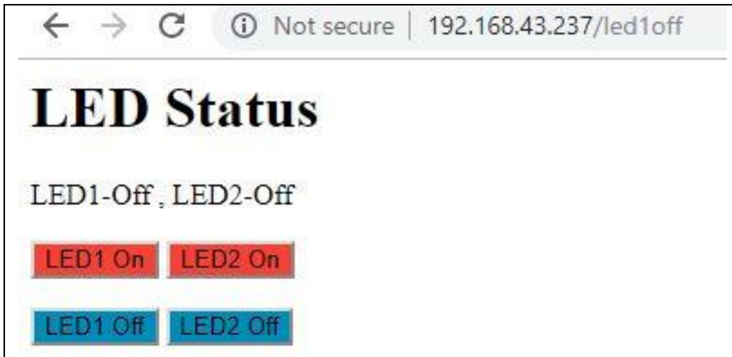
แนวทางการใช้งานอินเทอร์เน็ตของสรรพสิ่งในระบบการผลิต IoT Approaches to Manufacturing System
ชื่อ-สกุล : ณัฐพงศ์ โต๊ะแอ   รหัสนักศึกษา : B6310158

4/4. คำถามท้ายบทเพื่อทดสอบความเข้าใจ

Quiz_201 – Web Control 2 LED

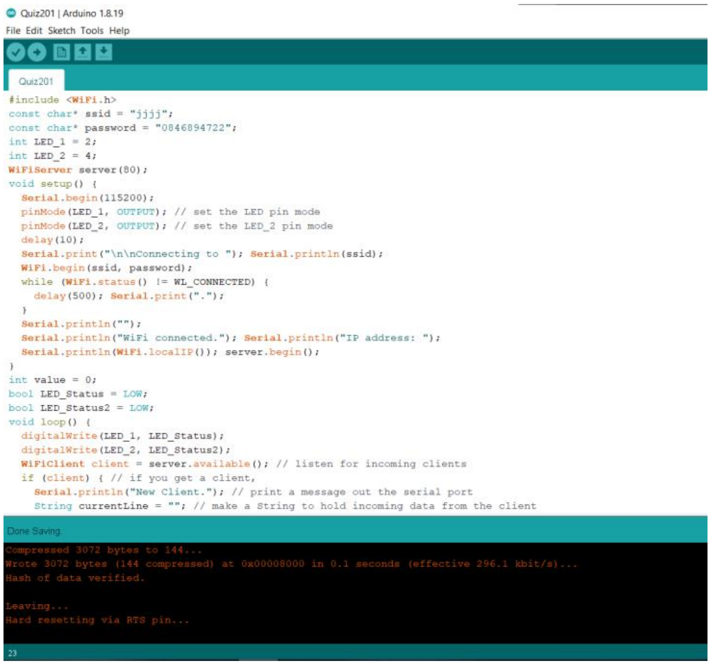
- อยากได้ปุ่มสำหรับคุมปิด-เปิด หลอดไฟ LED 2 ดวง
- https://www.colorhexa.com/008cba?fbclid=IwAR3dIZ_gRgDWmREmnzuknLbMxV3pOHy4YIPuLEz8-ZzTOX2VhWxcH2QjLGk



LED Status

LED1-Off , LED2-Off

< Test Code >



```
#include <WiFi.h>
const char* ssid = "jjjj";
const char* password = "0846894722";
int LED_1 = 2;
int LED_2 = 4;
WiFiServer server(80);
void setup() {
  Serial.begin(115200);
  pinMode(LED_1, OUTPUT); // set the LED pin mode
  pinMode(LED_2, OUTPUT); // set the LED_2 pin mode
  delay(10);
  Serial.print("\n\nConnecting to "); Serial.println(ssid);
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500); Serial.print(".");
  }
  Serial.println("");
  Serial.println("WiFi connected."); Serial.println("IP address: ");
  Serial.println(WiFi.localIP()); server.begin();
}
int value = 0;
bool LED_Status = LOW;
bool LED_Status2 = LOW;
void loop() {
  digitalWrite(LED_1, LED_Status);
  digitalWrite(LED_2, LED_Status2);
  WiFiClient client = server.available(); // listen for incoming clients
  if (client) { // if you get a client,
    Serial.println("New Client."); // print a message out the serial port
    String currentLine = ""; // make a String to hold incoming data from the client
```

Done Saving.

```
Compressed 3072 bytes to 144...
Wrote 3072 bytes (144 compressed) at 0x00008000 in 0.1 seconds (effective 296.1 kbit/s)...
Hash of data verified.

Leaving...
Hard resetting via RTS pin...
```

