

WiFi Manager

WiFi Manager เป็นไลบรารีตัวหนึ่งที่ช่วยให้พัฒนาหรือผู้ใช้งานสามารถที่จะจัดการเรื่อง WiFi ให้กับอุปกรณ์ได้ง่ายขึ้น

ข้อดี

1. มี Captive Portal ช่วยให้เราสามารถระบุ ssid และ password ให้กับอุปกรณ์ได้โดยง่าย เพียงแค่เลือกเมนูและกรอกข้อมูลที่ต้องการแล้วกดบันทึก
2. สะดวกสบายไม่ต้องแก้ไขโค้ดโปรแกรมแล้วอัปโหลดใหม่ให้ยุ่งยาก เขียนโค้ดอัปโหลดให้แค่ครั้งเดียว
3. ช่วยดูแลเรื่องการเชื่อมต่อ WiFi เมื่อมีปัญหา เช่น Access Point หาย ใช้งานไม่ได้ หรืออินเทอร์เน็ตหลุด ตัว WiFiManager จะทำการเปลี่ยนโหมดตัวเองให้เป็น Access Point เพื่อให้ผู้ใช้งานสามารถที่จะเชื่อมต่ออุปกรณ์เข้าไปเปลี่ยน ssid และ password ให้ใหม่

ข้อเสีย

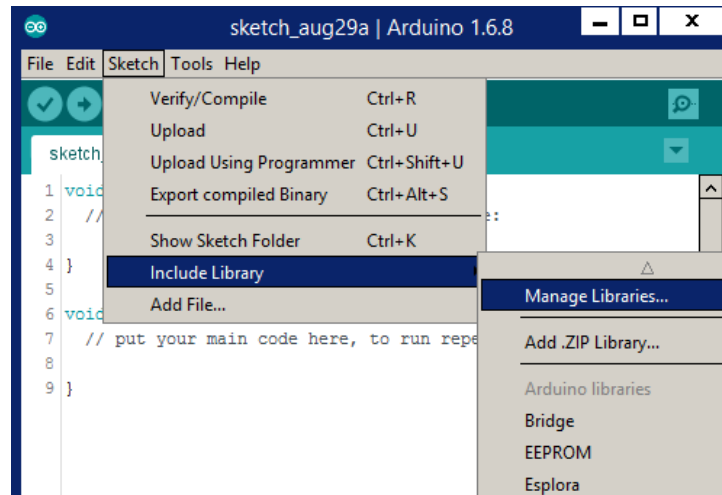
WiFiManager ช่วยดูแลการเชื่อมต่อให้ได้ในระดับหนึ่ง แต่การเปลี่ยนโหมดจาก Client เป็น AP นั้นจะมีการเรียก ESP.reset() ซึ่งการเรียกบ่อยๆ อาจจะมีผลทำให้อุปกรณ์ค้าง

ข้อควรระวัง

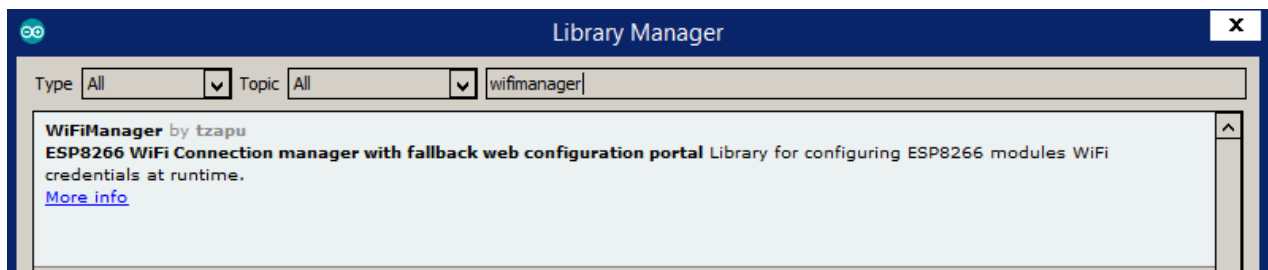
WiFiManager ใช้งานง่ายเพียงแค่เชื่อมต่อกับอุปกรณ์ แต่ในขณะที่อุปกรณ์นั้นเปลี่ยนตัวเองให้เป็นโหมด Access Point ซึ่งผู้ใช้งานคนอื่นก็สามารถมองเห็นและเข้าถึงตัวอุปกรณ์ได้ ดังนั้นควรจะต้องตั้ง password ของ AP ด้วย ในกรณีที่ทำไปใช้งานจริง เพื่อความปลอดภัยของตัวอุปกรณ์ และมั่นใจได้ว่าไม่มีใครสามารถเข้าไปแก้ไขหรือสั่งงานอุปกรณ์ได้

