

อ่านค่า ADC แล้วส่งออกทาง server

วัตถุประสงค์

- รู้จักการเพิ่มเครื่องมือใน Arduino IDE
- เข้าใจถึงการใช้งาน ESP8266FS
- เก็บภาพ (jpg, png)] ลงในหน้าเว็บที่ได้จาก ESP8266

อุปกรณ์

1. ESP8266 NodeMCU
2. บอร์ดทดลอง (Breadboard)

วงจร

ประกอบ ESP8266 NodeMCU ลงบอร์ด แล้วต่อสาย USB เข้ากับคอมพิวเตอร์

การทดลองโปรแกรมแสดงภาพบนเว็บเพจด้วยภาษา HTML

โปรแกรมที่ 1 ทดลองเขียน HTML เพื่อแสดงรูปภาพออกทาง เว็บเบราว์เซอร์ ดังนี้

โปรแกรม *index.html*

```
<!DOCTYPE html>
<html>
<body>

<p>This example shows image loading from ESP8266 Web server,<br>
Image and HTML web page files are uploaded in ESP Flash using SPIFFS tool:</p>

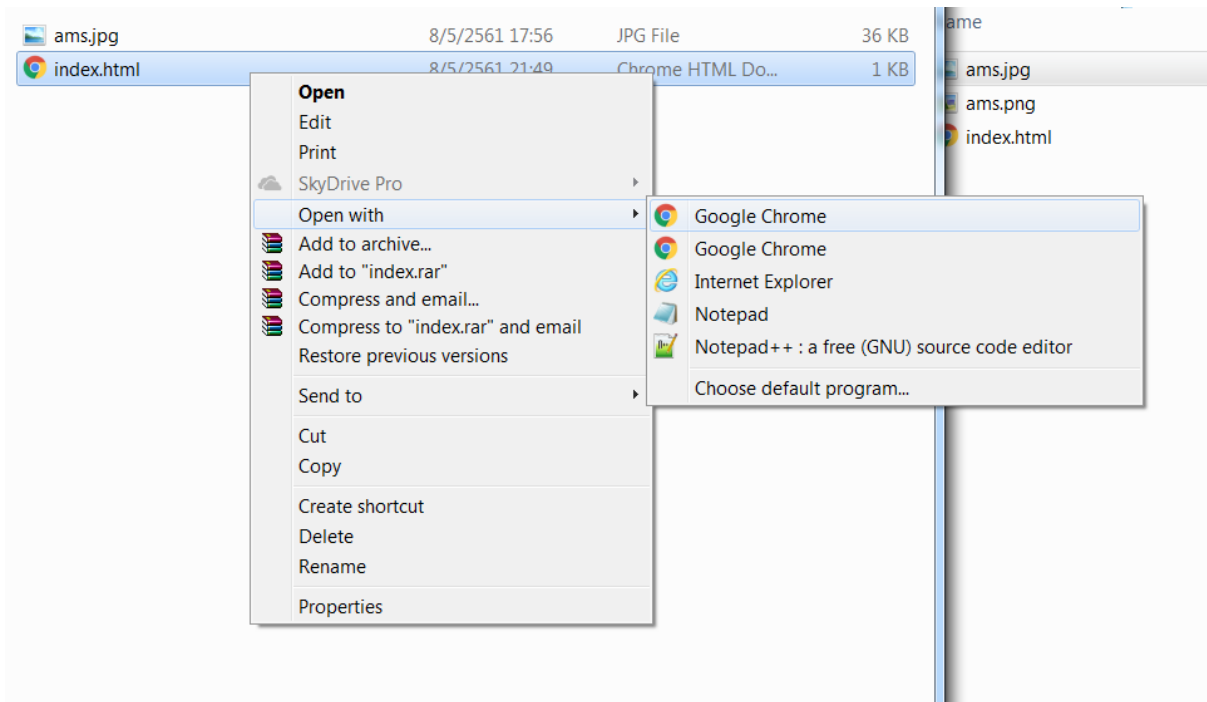


</body>
</html>
```

