

### การทดลองที่ 3 แสดงการควบคุมการติดดับของ LED โดยใช้เว็บเซิร์ฟเวอร์ (Web Server)

#### หมายเหตุ

ควรศึกษาตั้งแต่หัวข้อมี 14.1 ถึง 14.5

ฟังก์ชันสำหรับการรับค่าจาก web Client เมื่อมีการเรียกหน้าเว็บ

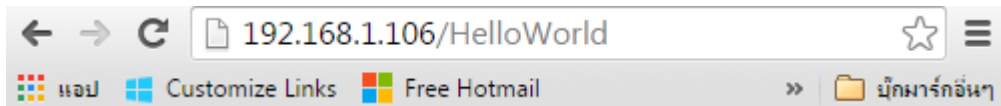
#### ฟังก์ชัน `client.readStringUntil('\r')`

เป็น Function สำหรับอ่านค่าสตริงที่ได้รับจากclient จากข้อมูลตัวแรกถึง '\r' การกระทำนี้จะเกิดเมื่อเรียกเปิดหน้าเว็บ

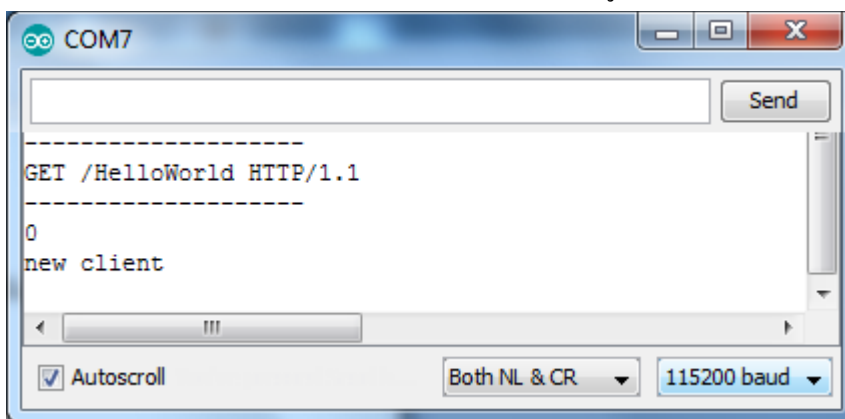
ตัวอย่าง

```
String req = client.readStringUntil('\r');  
Serial.println("-----");  
Serial.println(req);  
Serial.println("-----");
```

เมื่อเรียกหน้าเว็บด้วยค่า IPและตามด้วย /HelloWorld



ค่าที่รับได้จะเป็น “GET /HelloWorld HTTP/1.1” ตามรูป



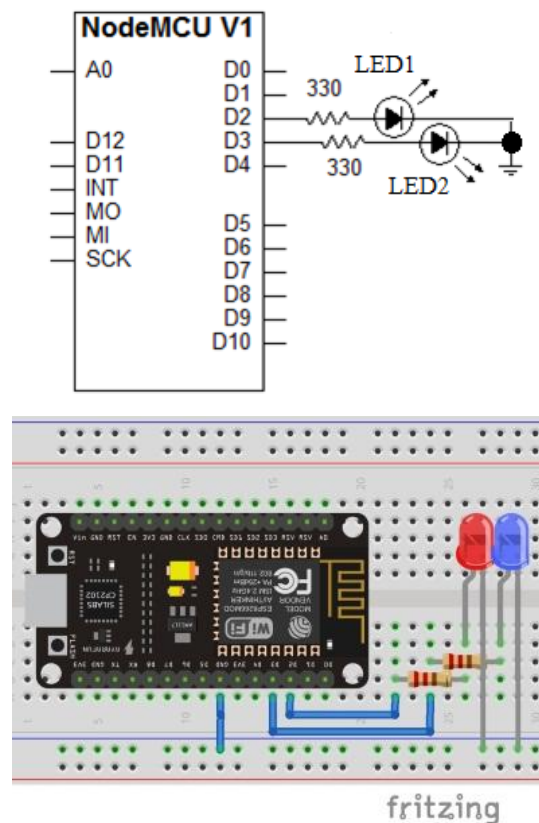
ในตัวอย่างนี้เลข 1 ถึง 4 แสดงการควบคุม LED 2 ดวง โดยจะพิมพ์เพียง <http://192.168.1.106/1> (เลข 1 เปลี่ยนได้ตั้งแต่ 1 ถึง 4) แล้วจะใช้ฟังก์ชัน `substring()` และฟังก์ชัน `toInt()` จำแนกตัวเลขเอามาใช้งานต่อไป