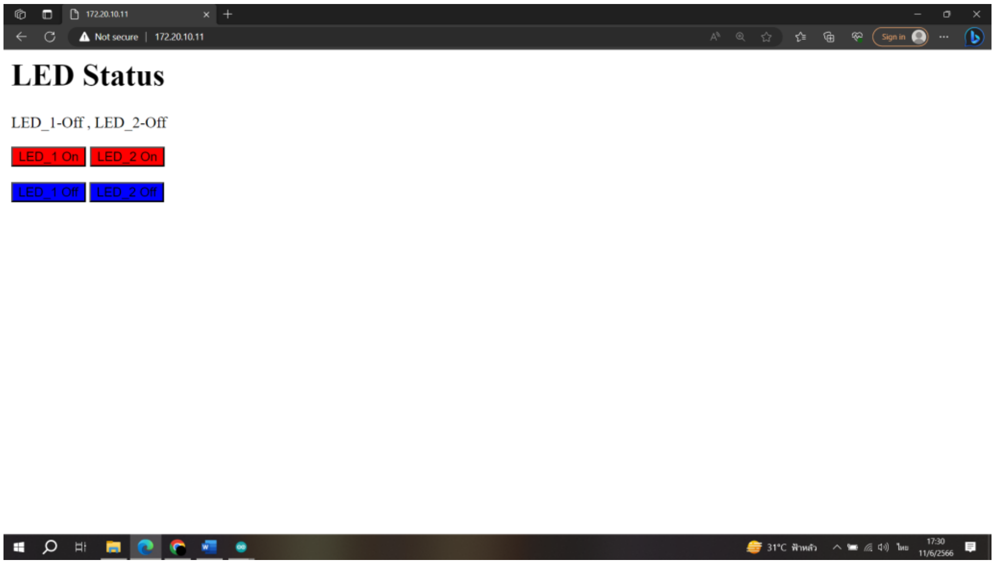
```
#include <WiFi.h>
const char* ssid = "jjjj";
const char* password = "0846894722";
```

```
int LED_1 = 2;
int LED_2 = 4;
WiFiServer server(80);
void setup() {
  Serial.begin(115200);
  pinMode(LED_1, OUTPUT); // set the LED pin mode
  pinMode(LED_2, OUTPUT); // set the LED_2 pin mode
  delay(10);
  Serial.print("\n\nConnecting to "); Serial.println(ssid);
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500); Serial.print(".");
  }
  Serial.println("");
  Serial.println("WiFi connected."); Serial.println("IP address: ");
  Serial.println(WiFi.localIP()); server.begin();
}
int value = 0;
bool LED_Status = LOW;
bool LED_Status2 = LOW;
void loop() {
  digitalWrite(LED_1, LED_Status);
  digitalWrite(LED_2, LED_Status2);
  WiFiClient client = server.available(); // listen for incoming clients
  if (client) { // if you get a client,
    Serial.println("New Client."); // print a message out the serial port
    String currentLine = ""; // make a String to hold incoming data from the client
    while (client.connected()) { // loop while the client's connected
      if (client.available()) { // if there's bytes to read from the client,
        char c = client.read(); // read a byte, then
        Serial.write(c); // print it out the serial monitor
        if (c == '\n') { // if the byte is a newline character
          if (currentLine.length() == 0) {
            client.println("HTTP/1.1 200 OK");
            client.println("Content-type:text/html");
            client.println();
            client.println("<html>");
            client.println("<body>");
            client.println("<h1>LED Status</h1>");
            client.println("<p>");
            if (LED_Status == HIGH)
              client.println("LED_1-On ,");
            else
              client.println("LED_1-Off ,");
            if (LED_Status2 == HIGH)
              client.println("LED_2-On");
            else
              client.println("LED_2-Off");
            client.println("<p>");
            client.println("<a href=\"/LED_1on\"><button style = \"background-color:
#ff0000;\">LED_1 On</button></a>");
            client.println("<a href=\"/LED_2on\"><button style = \"background-color:
#ff0000;\">LED_2 On</button></a>");
            client.println("</p>");
            client.println("<a href=\"/LED_1off\"><button style = \"background-color:
#0000ff;\">LED_1 Off</button></a>");
            client.println("<a href=\"/LED_2off\"><button style = \"background-color:
#0000ff;\">LED_2 Off</button></a>");
            client.println("<body>");
      client.println("<html>");
      break;
     } else {
      currentLine = "";
     }
    } else if (c != '\r') {
     currentLine += c;
    }
    if (currentLine.endsWith("GET /LED_1on")) LED_Status = HIGH;
    if (currentLine.endsWith("GET /LED_1off")) LED_Status = LOW;
    if (currentLine.endsWith("GET /LED_2on")) LED_Status2 = HIGH;
    if (currentLine.endsWith("GET /LED_2off")) LED_Status2 = LOW;
   }
  }
  client.stop(); // close the connection:
  Serial.println("Client Disconnected.");
 }
}
```

รูปการต่อวงจร – 1

รูปการต่อวงจร – 2

หน้าจอ Web Control

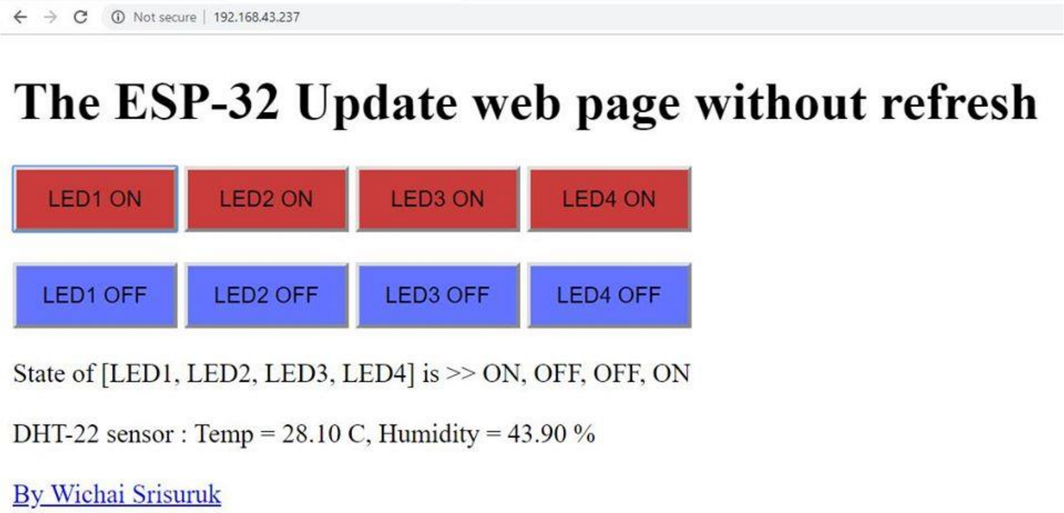


LED Status

LED_1-Off , LED_2-Off

Quiz_202 – Web Control 4 LED and Monitor Humid/Temperature

- เพิ่มเติมจาก Q202 อยากได้ปุ่มสำหรับคุมปิด-เปิด หลอดไฟ LED 4 ดวง
- อยากมีกด Link ไปที่หน้า FB ของตัวเอง