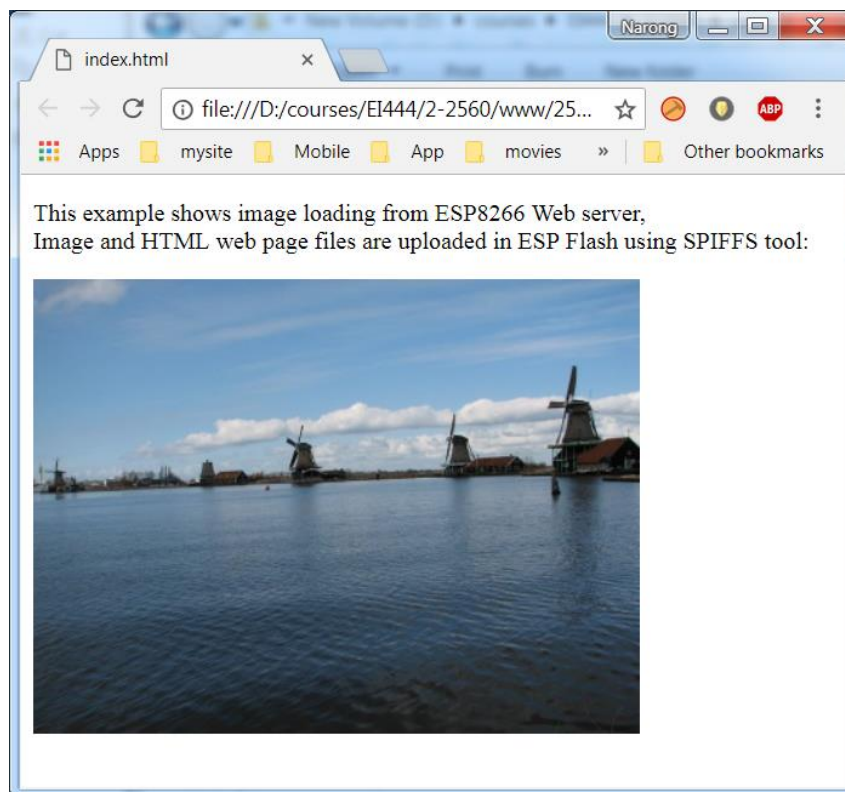
บันทึกโปรแกรมเป็นชื่อ *index.html* ไว้ที่ไดรฟ์ และบันทึกไฟล์ภาพชื่อ *ans.jpg* ไว้ที่โฟลเดอร์เดียวกัน

การทดสอบ

ให้เปิดไฟล์ *index.html* ด้วยเว็บเบราว์เซอร์ โดยการคลิกเมาส์ปุ่มขวา แล้วเลือกเปิดด้วยเว็บเบราว์เซอร์
อะไรก็ได้ เช่นตามรูปที่ 1 เปิดด้วย Google Chrome ผลลัพธ์ควรได้ดังรูปที่ 2



รูปที่ 1 วิธีเปิดไฟล์ .html ด้วยการคลิกขวา



รูปที่ 2 ผลการทำงานของโปรแกรม index.html

หมายเหตุ ใช้ภาพอะไรก็ได้ แต่ไม่ควรมีขนาดใหญ่ เพราะ esp8266 NodeMcu มีขนาดหน่วยความจำไม่มาก (ตรวจสอบได้จาก datasheet) และให้เปลี่ยนชื่อไฟล์เป็น ams.jpg (ที่จริงสามารถชื่ออะไรก็ได้ แต่ต้องแก้ชื่อไฟล์ในโปรแกรมให้ตรงกัน และเป็น jpg หรือ png)

การเพิ่ม เครื่องมือ ESP8266FS ลงใน Arduino IDE

ESP8266FS เป็นเครื่องมือที่ใช้งานร่วมกับ Arduino IDE เพื่อใช้ในการอัปเดตข้อมูลของ sketch ที่อยู่ในโฟลเดอร์ Data เพื่อเก็บไว้ใน ไฟล์ระบบบน flash