การติดตั้งและการใช้งาน WiFiManager Library

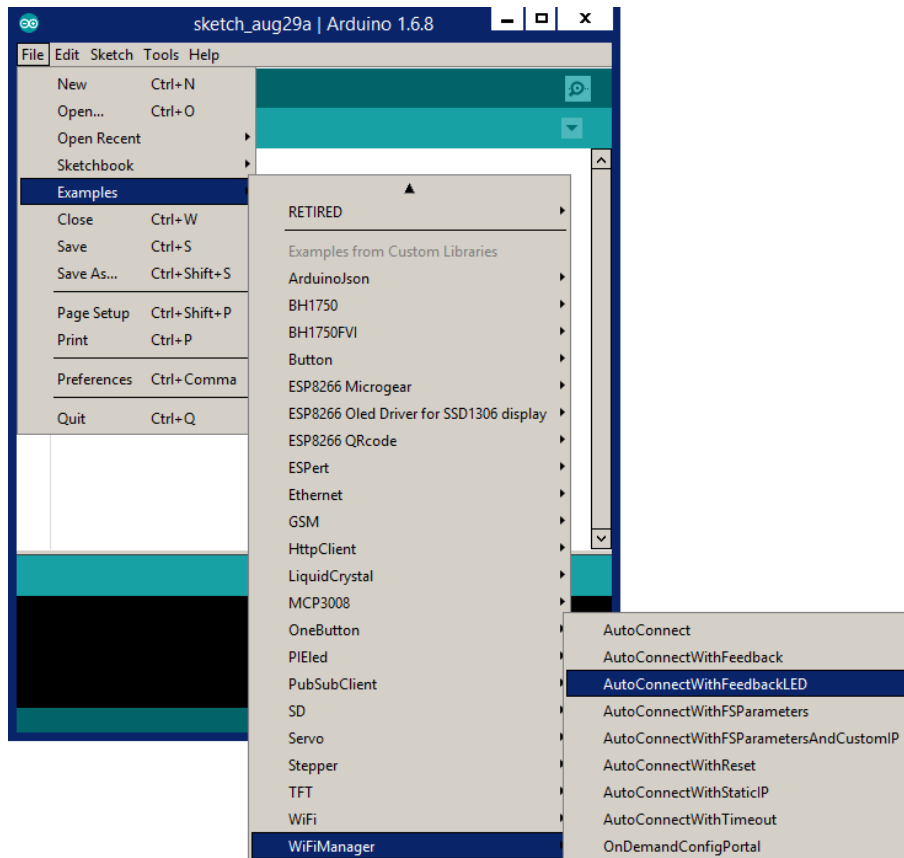
1. เปิดโปรแกรม Arduino IDE
2. เข้าไปที่ Sketch -> Include Library -> Manage Libraries...



3. พิมพ์คำว่า wifimanager ในช่อง Filter your search... เมื่อมีค้นหาเจอแล้วให้คลิกบน Label ของ WiFiManager และคลิก Install



4. เมื่อติดตั้งเรียบร้อยแล้ว กด Close ได้เลย และลองดูตัวอย่าง WiFiManager เข้าไปที่ File -> Examples -> WiFiManager -> AutoConnectWithFeedbackLED