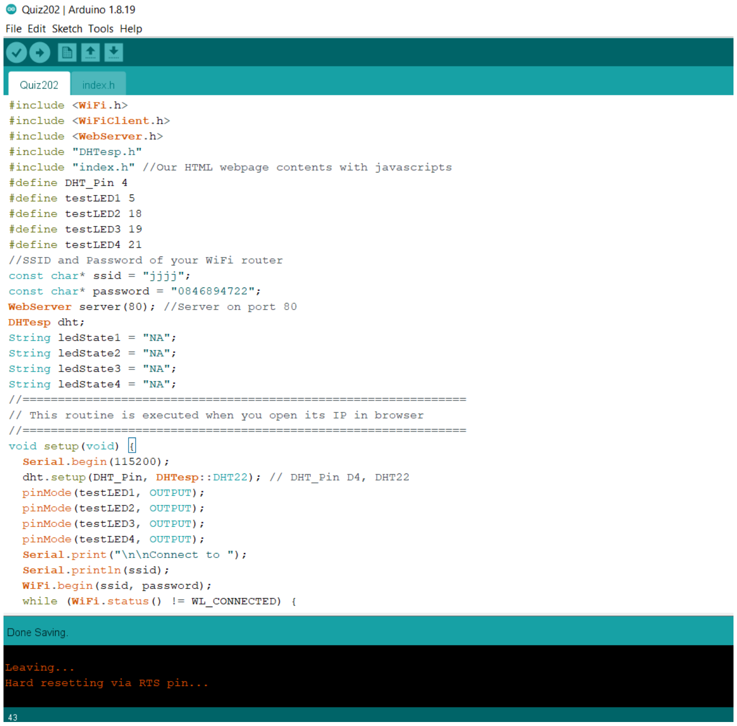
The ESP-32 Update web page without refresh

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

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

By Wichai Srisuruk

< Test Code >



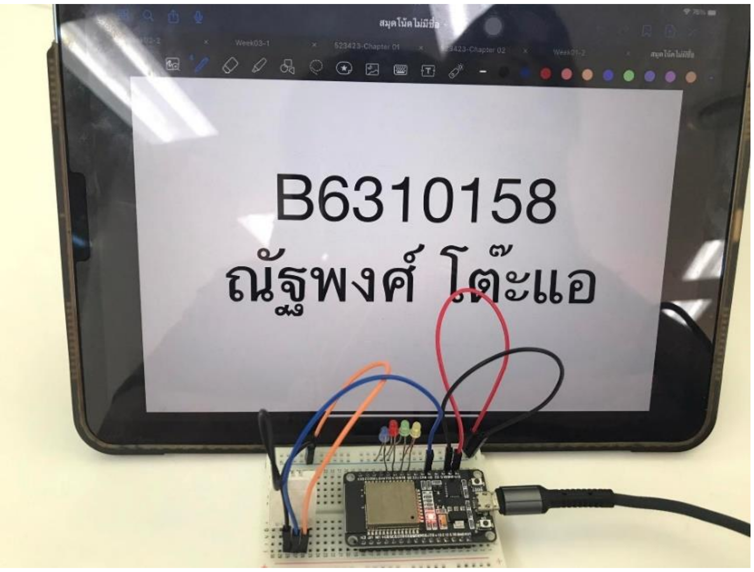
```
#include <WiFi.h>
#include <WiFiClient.h>
#include <WebServer.h>
#include "DHTesp.h"
#include "index.h" //Our HTML webpage contents with javascripts
#define DHT_Pin 4
#define testLED1 5
#define testLED2 18
#define testLED3 19
#define testLED4 21
//SSID and Password of your WiFi router
const char* ssid = "jjjj";
const char* password = "0846894722";
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 setup(void) {
  Serial.begin(115200);
  dht.setup(DHT_Pin, DHTesp::DHT22); // DHT_Pin D4, DHT22
  pinMode(testLED1, OUTPUT);
  pinMode(testLED2, OUTPUT);
  pinMode(testLED3, OUTPUT);
  pinMode(testLED4, OUTPUT);
  Serial.print("\n\nConnect to ");
  Serial.println(ssid);
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
```

```
Done Saving.

Leaving...
Hard resetting via RTS pin...

43
```

