

การทดลองที่ 2 การเชื่อมต่อ NodeMCU เข้ากับระบบ Network และการสร้างเว็บเซิร์ฟเวอร์ (Web Server)

ฟังก์ชันสำหรับเชื่อมต่อกับระบบเน็ตเวิร์ค

ฟังก์ชัน void mode(WiFiMode)

เป็น Function สำหรับเลือกโหมดการทำงานให้ ESP8266 โดยมีอยู่ด้วยกัน 3 โหมด ได้แก่ Station, Access Point, Station และ Access Point

Parameter

WIFI_STA = สั่งให้ทำงานเป็น Station

WIFI_AP = สั่งให้ทำงานเป็น Access Point

WIFI_AP_STA = สั่งให้ทำงานเป็นทั้ง Station และ Access Point

ใน IEEE 802.11 (Wi-Fi) คำว่า station (STA) หมายถึงอุปกรณ์ที่สามารถใช้โปรโตคอล 802.11 ได้ เช่น เครื่องคอมพิวเตอร์แบบตั้งโต๊ะ เครื่องคอมพิวเตอร์แบบ โน้ตบุ๊ก แท็บเล็ต และ สมาร์ทโฟน เป็นต้น ซึ่งอุปกรณ์เหล่านี้สามารถเป็นได้ทั้งตัวรับและตัวส่งสัญญาณ

ส่วนคำว่า Access Point หรือ AP หรือ a wireless access point (WAP) หมายถึงอุปกรณ์ทางฮาร์ดแวร์ที่ยอมให้อุปกรณ์อื่นๆเชื่อมต่อกับระบบเน็ตเวิร์คผ่านทาง WiFi โดยปกติ AP ต่ออยู่กับเราเตอร์ (router เป็นอุปกรณ์คอมพิวเตอร์ที่ทำหน้าที่หาเส้นทางและส่ง(forward)แพ็กเก็ตข้อมูลระหว่างเครือข่ายคอมพิวเตอร์ ไปยังเครือข่ายปลายทางที่ต้องการ)

ฟังก์ชัน int begin(const char* ssid, const char *passphrase)

เป็น Function สำหรับเชื่อมต่อ ESP8266 เข้ากับ AP โดยหากใช้งาน Function เพียงอย่างเดียว จะเป็นการใช้งาน DHCP โดย ESP8266 จะได้ IP ตามที่ AP เป็นคนจัดสรรให้ Parameter

ssid = ชื่อของ AP ที่ต้องการ join

passphrase = password สำหรับ join เข้า AP

ในกรณีที่ AP ที่ต้องการเชื่อมต่อเป็นแบบ open (ไม่มี password) ให้ใส่แค่ ssid อย่างเดียว เช่น

```
WiFi.begin("wifi_name");
```

แบบที่มี Password

```
WiFi.begin("wifi_name","pass1234");
```

แต่หากต้องการใช้งานเป็น **static IP** ต้องใช้งานร่วมกับ Function config ในคำสั่งต่อไป

ฟังก์ชัน void config(IPAddress local_ip, IPAddress gateway, IPAddress subnet)

เป็น Function สำหรับตั้งค่า IP , gateway , subnet ให้กับ ESP8266 ในกรณีที่ต้องการใช้งานเป็นแบบ Static IP

Parameter

local_ip = สำหรับตั้ง IP

gateway = สำหรับตั้ง gateway

subnet = สำหรับตั้ง subnet

ตัวอย่างการใช้งาน

```
IPAddress local_ip = {192,168,1,144};
```

```
IPAddress gateway = {192,168,1,1};
```

```
IPAddress subnet = {255,255,255,0};
```

```
WiFi.config(local_ip,gateway,subnet);
```

ฟังก์ชัน int disconnect(void)

เป็น Function สำหรับตัดการเชื่อมต่อกับ AP

ฟังก์ชันนี้เป็น Function สำหรับเชื่อมต่อ ESP8266 เข้ากับ AP โดยหากใช้งาน Function เพียงอย่างเดียว จะเป็นการใช้งาน DHCP โดย ESP8266 จะได้ IP ตามที่ AP เป็นคนจัดสรรให้ หากต้องการใช้งานเป็น static IP ต้องใช้งานร่วมกับ Function config ในคำสั่งต่อไป

Parameter

ssid = ชื่อของ AP ที่ต้องการ join

passphrase = password สำหรับ join เข้า AP

ในกรณีที่ AP ที่ต้องการเชื่อมต่อเป็นแบบ open (ไม่มี password) ให้ใส่แค่ ssid อย่างเดียว เช่น

```
WiFi.begin("ThaiEasyElec");
```

