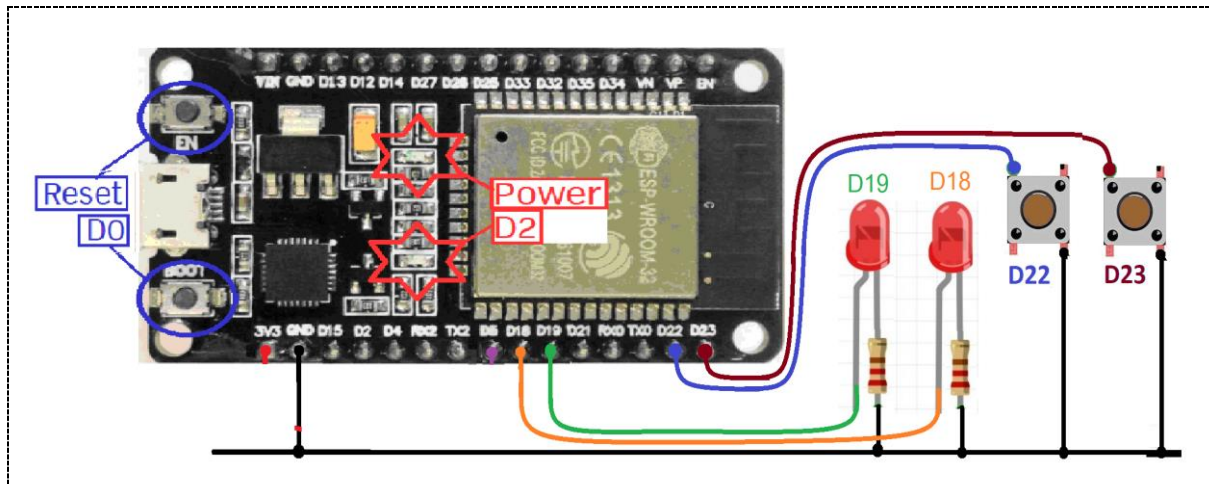


การใช้งาน ThingsBoard IoTs Platform เพื่อสร้างและจัดการระบบอัจฉริยะ
ThingsBoard IoTs Platform for smart system

ชื่อ-สกุล : นายรชพล พงศ์กิตติศักดิ์

Quiz_101 – กดติด กดดับ 2 ชุด

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



Code

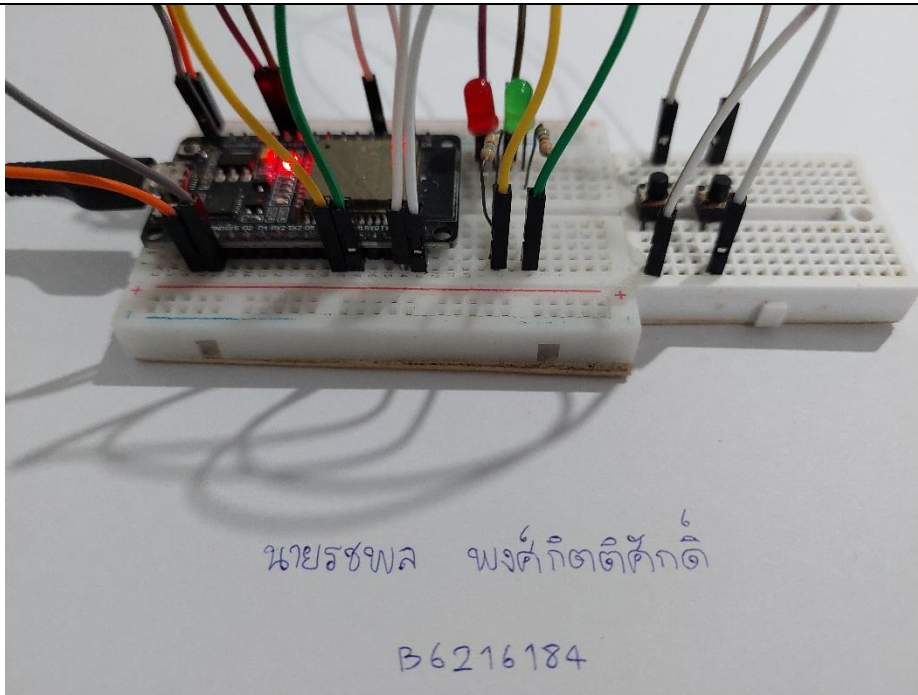
```
#define LED1 19
#define LED2 21
#define SW1 22
#define SW2 23
bool LED_Status[] = {0, 0};

void setup() {
  pinMode(LED1, OUTPUT);
  pinMode(LED2, OUTPUT);
  pinMode(SW1, INPUT_PULLUP);
  pinMode(SW2, INPUT_PULLUP);
}

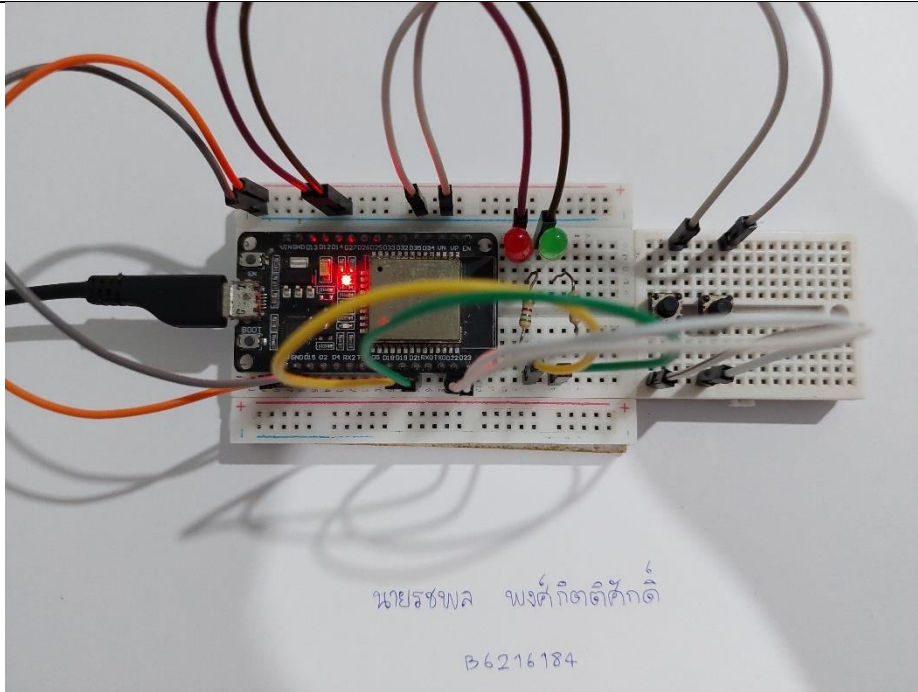
void loop() {
  if(digitalRead(SW1) == LOW) {
    delay(20);
    LED_Status[0] = 1 - LED_Status[0];
```

```
digitalWrite(LED1, LED_Status[0]);  
while(digitalRead(SW1) == LOW) delay(20);  
}  
  
if(digitalRead(SW2) == LOW) {  
    delay(20);  
    LED_Status[1] = 1 - LED_Status[1];  
    digitalWrite(LED2, LED_Status[1]);  
    while(digitalRead(SW2) == LOW) delay(20);  
}  
}
```

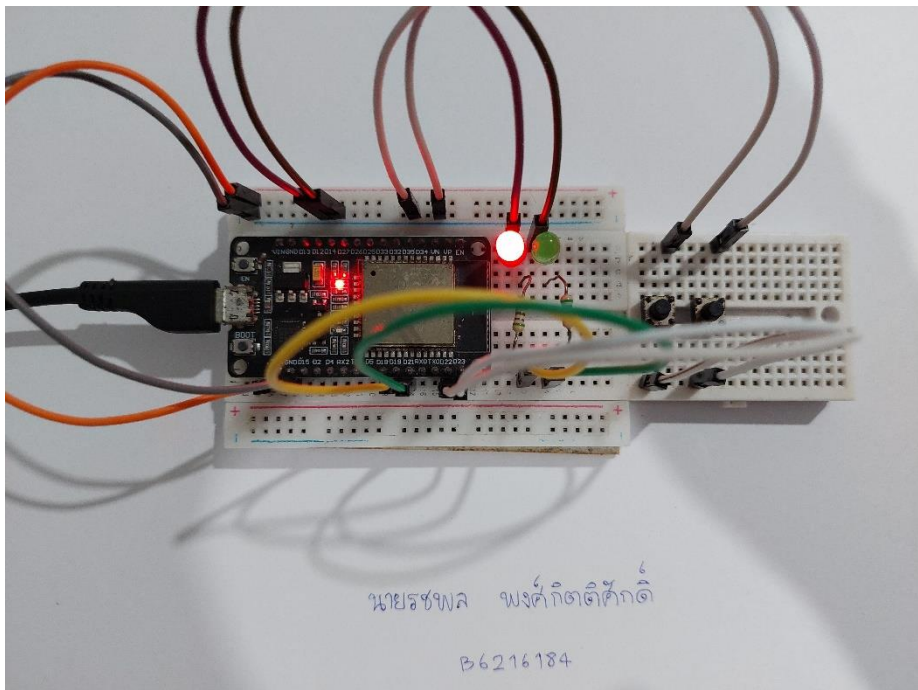
รูปการต่อวงจร 1 – ด้านข้าง



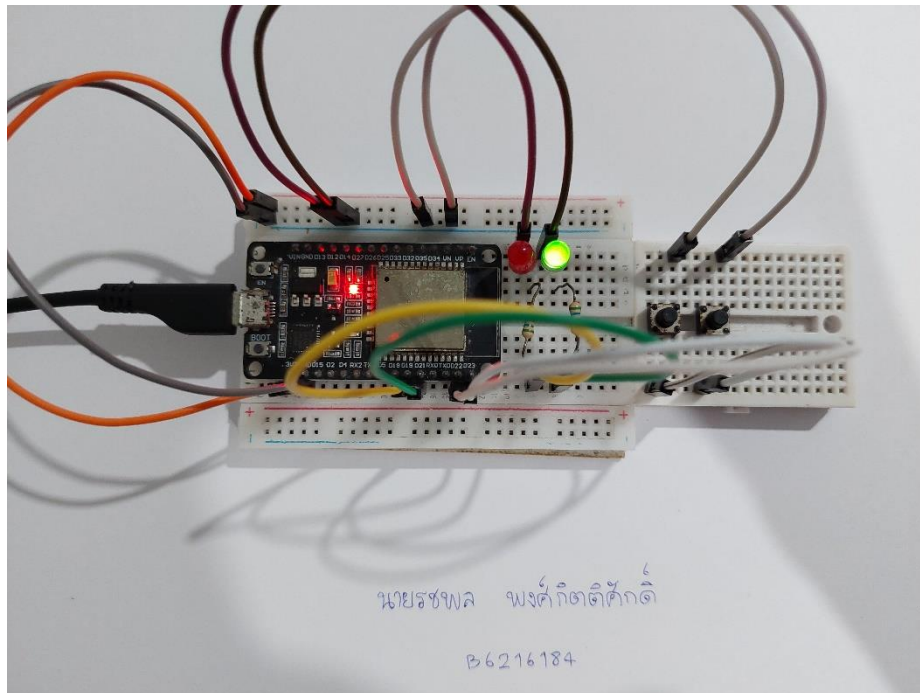
รูปการต่อวงจร 2 – ด้านบน



รูปการทดสอบ 1 – เมื่อกดสวิตช์ด้านซ้ายมือ 1 ครั้ง



รูปการทดสอบ 2 – เมื่อกดสวิตช์ด้านซ้ายมือ 1 ครั้ง และกดสวิตช์ด้านขวามือ 1 ครั้ง



Quiz_102 – Web Control 4 LED and Monitor Humid/Temperature

- เพิ่มเติมจาก Q202 อยากได้ปุ่มสำหรับคุมปิด-เปิด หลอดไฟ LED 4 ดวง
- อยากมีกด Link ไปที่หน้า FB ของตัวเอง
- https://www.colorhexa.com/008cba?fbclid=IwAR3dlZ_gRgDWmREmnzknLbMxV3pOHY4YIPuLEz8-ZzTOX2VhWxcH2QjLGk