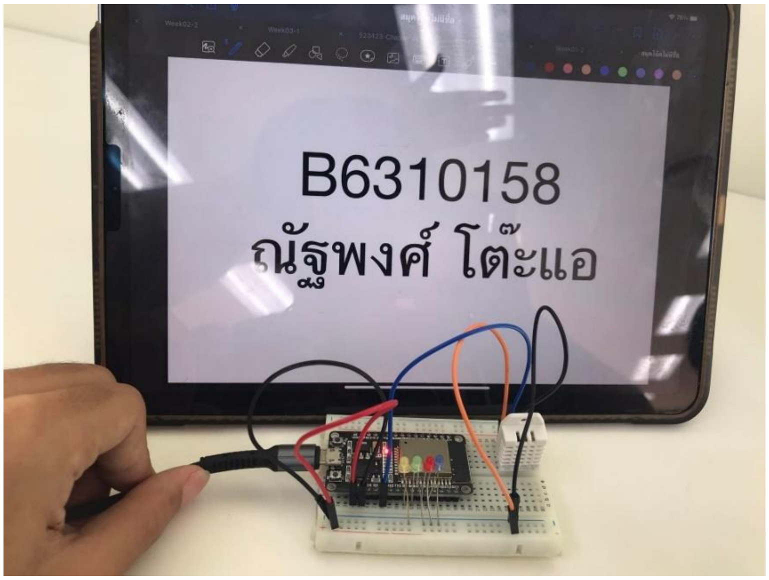
```
#include <WiFi.h>
#include <WiFiClient.h>
#include <WebServer.h>
#include "DHTesp.h"
#include "index.h" //Our HTML webpage contents with javascripts
#define DHT_Pin 4
#define testLED1 5
#define testLED2 18
```

```
#define testLED3 19
#define testLED4 21

//SSID and Password of your WiFi router
const char* ssid = "jjjj";
const char* password = "0846894722";
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 setup(void) {
  Serial.begin(115200);
  dht.setup(DHT_Pin, DHTesp::DHT22); // DHT_Pin D4, DHT22
  pinMode(testLED1, OUTPUT);
  pinMode(testLED2, OUTPUT);
  pinMode(testLED3, OUTPUT);
  pinMode(testLED4, 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 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/plane", 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(testLED1, HIGH);  //Feedback parameter
    ledState1 = "ON";
  }
  if (t_state == "10") {
    digitalWrite(testLED1, LOW);  //Feedback parameter
    ledState1 = "OFF";
  }
  if (t_state == "21") {
    digitalWrite(testLED2, HIGH);  //Feedback parameter
    ledState2 = "ON";
  }
  if (t_state == "20") {
    digitalWrite(testLED2, LOW);  //Feedback parameter
    ledState2 = "OFF";
  }
  if (t_state == "31") {
    digitalWrite(testLED3, HIGH);  //Feedback parameter
    ledState3 = "ON";
  }
  if (t_state == "30") {
    digitalWrite(testLED3, LOW);  //Feedback parameter
    ledState3 = "OFF";
  }
  if (t_state == "41") {
    digitalWrite(testLED4, HIGH);  //Feedback parameter
    ledState4 = "ON";
  }
  if (t_state == "40") {
    digitalWrite(testLED4, LOW);  //Feedback parameter
    ledState4 = "OFF";}
  server.send(200, "text/plane", ledState1 + ", " + ledState2 + ", " + ledState3 + ", " +
ledState4); //Send web page
}
void loop(void) {
  server.handleClient(); //Handle client requests
}
```

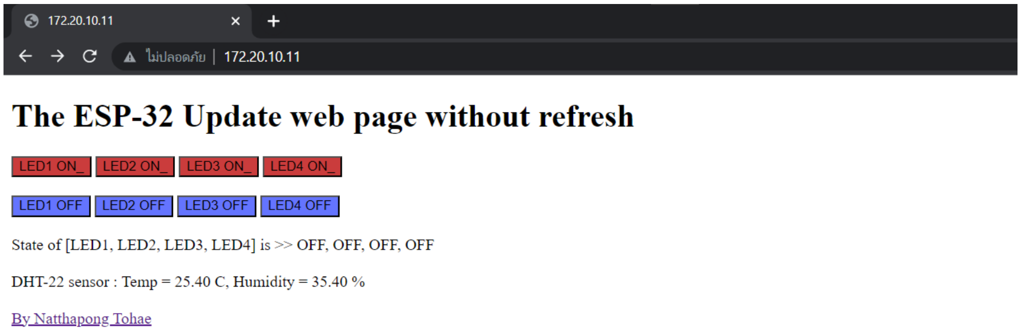
รูปการต่อวงจร – 1



รูปการต่อวงจร – 2



หน้าจอ Web Control



The ESP-32 Update web page without refresh

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

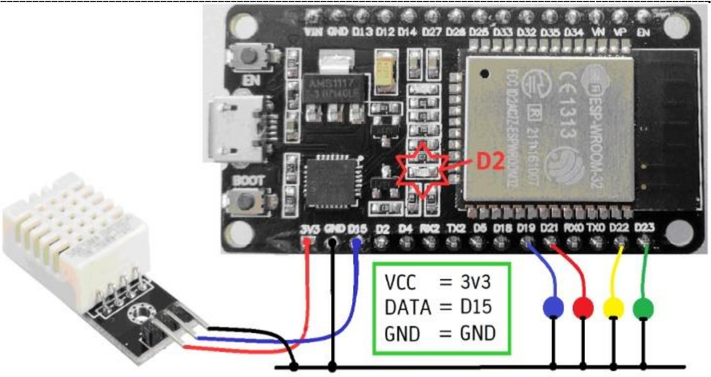
DHT-22 sensor : Temp = 25.40 C, Humidity = 35.40 %

By Natthapong Tohae

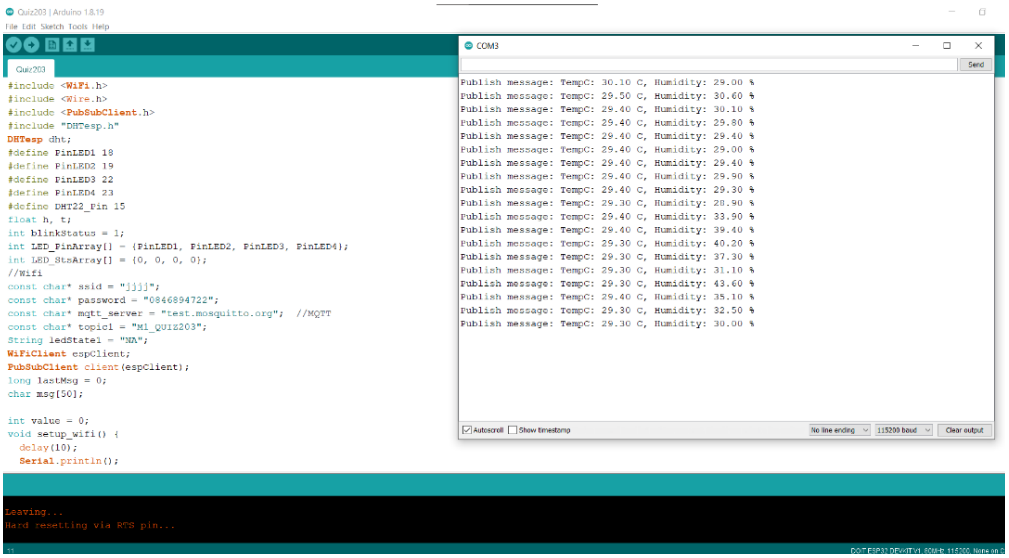
Quiz_203 – Publish

- อ่านค่า DHT-22 แล้วส่งไปยัง MQTT Broker ทุกๆ 5 วินาที
- ควบคุมการแสดงผลให้ 4 LED แสดงผลตามข้อกำหนดดังนี้

