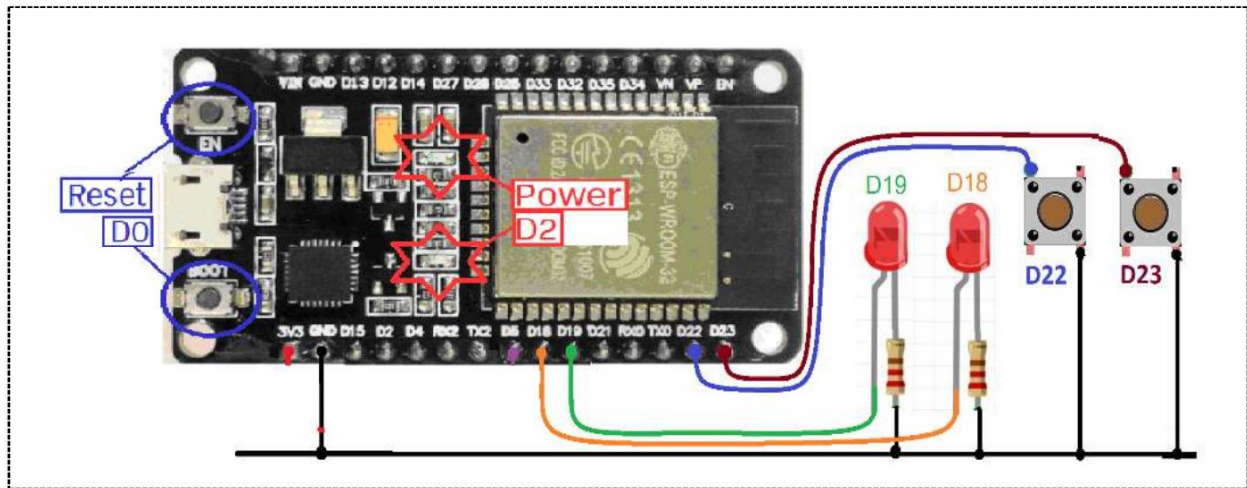


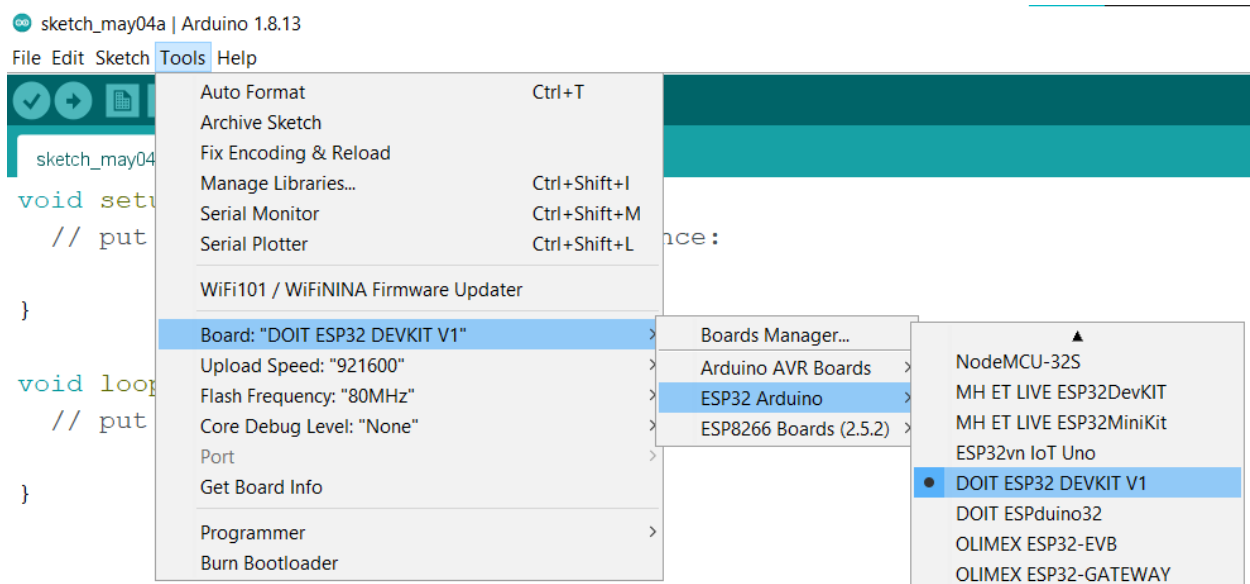
D1 การทดลองที่ 1

- หากต้องการให้ใช้ 1 สวิตช์ ควบคุม 1 LED แบบกดติด-กดดับ จำนวน 2 วงจรจะต้องวงจรและเขียนโปรแกรมอย่างไร {SW-D22 -- LED-D19, SW-D23 -- LED-D18}



1.ต่อวงจรดังรูปด้านบน

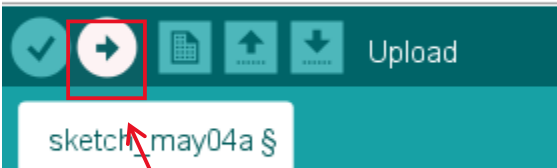
2.หน้าต่าง Arduino IDE ตั้งค่า board เป็น DOIT ESP32 DEVKIT V1