**Code – Main**

```
#include <WiFi.h>
#include <WiFiClient.h>
#include <WebServer.h>
#include <Wire.h>
#include "SHTC3.h"
#include "index.h" //Our HTML webpage contents with javascripts
#define WIFI_NAME "your-wifi-name"
#define WIFI_PASS "your-wifi-pass"

WebServer server(80); //Server on port 80
SHTC3 sht(Wire);
byte LED[] = {5, 18, 19, 21};
String LED_State[] = {"NA", "NA", "NA", "NA"};

void handleRoot() {
  String s = MAIN_page; //Read HTML contents
  server.send(200, "text/html", s); //Send web page
}
```

```

void handleADC() {
  sht.begin();
  sht.sample();
  float h = sht.readHumidity();
  float t = sht.readTempC();
  if(t < -40) return;
  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 led_state = server.arg("LEDstate"); //Refer xhttp.open("GET",
  "setLED?LEDstate="+led, true);
  Serial.println(led_state);
  if (led_state == "11") {
    digitalWrite(LED[0], HIGH); //Feedback parameter
    LED_State[0] = "ON";
  }
  if (led_state == "10") {
    digitalWrite(LED[0], LOW); //Feedback parameter
    LED_State[0] = "OFF";
  }
  if (led_state == "21") {
    digitalWrite(LED[1], HIGH); //Feedback parameter
    LED_State[1] = "ON";
  }
  if (led_state == "20") {
    digitalWrite(LED[1], LOW); //Feedback parameter
    LED_State[1] = "OFF";
  }
  if (led_state == "31") {
    digitalWrite(LED[2], HIGH); //Feedback parameter
    LED_State[2] = "ON";
  }
  if (led_state == "30") {
    digitalWrite(LED[2], LOW); //Feedback parameter
    LED_State[2] = "OFF";
  }
}

```



```

if (led_state == "41") {
    digitalWrite(LED[3], HIGH); //Feedback parameter
    LED_State[3] = "ON";
}
if (led_state == "40") {
    digitalWrite(LED[3], LOW); //Feedback parameter
    LED_State[3] = "OFF";
}
server.send(200, "text/plain", LED_State[0]+", "+LED_State[1]+",
"+LED_State[2]+", "+LED_State[3]); //Send web page
}

void getLEDstate() {
    server.send(200, "text/plain", LED_State[0]+", "+LED_State[1]+",
"+LED_State[2]+", "+LED_State[3]);
}

void setup() {
    Serial.begin(115200);
    Wire.begin(22, 23);
    for(int i = 0 ; i < 4 ; i++) {
        pinMode(LED[i], OUTPUT);
    }
    Serial.print("\n\nConnect to ");
    Serial.println(WIFI_NAME);
    WiFi.begin(WIFI_NAME, WIFI_PASS);
    while (WiFi.status() != WL_CONNECTED) {
        delay(500); Serial.print(".");
    }
    Serial.print("\nConnected ");
    Serial.println(WIFI_NAME);
    Serial.print("IP address: ");
    Serial.println(WiFi.localIP());
    server.on("/", handleRoot);
    server.on("/setLED", handleLED);
    server.on("/readADC", handleADC);
    server.on("/getLED", getLEDstate);
    server.begin();
    Serial.println("HTTP server started");
}

```

```
void loop() {
  server.handleClient(); //Handle client requests
}
```

ที่ Highlight สีฟ้าไว้ให้เปลี่ยนเป็น WiFi Name และ WiFi Password ของตนเองได้

Code - Main กับ Code - Index ต้องอยู่ใน Folder เดียวกัน

Code – Index

```
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);
height: 40px; width: 100px">LED1 ON</button>
<button type="button" onclick="sendData(21)" style="background: rgb(202, 60,
60);
height: 40px; width: 100px">LED2 ON</button>
<button type="button" onclick="sendData(31)" style="background: rgb(202, 60,
60);
height: 40px; width: 100px">LED3 ON</button>
<button type="button" onclick="sendData(41)" style="background: rgb(202, 60,
60);
height: 40px; width: 100px">LED4 ON</button><br><br>
<button type="button" onclick="sendData(10)" style="background:
rgb(100,116,255);
height: 40px; width: 100px">LED1 OFF</button>
<button type="button" onclick="sendData(20)" style="background:
rgb(100,116,255);
height: 40px; width: 100px">LED2 OFF</button>
<button type="button" onclick="sendData(30)" style="background:
rgb(100,116,255);
height: 40px; width: 100px">LED3 OFF</button>
<button type="button" onclick="sendData(40)" style="background:
rgb(100,116,255);
height: 40px; width: 100px">LED4 OFF</button><br><br>
State of [LED1, LED2, LED3, LED4] is >> <span id="LEDState">NA, NA, NA,
NA</span><br>
```



```

</div>
<div>
<br>SHTC-3 sensor : <span id="ADCValue">0</span><br>
</div>
<script>
function ready() {
    var xhttp = new XMLHttpRequest();
    xhttp.onreadystatechange = function() {
        if(this.responseText) {
            document.getElementById("LEDState").innerHTML = this.responseText;
        }
    };
    xhttp.open("GET", "getLED", true);
    xhttp.send();
}

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();
}

document.addEventListener("DOMContentLoaded", ready);

```

```

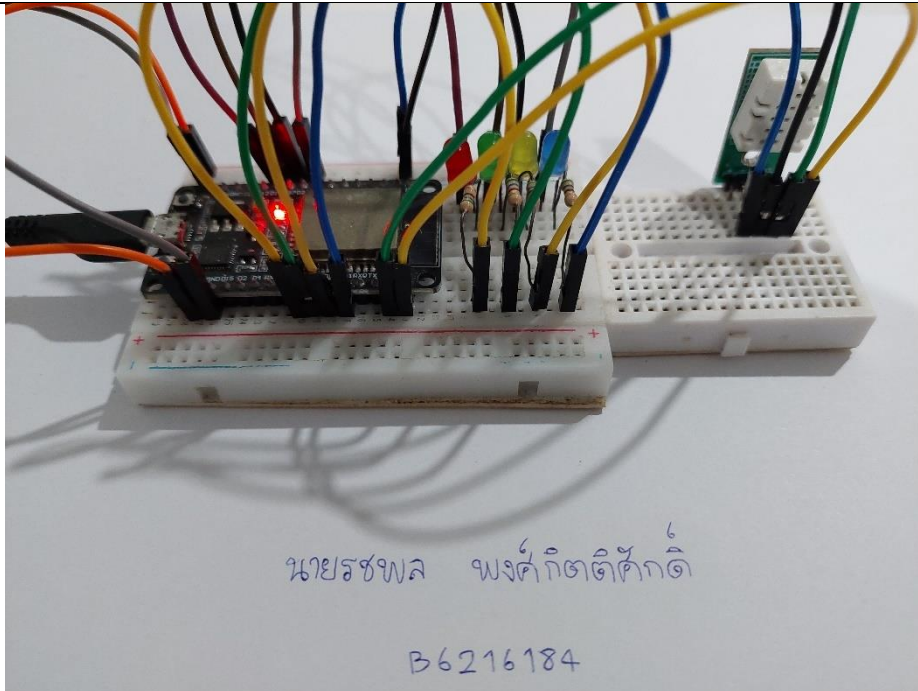
</script>
<br><a href="https://www.facebook.com/profile.php?id=your-id">By
Me</a>
</body>
</html>
)=====";

```

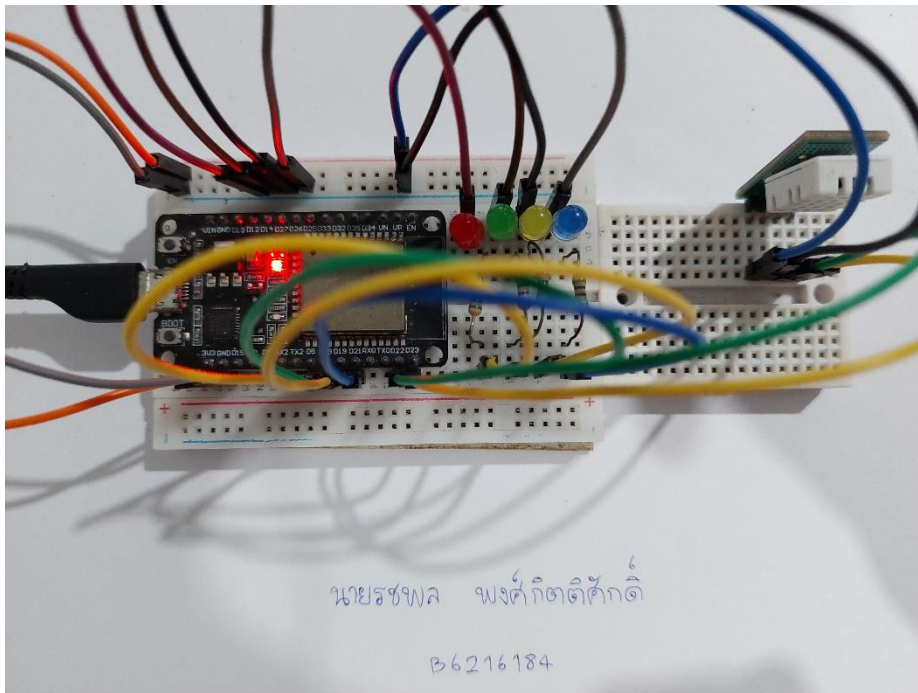
ที่ Highlight สีฟ้าไว้สามารถเปลี่ยนเป็น Facebook ของตนเองได้

Code - Main กับ Code - Index ต้องอยู่ใน Folder เดียวกัน

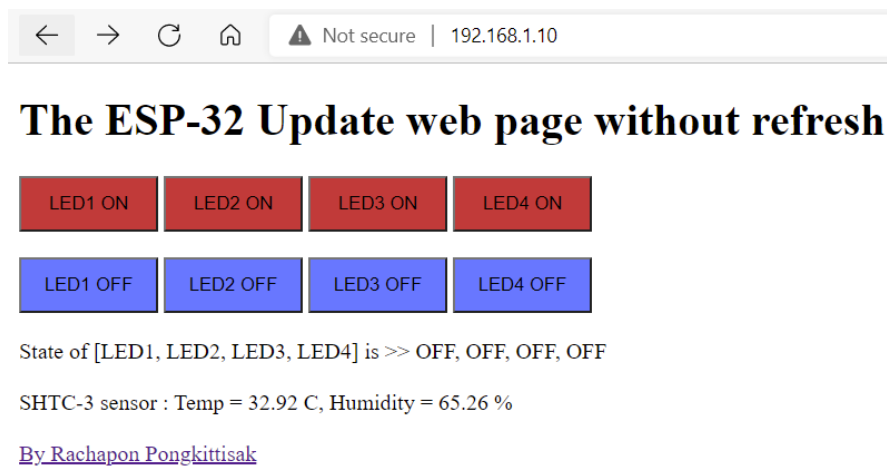
รูปการต่อวงจร 1 – ด้านข้าง



รูปการต่อวงจร 2 – ด้านบน



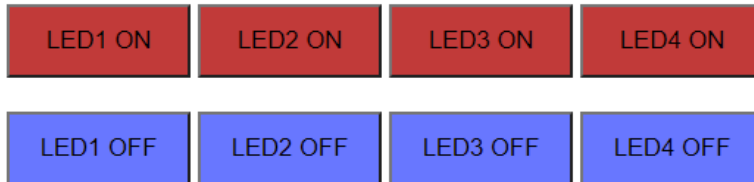
รูปถ่ายหน้า Web Browser – เมื่อใช้ Web Browser เข้าไปตามเลข IP Address ของ ESP32



IP Address ของ ESP32 จะแตกต่างกันไปขึ้นอยู่กับหลายปัจจัย ดังนั้นจึงต้องทำการตรวจสอบ IP Address ของ ESP32 ก่อนทุกครั้งเมื่อ ESP32 เริ่มทำงาน

