

# 其他元件 – Wi-Fi 通訊

作者：林煜衡

## 本章介紹

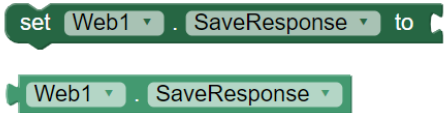
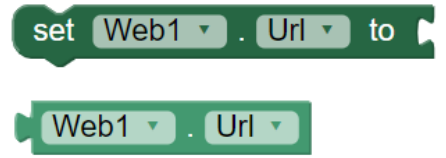
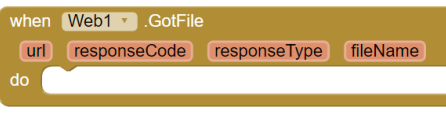
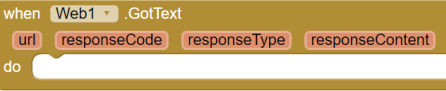
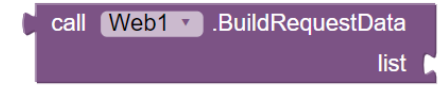

因為 App Inventor 2 中的 Wi-Fi 的模組是控制 Wi-Fi 開關或者連接，沒辦法傳送或接收資料來控制 Arduino，所以是透過 Web 元件與 Button 開啟網頁，傳送資料，並在 Arduino 製作一個簡易的 web server，使 App Inventor 2 的 Button 連接網頁，或是在 URL 中直接輸入 IP 位置開啟網頁，以達成 Arduino 的 LED 開關控制。

## Wi-Fi

Wi-Fi 在中文裡又稱作「無線熱點」，是一個無線網路通信技術的品牌，由 Wi-Fi 聯盟 ( Wi-Fi Alliance ) 所持有，由無線乙太網相容聯盟所發佈的業界術語，指的是無線資料傳輸技術與規格。Wi-Fi 採用 IEEE 802.11 系列標準之無線區域網路 ( Wireless Local Area Network / Wireless LAN; WLAN )。使用 Wi-Fi 無線區域網路可讓使用者透過數據機在 300 英尺範圍內(約 91 公尺)無線上網。Wi-Fi 網路傳輸速度快，但距離短，多半設在咖啡店或機場等處，提供使用者穩定平順的無線上網通訊。

Web 元件常用介紹：

屬性	功能
	設定是否可儲存網頁回應的 cookies，並用於後續的網路要求。只有 Android 2.3 版以後才支援 cookies。
	Request header 是一個包含兩個子清單的清單。  每個子清單的第一個元素是欄位名稱；第二個元素則是欄位值，可能是單一值或是一個包含多值的清單。
	用來儲存網頁回應的檔案名稱。如果 SaveResponse 欄位設定為 true 但未指定 ResponseFileName 的話，就會自動

	產生新的檔名。
	設定是否要將網頁回應存在檔案中。
	網路要求的 URL 路徑。
事件	功能
	執行完一次網路取得檔案之後，呼叫本事件。
	執行完一次網路取得文字檔案之後，呼叫本事件。
方法	功能
	將一個包含兩個子清單的清單轉換為應用程式格式的字串，可用來傳給 <b>PostText</b> 。
	清除本 Web 元件的所有 cookies。