✱○○○(Blink)	หากการอ่านค่าแล้วเป็น null, หรือไม่มีเซ็นเซอร์
●○○○	ช่วงของอุณหภูมิ (-∞, 24)
●●○○	ช่วงของอุณหภูมิ [24,26)
●●●○	ช่วงของอุณหภูมิ [26,28)
●●●●	ช่วงของอุณหภูมิ [28,30)
✱✱✱✱(Blink)	ช่วงของอุณหภูมิ [30,∞)


VIN GND D13 D12 D14 D27 D26 D25 D33 D32 D35 D34 VN VP EN
EN
BOOT
ESP-WROOM-32
D2
3V3 GND D15 D2 D4 RX2 TX2 D5 D18 D19 D21 RX0 TX0 D22 D23
VCC = 3v3
DATA = D15
GND = GND

< Test Code >



```
#include <WiFi.h>
#include <Wire.h>
#include <PubSubClient.h>
#include "DHTesp.h"
DHTesp dht;
#define PinLED1 18
#define PinLED2 19
#define PinLED3 22
#define PinLED4 23
#define DHT22_Pin 15
float h, t;
int blinkStatus = 1;
int LED_PinArray[] = {PinLED1, PinLED2, PinLED3, PinLED4};
int LED_StsArray[] = {0, 0, 0, 0};
//Wifi
const char* ssid = "jjjj";
const char* password = "0846894722";
const char* mqtt_server = "test.mosquitto.org";  //MQTT
const char* topic1 = "M1_QUIZ203";
String ledState1 = "NA";
WiFiClient espClient;
PubSubClient client(espClient);
long lastMsg = 0;
char msg[50];

int value = 0;
void setup_wifi() {
  delay(10);
  Serial.println();
```

```
Publish message: TempC: 30.10 C, Humidity: 29.00 %
Publish message: TempC: 29.50 C, Humidity: 30.60 %
Publish message: TempC: 29.40 C, Humidity: 30.10 %
Publish message: TempC: 29.40 C, Humidity: 29.80 %
Publish message: TempC: 29.40 C, Humidity: 29.40 %
Publish message: TempC: 29.40 C, Humidity: 29.00 %
Publish message: TempC: 29.40 C, Humidity: 29.40 %
Publish message: TempC: 29.40 C, Humidity: 29.90 %
Publish message: TempC: 29.40 C, Humidity: 29.30 %
Publish message: TempC: 29.30 C, Humidity: 28.90 %
Publish message: TempC: 29.40 C, Humidity: 33.90 %
Publish message: TempC: 29.40 C, Humidity: 39.40 %
Publish message: TempC: 29.30 C, Humidity: 40.20 %
Publish message: TempC: 29.30 C, Humidity: 37.30 %
Publish message: TempC: 29.30 C, Humidity: 31.10 %
Publish message: TempC: 29.30 C, Humidity: 43.60 %
Publish message: TempC: 29.40 C, Humidity: 35.10 %
Publish message: TempC: 29.30 C, Humidity: 32.50 %
Publish message: TempC: 29.30 C, Humidity: 30.00 %
```

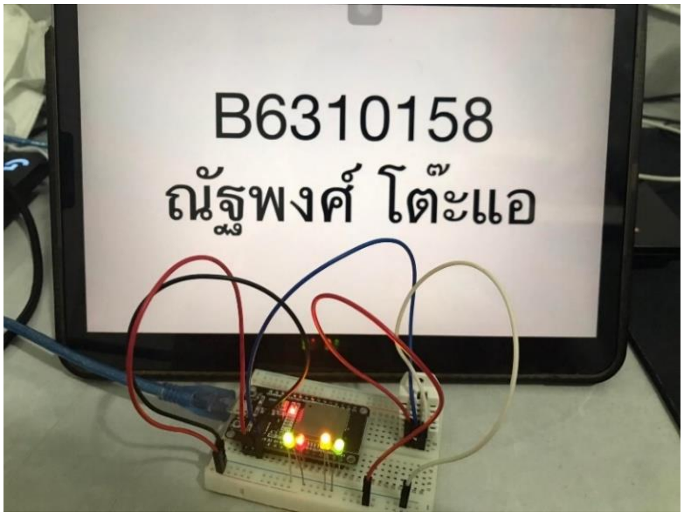
```
Leaving...
Hard resetting via RTS pin...
```

```
#include <WiFi.h>
#include <Wire.h>
#include <PubSubClient.h>
#include "DHTesp.h"
DHTesp dht;
#define PinLED1 18
```