รูปการทดสอบ 1 – กดเปิด/ปิด LED

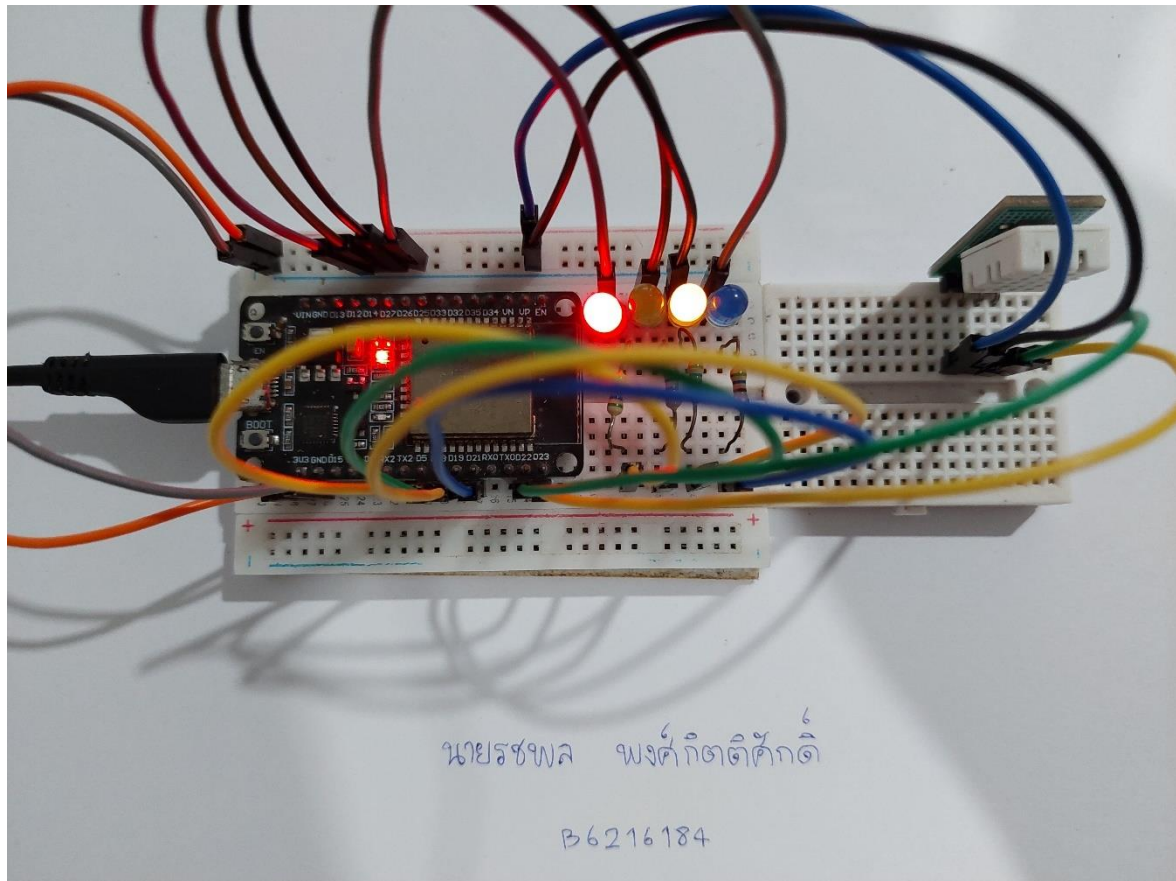
The ESP-32 Update web page without refresh



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

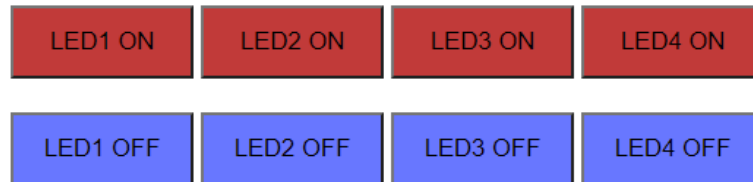
SHTC-3 sensor : Temp = 32.95 C, Humidity = 65.62 %

By [Rachapon Pongkittisak](#)



รูปการทดสอบ 2 – กดเปิด/ปิด LED

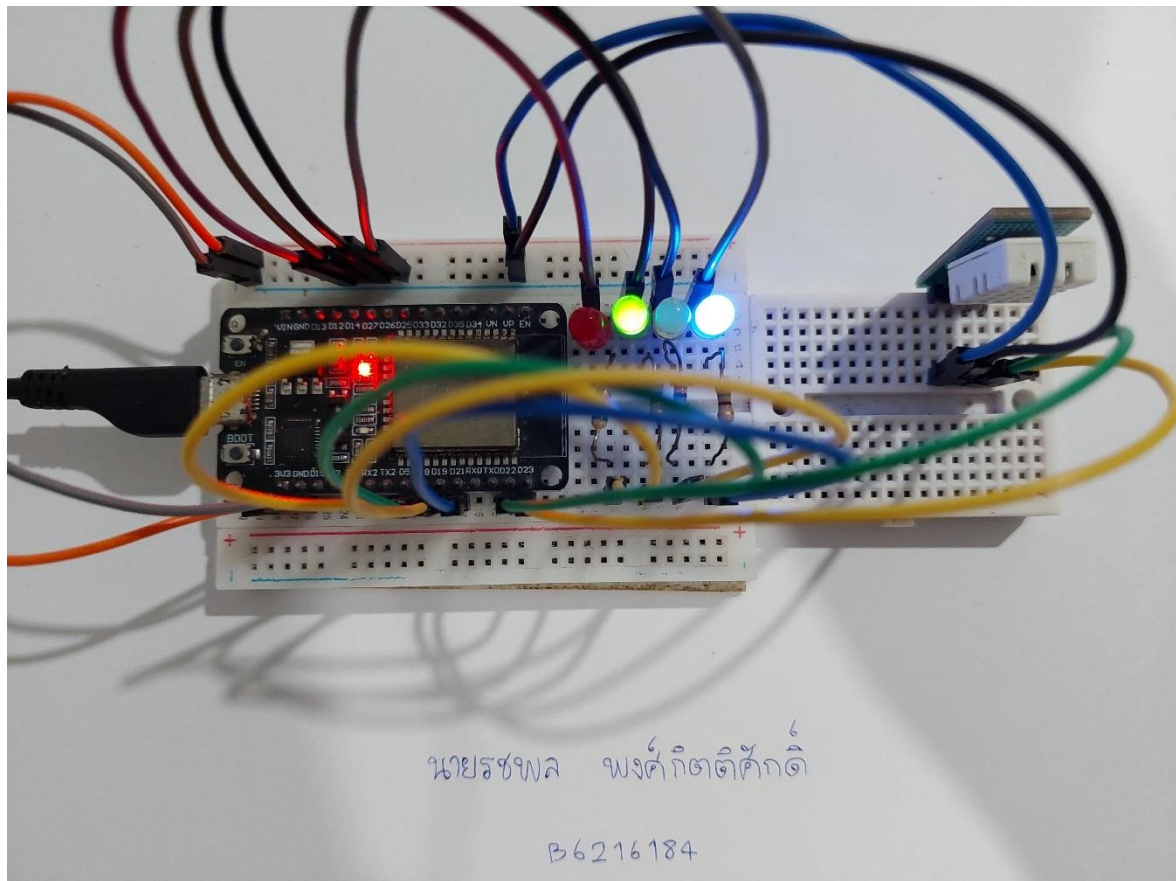
The ESP-32 Update web page without refresh



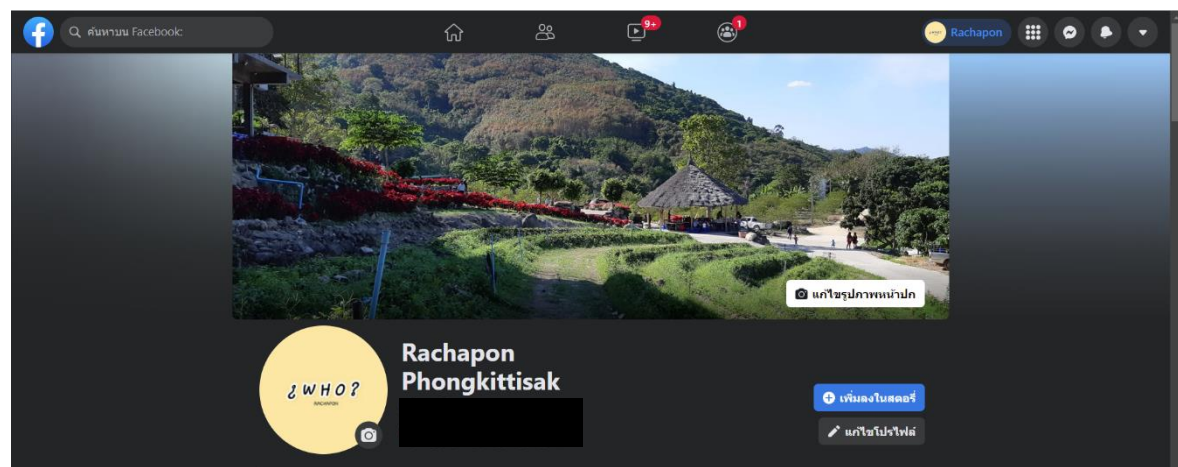
State of [LED1, LED2, LED3, LED4] is >> OFF, ON, OFF, ON

SHTC-3 sensor : Temp = 33.11 C, Humidity = 65.05 %

By [Rachapon Pongkittisak](#)

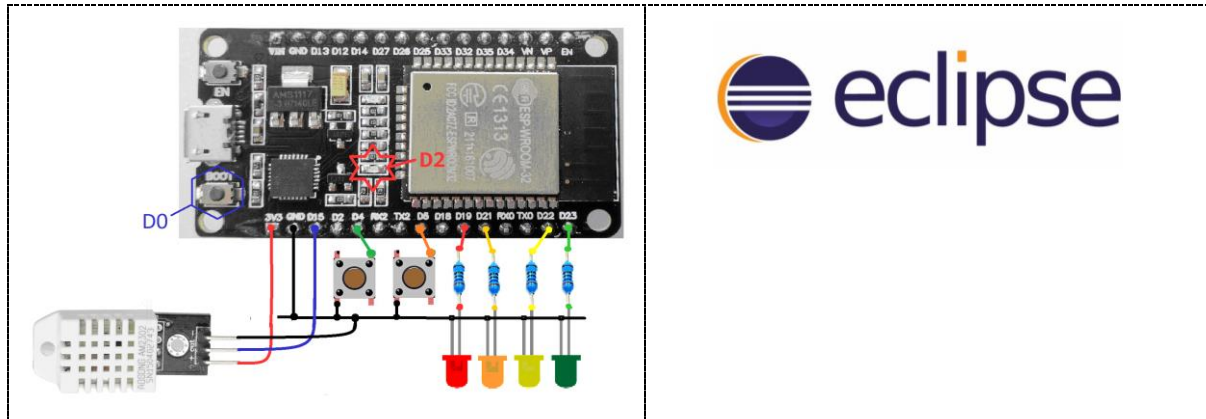


รูปการทดสอบ 3 – กดเข้าไปที่ Link Facebook ของตนเอง



Quiz_103 – Pub/Sub Data from (DHT22 + 4 LED + 2 Switch)

- อ่านค่า DHT-22 แล้วส่งไปยัง MQTT Broker ทุกๆ 5 วินาที
- กำหนดให้ใช้ mqtt.eclipse.org เป็น Broker (ใช้ mqtt.eclipseprojects.io เป็น Broker แทน)
- ควบคุมการปิดเปิด 4 LED
- รับค่าสวิตช์กำหนด SW1 แจ้ง Overheat Alarm, SW2 แจ้ง Intruders Alarm



Code

```
#include <WiFi.h>
#include <PubSubClient.h>
#include <Wire.h>
#include "SHTC3.h"
#define WIFI_NAME "your-wifi-name"
#define WIFI_PASS "your-wifi-pass"
#define MQTT_SERVER "mqtt.eclipseprojects.io"
#define MQTT_PORT 1883
#define TOPIC1 "LED1_SxWIBVzDd2ufJxmIWngGzwl9MSD"
#define TOPIC2 "LED2_os28V4zQSGQOAdRTIwQJdWna2Ws"
#define TOPIC3 "LED3_IgoItb5llzdm3SQ9T2a2Vim4xHq"
#define TOPIC4 "LED4_J04ZcZCQL0rBTsPFEosIWtQ2LrA"
#define TOPIC5 "Temp_CPQUI7Vv8AzQ0QaX1AopwhgZJMe"
#define TOPIC6 "Humid_ORic62VRNxGbKOMzgdVAICl0S5q"
#define TOPIC7 "Alarm1_AzBAF6LbSmcrUExityukWg6MEHe"
#define TOPIC8 "Alarm2_QMGLPksPUMerF7E8neUVKxIrng3"

byte LED[] = {19, 21, 22, 23};
byte button[] = {4, 5};

SHTC3 sht(Wire);
WiFiClient espClient;
```



```

PubSubClient client(espClient);

void setup() {
  Serial.begin(115200);
  Wire.begin(15, 18);
  for(int i = 0 ; i < 4 ; i++) {
    if(i < 2) pinMode(button[i], INPUT_PULLUP);
    pinMode(LED[i], OUTPUT);
  }
  WiFi.begin(WIFI_NAME, WIFI_PASS);
  while(WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  randomSeed(micros());
  Serial.println("\nWiFi Connected");
  Serial.print("IP Address: ");
  Serial.println(WiFi.localIP());
  client.setServer(MQTT_SERVER, MQTT_PORT);
  client.setCallback(controlLED);
  xTaskCreatePinnedToCore(
    alarm,    // Task function
    "alarm",  // name of task
    1000,    // Stack size of task
    NULL,    // parameter of task
    1,       // priority of the task
    NULL,    // Task handle to keep track of created task
    1        // pin task to core 1
  );
}

void reconnect() {
  while(!client.connected()) {
    Serial.print("Attempting MQTT connection...");
    String clientID = "ESP32 Client-";
    clientID += String(random(0xffff), HEX);

    if(client.connect(clientID.c_str())) {
      Serial.println("connected");
      client.subscribe(TOPIC1);
      client.subscribe(TOPIC2);
    }
  }