3. อัปโหลดโค้ดลงบอร์ด

sketch_may04a | Arduino 1.8.13

File Edit Sketch Tools Help



```
#define LED_1 18
#define LED_2 19
#define BUTTON_1 22
#define BUTTON_2 23

int buttonState1 = 0;
int buttonState2 = 0;

void setup(){
  pinMode(BUTTON_1, INPUT_PULLUP);
  pinMode(BUTTON_2, INPUT_PULLUP);
  pinMode(LED_1, OUTPUT);
  pinMode(LED_2, OUTPUT);
  Serial.begin(115200);
  delay(10);
}

void loop() {
  if (digitalRead(BUTTON_1) == LOW) {
    delay(20);
    buttonState1 = 1 - buttonState1;
    digitalWrite(LED_1, buttonState1);
    while (digitalRead(BUTTON_1) == LOW);
    delay(20);
  }
  if (digitalRead(BUTTON_2) == LOW) {
    delay(20);
    buttonState2 = 1 - buttonState2;
    digitalWrite(LED_2, buttonState2);
    while (digitalRead(BUTTON_2) == LOW);
    delay(20);
  }
}
```

D1 การทดลองที่ 2

- เพิ่มเติมจาก Q202 อยากได้ปุ่มสำหรับคุมปิด-เปิด หลอดไฟ LED 4 ดวง
- อยากฝึก Link ไปที่หน้า FB ของตัวเอง
- https://www.colorhexa.com/008cba?fbclid=IwAR3dIZ_gRgDWmREmnzukuLbMxV3pOHY4YIPuLEz8-ZzTOX2VhWxcH2QjLGk

← → ⓘ Not secure | 192.168.43.237

The ESP-32 Update web page without refresh

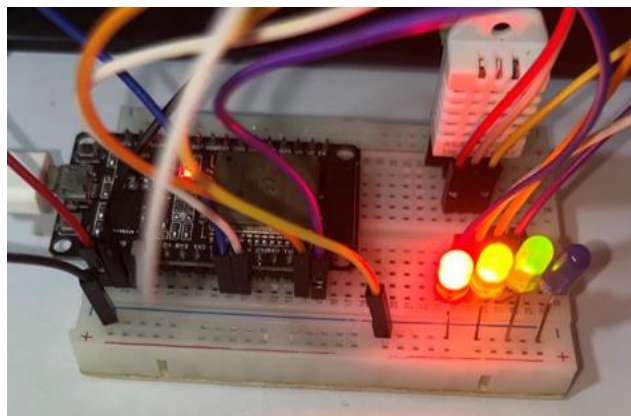


State of [LED1, LED2, LED3, LED4] is >> ON, OFF, OFF, ON

DHT-22 sensor : Temp = 28.10 C, Humidity = 43.90 %

By Wichai Srisuruk

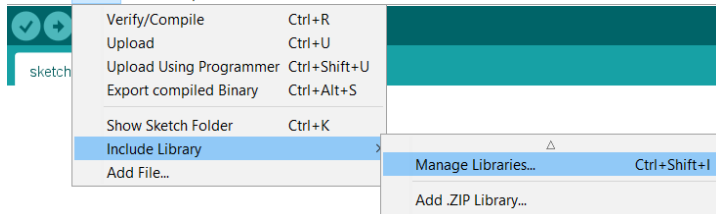
1. ต่อดังรูป



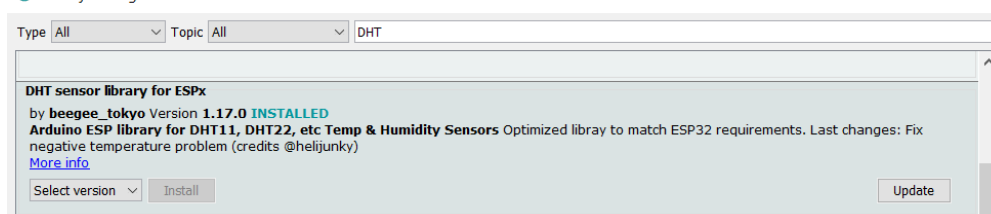
2. ติดตั้งไลบรารีของ DHT-22 ชื่อ DHT sensor library for ESPx

sketch_may04a | Arduino 1.8.13

File Edit Sketch Tools Help



Library Manager



3.ในการ update อุณหภูมิและความชื้นจากเซ็นเซอร์ทุกๆ 2วินาที จะใช้ AJAX เข้ามาช่วยในการรีเฟรชหน้าเว็บเพื่อลดภาระของเซิร์ฟเวอร์

4.สร้างไฟล์ชื่อ Test2.ino โดยการเปิด Arduino IDE สร้างไฟล์ใหม่แล้วบันทึกด้วยชื่อที่กล่าวมา

