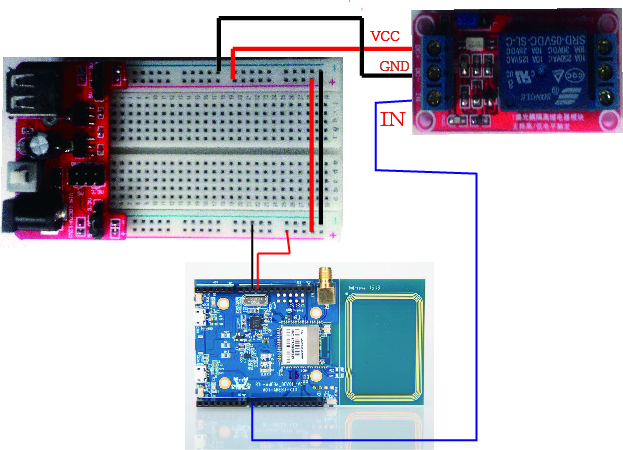
程式：WebServerControlRelay 使用Client模式建立網頁伺服器

**開啟程式**WebServerControlRelay

**程式位址：**https://github.com/brucetsao/BruceCourses/blob/master/105ANQU\_IOT/Code/WebServerControlRelay/WebServerControlRelay.ino

****

|  |
| --- |
| #include <WiFi.h>  char ssid[] = "Apple\_TC\_Wi-Fi"; // your network SSID (name)  char pass[] = "27541147"; // your network password  int keyIndex = 0; // your network key Index number (needed only for WEP)  int status = WL\_IDLE\_STATUS;  WiFiServer server(80);  #define RelayPin 8  #define Turnon HIGH  #define Turnoff LOW  void setup() {  Serial.begin(9600); // initialize serial communication  pinMode(RelayPin, OUTPUT); // set the LED pin mode  // check for the presence of the shield:  if (WiFi.status() == WL\_NO\_SHIELD) {  Serial.println("WiFi shield not present");  while (true); // don't continue  }    String fv = WiFi.firmwareVersion();  if (fv != "1.1.0") {  Serial.println("Please upgrade the firmware");  }  // attempt to connect to Wifi network:  while (status != WL\_CONNECTED) {  Serial.print("Attempting to connect to Network named: ");  Serial.println(ssid); // print the network name (SSID);  // Connect to WPA/WPA2 network. Change this line if using open or WEP network:  // status = WiFi.begin(ssid); //no pass  status = WiFi.begin(ssid, pass); //wpa use  // wait 10 seconds for connection:  delay(10000);  }  server.begin(); // start the web server on port 80  printWifiStatus(); // you're connected now, so print out the status  }  void loop() {  WiFiClient client = server.available(); // listen for incoming clients  if (client) { // if you get a client,  Serial.println("new client"); // print a message out the serial port  String currentLine = ""; // make a String to hold incoming data from the client  while (client.connected()) { // loop while the client's connected  if (client.available()) { // if there's bytes to read from the client,  char c = client.read(); // read a byte, then  Serial.write(c); // print it out the serial monitor  if (c == '\n') { // if the byte is a newline character  // if the current line is blank, you got two newline characters in a row.  // that's the end of the client HTTP request, so send a response:  if (currentLine.length() == 0) {  // HTTP headers always start with a response code (e.g. HTTP/1.1 200 OK)  // and a content-type so the client knows what's coming, then a blank line:  client.println("HTTP/1.1 200 OK");  client.println("Content-type:text/html");  client.println();  // the content of the HTTP response follows the header:  client.print("Click <a href=\"/H\">here</a> turn Power on<br>");  client.print("Click <a href=\"/L\">here</a> turn Power off<br>");  // The HTTP response ends with another blank line:  client.println();  // break out of the while loop:  break;  } else { // if you got a newline, then clear currentLine:  currentLine = "";  }  } else if (c != '\r') { // if you got anything else but a carriage return character,  currentLine += c; // add it to the end of the currentLine  }  // Check to see if the client request was "GET /H" or "GET /L":  if (currentLine.endsWith("GET /H")) {  digitalWrite(RelayPin, Turnon); // GET /H turns the LED on  }  if (currentLine.endsWith("GET /L")) {  digitalWrite(RelayPin, Turnoff); // GET /L turns the LED off  }  }  }  // close the connection:  client.stop();  Serial.println("client disonnected");  }  }  void printWifiStatus() {  // print the SSID of the network you're attached to:  Serial.print("SSID: ");  Serial.println(WiFi.SSID());  // print your WiFi shield's IP address:  IPAddress ip = WiFi.localIP();  Serial.print("IP Address: ");  Serial.println(ip);  // print the received signal strength:  long rssi = WiFi.RSSI();  Serial.print("signal strength (RSSI):");  Serial.print(rssi);  Serial.println(" dBm");  // print where to go in a browser:  Serial.print("To see this page in action, open a browser to http://");  Serial.println(ip);  } |

WebServerControlRelay**程式重點解說**

* #include <WiFi.h> 使用網路必要函數
* WiFiServer server(80); 啟動PORT 80進行
* status = WiFi.begin(ssid); 不使用加密連AP
* status = WiFi.begin(ssid, pass); 使用加密連AP
* server.begin(); 開始啟動PORT 80 傾聽
* printWifiStatus(); 列印網路資訊
* WiFiClient client = server.available(); 有人連接Port 80
* client.connected() 有用戶連接中
* client.available() 用戶送資料進來
* char c = client.read(); 讀出用戶送的資料(一個位元組)
* client.println(“HTTP/1.1 200 OK”); …. 送給用戶端一段HTML碼，用瀏覽器方能顯示
* currentLine.endsWith(“GET /H”) 判斷是否用/H 結束
* currentLine.endsWith(“GET /L”) 判斷是否用/L 結束
* client.stop(); 與用戶連線停止傳輸資料