```

```

    client.subscribe(TOPIC3);
    client.subscribe(TOPIC4);
}
else {
    Serial.print("\nfailed, rc=");
    Serial.print(client.state());
    Serial.println("\ntry again in 5 seconds");
    delay(5000);
}
}
}

void alarm(void *pvParameters) {
    while(true) {
        if(digitalRead(button[0]) == LOW) {
            vTaskDelay(20/portTICK_PERIOD_MS);
            client.publish(TOPIC7, "Overheat Alarm");
            Serial.println("Alarm: Overheat Alarm");
            while(digitalRead(button[0]) == LOW) vTaskDelay(20/portTICK_PERIOD_MS);
        }
        if(digitalRead(button[1]) == LOW) {
            vTaskDelay(20/portTICK_PERIOD_MS);
            client.publish(TOPIC8, "Intruders Alarm");
            Serial.println("Alarm: Intruders Alarm");
            while(digitalRead(button[1]) == LOW) vTaskDelay(20/portTICK_PERIOD_MS);
        }
    }
}

void controlled(char *topic, byte *payload, unsigned int length) {
    char localPayload[length + 1];
    Serial.println("Message arrived ["+String(topic)+"]");
    Serial.print("-> ");
    for(int i = 0 ; i < length ; i++) {
        localPayload[i] = (char)payload[i];
        localPayload[i+1] = '\0';
        Serial.print(localPayload[i]);
    }
    Serial.println();

    if(String(topic) == String(TOPIC1)) {

```

```

    if(String(localPayload) == "ON") digitalWrite(LED[0], HIGH);
    if(String(localPayload) == "OFF") digitalWrite(LED[0], LOW);
  }
  if(String(topic) == String(TOPIC2)) {
    if(String(localPayload) == "ON") digitalWrite(LED[1], HIGH);
    if(String(localPayload) == "OFF") digitalWrite(LED[1], LOW);
  }
  if(String(topic) == String(TOPIC3)) {
    if(String(localPayload) == "ON") digitalWrite(LED[2], HIGH);
    if(String(localPayload) == "OFF") digitalWrite(LED[2], LOW);
  }
  if(String(topic) == String(TOPIC4)) {
    if(String(localPayload) == "ON") digitalWrite(LED[3], HIGH);
    if(String(localPayload) == "OFF") digitalWrite(LED[3], LOW);
  }
}

void loop() {
  if(!client.connected()) reconnect();
  client.loop();
  sht.begin();
  sht.sample();
  float t = sht.readTempC();
  float h = sht.readHumidity();
  if(t >= -40 && t <= 125) {
    client.publish(TOPIC5, String(t).c_str());
    client.publish(TOPIC6, String(h).c_str());
    Serial.println("temperature: " + String(t) + ", humidity: " + String(h));
  }
  delay(5000);
}

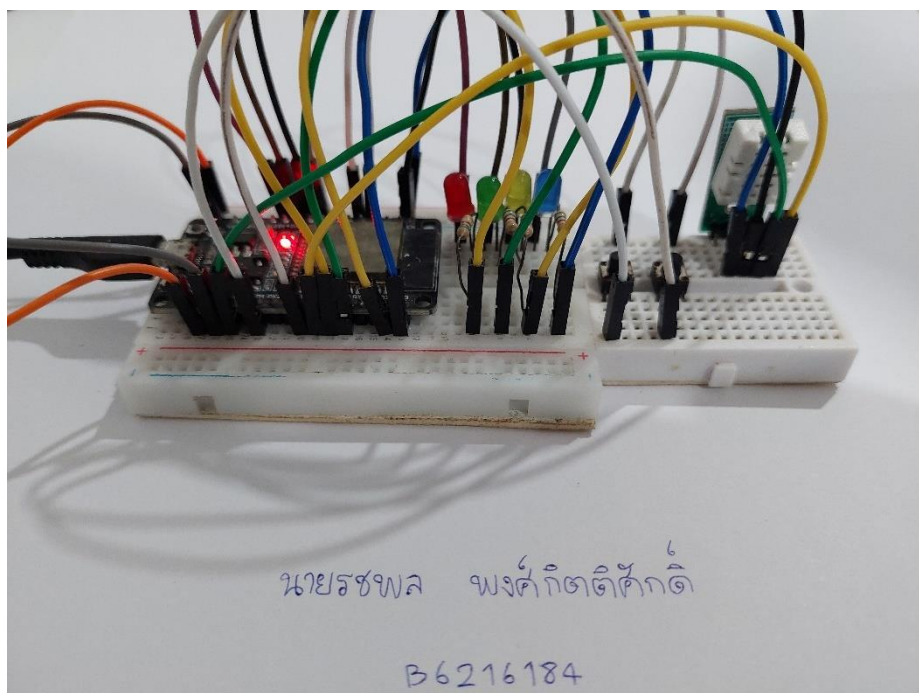
```

เนื่องจาก mqtt.eclipse.org ยกเลิกใช้งานไปแล้ว จึงเปลี่ยนไปใช้เป็น mqtt.eclipseprojects.io แทน

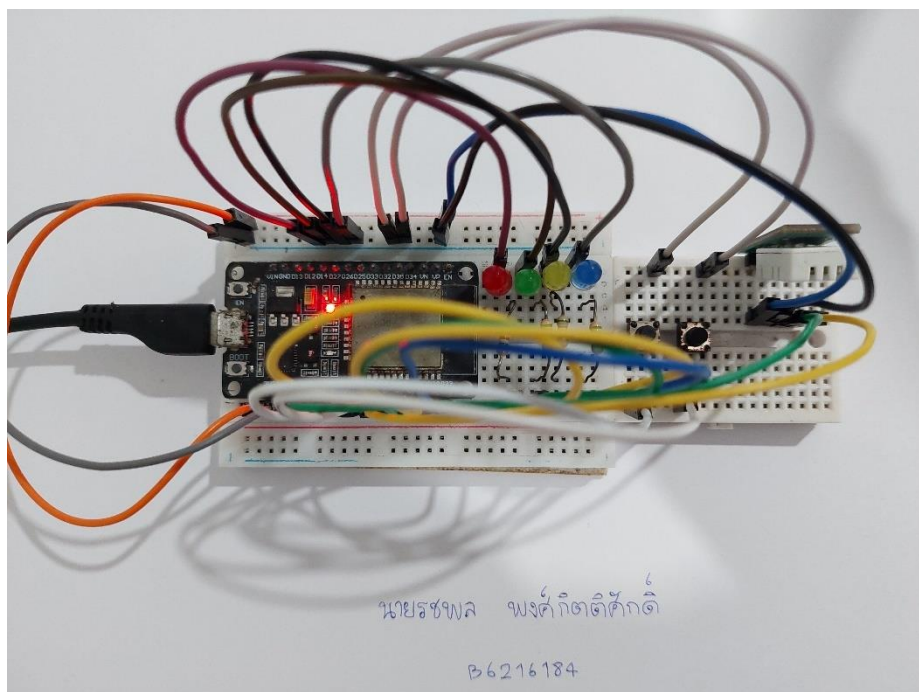
ที่ Highlight สีฟ้าไว้ให้เปลี่ยนเป็น WiFi Name และ WiFi Password ของตนเองได้

ชื่อ Topic สามารถเปลี่ยนได้ตามอิสระ

รูปการต่อวงจร 1 – ด้านข้าง



รูปการต่อวงจร 2 – ด้านบน



รูปหน้าจอ MQTT Lens – ส่งข้อมูลการเปิด/ปิด LED ทั้ง 4 ไปยัง ESP32

Topic: "LED1_SxWIBVzDd2ufJxmiWngGzwI9MSD" Showing the last 1 messages — + Messages: 0/4

Time Topic QoS
3 9:21:24 LED1_SxWIBVzDd2ufJxmiWngGzwI9MSD 0

Message: ON

Topic: "LED2_os28V4zQSGQOAdRTlwQJdWna2Ws" Showing the last 1 messages — + Messages: 0/4

Time Topic QoS
3 9:21:28 LED2_os28V4zQSGQOAdRTlwQJdWna2Ws 0

Message: ON

Topic: "LED3_Igoltb5llzdm3SQ9T2a2Vim4xHq" Showing the last 1 messages — + Messages: 0/4

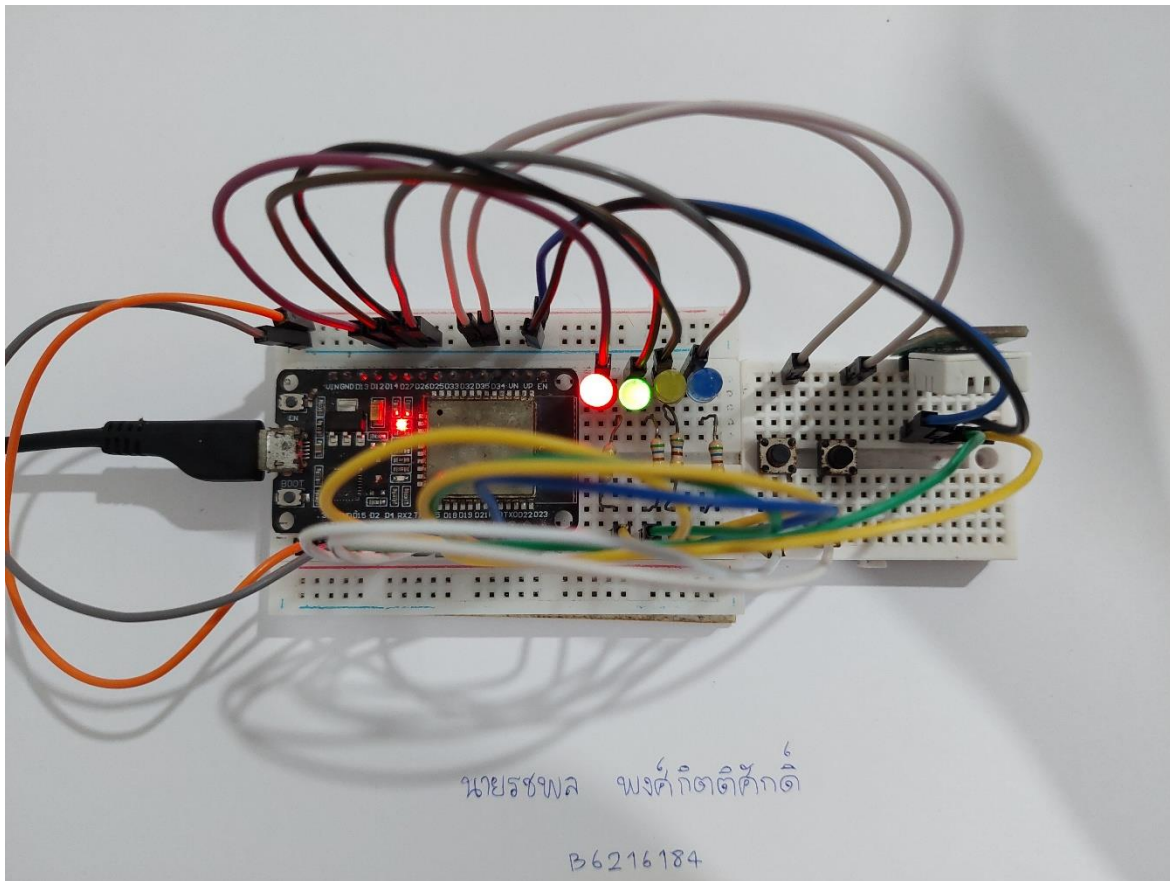
Time Topic QoS
3 9:21:36 LED3_Igoltb5llzdm3SQ9T2a2Vim4xHq 0

Message: OFF

Topic: "LED4_JO4ZcZCQL0rBTsPFEosiWtQ2LrA" Showing the last 1 messages — + Messages: 0/4