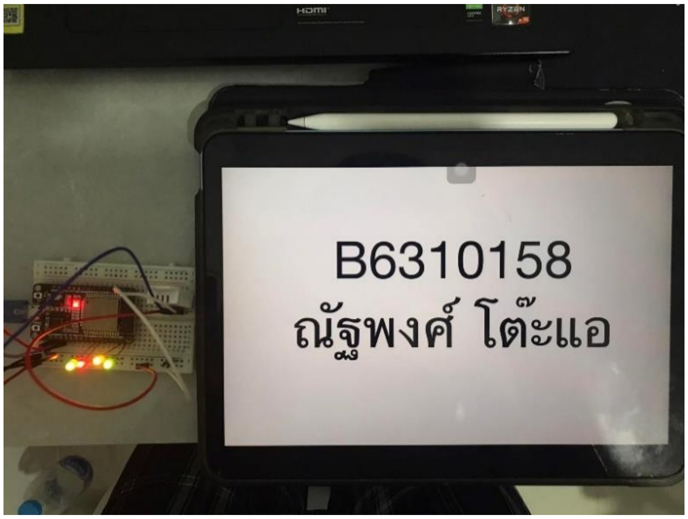
```
#define PinLED2 19
#define PinLED3 22
#define PinLED4 23
#define DHT22_Pin 15
float h, t;
int blinkStatus = 1;
int LED_PinArray[] = {PinLED1, PinLED2, PinLED3, PinLED4};
int LED_StsArray[] = {0, 0, 0, 0};
//Wifi
const char* ssid = "jjjj";
const char* password = "0846894722";
const char* mqtt_server = "test.mosquitto.org";  //MQTT
const char* topic1 = "M1_QUIZ203";
String ledState1 = "NA";
WiFiClient espClient;
PubSubClient client(espClient);
long lastMsg = 0;
char msg[50];

int value = 0;
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 = "ESP32Client-";
    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 Pk007"); // ... and resubscribe
      client.subscribe(topic1);
    } else
    { Serial.print("failed, rc=");
      Serial.print(client.state());
      Serial.println(" try again in 5 seconds");
      delay(5000);
    }
  }
}
```

```
void LEDShowStatus(void) {
  if (isnan(t)) {
    blinkStatus = 1 - blinkStatus;
    LED_StsArray[0] = 1;
    LED_StsArray[1] = 0;
    LED_StsArray[2] = 0;
    LED_StsArray[3] = 0;
  }
  if (t < 24) {
    blinkStatus = 1;
    LED_StsArray[0] = 1;
    LED_StsArray[1] = 0;
    LED_StsArray[2] = 0;
    LED_StsArray[3] = 0;
  }
  if (t >= 24) {
    LED_StsArray[0] = 1;
    LED_StsArray[1] = 1;
    LED_StsArray[2] = 0;
    LED_StsArray[3] = 0;
  }
  if (t < 26) {
    blinkStatus = 1;
    LED_StsArray[0] = 1;
    LED_StsArray[1] = 1;
    LED_StsArray[2] = 0;
    LED_StsArray[3] = 0;
  }
  if (t >= 26) {
    LED_StsArray[0] = 1;
    LED_StsArray[1] = 1;
    LED_StsArray[2] = 1;
    LED_StsArray[3] = 0;
  }
  if (t < 28) {
    blinkStatus = 1;
    LED_StsArray[0] = 1;
    LED_StsArray[1] = 1;
    LED_StsArray[2] = 1;
    LED_StsArray[3] = 0;
  }
  if (t >= 28) {
    LED_StsArray[0] = 1;
    LED_StsArray[1] = 1;
    LED_StsArray[2] = 1;
    LED_StsArray[3] = 1;
  }
  if (t < 30) {
    blinkStatus = 1;
    LED_StsArray[0] = 1;
    LED_StsArray[1] = 1;
    LED_StsArray[2] = 1;
    LED_StsArray[3] = 1;
  }
  if (t >= 30){
    blinkStatus = 1 - blinkStatus;
    LED_StsArray[0] = 1;
    LED_StsArray[1] = 1;
    LED_StsArray[2] = 1;
    LED_StsArray[3] = 1;
  }
  for (int i = 0; i < 4; i++)
    digitalWrite(LED_PinArray[i], LED_StsArray[i] & blinkStatus);
}
void setup()
{ Serial.begin(115200);
  setup_wifi();
 //Wire.begin(22, 23);
  client.setServer(mqtt_server, 1883);
  dht.setup(DHT22_Pin, DHTesp::DHT22);
  for (int i = 0; i < 4; i++) {
    pinMode(LED_PinArray[i], OUTPUT);
  }
}
void loop()
{
  if (!client.connected()) reconnect();
  client.loop();
  long now = millis();
  if (now - lastMsg > 5000)
  { lastMsg = now;
   ++value;
   //float t = s.readTempC();
   //float h = s.readHumidity();
    delay(dht.getMinimumSamplingPeriod());
    h = dht.getHumidity();
    t = dht.getTemperature();

    sprintf (msg, "TempC: %.2f C, Humidity: %.2f %%", t, h);
    Serial.print("Publish message: ");
    Serial.println(msg);
    client.publish(topic1, msg);

  }
  LEDShowStatus(); delay(250);
  LEDShowStatus(); delay(250);
  LEDShowStatus(); delay(250);
  LEDShowStatus(); delay(250);
  LEDShowStatus(); delay(250);
  LEDShowStatus(); delay(250);
}
```