โค้ดตัวอย่าง

```
#include <ESP8266WiFi.h>           //https://github.com/esp8266/Arduino

#include <DNSServer.h>

#include <ESP8266WebServer.h>

#include <WiFiManager.h>           //https://github.com/tzapu/WiFiManager

#include <Ticker.h>

Ticker ticker;

void tick() { // เป็นฟังก์ชันไว้สั่งงานแสดงสถานะ led

  int state = digitalRead(BUILTIN_LED); // get the current state of GPIO1 pin

  digitalWrite(BUILTIN_LED, !state);    // set pin to the opposite state

}
```

```

void configModeCallback (WiFiManager *myWiFiManager) { // callback เมื่อเชื่อม access point ไม่สำเร็จ จะเข้าสู่โหมด AP

  Serial.println("Entered config mode");

  Serial.println(WiFi.softAPIP());

  Serial.println(myWiFiManager->getConfigPortalSSID());

  ticker.attach(0.2, tick); // กำหนดสถานะกะพริบของ led ติดดับสลับกัน 200 มิลลิวินาที
}

void setup() {

  Serial.begin(115200);

  pinMode(BUILTIN_LED, OUTPUT);

  ticker.attach(0.6, tick); // กำหนดสถานะกะพริบของ led ติดดับสลับกัน 600 มิลลิวินาที

  WiFiManager wifiManager; // ประกาศใช้งาน WiFiManager

  wifiManager.resetSettings(); // ฟังก์ชันสำหรับลบการตั้งค่า ssid, password

  wifiManager.setTimeout(180); // ฟังก์ชันสำหรับกำหนด timeout ของโหมด AP อุปกรณ์

  wifiManager.setAPCallback(configModeCallback); // กำหนด callback ของ AP ไปที่ฟังก์ชัน configModeCallback

  if (!wifiManager.autoConnect("ESP_NodeMCU")) { // กำหนดชื่อ AP ให้อุปกรณ์ เมื่อเป็นโหมด AP, SSID ชื่อ ESP_NodeMCU

    Serial.println("failed to connect and hit timeout");

    ESP.reset(); // reset อุปกรณ์

    delay(1000);

  }

  Serial.println("connected...yeey :)");

  ticker.detach(); // ยกเลิกการใช้ tricker สั่งงานเปลี่ยนสถานะ led

  digitalWrite(BUILTIN_LED, LOW);

}

void loop() {

  // put your main code here, to run repeatedly:

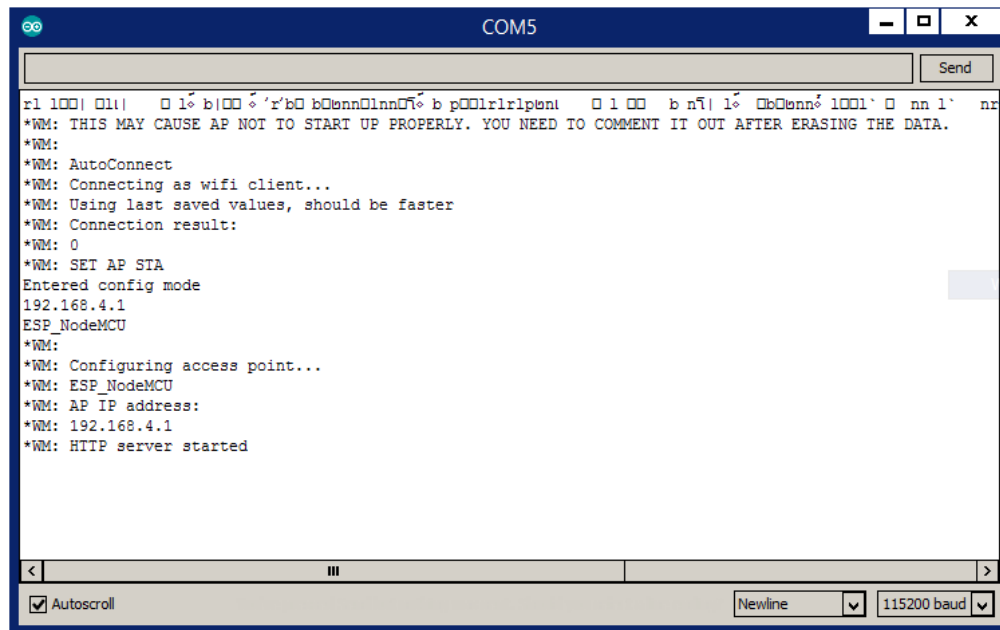
}

```

5. ให้เลือก board ที่จะใช้งาน ในตอนนี้จะใช้เป็น NodeMCU โดยคลิกที่ Tools -> Board: “???????” -> NodeMCU 1.0 และเลือก Port ที่เชื่อมต่อ

6. คลิก Sketch -> Upload หรือกดคีย์ลัด Ctrl+U

7. หลังจาก upload เสร็จแล้ว ดูที่ Serial Monitor โดยคลิกที่รูปแว่นขยาย หรือเข้าไปที่ Tools -> Serial Monitor เมื่ออุปกรณ์เข้าสู่โหมด AP จะได้ output ดังภาพด้านล่าง



8. ให้นำมือถือหรือคอมพิวเตอร์เชื่อมต่อไปที่ AP ชื่อ ESP_NodeMCU หลังจากเชื่อมต่อเรียบร้อยแล้ว จะมี Captive Portal เด้งขึ้นมา ถ้าหากไม่มีการตอบสนอง ให้เปิด Browser ขึ้นมา และพิมพ์ address : 192.168.4.1