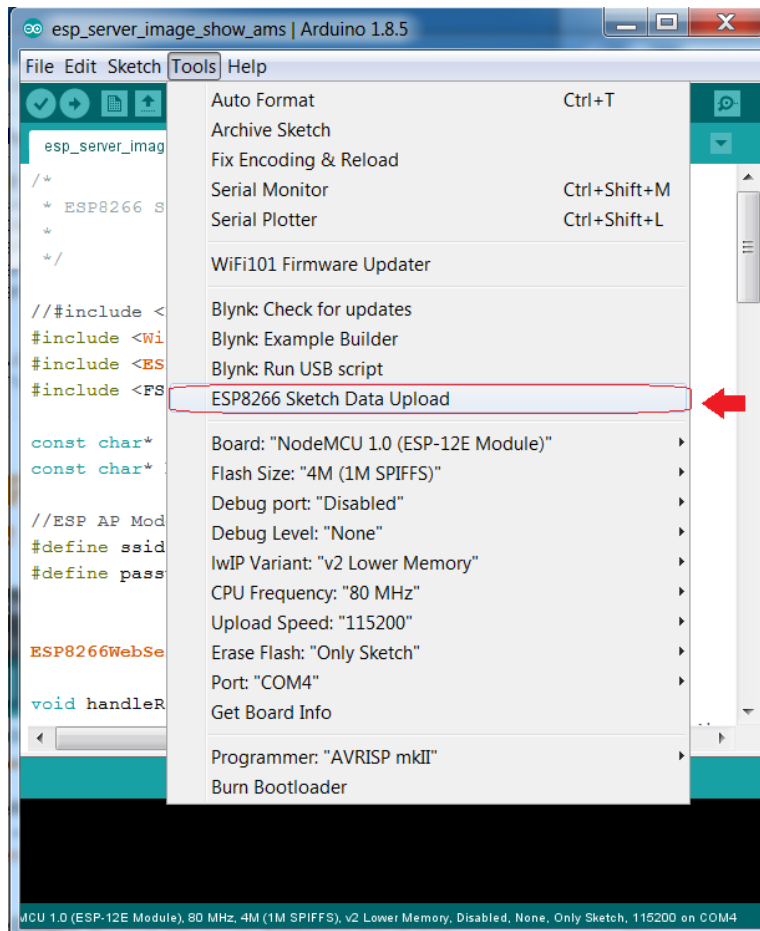
ให้ดาวน์โหลดโปรแกรม ESP8266FS-0.1.3.zip มาจาก <https://github.com/esp8266/arduino-esp8266fs-plugin/releases/download/0.1.3/ESP8266FS-0.1.3.zip>.

แล้วขยายไฟล์ที่ได้โฟลเดอร์ ESP8266FS -> tool -> esp8266fs.jar นำโฟลเดอร์ ESP8266FS นี้ไปไว้ในโฟลเดอร์ชื่อ tools ซึ่งอยู่ภายในโฟลเดอร์ Sketch ของ arduino sketch เพื่อเก็บแล้วจะอยู่ในตำแหน่งคล้ายๆรูปที่ 3



รูปที่ 3

มาถึงตอนนี้สามารถเรียกใช้ ESP8266FS ได้แล้ว โดยเปิด arduino IDE (ถ้าเปิดอยู่ก่อนให้ปิดแล้วเปิดใหม่) แล้วเข้าเมนู Tools จะปรากฏคำสั่ง ESP8266 Sketch Data Upload พร้อมใช้งาน ตามรูปที่ 4



รูปที่ 4

การทดลอง สร้างเว็บเพจเพื่อให้เห็นภาพ

ให้ ESP8266 NodeMCU ทำหน้าที่เป็นเว็บเซิร์ฟเวอร์สำหรับส่งหน้าเว็บเพจเพื่อแสดงภาพ ในการทดลองนี้ จะทดลองโปรแกรม 2 แบบ แบบแรกให้ ESP8266 NodeMCU ทำหน้าที่เป็นเว็บเซิร์ฟเวอร์อย่างเดียว ส่วนแบบที่ 2 ทำหน้าที่เป็น Access Point ด้วย