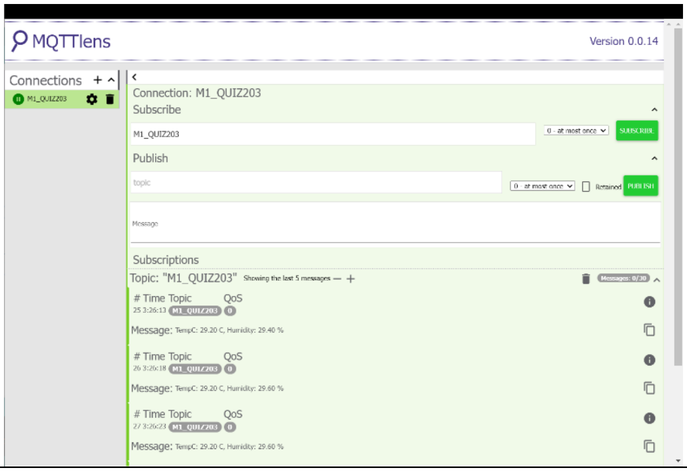
รูปการต่อวงจร – 1



รูปการต่อวงจร – 2



หน้าจอ MQTT Lens



Connection: M1_QUIZ203

Subscribe: M1_QUIZ203 — 0 - at most once

Publish: topic — 0 - at most once — □ Retained

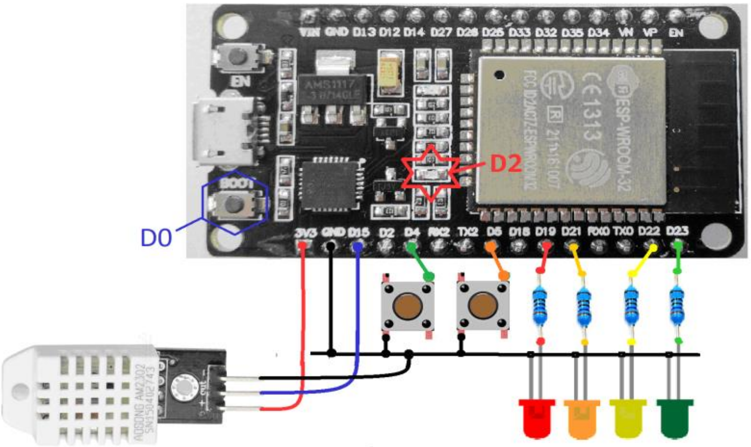
Message

Topic: "M1_QUIZ203" Showing the last 5 messages — + Messages: 0/30

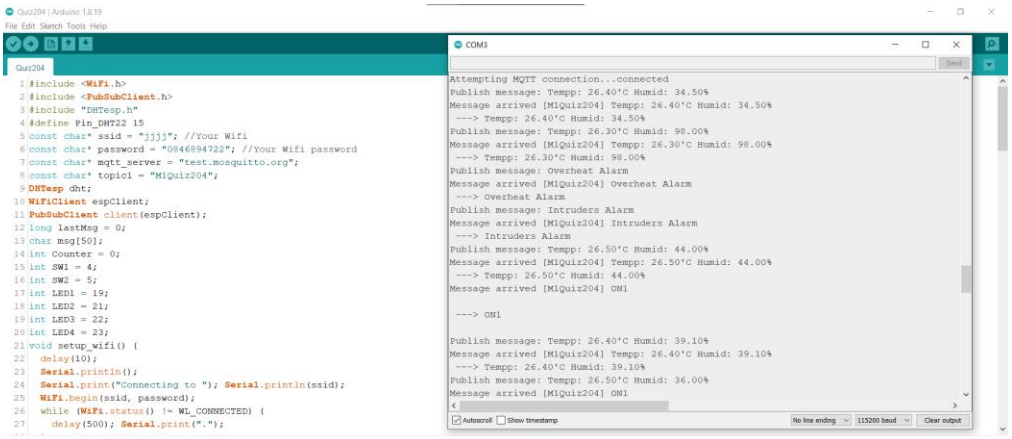
```
# Time Topic        QoS
25 3:26:13 M1_QUIZ203 0
Message: TempC: 29.20 C, Humidity: 29.40 %

# Time Topic        QoS
26 3:26:18 M1_QUIZ203 0
Message: TempC: 29.20 C, Humidity: 29.60 %

# Time Topic        QoS
27 3:26:23 M1_QUIZ203 0
Message: TempC: 29.20 C, Humidity: 29.60 %
```

Quiz_204 – Publish and Subscribe

- อ่านค่า DHT-22 แล้วส่งไปยัง MQTT Broker ทุกๆ 5 วินาที
- ควบคุมการปิดเปิด 4 LED
- รับค่าสวิตซ์กำหนด SW1 แจ้ง Overheat Alarm, SW2 แจ้ง Intruders Alarm


VIN GND D13 D12 D14 D27 D26 D25 D33 D32 D35 D34 VN VP EN
EN
BOOT
D0
D2
3V3 GND D15 D2 D4 RX2 TX2 D5 D18 D19 D21 RX0 TX0 D22 D23
ESP-WROOM-32

< Test Code >



```
#include <WiFi.h>
#include <PubSubClient.h>
#include "DHTesp.h"
#define Pin_DHT22 15
const char* ssid = "jjjj"; //Your Wifi
const char* password = "0846894722"; //Your Wifi password
const char* mqtt_server = "test.mosquitto.org";
const char* topic1 = "M1Quiz204";
DHTesp dht;
WiFiClient espClient;
PubSubClient client(espClient);
long lastMsg = 0;
char msg[50];
int Counter = 0;
int SW1 = 4;
int SW2 = 5;
int LED1 = 19;
int LED2 = 21;
int LED3 = 22;
int LED4 = 23;
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(".");
```

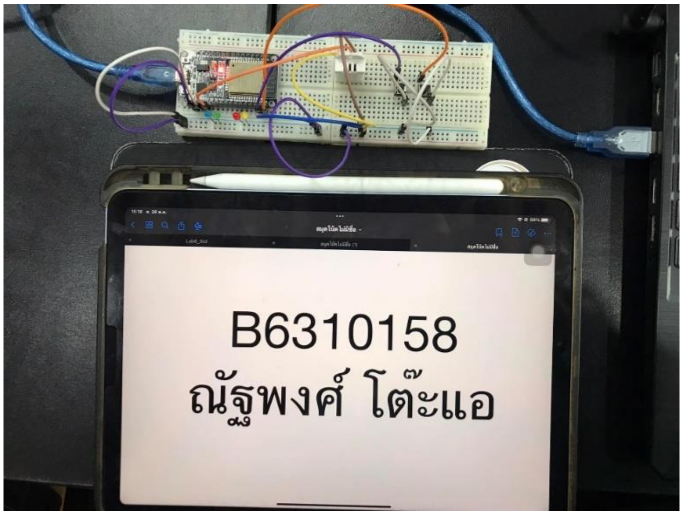
```
Attempting MQTT connection...connected
Publish message: Tempp: 26.40'C Humid: 34.50%
Message arrived [M1Quiz204] Tempp: 26.40'C Humid: 34.50%
 ---> Tempp: 26.40'C Humid: 34.50%
Publish message: Tempp: 26.30'C Humid: 98.00%
Message arrived [M1Quiz204] Tempp: 26.30'C Humid: 98.00%
 ---> Tempp: 26.30'C Humid: 98.00%
Publish message: Overheat Alarm
Message arrived [M1Quiz204] Overheat Alarm
 ---> Overheat Alarm
Publish message: Intruders Alarm
Message arrived [M1Quiz204] Intruders Alarm
 ---> Intruders Alarm
Publish message: Tempp: 26.50'C Humid: 44.00%
Message arrived [M1Quiz204] Tempp: 26.50'C Humid: 44.00%
 ---> Tempp: 26.50'C Humid: 44.00%
Message arrived [M1Quiz204] ON1

 ---> ON1

Publish message: Tempp: 26.40'C Humid: 39.10%
Message arrived [M1Quiz204] Tempp: 26.40'C Humid: 39.10%
 ---> Tempp: 26.40'C Humid: 39.10%
Publish message: Tempp: 26.50'C Humid: 36.00%
Message arrived [M1Quiz204] ON1
```