5.สร้างไฟล์ชื่อ index.h โดยการเปิด notepad แล้วบันทึกไฟล์ด้วยชื่อที่กล่าวมา

6.อัปโหลดโค้ดด้านล่างลงใน ESP32

ไฟล์ Test2.ino

```
#include <WiFi.h>
#include <WiFiClient.h>
#include <WebServer.h>
#include "DHTesp.h"
#include "index.h" //Our HTML webpage contents with javascripts
#define DHT_Pin 4
#define LED1 18
#define LED2 19
#define LED3 22
#define LED4 23
//SSID and Password of your WiFi router
const char* ssid = "network";
const char* password = "password";
WebServer server(80); //Server on port 80
DHTesp dht;
String ledState1 = "NA";
String ledState2 = "NA";
String ledState3 = "NA";
String ledState4 = "NA";
//=====
// This routine is executed when you open its IP in browser
//=====
void handleRoot() {
String s = MAIN_page; //Read HTML contents
server.send(200, "text/html", s); //Send web page
}
void handleADC() {
float h = dht.getHumidity();
```

```

float t = dht.getTemperature();
String tmpValue = "Temp = ";
tmpValue += String(t) + " C, Humidity = ";
tmpValue += String(h) + " %";
server.send(200, "text/plain", tmpValue); //Send value to client ajax request
}

void handleLED() {
String t_state = server.arg("LEDstate"); //Refer xhttp.open("GET", "setLED?LEDstate="+led, true);
Serial.println(t_state);
if (t_state == "11") { digitalWrite(LED1, HIGH); ledState1 = "ON"; } //Feedback parameter
if (t_state == "10") { digitalWrite(LED1, LOW); ledState1 = "OFF"; } //Feedback parameter
if (t_state == "21") { digitalWrite(LED2, HIGH); ledState2 = "ON"; } //Feedback parameter
if (t_state == "20") { digitalWrite(LED2, LOW); ledState2 = "OFF"; } //Feedback parameter
if (t_state == "31") { digitalWrite(LED3, HIGH); ledState3 = "ON"; } //Feedback parameter
if (t_state == "30") { digitalWrite(LED3, LOW); ledState3 = "OFF"; } //Feedback parameter
if (t_state == "41") { digitalWrite(LED4, HIGH); ledState4 = "ON"; } //Feedback parameter
if (t_state == "40") { digitalWrite(LED4, LOW); ledState4 = "OFF"; } //Feedback parameter
server.send(200, "text/plain", ledState1 + " , " + ledState2 + " , " + ledState3 + " , " + ledState4); //Send
web page
}

void setup(void) {
Serial.begin(115200);
dht.setup(DHT_Pin, DHTesp::DHT22); // DHT_Pin D4, DHT22
pinMode(LED1, OUTPUT);
pinMode(LED2, OUTPUT);
pinMode(LED3, OUTPUT);
pinMode(LED4, OUTPUT);
Serial.print("\n\nConnect to ");
Serial.println(ssid);
WiFi.begin(ssid, password);
while (WiFi.status() != WL_CONNECTED) {
delay(500); Serial.print(".");
}
Serial.print("\nConnected "); Serial.println(ssid);
Serial.print("IP address: "); Serial.println(WiFi.localIP());
server.on("/", handleRoot);
server.on("/setLED", handleLED);
server.on("/readADC", handleADC);

```

```

server.begin();
Serial.println("HTTP server started");
}
void loop(void) {
server.handleClient(); //Handle client requests
}

```

ไฟล์ index.h

```

const char MAIN_page[] PROGMEM = R"=====(
<!DOCTYPE html>
<html>
<body>
<div id="demo">
<h1>The ESP-32 Update web page without refresh</h1>
<button type="button" onclick="sendData(11)" style="background: rgb(202, 60, 60);">LED1
ON</button>
<button type="button" onclick="sendData(21)" style="background: rgb(202, 60, 60);">LED2
ON</button>
<button type="button" onclick="sendData(31)" style="background: rgb(202, 60, 60);">LED3
ON</button>
<button type="button" onclick="sendData(41)" style="background: rgb(202, 60, 60);">LED4
ON</button><br><br>
<button type="button" onclick="sendData(10)" style="background: rgb(100,116,255);">LED1
OFF</button>
<button type="button" onclick="sendData(20)" style="background: rgb(100,116,255);">LED2
OFF</button>
<button type="button" onclick="sendData(30)" style="background: rgb(100,116,255);">LED3
OFF</button>
<button type="button" onclick="sendData(40)" style="background: rgb(100,116,255);">LED4
OFF</button><br><br>
State of [LED1, LED2, LED3, LED4] is >> <span id="LEDState">NA</span><br>
</div>
<div>
<br>DHT-22 sensor : <span id="ADCValue">0</span><br>
</div>