โปรแกรมแบบที่ 1 ให้ ESP8266 NodeMCU ทำหน้าที่เป็นเว็บเซิร์ฟเวอร์ อย่างเดียว

ขั้นตอนที่ 1 เขียนโปรแกรม

โปรแกรม *esp_server_image_show_ams.ino*

```
#include <WiFiClient.h>
#include <ESP8266WebServer.h>
#include <FS.h> //Include File System Headers

const char* imagefile = "/ams.jpg";
const char* htmlfile = "/index.html";

//ESP AP Mode configuration
#define ssid "..." // WiFi SSID
#define password "..." // WiFi password
```

```

ESP8266WebServer server(80);

void handleRoot(){
  server.sendHeader("Location", "/index.html",true); //Redirect to our html web page
  server.send(302, "text/plain","");
}

void handleWebRequests(){
  if(loadFromSpiffs(server.uri())) return;
  String message = "File Not Detected\n\n";
  message += "URI: ";
  message += server.uri();
  message += "\nMethod: ";
  message += (server.method() == HTTP_GET)?"GET":"POST";
  message += "\nArguments: ";
  message += server.args();
  message += "\n";
  for (uint8_t i=0; i<server.args(); i++){
    message += " NAME:"+server.argName(i) + "\n VALUE:" + server.arg(i) + "\n";
  }
  server.send(404, "text/plain", message);
  Serial.println(message);
}

void setup() {
  delay(1000);
  Serial.begin(115200);
  WiFi.begin(ssid, password);
  while ( WiFi.status() != WL_CONNECTED ) {
    delay ( 500 ); Serial.print ( "." );
  }
  // WiFi connecton is OK
  Serial.println ( "" );
  Serial.print ( "Connected to " ); Serial.println ( ssid );
  Serial.print ( "IP address: " ); Serial.println ( WiFi.localIP() );

  //Initialize File System
  SPIFFS.begin();
  Serial.println("File System Initialized");

  //Initialize Webserver
  server.on("/",handleRoot);
  server.onNotFound(handleWebRequests); //Set setver all paths are not found so we can handle as per URI
  server.begin();
}

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

bool loadFromSpiffs(String path){
  String dataType = "text/plain";
  if(path.endsWith("/")) path += "index.htm";

  if(path.endsWith(".src")) path = path.substring(0, path.lastIndexOf("."));
  else if(path.endsWith(".html")) dataType = "text/html";
  else if(path.endsWith(".htm")) dataType = "text/html";
  else if(path.endsWith(".css")) dataType = "text/css";
  else if(path.endsWith(".js")) dataType = "application/javascript";
  else if(path.endsWith(".png")) dataType = "image/png";
}

```

```

else if(path.endsWith(".gif")) dataType = "image/gif";
else if(path.endsWith(".jpg")) dataType = "image/jpeg";
else if(path.endsWith(".ico")) dataType = "image/x-icon";
else if(path.endsWith(".xml")) dataType = "text/xml";
else if(path.endsWith(".pdf")) dataType = "application/pdf";
else if(path.endsWith(".zip")) dataType = "application/zip";
File dataFile = SPIFFS.open(path.c_str(), "r");
if (server.hasArg("download")) dataType = "application/octet-stream";
if (server.streamFile(dataFile, dataType) != dataFile.size()) {
}

dataFile.close();
return true;
}

```

ขั้นตอนที่ 2 เตรียมข้อมูลเก็บลงโฟลเดอร์ data

เมื่อเขียนเสร็จแล้วอย่างเพิ่ง upload ลงบอร์ดให้ save เก็บลง sketch ก่อน เมื่อเซฟแล้วในโฟลเดอร์ของโปรแกรมนี้ให้สร้างโฟลเดอร์ย่อยชื่อ data ขึ้นมาแล้วนำไฟล์ภาพเก็บลงที่โฟลเดอร์ data ส่วนไฟล์ index.html ให้แก้ไขตรงบรรทัด

```



