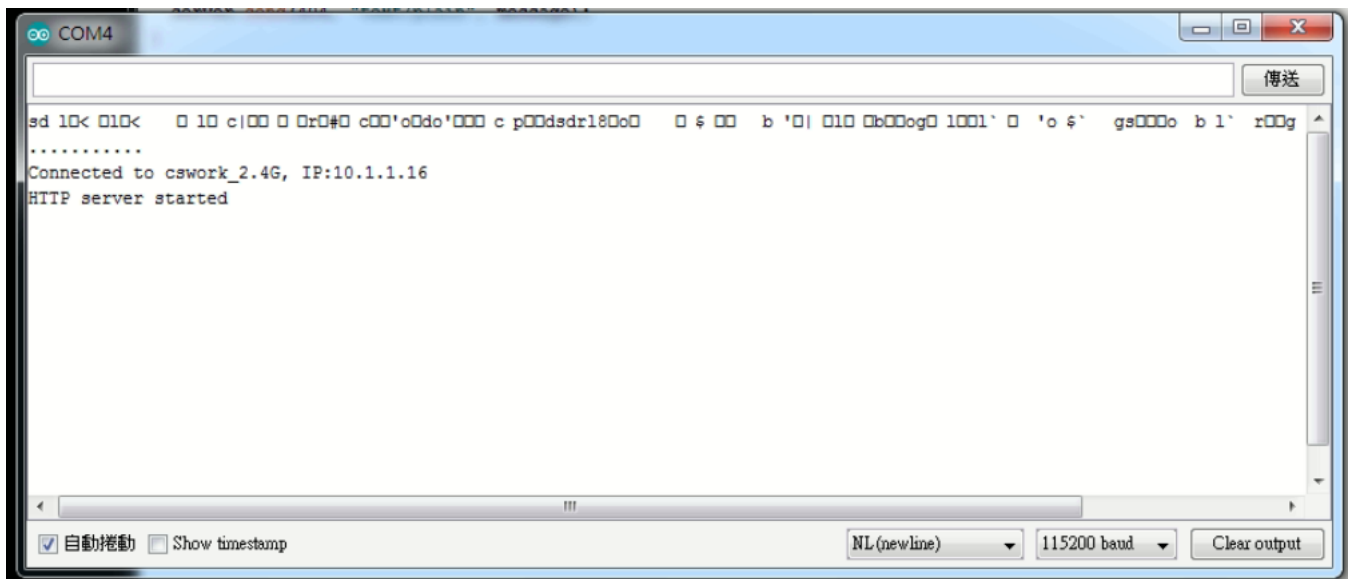
void loop(void) {
  //檢查是否有連線進入
  server.handleClient();
}
```

從程式碼可看出：

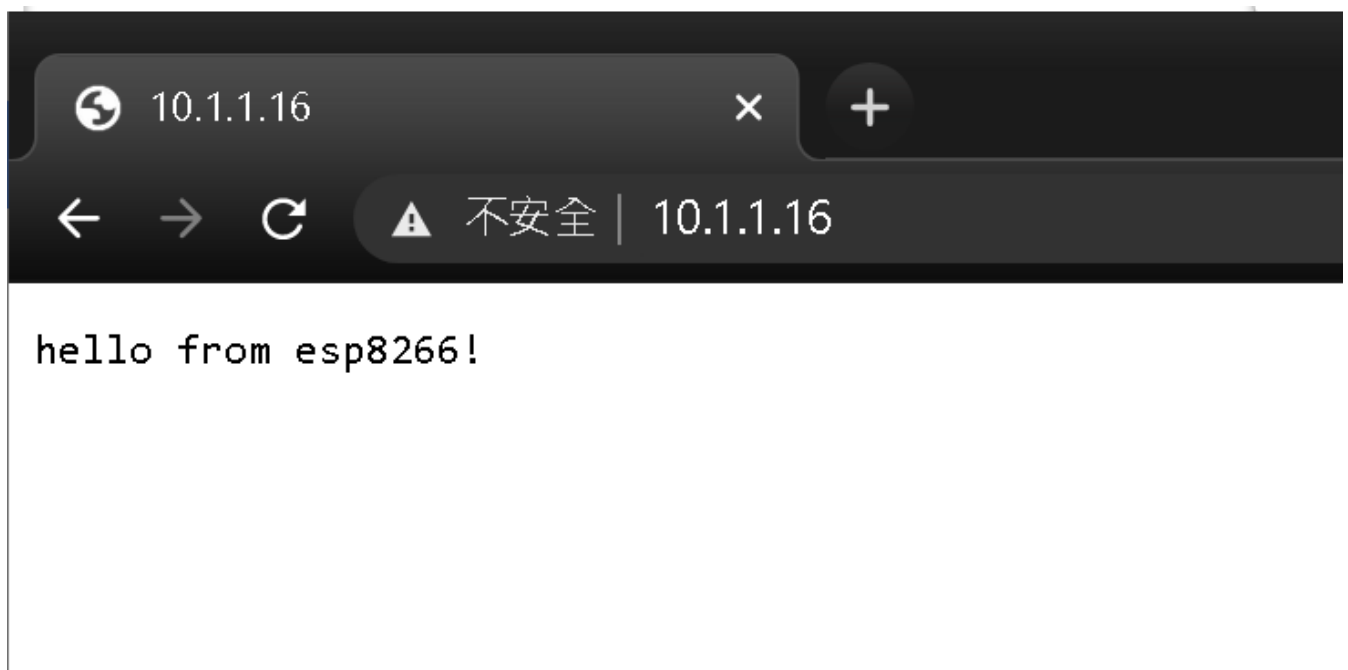
1.ESP8266連上WIFI得到IP

2.HTTP Server除了首頁外，還定義了inline頁面，以及當 HTTP 路由找不到時，回傳 Not Found的頁面。

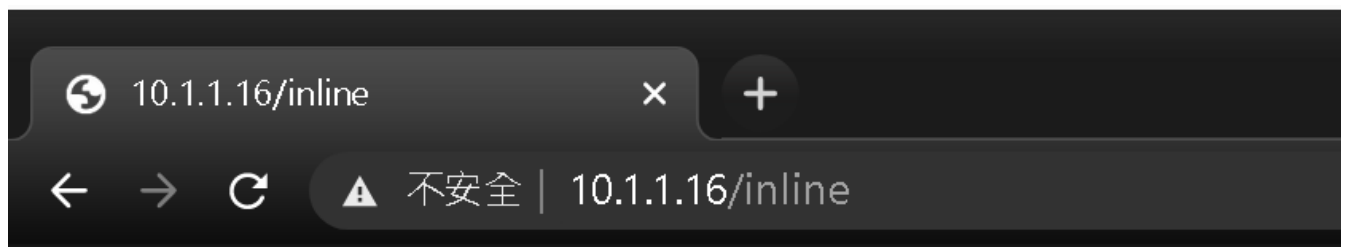
2. 觀看結果



ESP8266 連上 WiFi-AP 取得 IP(10.1.1.16)後，啟動HTTP Server



連上首頁



this works as well

連上 inline 頁面



連上HTTP 路由找不到的頁面，得到 Not Found