```

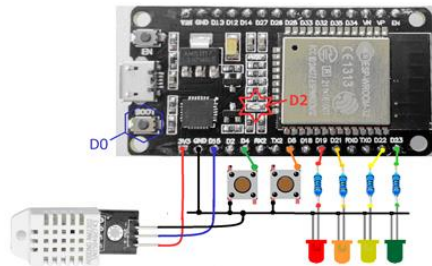
```

<script>
function sendData(led) {
var xhttp = new XMLHttpRequest();
xhttp.onreadystatechange = function() {
if (this.readyState == 4 && this.status == 200) {
document.getElementById("LEDState").innerHTML =
this.responseText;
}
};
xhttp.open("GET", "setLED?LEDstate="+led, true);
xhttp.send();
}
setInterval(function() {
// Call a function repetatively with 2 Second interval
getData();
}, 2000); //2000mSeconds update rate
function getData() {
var xhttp = new XMLHttpRequest();
xhttp.onreadystatechange = function() {
if (this.readyState == 4 && this.status == 200) {
document.getElementById("ADCValue").innerHTML =
this.responseText;
}
};
xhttp.open("GET", "readADC", true);
xhttp.send();
}
</script>
<br><a href="facebook link">By "Name"</a>
</body>
</html>
)=====";

```

D1 การทดลองที่ 3

- อ่านค่า DHT-22 แล้วส่งไปยัง MQTT Broker ทุกๆ 5 วินาที
- กำหนดให้ใช้ mqtt.eclipse.org เป็น Broker
- ควบคุมการปิดเปิด 4 LED
- รับคำสั่งกำหนด SW1 แฉ่ง Overheat Alarm, SW2 แฉ่ง Intruders Alarm



1.ต่อวงจรดังรูปด้านบน

2.การทดลองนี้จะใช้ MQTTLens ที่เป็นส่วนขยายของ chrome โดยจะทำการตั้งค่าการเชื่อมต่อดังรูป (เนื่องจากมีปัญหาในการใช้งาน mqtt.eclilpes.org จึงเปลี่ยนไปใช้ broker.mqttdashboard.com เป็น broker แทน)

Connection Details

Connection name	TestM2	Connection color scheme	
Hostname	tcp:// broker.mqttdashboard.com	Port	1883
Client ID	lens_cl4PZ5hqMBJo2e3GVAIeBuLCUqm Generate a random ID		
Session	Automatic Connection	Keep Alive	120 seconds
<input checked="" type="checkbox"/> Clean Session	<input checked="" type="checkbox"/> Automatic Connection		

Credentials

Username	Password
<input type="text" value="Enter username"/>	<input type="text" value="Enter password"/>

3.ติดตั้งไลบรารีชื่อ PubSubClient เพิ่มเติมจากนั้นอัปโหลดโค้ดด้านล่าง

```
#include <WiFi.h>
#include <PubSubClient.h>
#include "DHTesp.h"
#define DHT22_Pin 15
DHTesp dht;
```



```
const char* ssid = "network";
const char* password = "password";
const char* mqtt_server = "broker.mqttdashboard.com";
const char* topic1 = "TestM2";
WiFiClient espClient;
PubSubClient client(espClient);
long lastMsg = 0;
char msg[50];
int LED1 = 18;
int LED2 = 19;
int LED3 = 22;
int LED4 = 23;
int Button1 = 4;
int Button2 = 5;
void setup_wifi() {
  delay(10);
  Serial.println();
  Serial.print("Connecting to "); Serial.println(ssid);
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500); Serial.print(".");
  }
  randomSeed(micros());
  Serial.println(""); Serial.println("WiFi connected");
  Serial.println("IP address: "); Serial.println(WiFi.localIP());
}
void reconnect()
{ while (!client.connected()) // Loop until we're reconnected
{ Serial.print("Attempting MQTT connection...");
String clientId = "ESP32 Client-";
clientId += String(random(0xffff), HEX); // Create a random client ID
if (client.connect(clientId.c_str())) // Attempt to connect
{ Serial.println("connected"); // Once connected, publish an announcement...
client.publish(topic1, "Hello World ATP"); // ... and resubscribe
client.subscribe(topic1);
} else
{ Serial.print("failed, rc=");
Serial.print(client.state());
```

```

Serial.println(" try again in 5 seconds");
delay(5000);
}
}
}

void callback(char* topic, byte* payload, unsigned int length)
{ char myPayload[50];
Serial.print("Message arrived [");
Serial.print(topic);
Serial.print("] ");
for (int i = 0; i < length; i++){
Serial.print((char)payload[i]);
myPayload[i] = payload[i];
myPayload[i + 1] = '\0'; // End of String
}
Serial.print("\n ---> "); Serial.println(myPayload);
myPayload[8] = '\0'; // String less than 4 characters
if ((String)myPayload == "LED1ON") digitalWrite(LED1, HIGH);
if ((String)myPayload == "LED1OFF") digitalWrite(LED1, LOW);
if ((String)myPayload == "LED2ON") digitalWrite(LED2, HIGH);
if ((String)myPayload == "LED2OFF") digitalWrite(LED2, LOW);
if ((String)myPayload == "LED3ON") digitalWrite(LED3, HIGH);
if ((String)myPayload == "LED3OFF") digitalWrite(LED3, LOW);
if ((String)myPayload == "LED4ON") digitalWrite(LED4, HIGH);
if ((String)myPayload == "LED4OFF") digitalWrite(LED4, LOW);
}

void setup(){
Serial.begin(115200);
pinMode(LED1, OUTPUT);
pinMode(LED2, OUTPUT);
pinMode(LED3, OUTPUT);
pinMode(LED4, OUTPUT);
pinMode(Button1, INPUT_PULLUP);
pinMode(Button2, INPUT_PULLUP);
dht.setup(DHT22_Pin, DHTesp::DHT22);
setup_wifi();
client.setServer(mqtt_server, 1883);
client.setCallback(callback);
}

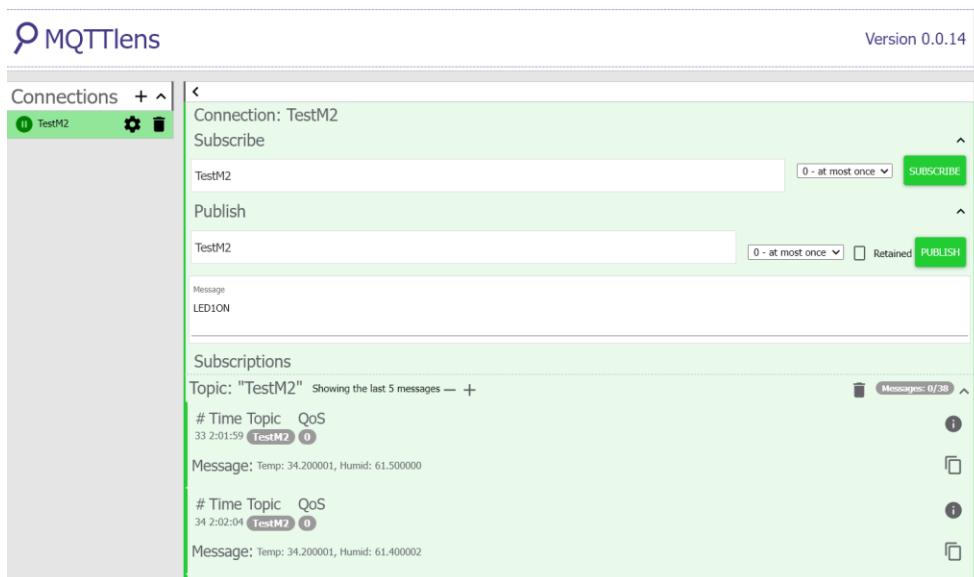
```

```

}
void loop(){
float temperature = dht.getTemperature();
float humidity = dht.getHumidity();
if (!client.connected()) reconnect();
client.loop();
if(digitalRead(Button1) == 0){
client.publish(topic1, "Overheat");
delay(1000);
}
if(digitalRead(Button2) == 0){
client.publish(topic1, "Intruders");
delay(1000);
}
sprintf (msg, "Temp: %f, Humid: %f", temperature, humidity);
Serial.print("Publish message: ");
Serial.println(msg);
client.publish(topic1, msg);
delay(5000);
}