```

ให้เปลี่ยนเป็น

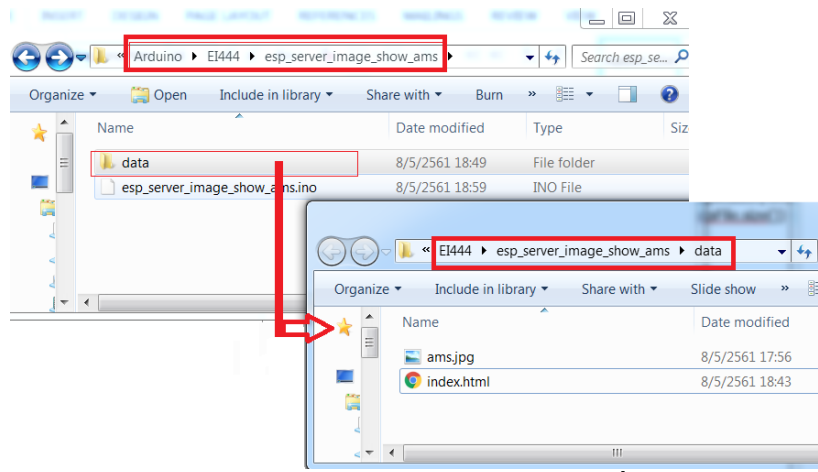
```



```

แล้วบันทึกลงในโฟลเดอร์ data ด้วย

ดังนั้นในโฟลเดอร์ data จะมีไฟล์อยู่ 2 ไฟล์ คือ ไฟล์ภาพ ams.jpg และ index.html



รูปที่ 5

ขั้นตอนที่ 3 เก็บข้อมูลของไฟล์ภาพและไฟล์ index.html ลงแฟลช ด้วย ESP8266FS

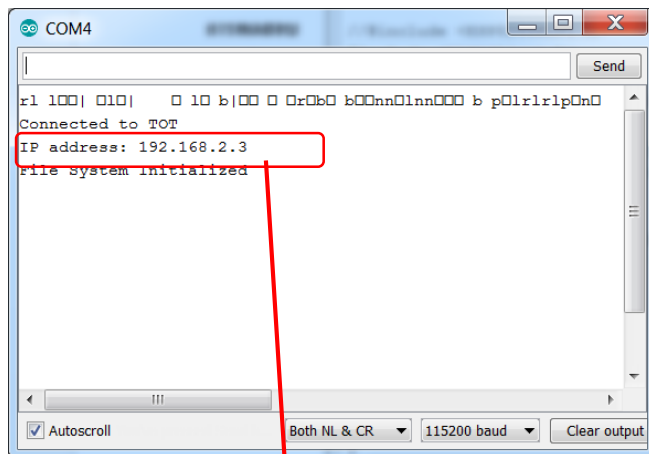
เมื่อมีไฟล์พร้อมให้กลับไปที่ arduino IDE ที่เปิดโปรแกรม esp_server_image_show_ams.ino นี้จะใช้ ESP8266FS ทำการอัปโหลดรูปภาพและไฟล์ index.html ลงแฟลช โดยคลิกคำสั่ง ESP8266 Sketch Data Upload ตามรูปที่ 4

ขั้นตอนที่ 4 แปลและอัปโหลดโปรแกรม `esp_server_image_show_ams.ino`

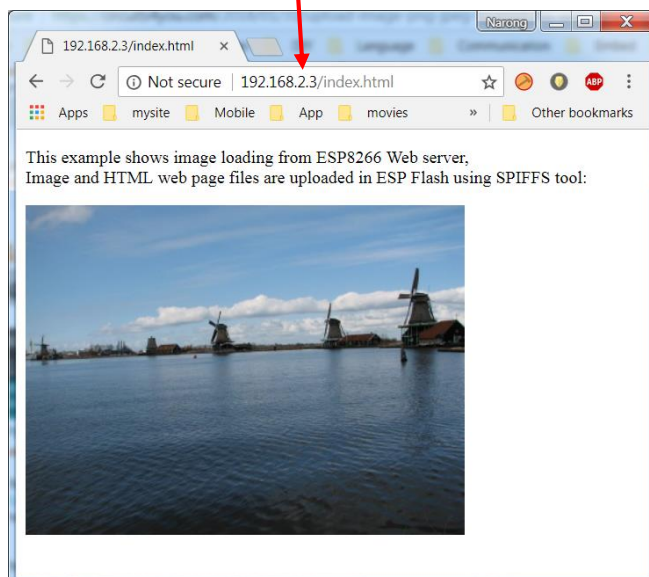
เมื่อเรียบร้อยแล้ว ให้ แปลโปรแกรม `esp_server_image_show_ams.ino` และอัปโหลด ลงไปบน ESP8266 NodeMCU