Time Topic QoS
3 9:21:40 LED4_JO4ZcZCQL0rBTsPFEosiWtQ2LrA 0

Message: OFF



รูปหน้าจอ MQTT Lens – ส่งข้อมูลการเปิด/ปิด LED ทั้ง 4 ไปยัง ESP32

Topic: "LED1_SxWIBVzDd2ufJxmiWngGzwI9MSD" Showing the last 1 messages — + Messages: 0/5

Time Topic QoS
4 9:22:40 LED1_SxWIBVzDd2ufJxmiWngGzwI9MSD 0

Message: OFF

Topic: "LED2_os28V4zQSGQOAdRTlwQJdWna2Ws" Showing the last 1 messages — + Messages: 0/5

Time Topic QoS
4 9:22:44 LED2_os28V4zQSGQOAdRTlwQJdWna2Ws 0

Message: OFF

Topic: "LED3_Igoltb5llzdm3SQ9T2a2Vim4xHq" Showing the last 1 messages — + Messages: 0/5

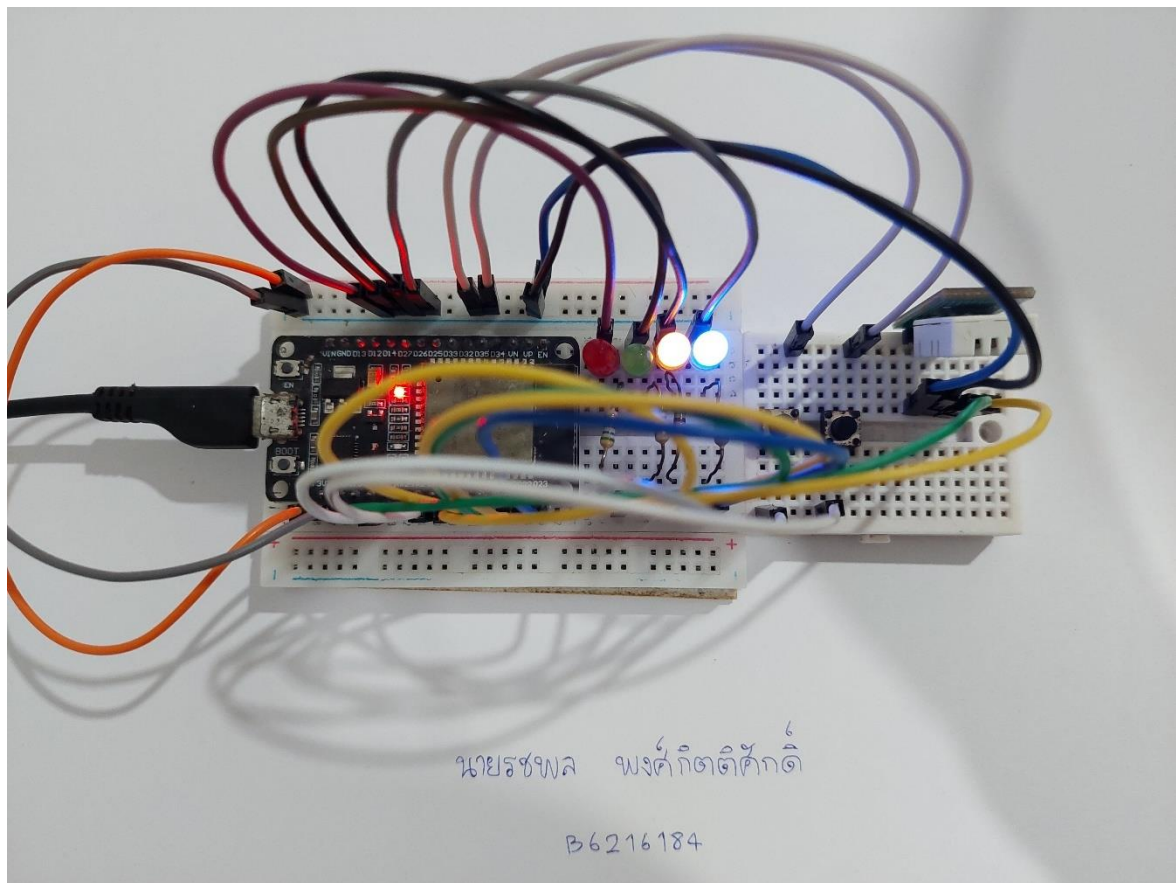
Time Topic QoS
4 9:22:49 LED3_Igoltb5llzdm3SQ9T2a2Vim4xHq 0

Message: ON

Topic: "LED4_JO4ZcZCQL0rBTsPFEosiWtQ2LrA" Showing the last 1 messages — + Messages: 0/5

Time Topic QoS
4 9:22:53 LED4_JO4ZcZCQL0rBTsPFEosiWtQ2LrA 0

Message: ON



หน้าจอ MQTT Lens – ส่งข้อมูลอุณหภูมิ และความชื้นไป MQTT Server

Topic: "Temp_CPQUI7Vv8AzQ0QaX1AopwhgZJMe" Showing the last 1 messages — + Messages: 0/494

#	Time	Topic	QoS
493	9:26:24	Temp_CPQUI7Vv8AzQ0QaX1AopwhgZJMe	0

Message: 31.93

Topic: "Humid_ORic62VRNxGbK0mzgDVAICl0S5q" Showing the last 1 messages — + Messages: 0/493

#	Time	Topic	QoS
492	9:26:24	Humid_ORic62VRNxGbK0mzgDVAICl0S5q	0

Message: 72.96

หน้าจอ MQTT Lens – ส่งข้อมูลการกดสวิตช์ทั้ง 2 ไป MQTT Server

Topic: "Alarm1_AzBAF6LbSmcrUEXityukWg6MEHe" Showing the last 1 messages — + Messages: 0/4

#	Time	Topic	QoS
3	9:26:47	Alarm1_AzBAF6LbSmcrUEXityukWg6MEHe	0

Message: Overheat Alarm

Topic: "Alarm2_QMGLPkSPUMerF7E8neUVKxlrng3" Showing the last 1 messages — + Messages: 0/5

#	Time	Topic	QoS
4	9:26:48	Alarm2_QMGLPkSPUMerF7E8neUVKxlrng3	0

Message: Intruders Alarm

SW1 ส่งข้อมูลไปที่ Topic: Alarm1 และ SW2 ส่งข้อมูลไปที่ Topic: Alarm2

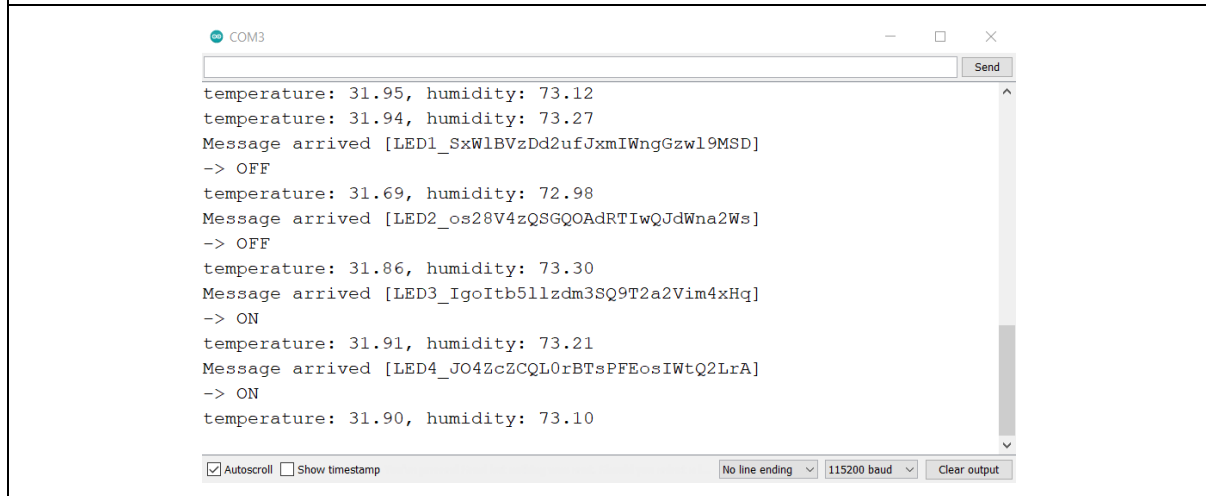
หน้าจอ Serial Monitor – ใช้ตรวจสอบการส่งข้อมูลการเปิด/ปิด LED ทั้ง 4 ไปยัง ESP32

COM3

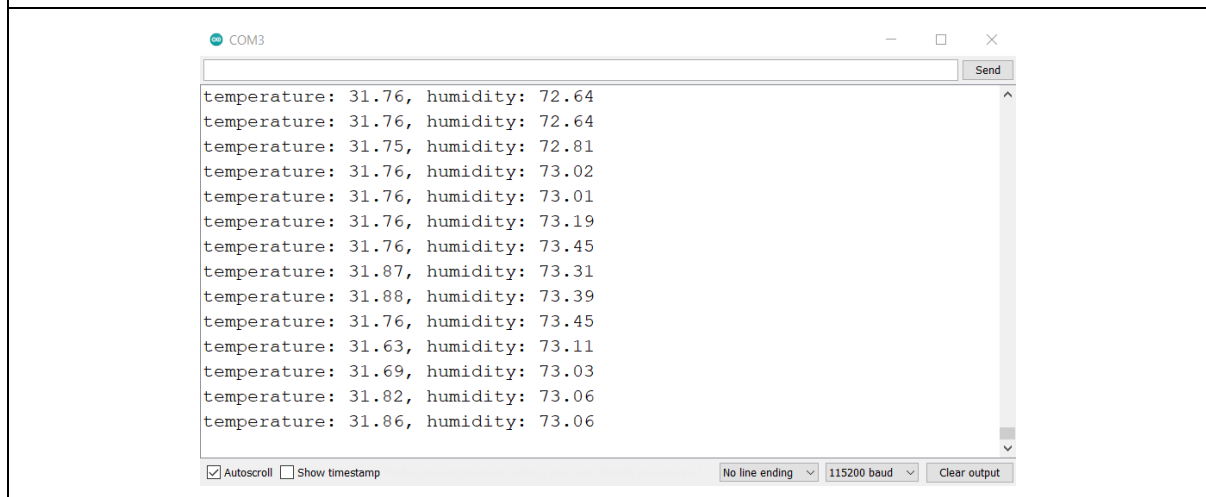
temperature: 31.82, humidity: 73.06
 temperature: 31.83, humidity: 72.89
 Message arrived [LED1_SxWlBVzDd2ufJxmIWngGzw19MSD]
 -> ON
 temperature: 31.76, humidity: 73.11
 Message arrived [LED2_os28V4zQSGQOAdRTIwQJdWna2Ws]
 -> ON
 temperature: 31.82, humidity: 73.19
 Message arrived [LED3_IgoItb5llzdm3SQ9T2a2Vim4xHq]
 -> OFF
 temperature: 31.86, humidity: 73.20
 Message arrived [LED4_JO4ZcZCQL0rBTsPFEosIWtQ2LrA]
 -> OFF
 temperature: 31.86, humidity: 73.35

☒ Autoscroll ☐ Show timestamp

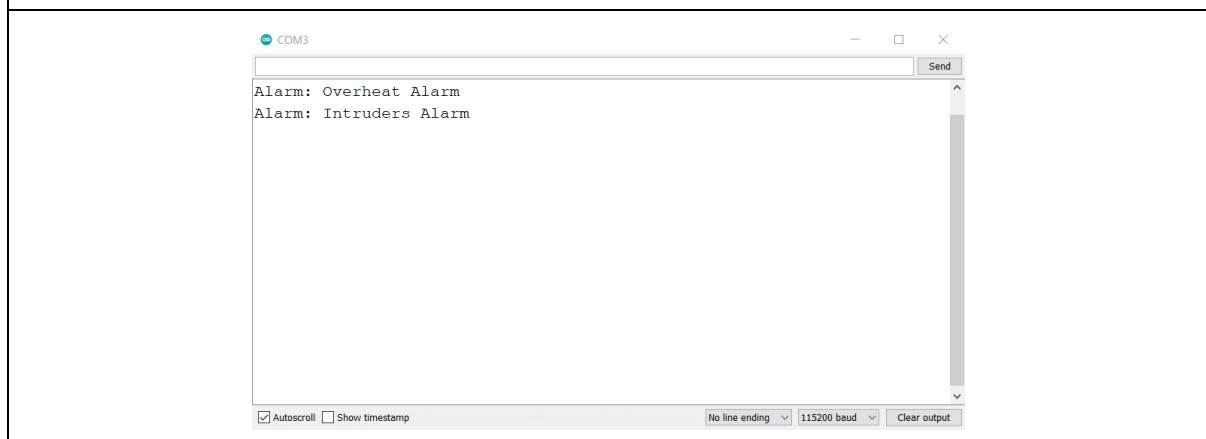
รูปหน้าจอ Serial Monitor – ใช้ตรวจสอบการส่งข้อมูลการเปิด/ปิด LED ทั้ง 4 ไปยัง ESP32