เมื่อเข้าถึงอุปกรณ์ได้แล้วก็ตั้งค่าให้กับอุปกรณ์ได้เลย

หน้า Captive Portal



ESP_NodeMCU

WiFiManager

Configure WiFi

Configure WiFi (No Scan)

Info

Reset

หน้า Configure WiFi และ Configure WiFi (No Scan)



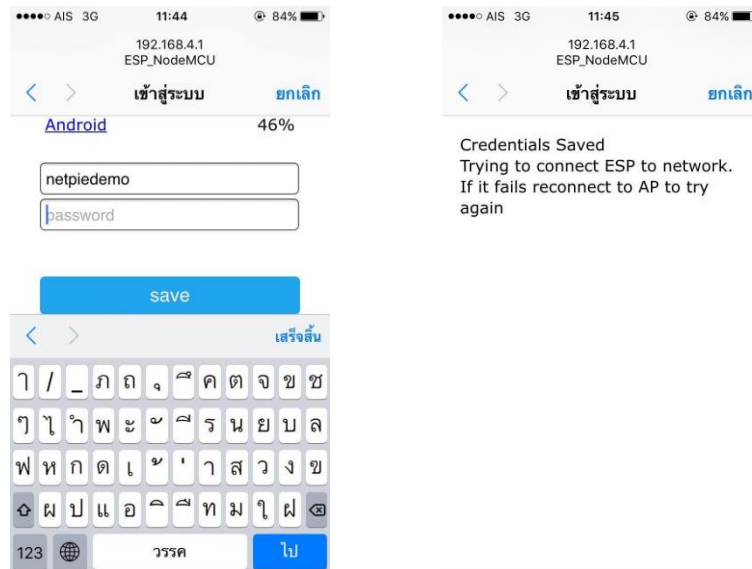
netpi_demo	🔒 100%
catbus	🔒 100%
pi_node01	100%
TSP-WIFI	100%
NSTDA-GUEST	100%
NSTDA-STAFF	🔒 96%
NSTDA-WIFI	92%
NOKIA_909_8052	🔒 78%
pi_node03	76%
NECTEC_Temp	72%
TMECIoT3	68%
MikroTik-C2CEFF	🔒 64%
PaidWifi	🔒 60%
pi_node02	58%



save

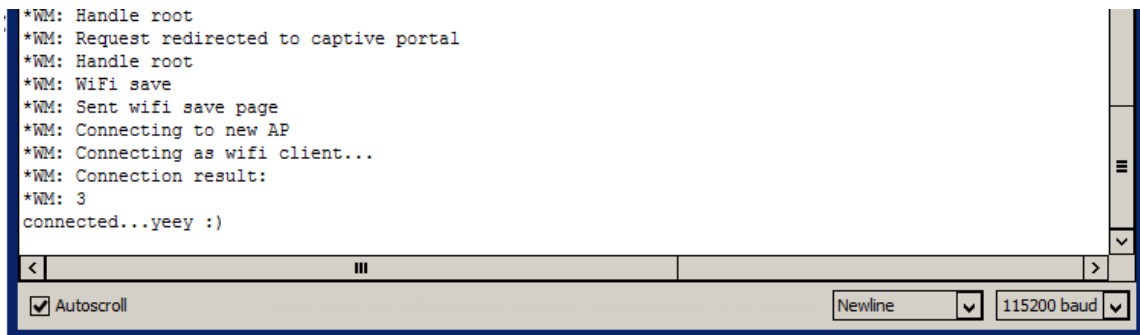
[Scan](#)

9. ลองตั้งค่า ssid และ password ให้อุปกรณ์



10. เมื่อกด Save ฟังก์ชันจะทำการบันทึก ssid และ password แล้วทำการเชื่อมต่อ Access Point ตามที่ระบุ