แบบที่มี Password

```
WiFi.begin("ThaiEasyElec","passwordxxxx");
```

ฟังก์ชัน WiFi.status() ใช้ตรวจสอบการเชื่อมต่อกับ AP ว่าได้หรือไม่

โปรแกรม

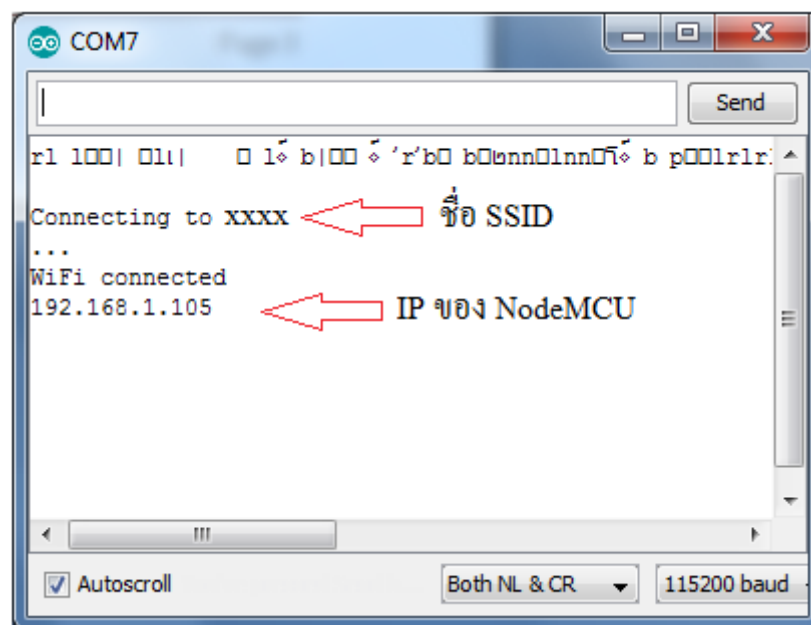
```
#include <ESP8266WiFi.h>           //เรียกใช้ไลบรารี
const char* ssid    = ".....";   //อย่าลืมแก้เป็นชื่อ SSID ของตัวเอง
const char* password = "....."; //อย่าลืมแก้เป็นชื่อ password ของตัวเอง

void setup()
{
  Serial.begin(115200);           //ตั้งค่าใช้งาน serial ที่ baudrate 115200
  delay(10);
  Serial.println();              //ส่งข้อมูลออกทางพอร์ตอนุกรม(Serial monitor)
  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) //รอนจนกว่าจะเชื่อมต่อสำเร็จ
  {
    delay(500);
    Serial.print(".");
  }
  Serial.println("");
  Serial.println("WiFi connected"); //แสดงข้อความเชื่อมต่อสำเร็จ
  Serial.println(WiFi.localIP());   //แสดงหมายเลข IP ของ ESP8266(DHCP)
}

void loop() {
}
```

เมื่อแปลและ Upload ลงไปบน NodeMCU ให้เปิด Serial Monitor เพื่อตรวจสอบผลการทำงาน โดยใช้ Baud Rate 115200 ถ้าเชื่อมต่อกับ AP ได้จะได้ผลลัพธ์ดังรูป



โปรแกรมที่ 2 ใช้ NodeMCU ทำหน้าที่เป็นเว็บเซิร์ฟเวอร์ ส่งหน้าเว็บไปที่เว็บไคลเอ็นท์ (Web Client) ด้วย html code

```
#include <ESP8266WiFi.h>
const char* ssid    = "wifi_name";           //อย่าลืมแก้เป็นชื่อ SSID ของตัวเอง
const char* password = "wifi_password";     //อย่าลืมแก้เป็นชื่อ password ของตัวเอง
WiFiServer server(80);                       //กำหนดใช้งาน TCP Server ที่ Port 80

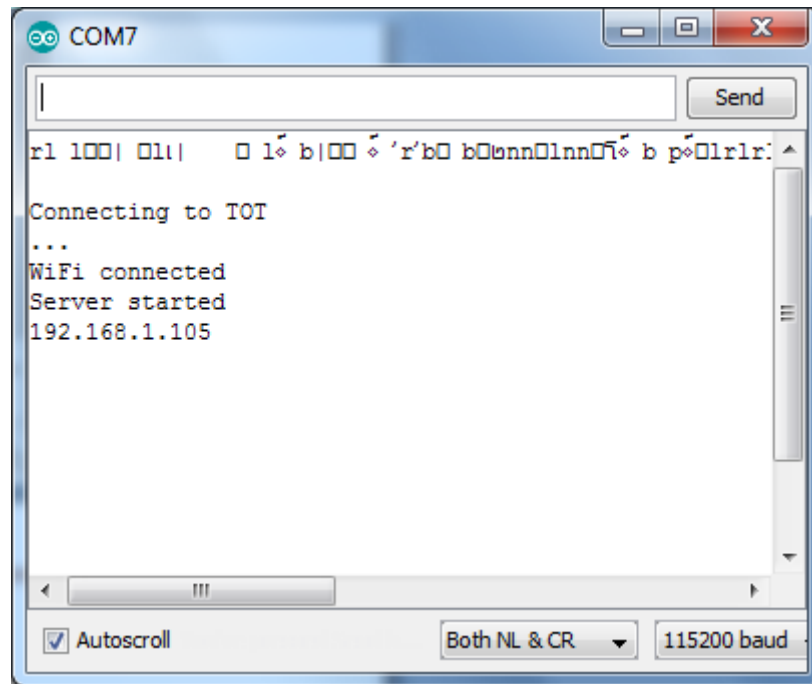
void setup()
{
  Serial.begin(115200);                       //ตั้งค่าใช้งาน serial ที่ baudrate 115200
  delay(10);
  Serial.println();
  Serial.println();
  Serial.print("Connecting to ");           //แสดงข้อความ "Connecting to"
  Serial.println(ssid);                     //แสดงข้อความ ชื่อ SSID
 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

  while (WiFi.status() != WL_CONNECTED)     //รอนจนกว่าจะเชื่อมต่อสำเร็จ
  {
    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);
  }
  client.flush();
  //----- print HTML Code to Client -----//
  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("</p>");
  client.println("</body>");
  client.println("</html>");
}
```

เมื่อแปลและ Upload ลงไปบน NodeMCU ให้เปิด Serial Monitor เพื่อตรวจสอบผลการทำงาน โดยใช้ Baud Rate 115200 ถ้าเชื่อมต่อกับ AP ได้จะได้ผลลัพธ์ดังรูป



เปิดเว็บเบราว์เซอร์ ตาม IP ที่ได้ จะได้หน้าเว็บตามนี้

