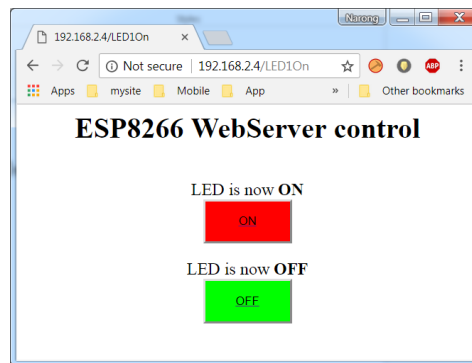


การควบคุมผ่านทาง web server



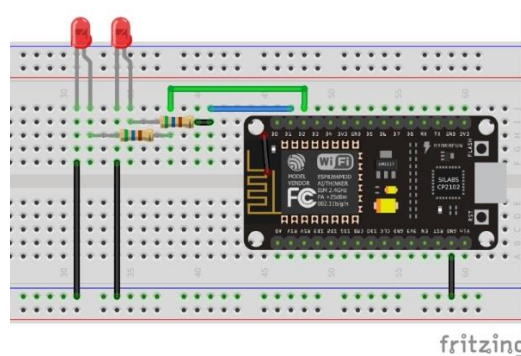
วัตถุประสงค์

สร้าง Server เพื่อควบคุมการทำงานของ LED 2 ดวง ผ่านทางเว็บ เว็บเบราว์เซอร์

อุปกรณ์

1. ESP8266 NodeMCU
2. LED 2 ตัว
3. ความต้านทาน 560 โอห์ม 2 ตัว
4. บอร์ดทดลอง (Breadboard)

วงจร



โปรแกรม ให้ ESP8266 NodeMCU ทำหน้าที่เป็นเว็บเซิร์ฟเวอร์ เมื่อมีการร้องขอมาจาก เว็บไคลเอ็นท์ (Web Client) ให้ ESP8266 NodeMCU ส่งคำสั่งในภาษา HTML ไปยังเว็บไคลเอ็นท์ เพื่อแสดงเป็นหน้าเว็บเพจ โดยคำสั่งที่สำคัญคือการทำเป็นปุ่มลิงค์ (Button Link) เมื่อมีการคลิกที่ปุ่มลิงค์ใดปุ่มลิงค์หนึ่ง ก็ให้เว็บเซิร์ฟเวอร์ ส่งคำสั่งสำหรับสร้างหน้าเพจใหม่ไป

คำสั่งภาษา HTML ที่เกี่ยวกับปุ่มลิ่งค์

```
"<a href=\"LED1Off\"><button style=\"display: block; background-color: #FF0000; height: 50px; width: 100px;\">ON</button></a><br/>"
```

ความหมาย

ถ้าคลิกปุ่มให้ไปที่ URL "LED1Off"

ลักษณะของปุ่มให้เป็นบล็อกรหัสสีแดง (FF0000 = red = 255 green = 00 blue = 00)

ปุ่มมีขนาด สูง 50 พิกเซล กว้าง 100 พิกเซล

และใส่คำว่า "ON" ไว้ที่ปุ่ม

และอีกหน้าที่หนึ่งของ ESP8266 NodeMCU คือการตรวจสอบชื่อที่อยู่ของหน้าเว็บนั้น หรือ URL ของเว็บว่ามีตัวอักษรที่สำคัญอะไร ถ้ามีตรงกับคีย์เวิร์ด ที่กำหนดไว้ก็ให้ LED ทำงาน และเตรียม URL ของหน้าเว็บใหม่แก่ปุ่มลิ่งค์ต่อไป

คำสั่งที่เกี่ยวกับการตรวจสอบ URL

```
server.on("/", handleRoot); //มีเพียงคำว่า "/" ให้ทำฟังก์ชัน handleLedRoot
server.on("/LED1On", handleLed1On); //มีคำว่า "/LED1On" ให้ทำฟังก์ชัน handleLed1On
server.on("/LED1Off", handleLed1Off); //มีคำว่า "/LED1Off" ให้ทำฟังก์ชัน handleLed1Off
server.on("/LED2On", handleLed2On); //มีคำว่า "/LED2On" ให้ทำฟังก์ชัน handleLed2On
server.on("/LED2Off", handleLed2Off); //มีคำว่า "/LED2Off" ให้ทำฟังก์ชัน handleLed2Off
```