```

4. สังเกตการทำงานที่ MQTTLens จะพบว่าหัวข้อที่ทำการ subscribe ไว้จะมีข้อความแสดงอุณหภูมิ และความชื้นปรากฏขึ้น และเมื่อทำการ publish ที่หัวข้อนี้ด้วยข้อความ LED1ON ให้ทำการสังเกตการทำงานที่ serial monitor ต่อไป

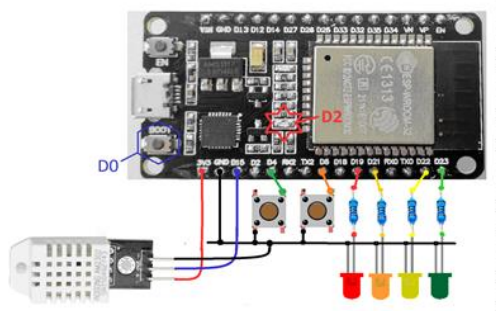


5.สังเกตการทำงาน serial monitor จะพบว่าเมื่อมีการ publish จาก MQTTLens ด้วยข้อความว่า LED1ON ทางฝั่งของ ESP32 ที่ subscribe หัวข้อเดียวกันจะได้รับข้อความแล้วนำไปสั่งการ LED ต่อไป

```
COM7
Send
Message arrived [TestM2] Temp: 34.200001, Humid: 61.700001
---> Temp: 34.200001, Humid: 61.700001
Publish message: Temp: 34.200001, Humid: 61.200001
Message arrived [TestM2] Temp: 34.200001, Humid: 61.500000
---> Temp: 34.200001, Humid: 61.500000
Publish message: Temp: 34.200001, Humid: 61.500000
Message arrived [TestM2] LED1ON
---> LED1ON
Publish message: Temp: 34.200001, Humid: 61.599998
Message arrived [TestM2] Temp: 34.200001, Humid: 61.200001
---> Temp: 34.200001, Humid: 61.200001
Publish message: Temp: 34.099998, Humid: 61.200001
Message arrived [TestM2] Temp: 34.200001, Humid: 61.500000
---> Temp: 34.200001, Humid: 61.500000
Publish message: Temp: 34.200001, Humid: 61.299999
Message arrived [TestM2] Temp: 34.200001, Humid: 61.599998
```