	<p>執行一次 HTTP DELETE 要求，須設定 <code>Url</code> 屬性之後才能取得回應。如果 <code>SaveResponse</code> 屬性為 <code>true</code>，回應會存在某個檔案之中並呼叫 <code>GotFile</code> 事件。</p> <p><code>ResponseFileName</code> 屬性可用來指定該檔案檔名。如果 <code>SaveResponse</code> 屬性為 <code>false</code>，就會呼叫 <code>GotText</code> 事件。</p>
	<p>執行一次 HTTP GET 要求，須設定 <code>Url</code> 屬性之後才能取得回應。如果 <code>SaveResponse</code> 屬性為 <code>true</code>，回應會存在某個檔案之中並呼叫 <code>GotFile</code> 事件。</p> <p><code>ResponseFileName</code> 屬性可用來指定該檔案檔名。如果 <code>SaveResponse</code> 屬性為 <code>false</code>，就會呼叫 <code>GotText</code> 事件。</p>
	<p>對指定 HTML 文字進行解碼。像 <code>&amp;amp;</code>、<code>&amp;lt;</code>、<code>&amp;gt;</code>、<code>&amp;apos;</code> 以及 <code>&amp;quot;</code> 這樣的 HTML 字元會被解析為 <code>&amp;</code>、<code>&lt;</code>、<code>&gt;</code>、<code>'</code> 與 <code>"</code>。像 <code>&amp;#xhhhh</code> 與 <code>&amp;#nnnn</code> 會被解析為對應的字元。</p>
	<p>解碼指定的 JSON 物件以產生對應的 AppInventor 值。一個 JSON 清單 <code>[x, y, z]</code> 會被解碼為 <code>(x y z)</code> 這樣的清單。</p> <p>具有 <code>name A</code> 與 <code>value B</code> 的 JSON 物件，例如 <code>{name:123}</code>，會被解碼為會被解碼為 <code>((name 123))</code> 這樣的清單，也就是一個包含一個二元素清單 <code>(name 123)</code> 的清單。</p>

	<p>執行一次 HTTP POST 要求，須設定 <b>Url</b> 屬性與指定檔案的路徑資料。如果 <b>SaveResponse</b> 屬性為 <b>true</b>，回應會存在某個檔案之中並呼叫 <b>GotFile</b> 事件。</p> <p><b>ResponseFileName</b> 屬性可用來指定該檔案檔名。如果 <b>SaveResponse</b> 屬性為 <b>false</b>，就會呼叫 <b>GotText</b> 事件。</p>
	<p>執行一次 HTTP POST 要求，須設定 <b>Url</b> 屬性與指定文字。文字字元會以 <b>UTF-8</b> 進行編碼。如果 <b>SaveResponse</b> 屬性為 <b>true</b>，回應會存在某個檔案之中並呼叫 <b>GotFile</b> 事件。</p> <p><b>ResponseFileName</b> 屬性可用來指定該檔案檔名。如果 <b>SaveResponse</b> 屬性為 <b>false</b>，就會呼叫 <b>GotText</b> 事件。</p>
	<p>執行一次 HTTP POST 要求，需設定 <b>Url</b> 屬性與指定文字。文字字元會以指定格式來進行編碼。如果 <b>SaveResponse</b> 屬性為 <b>true</b>，回應會存在某個檔案之中並呼叫 <b>GotFile</b> 事件。</p> <p><b>ResponseFileName</b> 屬性可用來指定該檔案檔名。如果 <b>SaveResponse</b> 屬性為 <b>false</b>，就會呼叫 <b>GotText</b> 事件。</p>

	<p>執行一次 HTTP PUT 要求，須設定 Url 屬性與指定檔案的路徑資料。如果 SaveResponse 屬性為 true，回應會存在某個檔案之中並呼叫 GotFile 事件。</p> <p>ResponseFileName 屬性可用來指定該檔案檔名。如果 SaveResponse 屬性為 false，就會呼叫 GotText 事件。</p>
	<p>執行一次 HTTP PUT 要求，須設定 Url 屬性與指定文字。文字字元會以 UTF-8 進行編碼。如果 SaveResponse 屬性為 true，回應會存在某個檔案之中並呼叫 GotFile 事件。</p> <p>ResponseFileName 屬性可用來指定該檔案檔名。如果 SaveResponse property 為 false，就會呼叫 GotText 事件。</p>
	<p>執行一次 HTTP PUT 要求，須設定 Url 屬性與指定文字。文字中的字元會以指定編碼格式來進行編碼。如果 SaveResponse 屬性為 true，回應會存在某個檔案之中並呼叫 GotFile 事件。</p> <p>ResponseFileName 屬性可用來指定該檔案檔名。如果 SaveResponse 屬性為 false，就會呼叫 GotText 事件。</p>
	<p>將指定文字進行編碼，讓它可用在 URL 之中。</p> <p>在某些情況下會需要在 URL 中輸入中文字，但是直接輸入中文字可能會因為編碼問題而產生亂碼。因此需要此方法轉換編碼。</p>