ขั้นตอนที่ 5 ทดสอบ

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 จะเปลี่ยนตามการปรับและทุกครั้งทีหน้าเพจรีเฟรช



โปรแกรมแบบที่ 2 ให้ ESP8266 NodeMCU ทำหน้าที่เป็น Access Point (STA&AP) แล้วทำตัวเป็นเว็บเซิร์ฟเวอร์ เพื่อส่งภาพ ไปที่เว็บไคลเอนท์ (Web Client) ด้วย html code

โปรแกรม *esp_ap_server_image_show_ams.ino*

```
/*
 * ESP8266 SPIFFS HTML Web Page with JPEG, PNG Image
 *
 */

#include <ESP8266WiFi.h>
#include <ESP8266WebServer.h>
#include <FS.h> //Include File System Headers

const char* imagefile = "/ams.jpg";
const char* htmlfile = "/index.html";

//ESP AP Mode configuration
const char *ssid = "NodeAPtest";
const char *password = "password";

ESP8266WebServer server(80);

void handleRoot(){
  server.sendHeader("Location", "/index.html",true); //Redirect to our html web page
  server.send(302, "text/plain","");
}

void handleWebRequests(){
  if(loadFromSpiiffs(server.uri())) return;
  String message = "File Not Detected\n\n";
  message += "URI: ";
  message += server.uri();
  message += "\nMethod: ";
  message += (server.method() == HTTP_GET)?"GET":"POST";
  message += "\nArguments: ";
  message += server.args();
  message += "\n";
  for (uint8_t i=0; i<server.args(); i++){
    message += " NAME:"+server.argName(i) + "\n VALUE:" + server.arg(i) + "\n";
  }
  server.send(404, "text/plain", message);
  Serial.println(message);
}

void setup() {
  delay(1000);
  Serial.begin(115200);
  WiFi.mode(WIFI_AP);

  //Initialize File System
  SPIFFS.begin();
  Serial.println("File System Initialized");
}
```



```

//Initialize AP Mode
WiFi.softAP(ssid); //Password not used
IPAddress myIP = WiFi.softAPIP();
Serial.print("Web Server IP:");
Serial.println(myIP);

//Initialize Webserver
server.on("/",handleRoot);
server.onNotFound(handleWebRequests); //Set setver all paths are not found so we can handle as
per URI
server.begin();
}

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

bool loadFromSpiffs(String path){
String dataType = "text/plain";
if(path.endsWith("/")) path += "index.htm";

if(path.endsWith(".src")) path = path.substring(0, path.lastIndexOf("."));
else if(path.endsWith(".html")) dataType = "text/html";
else if(path.endsWith(".htm")) dataType = "text/html";
else if(path.endsWith(".css")) dataType = "text/css";
else if(path.endsWith(".js")) dataType = "application/javascript";
else if(path.endsWith(".png")) dataType = "image/png";
else if(path.endsWith(".gif")) dataType = "image/gif";
else if(path.endsWith(".jpg")) dataType = "image/jpeg";
else if(path.endsWith(".ico")) dataType = "image/x-icon";
else if(path.endsWith(".xml")) dataType = "text/xml";
else if(path.endsWith(".pdf")) dataType = "application/pdf";
else if(path.endsWith(".zip")) dataType = "application/zip";
File dataFile = SPIFFS.open(path.c_str(), "r");
if (server.hasArg("download")) dataType = "application/octet-stream";
if (server.streamFile(dataFile, dataType) != dataFile.size()) {
}

dataFile.close();
return true;
}

```

ขั้นตอนต่างๆทำเช่นเดียวกับ โปรแกรมแบบที่ 1 แต่ขั้นตอนที่ 5 การทดสอบทำดังนี้

การทดสอบ

1. แพลและ 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 แล้วเปิดหน้าเว็บตามข้อ 2 ในแบบแรก จะได้ผลการทำงานเหมือน เดิม