รูปหน้าจอ Serial Monitor – ใช้ตรวจสอบการอ่านข้อมูลอุณหภูมิ และความชื้น

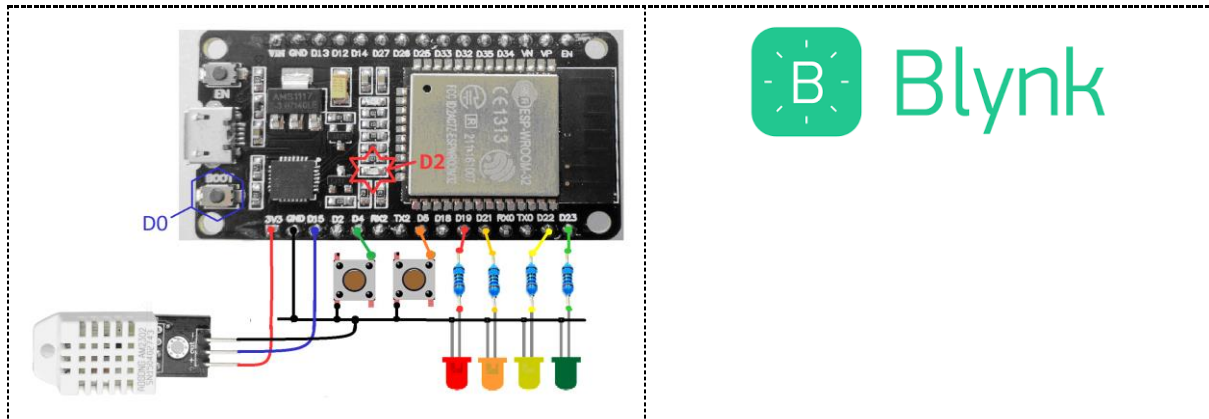


รูปหน้าจอ Serial Monitor – ใช้ตรวจสอบการการกดสวิตช์ทั้ง 2



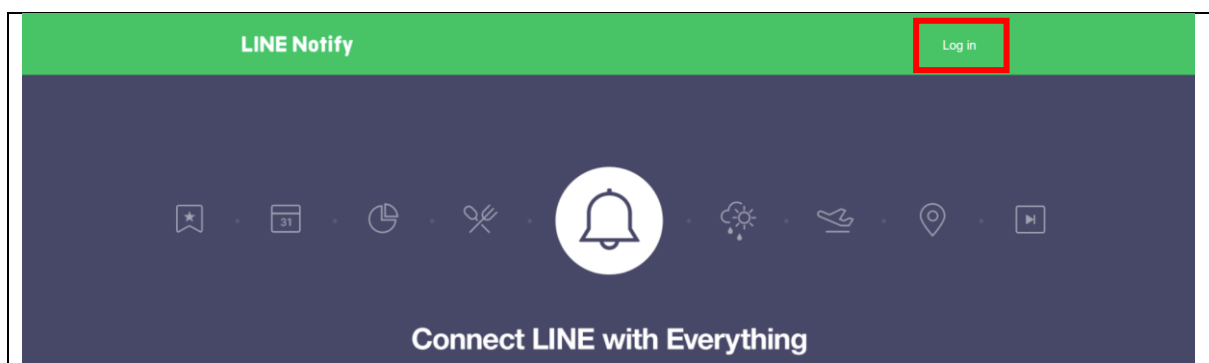
Quiz_104 – Blynk and LINE from (DHT22 + 4 LED + 2 Switch)

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

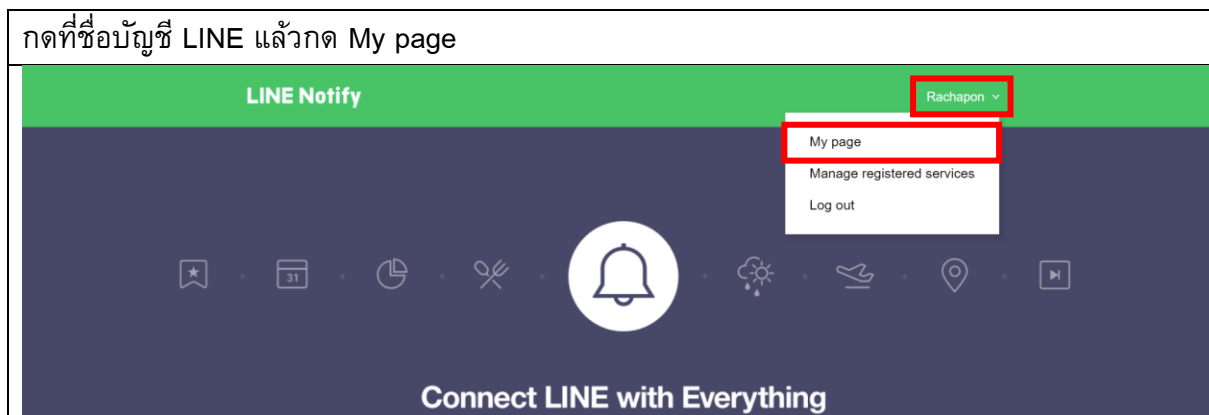


1. รับ LINE TOKEN เพื่อใช้ในการแจ้งเตือน

ไปที่ link: <https://notify-bot.line.me/en/> เพื่อรับ Token ของ LINE Notify จากนั้นให้กด Log in



กดที่ชื่อบัญชี LINE แล้วกด My page



เลื่อนหน้า Web ลงมาด้านล่างสุด แล้วกด Generate token

Generate access token (For developers)

By using personal access tokens, you can configure notifications without having to add a web service

Generate token

LINE Notify API Document

ตั้งชื่อเป็น Alarm แล้วกดเลือก 1-on1 chat with LINE Notify จากนั้นกด Generate token

Generate token

Please enter a token name to be displayed before each notification.

Alarm

Select a chat to send notifications to.

Search by group name

1-on-1 chat with LINE Notify

Note: Revealing your personal access token can allow a third party to obtain the names of your connected chats as well as your profile name.

Generate token

กดปุ่ม Copy เพื่อ Copy token เก็บเอาไว้

Your token is:

GPpdLgaC48s

RU6s13I

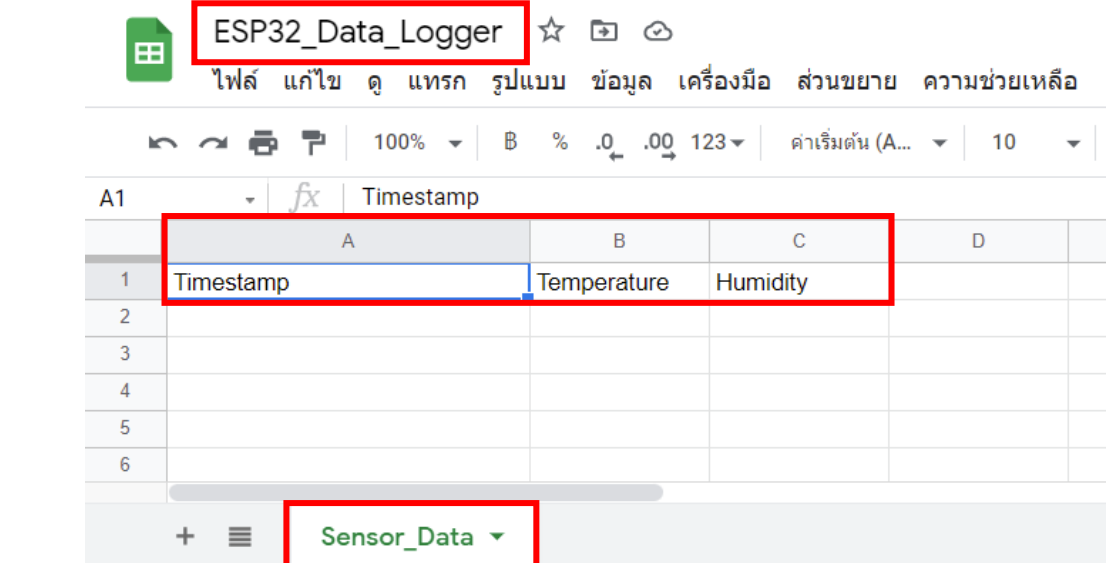
If you leave this page, you will not be able to view your newly generated token again. Please copy the token before leaving this page.

Copy

Close

2. สร้าง Google Spreadsheet สำหรับใช้เป็นที่เก็บข้อมูล

สร้าง Google Spreadsheet ให้มีรูปแบบตามนี้



The screenshot shows a Google Spreadsheet titled "ESP32_Data_Logger". The sheet name is "Sensor_Data". The columns are labeled as follows:

Column	Label
A	Timestamp
B	Temperature
C	Humidity

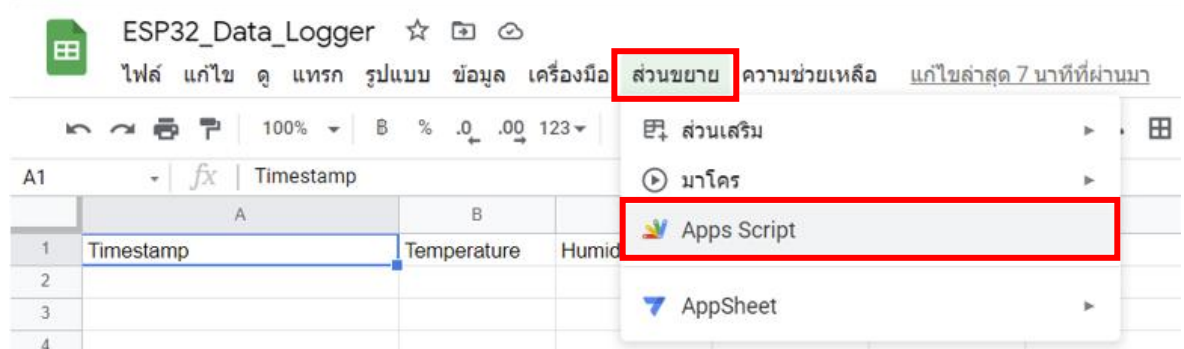
The sheet name "Sensor_Data" is highlighted in the bottom left corner.

Document name: ESP32_Data_Logger

Sheet name: Sensor_Data

Columns A: Timestamp Column B: Temperature Column C: Humidity

ไปที่ ส่วนขยาย -> Apps Script

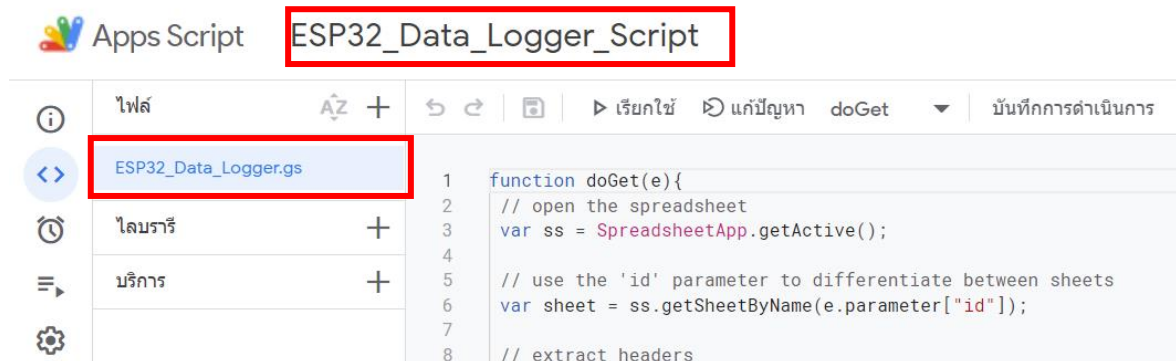


The screenshot shows the Google Spreadsheet with the "ส่วนขยาย" (Extensions) menu open. The "Apps Script" option is highlighted.