## 範例-利用手機 Wi-Fi 控制 LED 開關

### 步驟一.

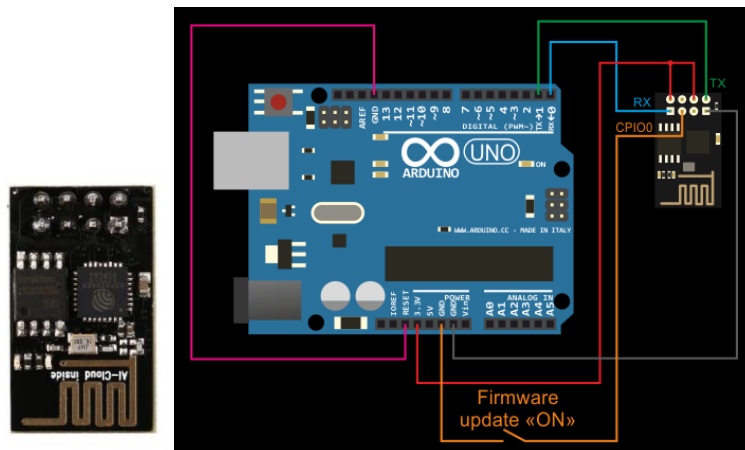
- 首先使用 Arduino 連接 Wi-Fi 模組 ( ESP-8266 )，目前在 Arduino 使用 Wi-Fi 模組的方法分為 2 種，如果開發板上沒有內建 Wi-Fi 晶片需要另外購買 Wi-Fi 模組連接 Arduino，連接時要請注意 **Wi-Fi 模組的正極只能連接 3.3V，否則會使晶片燒毀**。基本上兩種方式的程式碼概念是一樣的，只有在 Wi-Fi 的設定方式不同。

本範例使用內建有 Wi-Fi 模組的 NodeMcu 開發版，若使用無內建 Wi-Fi 模組的開發版則必須將 ESP-8266 的 RX 接 Arduino 的 0；TX 接 1；VCC 和 CH\_PD 接 3.3V；GND 接 GND。

LED 燈的正極則分別將接在開發版上的 PIN 接腳上、負極接 GND，一般開發版的接腳名稱為 1~13 的數字，部分開發版的接腳名稱會在數字前將上字母 D ( 例如：D1、D2、D3 )，通常 0、1 接腳是用來與其他開發版或通訊模組溝通在連接其他模組時盡量避免使用。



Wi-Fi 模組 ( ESP-8266、ESP-01 )  
內建 Wi-Fi 模組的開發板 ( Arduino UNO WIFI )