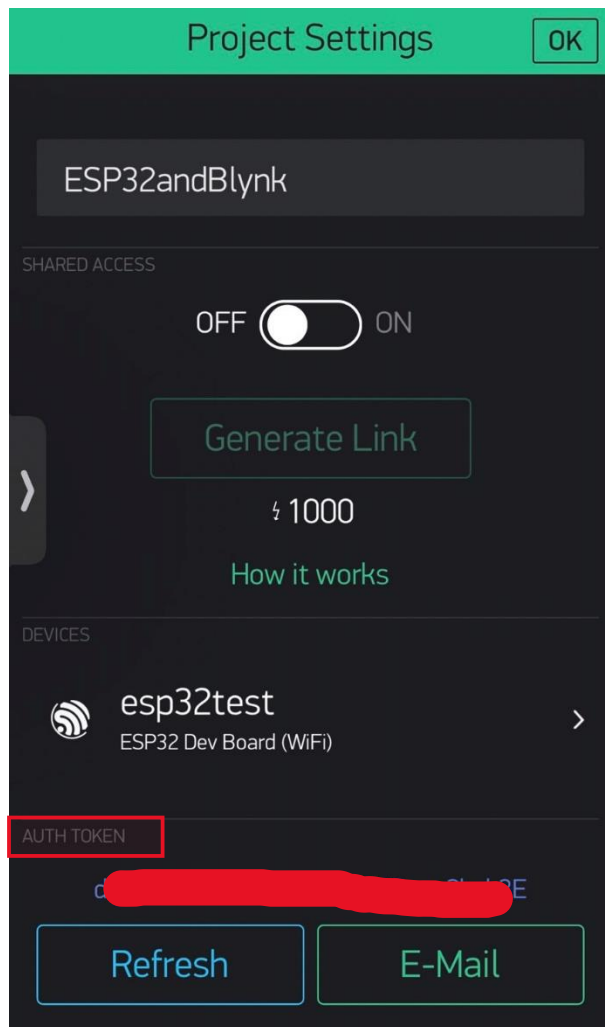
D1 การทดลองที่ 4

- ควบคุมการปิดเปิด 4 LED
- อ่านค่า DHT-22 แล้วส่งไปยัง Blynk ทุกๆ 5 วินาที
- บันทึกค่าไปยัง Google Sheet
- หากอุณหภูมิเกิน 28°C ให้แจ้งไปยัง LINE
- รับคำสั่งวิทยุกำหนด SW1 แจ้ง Overheat Alarm, SW2 แจ้ง Intruders Alarm ไปยัง LINE

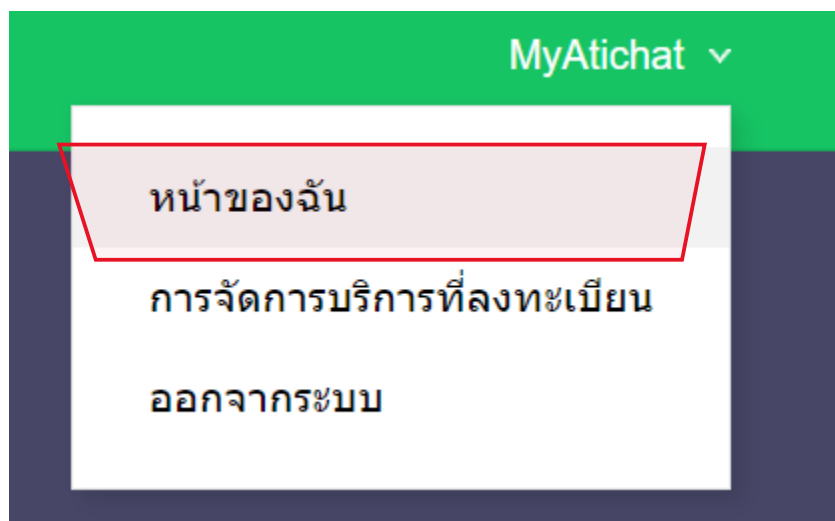


1.ต้องจตามรูป

2.ติดตั้งแอปพลิเคชัน Blynk บนสมาร์ตโฟนจากนั้นล็อกอินและสร้างโปรเจกจากนั้นตั้งค่าโปรเจกดังรูป
จะได้ Token จาก Blynk เพื่อนำไปใช้ในการอัปโหลดโค้ด

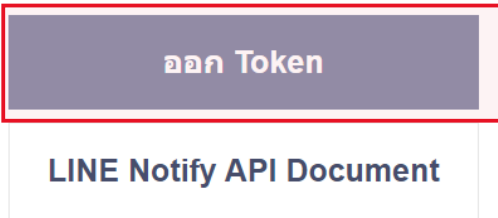


3.ต่อมาทำการเตรียมตั้งค่าการใช้งาน LINE notify โดยการเข้าไปที่เว็บของ line notify (<https://notify-bot.line.me/en/>) จากนั้นทำการออก Token เพื่อนำมาใช้งาน