ตัวอย่าง

```
status = req.substring(5,6).toInt();
```

↓ แปลงสตริงให้เป็นเลขจำนวนเต็ม  
↓ คัดลอกสตริงตัวที่ 5 ถึง 6 (เอามาตัวเดียว)  
↓ สตริงตัวเต็ม

วงจร



โปรแกรม

```
#include <ESP8266WiFi.h>
const char* ssid = "....."; //อย่าลืมแก้เป็นชื่อ SSID ของตัวเอง
const char* password = "....."; //อย่าลืมแก้เป็นชื่อ password ของตัวเอง
WiFiServer server(80); //กำหนดใช้งาน TCP Server ที่ Port 80
//-----
#define LED1 2
#define LED2 3
#define LED1_ON digitalWrite(LED1,HIGH)
#define LED1_OFF digitalWrite(LED1,LOW)
#define LED2_ON digitalWrite(LED2,HIGH)
#define LED2_OFF digitalWrite(LED2,LOW)
int status;
//-----
void setup()
{
  pinMode(LED1, OUTPUT); //กำหนด Pin ที่ต่อกับ LED เป็น Output
  pinMode(LED2, OUTPUT); //กำหนด Pin ที่ต่อกับ LED เป็น Output
```

```

//-----
Serial.begin(115200);    //ตั้งค่าใช้งาน serial ที่ baudrate 115200
delay(10);
Serial.println();
Serial.println();
Serial.print("Connecting to "); //แสดงข้อความ "Connecting to"
Serial.println(ssid);        //แสดงข้อความ ชื่อ SSID
WiFi.begin(ssid, password); // เชื่อมต่อไปยัง AP

while (WiFi.status() != WL_CONNECTED) //รอจนกว่าจะเชื่อมต่อสำเร็จ
{
    delay(500);
    Serial.print(".");
}
Serial.println("");
Serial.println("WiFi connected"); //แสดงข้อความเชื่อมต่อสำเร็จ
server.begin();                //เปิด TCP Server
Serial.println("Server started");
Serial.println(WiFi.localIP()); //แสดงหมายเลข IP ของ ESP8266(DHCP)
}

void loop() {
    WiFiClient client = server.available(); //รอรับ การเชื่อมต่อจาก Client
    if (!client) {                      //ถ้าไม่มี Client เข้ามาให้เริ่มกับไปวน loop รอรับใหม่
        return;
    }

    Serial.println("new client");
    while(!client.available())
    {