ESP-8266 與 Arduino 連接範例

## 步驟二.

- 連接成功後將 Arduino 程式碼上傳至開發板並找出 Arduino 的 IP，上傳完成後開啟 Serial Monitor 就能看到 Arduino 的 IP，不同的設備 IP 可能會不同，請確認 IP 位置無誤。

```

1 #include <SoftwareSerial.h>
2 #include <ESP8266WiFi.h>
3
4 ESP8266 WiFi(Serial1);
5
6 char ssid[] = "ssid"; //
7 char pass[] = "pass"; // you
8 int keyIndex = 0;
9
10 int status = WL_IDLE_STATUS;
11 WiFiServer server(80);
12
13 void setup() {
14   Serial.begin(9600); // initialize serial communication
15   pinMode(LED_BUILTIN, OUTPUT); // set the LED pin mode
16   pinMode(11, OUTPUT); //R
17   pinMode(9, OUTPUT); //G
18   pinMode(7, OUTPUT); //B
19
20   // attempt to connect to Wifi network:
21   while (status != WL_CONNECTED) {
22     Serial.print("Attempting to connect to Network named: ");
23     Serial.println(ssid); // print the network name (SSID);
24
25     // Connect to WPA/WPA2 network. Change this line if using open or WEP network:
26     status = WiFi.begin(ssid, pass);
27   }
28   server.begin(); // start the web server on port 80
29   printWifiStatus(); // you're connected now, so print out the status
30 }
31
32
33 void loop() {
34   WiFiClient client = server.available(); // listen for incoming clients
35
36   if (client) { // if you get a client,
37     Serial.println("new client"); // print a message out the serial port
38     String currentLine = ""; // make a String to hold incoming data from the client
39     while (client.connected()) { // loop while the client's connected
40       if (client.available()) { // if there's bytes to read from the client,
41         char c = client.read(); // read a byte, then
42         Serial.write(c); // print it out the serial monitor
43         if (c == '\n') { // if the byte is a newline character
44
45           // if the current line is blank, you got two newline characters in a row.
46           // that's the end of the client HTTP request, so send a response:
47           if (currentLine.length() == 0) {
48             // HTTP headers always start with a response code (e.g. HTTP/1.1 200 OK)
49             // and a content-type so the client knows what's coming, then a blank line:
50             client.println("HTTP/1.1 200 OK");
51             client.println("Content-type:text/html");
52             client.println();
53
54             // the content of the HTTP response follows the header:
55             client.print("Click <a href=\"/R/\">here</a> to light red<br>");
56             client.print("Click <a href=\"/G/\">here</a> to light green<br>");
57             client.print("Click <a href=\"/B/\">here</a> to light blue<br>");
58             client.print("Click <a href=\"/W/\">here</a> to light white<br>");
59             client.print("Click <a href=\"/o/\">here</a> to light off<br>");
60
61             // The HTTP response ends with another blank line:
62             client.println();
63             // break out of the while loop:
64             break;
65           } else { // if you got a newline, then clear currentLine:
66             currentLine = "";
67           }
68         } else if (c != '\r') { // if you got anything else but a carriage return character,
69           currentLine += c; // add it to the end of the currentLine
70         }
71

```

請輸入要連接的無線網路的名稱(SSID)及密碼(PASSWORD)

連接時請將 NodeMcu 與手機連接到同一無線網路設備，並確

認裝置在同一區域網路內。

本專案使用的 NodeMcu 的接腳為 D2、D4、D6。

```

72 // Check client request
73 if (currentLine.startsWith("GET /R")) { // GET /R to light red
74     led(1, 0, 0);
75 }
76 if (currentLine.startsWith("GET /G")) { // GET /G to light green
77     led(0, 1, 0);
78 }
79 if (currentLine.startsWith("GET /B")) { // GET /B to light blue
80     led(0, 0, 1);
81 }
82 if (currentLine.startsWith("GET /W")) { // GET /W to light white
83     led(1, 1, 1);
84 }
85 if (currentLine.startsWith("GET /o")) { // GET /o to light off
86     led(0, 0, 0);
87 }
88 }
89 }
90 // close the connection:
91 client.stop();
92 Serial.println("client disconnected");
93 }
94 }
95
96 void printWifiStatus() {
97     // print the SSID of the network you're attached to:
98     Serial.print("SSID: ");
99     Serial.println(WiFi.SSID());
100
101     // print your WiFi shield's IP address:
102     IPAddress ip = WiFi.localIP();
103     Serial.print("IP Address: ");
104     Serial.println(ip);
105
106     // print the received signal strength:
107     long rssi = WiFi.RSSI();
108     Serial.print("signal strength (RSSI):");
109     Serial.print(rssi);
110     Serial.println(" dBm");
111     // print where to go in a browser:
112     Serial.print("To see this page in action, open a browser to http://");
113     Serial.println(ip);
114 }
115
116 void led(int R, int G, int B) {
117     digitalWrite(11, R);
118     digitalWrite(9, G);
119     digitalWrite(7, B);
120     Serial.print(R);
121     Serial.print(",");
122     Serial.print(G);
123     Serial.print(",");
124     Serial.print(B);
125 }