ส่วนขยาย ความช่วยเหลือ แก้ไขล่าสุด 7 นาทีที่ผ่านมา

ส่วนเสริม มาโคร Apps Script AppSheet

ตั้งชื่อตามนี้



Project name: ESP32_Data_Logger_Script

Script name: ESP32_Data_Logger

ใส่ Code ตามนี้

```
function doGet(e){
  // open the spreadsheet
  var ss = SpreadsheetApp.getActive();

  // use the 'id' parameter to differentiate between sheets
  var sheet = ss.getSheetByName(e.parameter["id"]);
  // extract headers
  // getRange accepts row, col, number_of_rows and num_of_cols as argument
  // getLastColumn returns the position of the last column that has content
  var headers = sheet.getRange(1, 1, 1, sheet.getLastColumn()).getValues()[0];

  // store the position of the last row
  var lastRow = sheet.getLastRow();
  var cell = sheet.getRange('a1');
  var col = 0;
  var d = new Date();

  for (i in headers){
    // loop through the headers and if a parameter name matches the header name
    insert the value
    if (headers[i] == "Timestamp") {
      val = d.toString() + ", " + d.toLocaleTimeString();
    }
    else {
      val = e.parameter[headers[i]];
    }
    // append data to the last row
    cell.offset(lastRow, col).setValue(val);
    col++;
  }

  return ContentService.createTextOutput('success');
}
```

ไปที่ การทำให้ใช้งานได้ -> การทำให้ใช้งานได้รายการใหม่

Apps Script ESP32_Data_Logger_Script

ไฟล์ ESP32_Data_Loggers

```

1 // open the spreadsheet
2 var ss = SpreadsheetApp.getActive();
3
4 // use the 'id' parameter to differentiate between sheets
5 var sheet = ss.getSheetByName(e.parameter['id']);
6
7 // extract headers
8

```

การทำให้ใช้งานได้

- การทำให้ใช้งานได้รายการใหม่
- จัดการการทำให้ใช้งานได้
- การนำการทดสอบไปใช้งาน

กด -> เว็บแอป

การทำให้ใช้งานได้รายการใหม่

เลือกประเภท

- การกำหนดค่า
- เว็บแอป
- ไฟล์ดำเนินการของ API
- ส่วนเสริม
- ไลบรารี

โปรดเลือกประเภทการทำให้ใช้งานได้

ยกเลิก การทำให้ใช้งานได้

ให้ตั้งค่าตามนี้ และจากนั้นให้กด การทำให้ใช้งานได้

การทำให้ใช้งานได้รายการใหม่

เลือกประเภท

- การกำหนดค่า
- เว็บแอป

รายละเอียด

คำอธิบายใหม่

เว็บแอป

ดำเนินการในฐาน

จัน (@gmail.com)

ระบบจะตรวจสอบสิทธิ์เว็บแอปเพื่ออนุญาตให้ทำงานโดยใช้ข้อมูลบัญชีของคุณ

ผู้มีสิทธิ์เข้าถึง

ทุกคน

สามารถใช้โครงการนี้เป็นไลบรารีได้ด้วย ดูข้อมูลเพิ่มเติม

ยกเลิก การทำให้ใช้งานได้

ดำเนินการในฐาน: จัน

ผู้มีสิทธิ์เข้าถึง: ทุกคน

Copy รหัสการทำงานให้ใช้งานได้(GSS_TOKEN) และ URL เก็บไว้

การทำงานให้ใช้งานได้รายการใหม่

อัปเดตการทำงานให้ใช้งานได้เรียบร้อยแล้ว

เวอร์ชัน 1 วันที่ 3 พ.ค. เวลา 22:01

รหัสการทำงานให้ใช้งานได้

AKfycbznENzdwy_I2tzO9HY617yihqMhC3OwNNG1Hsdjd3OKX6oF6NHvqn8IMQgg4-Lalh9WRA

คัดลอก

เว็บแอป

URL

https://script.google.com/macros/s/AKfycbznENzdwy_I2tzO9HY617yihqMhC3OwNNG1Hsdjd3OKX6oF6NHvqn8IMQgg4-Lalh9WRA

คัดลอก

เสร็จสิ้น


URL: <https://script.google.com/macros/s/your-GSS-token/exec>

ทดสอบส่งข้อมูลไปที่ URL โดยใช้ Web Browser

https://script.google.com/macros/s/your-GSS-token/exec?id=Sensor_Data&Temperature=100&Humidity=100

← → ↻ 🏠 🔒 https://script.googleusercontent.com/macros/echo?user_content_key=:

success

 **ESP32_Data_Logger** ☆ 📁 📄 บันทึกไปยังไดรฟ์แล้ว

ไฟล์ แก้ไข ดู แทรก รูปแบบ ข้อมูล เครื่องมือ ส่วนขยาย

🔍 🔄 📄 📑 100% ▾ B % .0_ .00 123 ▾ ค่าเริ่มต้น (A...

G17 ▾ *fx* |

	A	B	C	D
1	Timestamp	Temperature	Humidity	
2	Sat Apr 23 2022	100	100	
3				
4				
5				

3. การโปรแกรม ESP32

Code

```
#include <WiFi.h>
#include <WiFiClientSecure.h>
#include <Wire.h>
#include <TridentTD_LineNotify.h>
#include <BlynkSimpleEsp32.h>
#include "SHTC3.h"
#define WIFI_NAME "your-wifi-name"
#define WIFI_PASS "your-wifi-pass"
#define BLYNK_TOKEN "your-blynk-token"
#define LINE_TOKEN "your-line-token"
#define GSS_TOKEN "your-GSS-token"
#define GSS_SHEET "Sensor_Data"
#define HOST "script.google.com"
#define HOST_PORT 443

byte LED[] = {19, 21, 22, 23};
byte button[] = {4, 5};
float temperature = -45;
float humidity = 0;
SHTC3 sht(Wire);
WiFiClientSecure client;

void setup() {
  Serial.begin(115200);
  Wire.begin(15, 18);
  for(int i = 0 ; i < 4 ; i++) {
    if(i < 2) pinMode(button[i], INPUT_PULLUP);
    pinMode(LED[i], OUTPUT);
  }
  Blynk.begin(BLYNK_TOKEN, WIFI_NAME, WIFI_PASS);
  LINE.setToken(LINE_TOKEN);
  client.setInsecure();
  xTaskCreatePinnedToCore(
    readSensorData, // Task function
    "readSensorData", // name of task
    1000, // Stack size of task
```

```

NULL,    // parameter of task
1,       // priority of the task
NULL,    // Task handle to keep track of created task
1        // pin task to core 1
);
xTaskCreatePinnedToCore(
  sendSensorData, // Task function
  "sendSensorData", // name of task
  5000,    // Stack size of task
  NULL,    // parameter of task
  1,       // priority of the task
  NULL,    // Task handle to keep track of created task
  1        // pin task to core 1
);
xTaskCreatePinnedToCore(
  sensorAlarm, // Task function
  "sensorAlarm", // name of task
  5000,    // Stack size of task
  NULL,    // parameter of task
  1,       // priority of the task
  NULL,    // Task handle to keep track of created task
  1        // pin task to core 1
);
xTaskCreatePinnedToCore(
  switchOverHeatAlarm, // Task function
  "switchAlarm", // name of task
  5000,    // Stack size of task
  NULL,    // parameter of task
  1,       // priority of the task
  NULL,    // Task handle to keep track of created task
  1        // pin task to core 1
);
xTaskCreatePinnedToCore(
  switchIntrudersAlarm, // Task function
  "switchIntrudersAlarm", // name of task
  5000,    // Stack size of task
  NULL,    // parameter of task
  1,       // priority of the task
  NULL,    // Task handle to keep track of created task
  1        // pin task to core 1
);

```

```

}

void readSensorData(void *pvParameters) {
    while(true) {
        sht.begin();
        sht.sample();
        temperature = sht.readTempC();
        humidity = sht.readHumidity();
        vTaskDelay(1000/portTICK_PERIOD_MS);
    }
}

void sensorAlarm(void *pvParameters) {
    while(true) {
        if(temperature > 28) {
            LINE.notify("\nTemperature is over 28 *C !\nCurrent temperature ->
"+String(temperature)+" *C");
            Serial.println("\nTemperature is over 28 *C !");
            Serial.println("Sending notification to line\n");
            vTaskDelay(30000/portTICK_PERIOD_MS);
        }
        vTaskDelay(1000/portTICK_PERIOD_MS);
    }
}

void switchOverHeatAlarm(void *pvParameters) {
    while(true) {
        if(digitalRead(button[0]) == LOW) {
            vTaskDelay(20/portTICK_PERIOD_MS);
            LINE.notify("\nOverheat Alarm");
            Serial.println("Alarm: Overheat Alarm!!!");
            while(digitalRead(button[0]) == LOW) vTaskDelay(20/portTICK_PERIOD_MS);
        }
    }
}

void switchIntrudersAlarm(void *pvParameters) {
    while(true) {
        if(digitalRead(button[1]) == LOW) {
            vTaskDelay(20/portTICK_PERIOD_MS);
            LINE.notify("\nIntruders Alarm");
        }
    }
}

```

```

    Serial.println("Alarm: Intruders Alarm!!!");
    while(digitalRead(button[1]) == LOW) vTaskDelay(20/portTICK_PERIOD_MS);
  }
}
}

void logSensorData() {
  if(!client.connect(HOST, HOST_PORT)) return;

  String strURL;
  strURL += "/macros/s/" + String(GSS_TOKEN) + "/exec?";
  strURL += "id=" + String(GSS_SHEET);
  strURL += "&Temperature=" + String(temperature);
  strURL += "&Humidity=" + String(humidity);
  Serial.print("requesting URL: "); Serial.println(strURL);

  client.print(
    String("GET ") + strURL + " HTTP/1.1\r\n" +
    "Host: " + String(HOST) + "\r\n" +
    "User-Agent: BuildFailureDetectorESP8266\r\n" +
    "Connection: close\r\n\r\n"
  );
  Serial.println("request sent");

  while (client.connected()) {
    String line = client.readStringUntil('\n');
    if (line == "\r") {
      Serial.println("headers received");
      break;
    }
  }

  String line = client.readStringUntil('\n');

  Serial.print("reply was : ");
  Serial.println(line);
  Serial.println("closing connection");
  client.stop();
  Serial.println();
}

```

```

void sendSensorData(void *pvParameters) {
    while(true) {
        if(temperature >= -40 && temperature <= 125) {
            logSensorData();
            Blynk.virtualWrite(V0, temperature);
            Blynk.virtualWrite(V1, humidity);
            Serial.println("Temp(*C) >> "+String(temperature));
            Serial.println("Humidity(%) >> "+String(humidity)+"\n");
        }
        vTaskDelay(5000/portTICK_PERIOD_MS);
    }
}

void loop() {
    Blynk.run();
}