        delay(1);
    }
    String req = client.readStringUntil('\r'); //อ่านค่าที่ได้รับจากclient จากข้อมูลแรกถึง '\r'
    Serial.println("-----");
    Serial.println(req);                    //แสดงค่าที่ได้รับทาง Serial
    Serial.println("-----");
    status = req.substring(5,6).toInt();
    Serial.println (status);

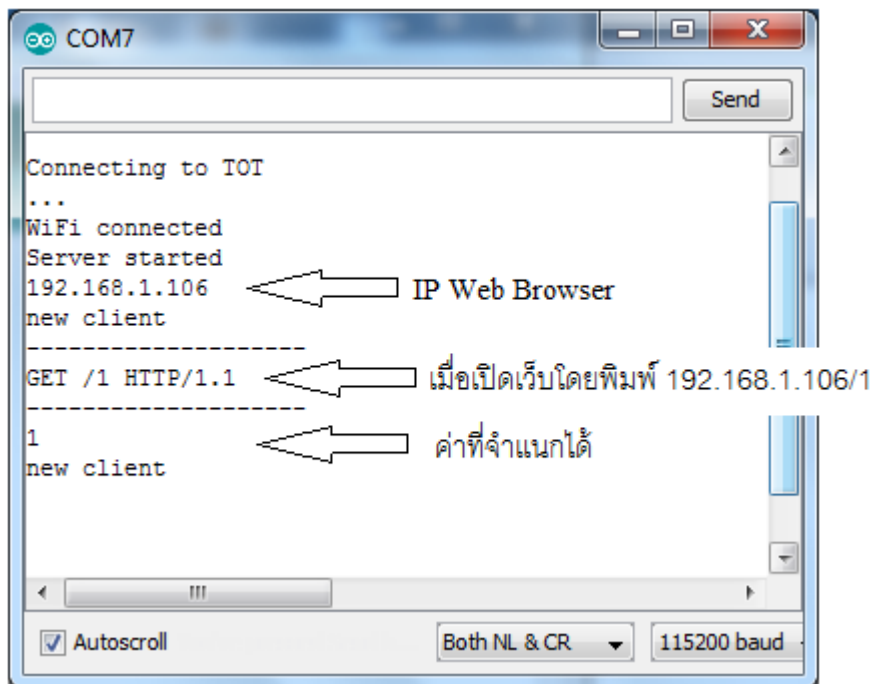
    client.flush();

    client.println("HTTP/1.1 200 OK\r\nContent-Type: text/html");
    client.println("");
    client.println("<html>");
    client.println("<title>Web Server Page Test</title>");
    client.println("</head>");
    client.println("<body>");
    client.println("<h1>EI444 Microprocessors in Automation System</h1>");
    client.println("<p>");
    client.println("<h2>Arduino NodeMCU ESP8266</h2>");
    client.println("Lab 3 Network and Web Server control<br>");
    client.println("</p>");
    switch (status) {
        case 0: LED1_OFF; client.println("LED 1 OFF<br>");break;
        case 1: LED1_ON; client.println("LED 1 ON<br>");break;
        case 2: LED2_OFF; client.println("LED 2 OFF<br>");break;
        case 3: LED2_ON; client.println("LED 2 ON<br>");break;
        default: break;
    }
    client.println("</body>");
    client.println("</html>");
}

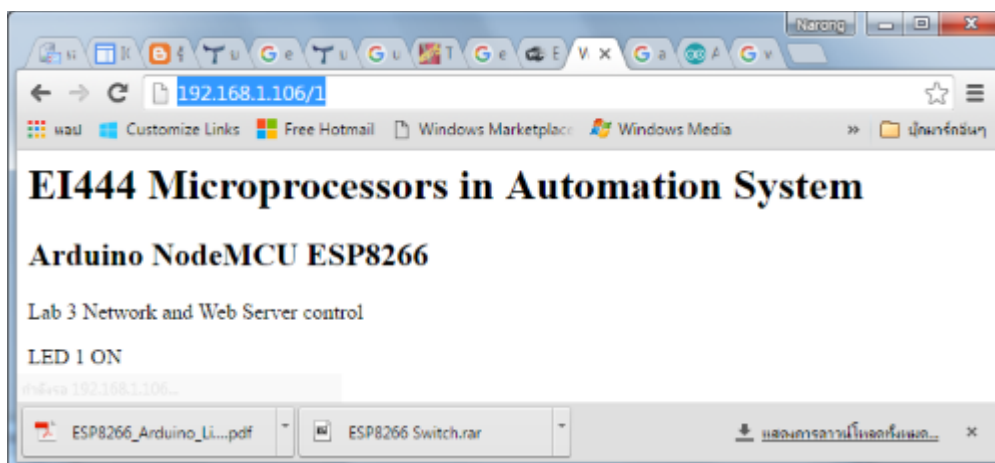
```

เมื่อแปลและ Upload ลงไปบน NodeMCU

1. ให้เปิด Serial Monitor เพื่อตรวจสอบผลการทำงาน โดยใช้ Baud Rate 115200
2. ให้เปิด web browser แล้วพิมพ์ IP ของเว็บ



3. เปิดเว็บเบราว์เซอร์ ตาม IP ที่ได้ จะได้หน้าเว็บตามนี้ พร้อมกับ LED1 ติด LED2 ดับ



ศึกษาข้อมูลเพิ่มเติมได้จาก

<http://thaieasyelec.com/article-wiki/embedded-electronics-application/getting-started-with-esp8266-nodemcu-ch5.html>