```

```

COM4 (LinkIt 7697)
Host: 192.168.1.73
Connection: keep-alive
Pragma: no-cache
Cache-Control: no-cache
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/88.0.4397.80 Safari/537.36
Save-Data: on
Accept: image/webp,image/apng,image/*,*/*;q=0.8
Referer: http://192.168.1.73/
Accept-Encoding: gzip, deflate
Accept-Language: zh-TW,zh;q=0.8,en-US;q=0.6,en;q=0.4,zh-CN;q=0.4

client disconnected
new client
client disconnected

```

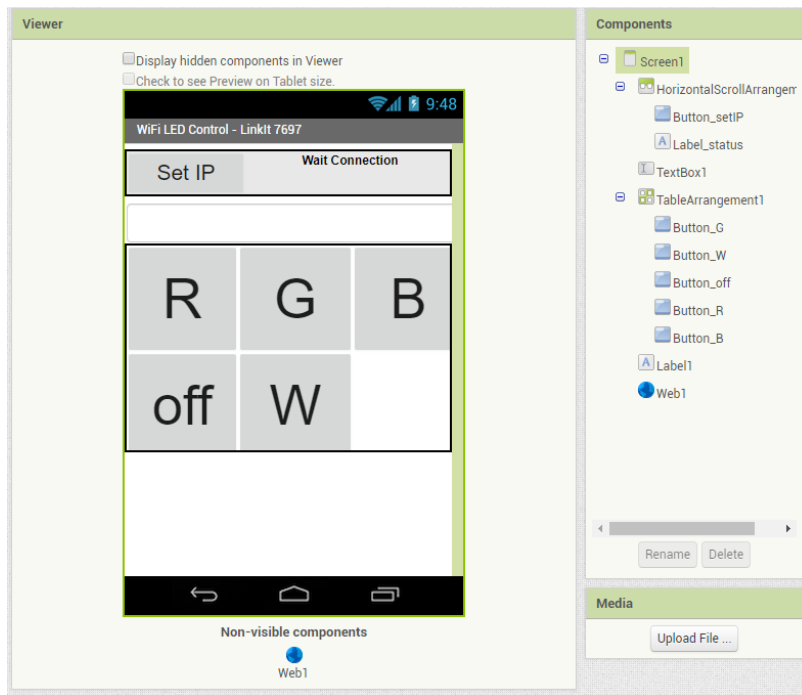


## 活動

請用電腦或手機連接 Arduino，並開啟瀏覽器輸入在步驟二得到的 IP 位置，測試是否有成功連接，如果有可以試著點擊網頁的超連結，看看 Arduino 上的變化。

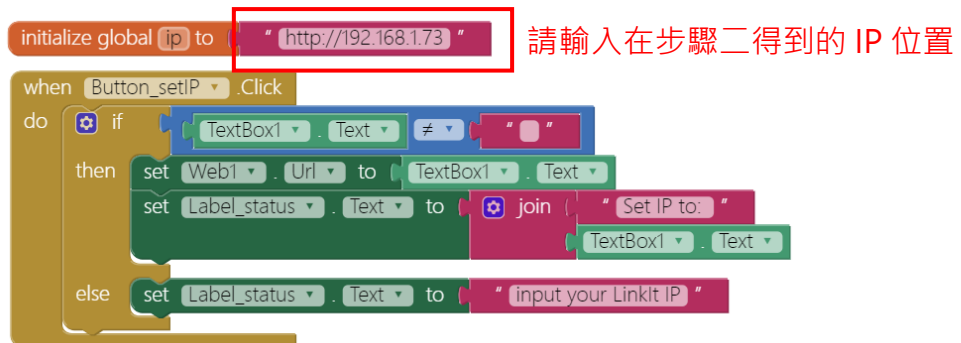
## 步驟三.