```

WIFI_NAME และ WIFI_PASS ให้เปลี่ยนเป็นของตนเอง

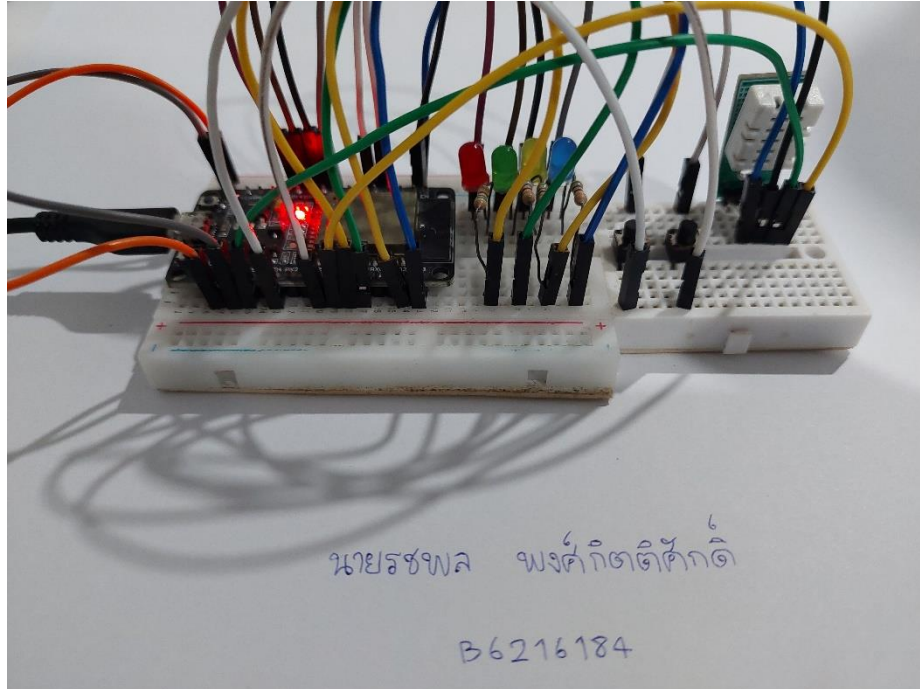
BLYNK_TOKEN ให้เปลี่ยนเป็นของตนเอง

LINE_TOKEN ให้เปลี่ยนเป็นของตนเอง

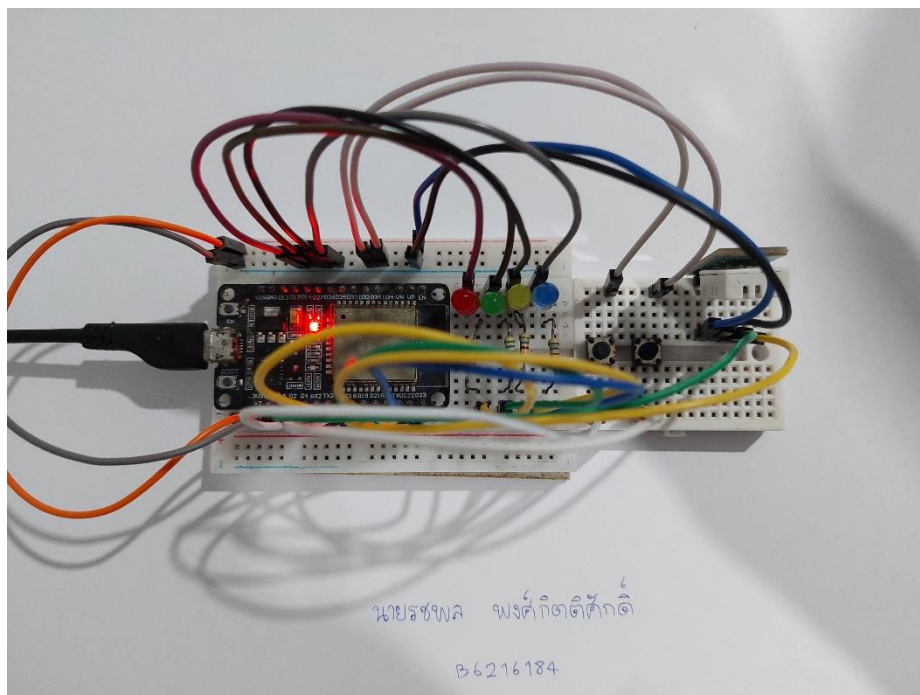
GSS_TOKEN (Google Spreadsheet) ให้เปลี่ยนเป็นของตนเอง

4. ผลลัพธ์

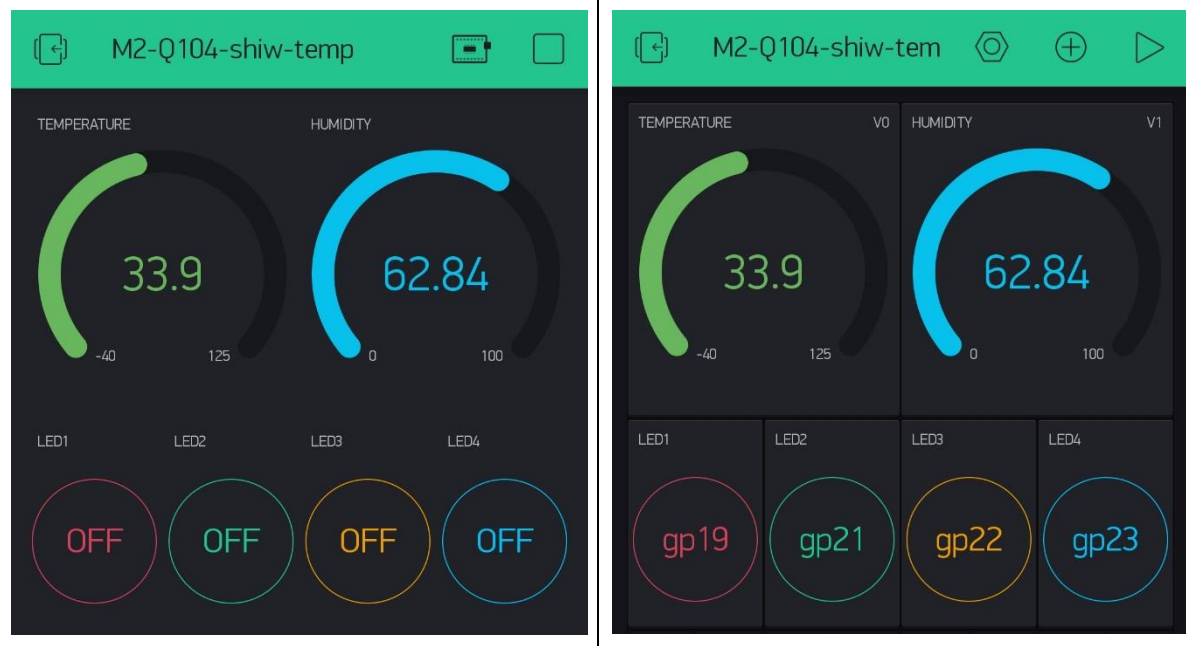
รูปการต่อวงจร 1 – ด้านข้าง



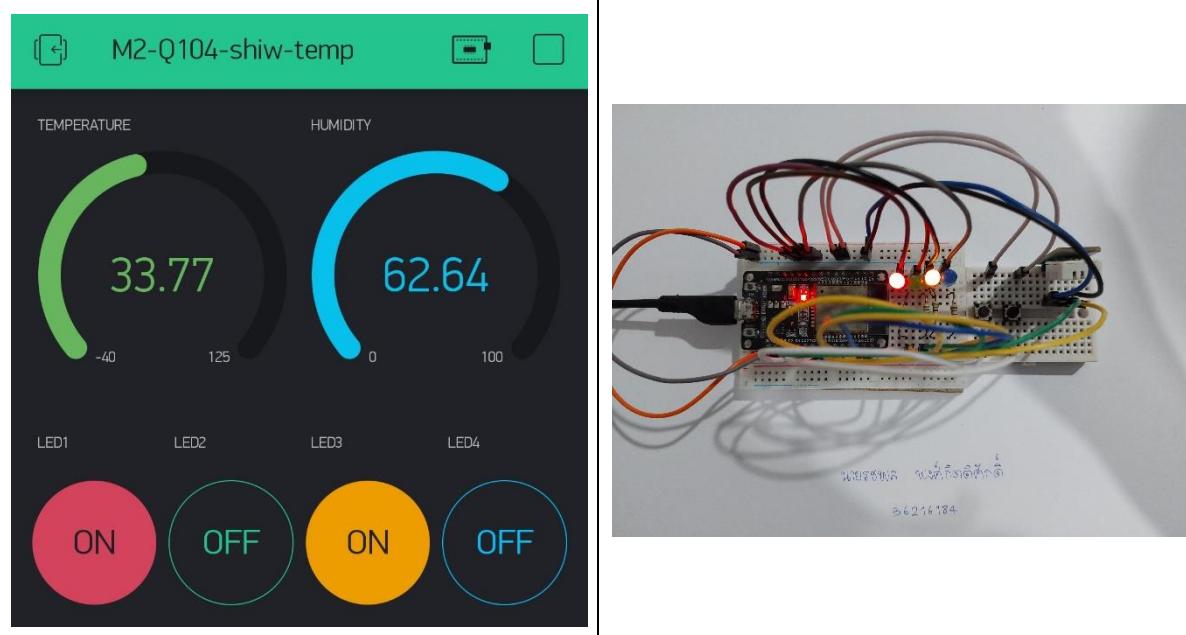
รูปการต่อวงจร 2 – ด้านบน



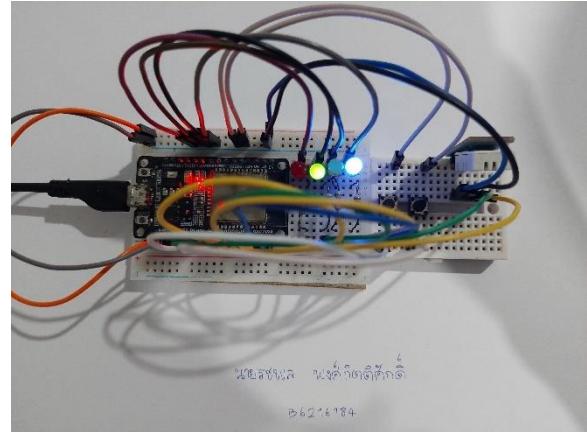
รูปหน้าจอ Blynk



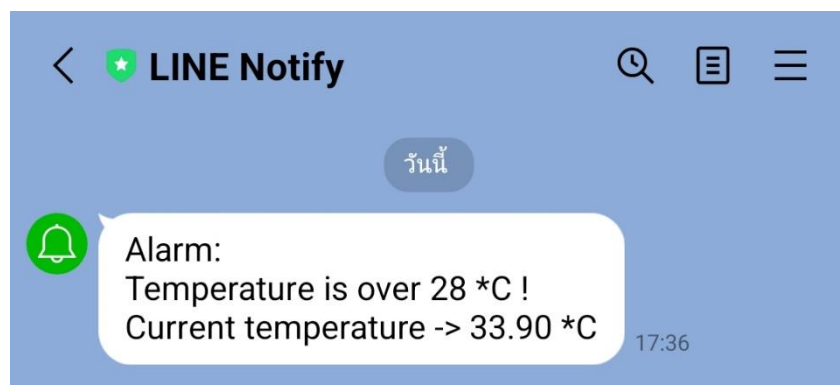
รูปหน้าจอ Blynk – เมื่อกดเปิด/ปิด LED ทั้ง 4



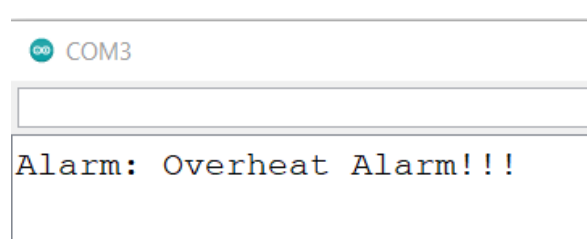
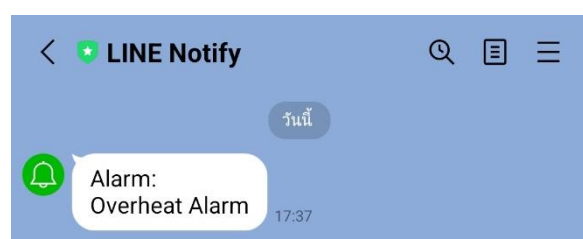
หน้าจอ Blynk – เมื่อกดเปิด/ปิด LED ทั้ง 4



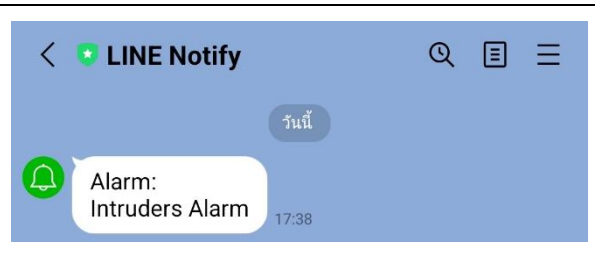
หน้าจอ LINE – แจ้งเตือนเมื่ออุณหภูมิ 28 องศาเซลเซียส



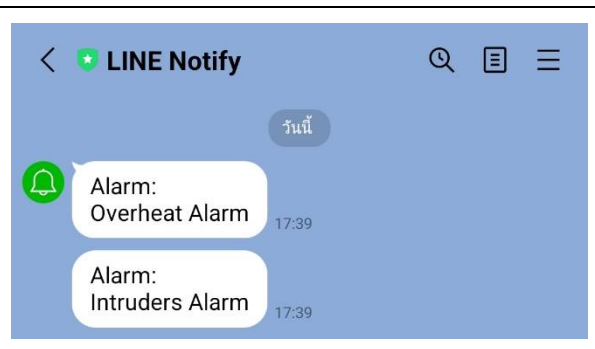
หน้าจอ LINE – แจ้งเตือนเมื่อมี Overheat Alarm



หน้าจอ LINE – แจ้งเตือนเมื่อมี Intruders Alarm

	<p>COM3</p> <hr/> <p>Alarm: Intruders Alarm!!!</p>
---	--

หน้าจอ LINE – แจ้งเตือนเมื่อมี Overheat Alarm กับ Intruders Alarm พร้อมกัน

	<p>COM3</p> <hr/> <p>Alarm: Overheat Alarm!!! Alarm: Intruders Alarm!!!</p>
--	---

สามารถสวิตช์ทั้ง 2 ปุ่มพร้อมกันแล้วเกิดการแจ้งเตือนมายัง LINE ได้

หน้าจอ LINE – แจ้งเตือนอุณหภูมิเกิน และ Alarm

