

อ่านค่า ADC แล้วส่งออกทาง server

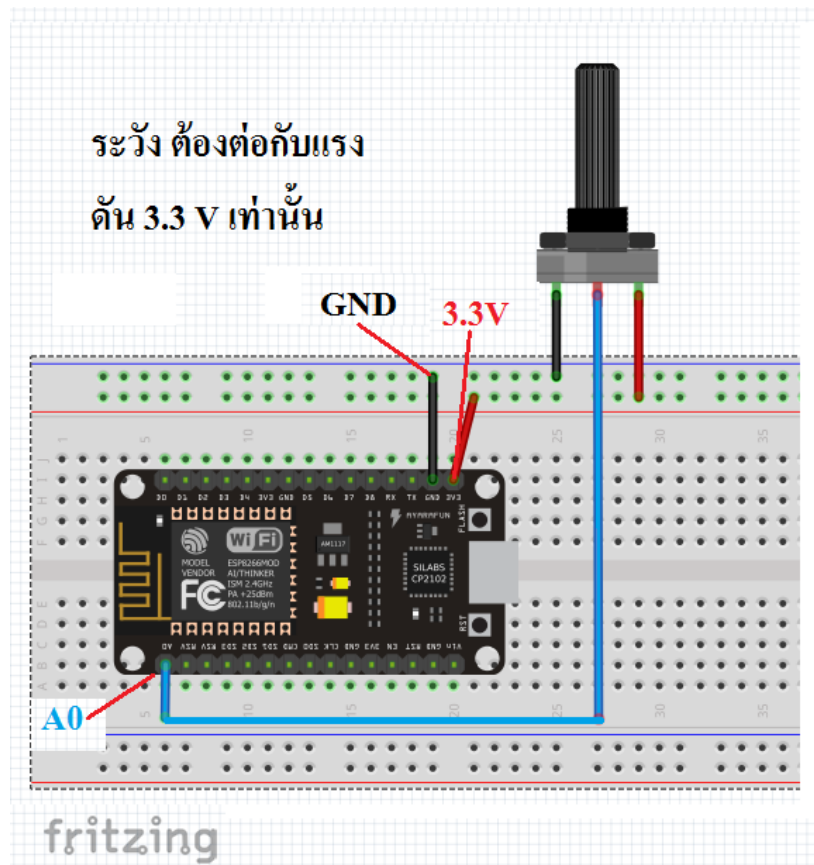
วัตถุประสงค์

สร้าง Server เพื่ออ่านค่าจากพอร์ทอนาล็อก แสดงผลทางเว็บไซต์ด้วยวิธีการรีเฟรชหน้าเว็บ (Refresh)

อุปกรณ์

1. ESP8266 NodeMCU
2. ความต้านทานปรับค่าได้ (Potentiometer)
3. บอร์ดทดลอง (Breadboard)

วงจร



โปรแกรมแบบที่ 1 ให้ ESP8266 NodeMCU ทำหน้าที่เป็นเว็บเซิร์ฟเวอร์ ส่งค่าอนาล็อกที่อ่านได้ออกเป็นหน้าเว็บไปที่เว็บไคลเอนท์ (Web Client) ด้วย html code โดยให้หน้าเว็บทำงานแบบ Auto refresh การรีเฟลช และระยะเวลาการรีเฟลช กำหนดด้วย `<meta http-equiv='refresh' content='3'/>` หน่วยเป็นวินาที ตามโค้ดนี้เป็น 3 วินาที ต่อครั้ง