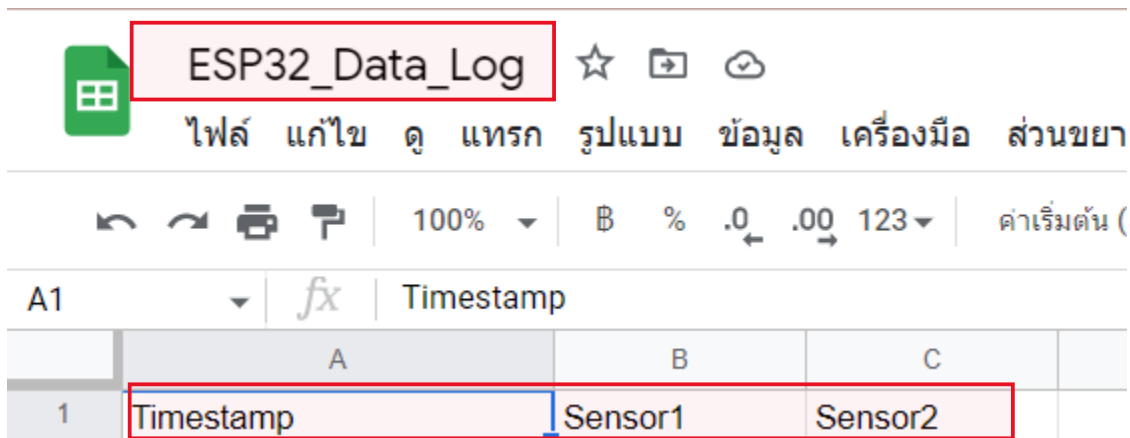


ออก Access Token (สำหรับผู้พัฒนา)

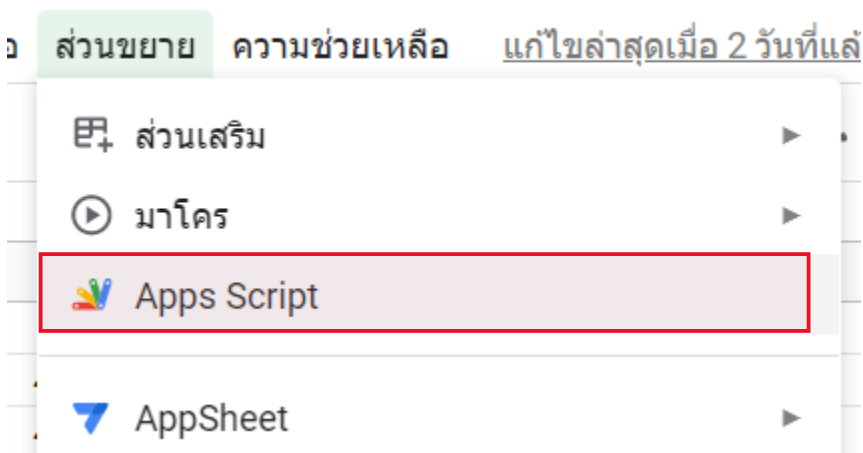
เมื่อใช้ Access Token แบบบุคคล จะสามารถตั้งค่าการแจ้งเตือนได้โดยไม่ต้องลงทะเบียนกับเว็บเซอร์วิส

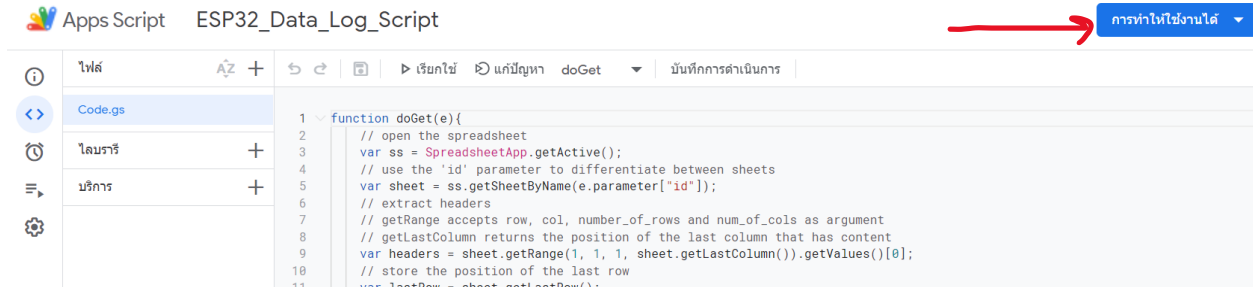


4.ต่อไปจะเป็นการเตรียมเอกสาร google sheet เพื่อบันทึกข้อมูลจากเซ็นเซอร์โดยการสร้างเอกสาร google sheet จากนั้นตั้งชื่อ ดังรูป



5.เพิ่มสคริปต์ในการบันทึกข้อมูลลงในชีตอย่างอัตโนมัติ โดยการเลือกที่ ส่วนขยาย -> Apps Script เมื่อเพิ่มสคริปต์แล้วให้กดที่ปุ่ม “การนำไปใช้งานได้”





6.กำหนดสิทธิ์การเข้าถึงจากนั้นกดที่ “การทำให้ใช้งานได้” จากนั้นจะได้ url ที่จะนำไปใช้งานในโค้ดต่อไป