- 建立 App Inventor 2 專案，新增版面及程式碼。



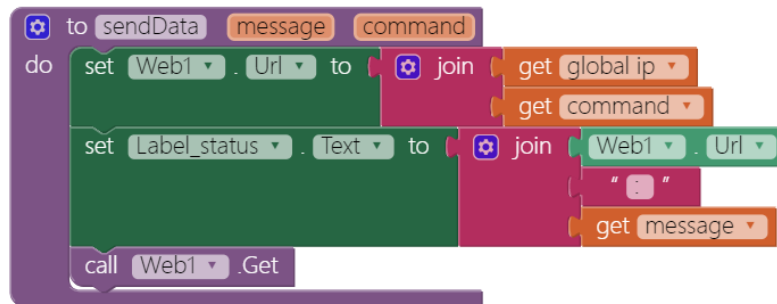
## 步驟四.

- 設定連接 Arduino 的 IP。



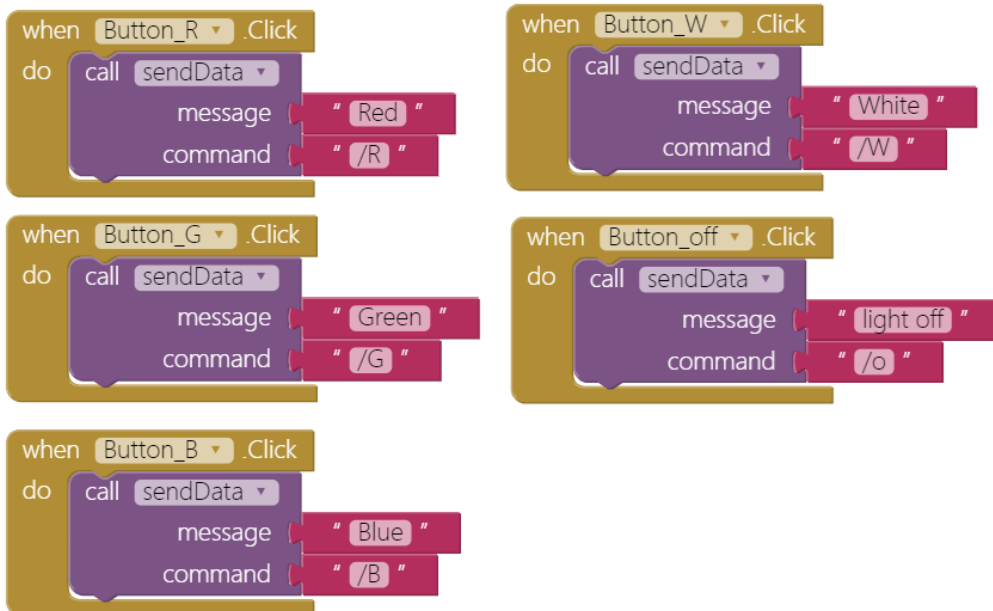
### 步驟五.

- 自訂傳送函數。



### 步驟六.

- 設定 Button 的點擊觸發事件。



### 步驟七.

- 藉由傳送不同 message 與 command 在 Arduino 中進行 LED 的開關與顏色控制。

```

// Check client request
if (currentLine.startsWith("GET /R")) { // GET /R to light red
  led(1,0,0);
}
if (currentLine.startsWith("GET /G")) { // GET /G to light green
  led(0,1,0);
}
if (currentLine.startsWith("GET /B")) { // GET /B to light blue
  led(0,0,1);
}
if (currentLine.startsWith("GET /W")) { // GET /W to light white
  led(1,1,1);
}
if (currentLine.startsWith("GET /o")) { // GET /o to light off
  led(0,0,0);
}

```

## 步驟八.

- 執行結果。