โปรแกรม esp_web_server_control.ino

```
// Load Wi-Fi library
#include <ESP8266WiFi.h>
//#include <WiFiClient.h>
#include <ESP8266WebServer.h>

const char* ssid = "....."; //ชื่อไวไฟ
const char* password = "....."; //รหัสผ่านของไวไฟ
ESP8266WebServer server(80); // Set web server port number to 80

// Variable to store the HTTP request
String header;

// Auxiliar variables to store the current output state
String output1State = "off";
String output2State = "off";

// Assign output variables to GPIO pins
const int output1 = D1;
const int output2 = D2;
```

```

#define output1_OFF digitalWrite(output1, LOW)
#define output1_ON  digitalWrite(output1, HIGH)
#define output2_OFF digitalWrite(output2, LOW)
#define output2_ON  digitalWrite(output2, HIGH)

//HTML code
const String HtmlHtml = "<html><head>"
  "<meta name=\"viewport\" content=\"width=device-width, initial-scale=1\" /></head>";
const String HtmlHtmlClose = "</html>";
const String HtmlTitle = "<h1>ESP8266 WebServer control</h1><br/>\n";
const String HtmlCenter = "<center>";
const String HtmlCloseCenter = "</center>";
const String HtmlButton1_ON = "<a href=\"LED1Off\"><button style=\"display: block; background-color: #FF0000; height: 50px; width: 100px;\">ON</button></a><br/>";
const String HtmlButton1_OFF = "<a href=\"LED1On\"><button style=\"display: block; background-color: #00FF00; height: 50px; width: 100px;\">OFF</button></a><br/>";
const String HtmlButton2_ON = "<a href=\"LED2Off\"><button style=\"display: block; background-color: #FF0000; height: 50px; width: 100px;\">ON</button></a><br/>";
const String HtmlButton2_OFF = "<a href=\"LED2On\"><button style=\"display: block; background-color: #00FF00; height: 50px; width: 100px;\">OFF</button></a><br/>";

//String HtmlButtons; = "<a href=\"LEDOff\"><button style=\"display: block; background-color: #00FF00; height: 50px; width: 100px;\">OFF</button></a><br/>";
String HtmlLedState1, HtmlButtons1, HtmlLedState2, HtmlButtons2;

void handleRoot() {
  output1_OFF;
  HtmlLedState1 = "<big>LED is now <b>OFF</b></big><br/>\n";
  HtmlButtons1 = HtmlButton1_OFF;
  output2_OFF;
  HtmlLedState2 = "<big>LED is now <b>OFF</b></big><br/>\n";
  HtmlButtons2 = HtmlButton2_OFF;
  response();
}

void handleLed1On() {
// stateLED = LOW;
  output1_ON;
  HtmlLedState1 = "<big>LED is now <b>ON</b></big><br/>\n";
  HtmlButtons1 = HtmlButton1_ON;

  response();
}

void handleLed1Off() {
// stateLED = HIGH;
  output1_OFF;
  HtmlLedState1 = "<big>LED is now <b>OFF</b></big><br/>\n";
  HtmlButtons1 = HtmlButton1_OFF;
  response();
}

void handleLed2On() {
// stateLED = LOW;
  output2_ON;
  HtmlLedState2 = "<big>LED is now <b>ON</b></big><br/>\n";
  HtmlButtons2 = HtmlButton2_ON;
  response();
}

void handleLed2Off() {
// stateLED = HIGH;
  output2_OFF;
  HtmlLedState2 = "<big>LED is now <b>OFF</b></big><br/>\n";
}

```

```

HtmlButtons2 = HtmlButton2_OFF;
response();
}

void response(){
String htmlRes = HtmlHtml + HtmlCenter;
htmlRes += HtmlTitle;
htmlRes += HtmlLedState1;
htmlRes += HtmlButtons1;
htmlRes += HtmlLedState2;
htmlRes += HtmlButtons2;
htmlRes += HtmlCloseCenter;
htmlRes += HtmlHtmlClose;

server.send (200, "text/html", htmlRes);
}

void setup() {
Serial.begin(115200);
// Initialize the output variables as outputs
pinMode(output1, OUTPUT);
pinMode(output2, OUTPUT);
// Set outputs to LOW
output1_OFF;
output2_OFF;

// Connect to Wi-Fi network with SSID and password
Serial.print("Connecting to ");
Serial.println(ssid);
WiFi.begin(ssid, password);
while (WiFi.status() != WL_CONNECTED) {
  delay(500);
  Serial.print(".");
}
// Print local IP address and start web server
Serial.println("");
Serial.println("WiFi connected.");
Serial.println("IP address: ");
Serial.println(WiFi.localIP());

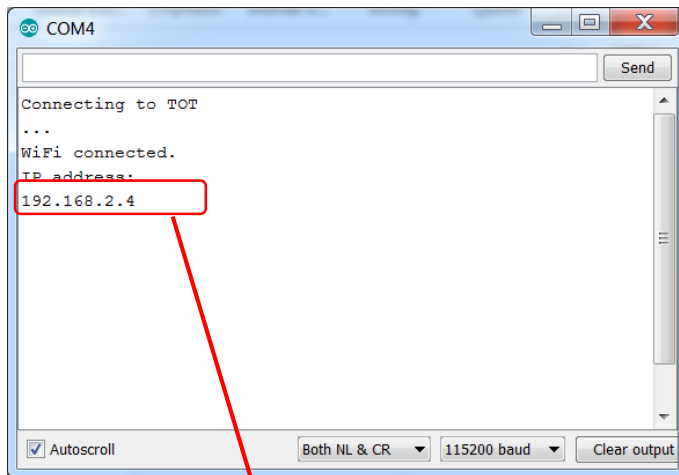
server.on("/", handleRoot);
server.on("/LED1On", handleLed1On);
server.on("/LED1Off", handleLed1Off);
server.on("/LED2On", handleLed2On);
server.on("/LED2Off", handleLed2Off);
server.begin();
}

void loop(){
  server.handleClient();
}

```

การทดสอบ

1. เมื่อแปลและ Upload ลงไปบน ESP8266 NodeMCU ให้เปิด Serial Monitor เพื่อตรวจดู IP address ของ Server ที่สร้างโดย ESP8266 NodeMCU (อย่าลืมใช้ Baud Rate 115200) ผลลัพธ์ควรได้ดังรูป



2. เปิดเว็บเบราว์เซอร์ แล้วเปิดหน้าเพจ โดย IP ที่ได้จากข้อ 1 ผลลัพธ์ควรได้ดังรูป เมื่อคลิกเมาส์ที่ปุ่มใดปุ่มหนึ่ง หน้าเว็บจะและ LED มีการเปลี่ยนแปลง