โปรแกรม `esp_server_adc.ino`

```
#include <WiFiClient.h>
#include <ESP8266WebServer.h>

#define ssid      "....."      // WiFi SSID
#define password  "....."      // WiFi password

int sensorValue;
float volts;
String volt_string;
ESP8266WebServer server ( 80 );

String getPage(){      //HTML Code
  String page = "<html><head><meta http-equiv='refresh' content='3'/>";
  page += "<title>ESP8266 ADC</title>";
  page += "<style> body { background-color: #ffffff; font-family: Arial, Helvetica, Sans-Serif; Color: #000088; }</style>";
  page += "</head><body><h1>ESP8266  ADC on Server</h1>";
  page += "<h3>ADC0 10 bit</h3>";
  page += "Sensor Value : ";
  page += sensorValue;
  page += ", Voltage = ";
  page += volts;
  page += " V";
  page += "</body></html>";
  return page;
}

void handleRoot(){
  server.send ( 200, "text/html", getPage() );
}

void setup() {
  Serial.begin ( 115200 );

  WiFi.begin ( ssid, password );
  // Attente de la connexion au réseau WiFi / Wait for connection
  while ( WiFi.status() != WL_CONNECTED ) {
    delay ( 500 ); Serial.print ( "." );
  }
  // Connexion WiFi établie / WiFi connexion is OK
  Serial.println ( "" );
  Serial.print ( "Connected to " ); Serial.println ( ssid );
  Serial.print ( "IP address: " ); Serial.println ( WiFi.localIP() );

  // On branche la fonction qui gère la première page / link to the function that manage launch page
  server.on ( "/", handleRoot );

  server.begin();
}
```

```

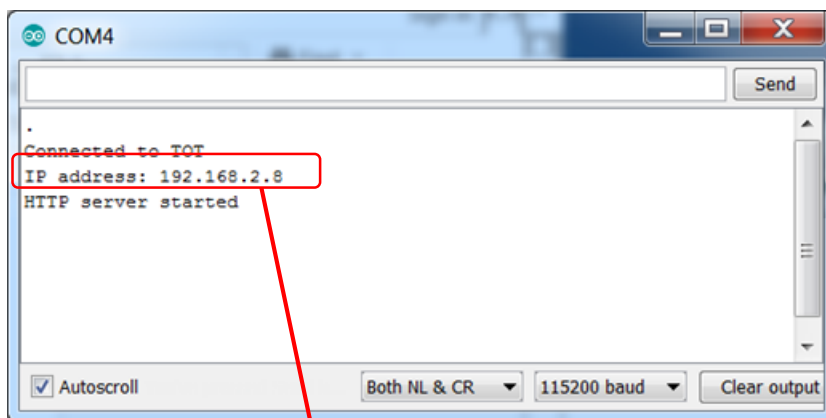
Serial.println ( "HTTP server started" );
}

void loop() {
  server.handleClient();
  sensorValue = analogRead(A0);
  volts = 3.30*(float)sensorValue/1023.00;
  delay(1000);
}

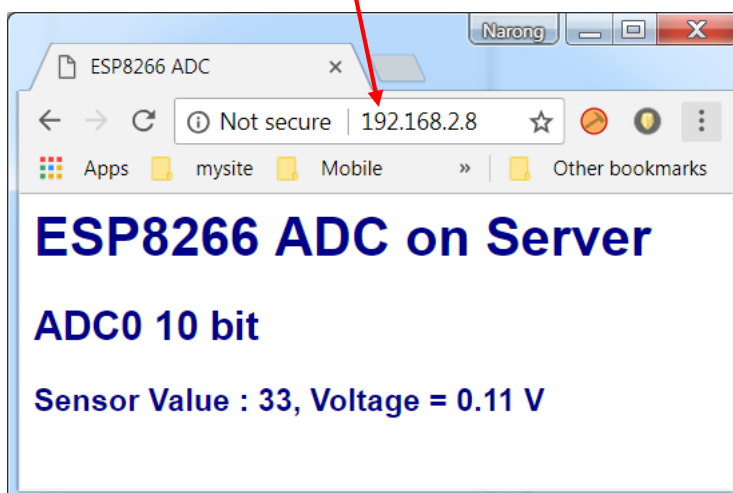
```

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 แล้วเปิดหน้าเพจโดย IP ที่ได้จากข้อ 1 ผลลัพธ์ควรได้ดังรูป เมื่อปรับค่าความต้านทาน ค่า Sensor และ Voltage จะเปลี่ยนตามการปรับและทุกครั้งที่หน้าเพจรีเฟรช



โปรแกรมแบบที่ 2 ให้ ESP8266 NodeMCU ทำหน้าที่เป็น Station และ Access Point (STA&AP) แล้วทำตัวเป็นเว็บเซิร์ฟเวอร์ ส่งค่าอนาล็อกที่อ่านได้ออกเป็นหน้าเว็บไปที่เว็บไคลเอนท์ (Web Client) ด้วย html code และให้หน้าเว็บทำงานแบบ Auto refresh เหมือนแบบที่ 1

esp_ap_server_adc.ino

```
#include <WiFiClient.h>
#include <ESP8266WebServer.h>

const char *ssid = "NodeAPtest";

int sensorValue;
float volts;
String volt_string;
ESP8266WebServer server ( 80 );

String getPage(){
  String page = "<html><head><meta http-equiv='refresh' content='5'/>";
  page += "<title>ESP8266 ADC</title>";
  page += "<style> body { background-color: #ffff; font-family: Arial, Helvetica, Sans-Serif; Color: #000088; }</style>";
  page += "</head><body><h1>ESP8266  ADC on Server</h1>";
  page += "<h2>ADC0 10 bit</h2>";
  page += "<h3>Sensor Value : ";
  page += sensorValue;
  page += ", Voltage = ";
  page += volts;
  page += " V</h3>";
  page += "</body></html>";
  return page;
}

void handleRoot(){
  server.send ( 200, "text/html", getPage() );
}

void setup() {
  Serial.begin ( 115200 );

  WiFi.mode(WIFI_AP);
  WiFi.softAP(ssid);
  IPAddress apip = WiFi.softAPIP();
  Serial.print("visit: \n");
  Serial.println(apip);
  server.begin();
  Serial.println ( "HTTP server started" );
}

void loop() {
  server.handleClient();
  sensorValue = analogRead(A0);
  volts = 3.30*(float)sensorValue/1023.00;
  delay(1000);
}
```

การทดสอบ

1. แอปและ Upload ลงไปบน ESP8266 NodeMCU ให้เปิด Serial Monitor เพื่อตรวจสอบ IP address ของ Server ที่สร้างโดย ESP8266 NodeMCU
2. ที่อุปกรณ์ไคลเอ็นท์ (อุปกรณ์ที่จะใช้เปิดหน้าเว็บ เช่น โทรศัพท์มือถือ) ให้ตั้งค่าสัญญาณไวไฟล์เป็นตามชื่อ ssid ในโปรแกรม เช่น NodeAPtest แล้วทำตามข้อ 2 ในแบบแรก จะได้ผลการทำงานเหมือนเดิม