*WM: 3 เป็นตัวเลขสถานะของการเชื่อมต่อ Access Point สำเร็จ



การใช้งาน WiFiManager Library + Microgear Library

ตัวอย่างนี้เป็นการใช้ WiFiManager ร่วมกับ Microgear Library (NETPIE)

```
/* NETPIE ESP8266 basic sample */
/* More information visit : https://netpie.io */

#include <ESP8266WiFi.h>      //https://github.com/esp8266/Arduino
#include <MicroGear.h>

//needed for library
#include <DNSServer.h>
#include <ESP8266WebServer.h>
#include <WiFiManager.h>      //https://github.com/tzapu/WiFiManager

//for LED status
#include <Ticker.h>
Ticker ticker;

#define APPID  "APPID"  // APPID บนหน้าเว็บ application
#define KEY    "APPKEY" // APPKEY บนหน้าเว็บ application
#define SECRET "APPSECRET" // APPSECRET บนหน้าเว็บ application
#define ALIAS  "esp8266"
#define AP_NAME "AP-XXXXX"

WiFiClient client;

int timer = 0;
MicroGear microgear(client);

void tick()
{
    //toggle state
    int state = digitalRead(BUILTIN_LED); // get the current state of GPIO1 pin
    digitalWrite(BUILTIN_LED, !state);    // set pin to the opposite state
}

//gets called when WiFiManager enters configuration mode
```



```

void configModeCallback (WiFiManager *myWiFiManager) {
  Serial.println("Entered config mode");
  Serial.println(WiFi.softAPIP());
  //if you used auto generated SSID, print it
  Serial.println(myWiFiManager->getConfigPortalSSID());
  //entered config mode, make led toggle faster
  ticker.attach(0.2, tick); // ticker ทำงานเรียกใช้ฟังก์ชัน tick ทุกๆ 200ms
}

/* If a new message arrives, do this */
void onMsgHandler(char *topic, uint8_t* msg, unsigned int msglen) {
  Serial.print("Incoming message --> ");
  msg[msglen] = '\0';
  Serial.println((char *)msg);
}

/* When a microgear is connected, do this */
void onConnected(char *attribute, uint8_t* msg, unsigned int msglen) {
  Serial.println("Connected to NETPIE...");
  /* Set the alias of this microgear ALIAS */
  microgear.setAlias(ALIAS);
  ticker.attach(1, tick); // ticker ทำงานเรียกใช้ฟังก์ชัน tick ทุกๆ 1s
}

void setup() {
  // put your setup code here, to run once:
  Serial.begin(115200);

  /* Add Event listeners */
  /* Call onMsgHandler() when new message arrives */
  microgear.on(MESSAGE,onMsgHandler);

  /* Call onConnected() when NETPIE connection is established */
  microgear.on(CONNECTED,onConnected);

  //set led pin as output
  pinMode(BUILTIN_LED, OUTPUT);
  ticker.attach(0.6, tick); // ticker ทำงานเรียกใช้ฟังก์ชัน tick ทุกๆ 600ms
}

```

```

//WiFiManager
//Local initialization. Once its business is done, there is no need to keep it around
WiFiManager wifiManager;
//reset settings - for testing
//wifiManager.resetSettings(); //

//set callback that gets called when connecting to previous WiFi fails, and enters Access Point mode
wifiManager.setAPCallback(configModeCallback);

//fetches ssid and pass and tries to connect
//if it does not connect it starts an access point with the specified name
//here "AutoConnectAP"
//and goes into a blocking loop awaiting configuration
if (!wifiManager.startConfigPortal(AP_NAME)) {
    Serial.println("failed to connect and hit timeout");
    delay(3000);
    //reset and try again, or maybe put it to deep sleep
    ESP.reset(); // สำหรับ library ตัวนี้จะไม่ใช้คำสั่ง ESP.reset()
    delay(5000);
}

//if you get here you have connected to the WiFi
Serial.println("connected...yeey :)");
ticker.detach(); // ยกเลิกคำสั่ง ticker
//keep LED on
digitalWrite(BUILTIN_LED, LOW);

/* Initial with KEY, SECRET and also set the ALIAS here */
microgear.init(KEY,SECRET,ALIAS);

/* connect to NETPIE to a specific APPID */
microgear.connect(APPID);
}

void loop() {
    /* To check if the microgear is still connected */
    if (microgear.connected()) {

```

```

Serial.println("connected");

/* Call this method regularly otherwise the connection may be lost */
microgear.loop();

if (timer >= 1000) {
    Serial.println("Publish...");

    /* Chat with the microgear named ALIAS which is myself */
    microgear.chat(ALIAS, "Hello");
    timer = 0;
}
else timer += 100;
}
else {
    Serial.println("connection lost, reconnect...");
    if (timer >= 5000) {
        microgear.connect(APPID);
        timer = 0;
    }
    else timer += 100;
    ticker.detach(); // ยกเลิกคำสั่ง ticker
}
delay(100);
}

```

4. กด upload และรอจนเสร็จ

สามารถปรับเปลี่ยนโค้ดตามการใช้งานได้ ขึ้นอยู่กับเงื่อนไขที่ใช้ ไม่จำเป็นต้องยึดตามรูปแบบโค้ดตัวอย่างก็ได้

ดาวน์โหลดโค้ดตัวอย่างได้ที่ https://github.com/anunpanya/WiFiManager_MicrogearNETPIE