The screenshot shows the 'การทำให้ใช้งานได้' (Make available) dialog box. It has a sidebar on the left with 'เลือกประเภท' (Select type) and 'การกำหนดค่า' (Settings). The main area is titled 'การทำให้ใช้งานได้รายการใหม่' (New item to make available). It contains two sections: 'เว็บแอป' (Web app) and 'ส่วนเสริม' (Add-on). In the 'เว็บแอป' section, there is a dropdown menu for 'ดำเนินการในฐานะ' (Perform as) set to 'จัน (atichat212@gmail.com)' and a dropdown for 'ผู้ที่มีสิทธิ์เข้าถึง' (Who can access) set to 'ทุกคน' (Everyone). Below these, there is a text box for 'ส่วนเสริม' (Add-on) with the text 'โครงการ Apps Script นี้ใช้โครงการ Google Cloud Platform (GCP) ที่จัดการโดย Apps Script'. At the bottom, there are two buttons: 'ยกเลิก' (Cancel) and 'การทำให้ใช้งานได้' (Make available).

7.ติดตั้งไลบรารี TridentTD_LineNotify และ Blynk เพิ่มเติมจากนั้นอัปโหลดโค้ด

```

#include <ssl_client.h>
#include <TridentTD_LineNotify.h>
#include <WiFi.h>
#include <WiFiClientSecure.h>
#include "DHTesp.h"
#include <BlynkSimpleEsp32.h>
#define BLYNK_PRINT Serial
#define SSID "network"
#define PASSWORD "password"
#define LINE_TOKEN "line Token"
#define DHT22_Pin 15

char auth[] = "blynk token";
int bnt1 = 4;

```

```

int bnt2 = 5;
String GAS_ID = "google sheet";
String GAS_Sheet = "Sensor_Data";
String t;
const char* host = "script.google.com";
const int httpsPort = 443;
long now = millis();
long lastMeasure = 0;
float temperature = 00.00, humidity = 00.00;

BlynkTimer timer;
WiFiClientSecure client;
DHTesp dht;

void myTimerEvent()
{
  temperature = dht.getTemperature();
  humidity = dht.getHumidity();
  Blynk.virtualWrite(V10, temperature);
  Blynk.virtualWrite(V11, humidity);
  Serial.print(" Temp('C) >> "); Serial.print(temperature, 1);
  Serial.print(", Humidity(%) >> "); Serial.println(humidity, 1);
}

void setup() {
  Serial.begin(115200); Serial.println();
  Serial.println(LINE.getVersion());

  client.setInsecure();
  // กำหนด Line Token
  LINE.setToken(LINE_TOKEN);
  dht.setup(DHT22_Pin, DHTesp::DHT22);
  pinMode(bnt1, INPUT_PULLUP);
  pinMode(bnt2, INPUT_PULLUP);
  Blynk.begin(auth, SSID, PASSWORD);
  timer.setInterval(1000L, myTimerEvent);
}

void loop() {

```



```

Blynk.run();
timer.run();

if(digitalRead(bnt1) == LOW){
    LINE.notify("Overheat Alarm");
    Serial.print("BNT1 ON");
}
if(digitalRead(bnt2) == LOW){
    LINE.notify("Intruders Alarm");
    Serial.print("BNT2 ON");
}
delay(5000);
now = millis();
if (now - lastMeasure > 5000) {
    lastMeasure = now;
    float humidity = dht.getHumidity();
    float temperature = dht.getTemperature();
    Serial.print("Temperature: " + String(temperature) + "C/");
    Serial.print("Humidity: " + String(humidity) + "%");

    sendData(temperature, humidity);

    if (temperature > 28.00) {
        String msg = "อุณหภูมิ: " + String(temperature) + " องศา";
        LINE.notify(msg);
    }
}

void sendData(float SValue1, float SValue2) {
    Serial.println("=====");
    Serial.print("connecting to "); Serial.println(host);
    //---- Connect to Google host
    if (!client.connect(host, httpsPort)) {
        Serial.println("connection failed");
        return;
    }
}

```

```

//---- Post Data
String url;
url += "/macros/s/" + GAS_ID + "/exec?";
url += "id=" + String(GAS_Sheet);
url += "&Sensor1=" + String(SValue1, 2);
url += "&Sensor2=" + String(SValue2, 2);
Serial.print("requesting URL: "); Serial.println(url);
client.print(String("GET ") + url + " HTTP/1.1\r\n" +
  "Host: " + host + "\r\n" +
  "User-Agent: BuildFailureDetectorESP8266\r\n" +
  "Connection: close\r\n\r\n");
Serial.println("request sent");
//---- Wait Echo
while (client.connected()) {
  String line = client.readStringUntil('\n');
  if (line == "\r") {
    Serial.println("headers received");
    break;
  }
}
String line = client.readStringUntil('\n');
if (line.startsWith("{\"state\":\"success\"}")) {
  Serial.println("ESP-32/Arduino CI successfull!");
} else {
  Serial.println("ESP-32/Arduino CI has failed");
}
Serial.print("reply was : ");
Serial.println(line);
Serial.println("closing connection");
Serial.println("=====");
Serial.println();
}

```