☑ Autoscroll □ Show timestamp No line ending 115200 baud Clear output

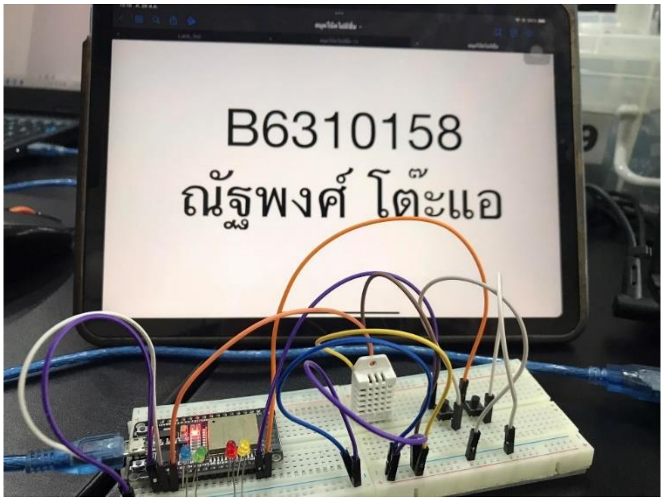
```
#include <WiFi.h>
#include <PubSubClient.h>
#include "DHTesp.h"
#define Pin_DHT22 15
const char* ssid = "jjjj"; //Your Wifi
const char* password = "0846894722"; //Your Wifi password
const char* mqtt_server = "test.mosquitto.org";
const char* topic1 = "M1Quiz204";
DHTesp dht;
WiFiClient espClient;
PubSubClient client(espClient);
long lastMsg = 0;
char msg[50];
int Counter = 0;
int SW1 = 4;
int SW2 = 5;
int LED1 = 19;
int LED2 = 21;
int LED3 = 22;
int LED4 = 23;
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 Akki"); // ... 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(topic1);
  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[4] = '\0'; // String lessthan 4 Charector
  if ((String)myPayLoad == "ON")
  { digitalWrite(LED1, HIGH);
    digitalWrite(LED2, HIGH);
    digitalWrite(LED3, HIGH);
    digitalWrite(LED4, HIGH);
  }
  if ((String)myPayLoad == "ON1") {
    digitalWrite(LED1, HIGH);
  }
  if ((String)myPayLoad == "ON2") {
    digitalWrite(LED2, HIGH);
  }
  if ((String)myPayLoad == "ON3") {
    digitalWrite(LED3, HIGH);
  }
  if ((String)myPayLoad == "ON4") {
    digitalWrite(LED4, HIGH);
  }
  if ((String)myPayLoad == "OFF")
  { digitalWrite(LED1, LOW); digitalWrite(LED2, LOW);
    digitalWrite(LED3, LOW); digitalWrite(LED4, LOW);
  }
  if ((String)myPayLoad == "OFF1") {
    digitalWrite(LED1, LOW);
  }
  if ((String)myPayLoad == "OFF2") {
    digitalWrite(LED2, LOW);
  }
  if ((String)myPayLoad == "OFF3") {
    digitalWrite(LED3, LOW);
  }
  if ((String)myPayLoad == "OFF4") {
    digitalWrite(LED4, LOW);
  }
}
void setup()
{ Serial.begin(115200);
  pinMode(LED1, OUTPUT);
  pinMode(LED2, OUTPUT);
  pinMode(LED3, OUTPUT);
  pinMode(LED4, OUTPUT);
  pinMode(SW1, INPUT_PULLUP);
  pinMode(SW2, INPUT_PULLUP);
  dht.setup(Pin_DHT22, DHTesp::DHT22);
  setup_wifi();
  client.setServer(mqtt_server, 1883);
  client.setCallback(callback);
}
void loop()
{ if (!client.connected()) reconnect();
  { client.loop();
    if (digitalRead(SW1) == 0)
    { client.loop();
      snprintf (msg, 75, " Overheat Alarm ");
      Serial.print("Publish message: ");
      Serial.println(msg);
      client.publish(topic1, msg);
      while (digitalRead(SW1) == 0);
      delay(100);
    }
    if (digitalRead(SW2) == 0)
    { client.loop();
      snprintf (msg, 75, " Intruders Alarm");
      Serial.print("Publish message: ");
      Serial.println(msg);
      client.publish(topic1, msg);
      while (digitalRead(SW2) == 0);
      delay(100);
    }
  }
  long now = millis();
  if (now - lastMsg > 5000)
  { lastMsg = now;
```

```
    float humid = dht.getHumidity();
    float tempp = dht.getTemperature();
    snprintf (msg, 75, "Tempp: %.2f'C Humid: %.2f%%", tempp, humid);
    Serial.print("Publish message: ");
    Serial.println(msg);
    client.publish(topic1, msg);
  }
}
```

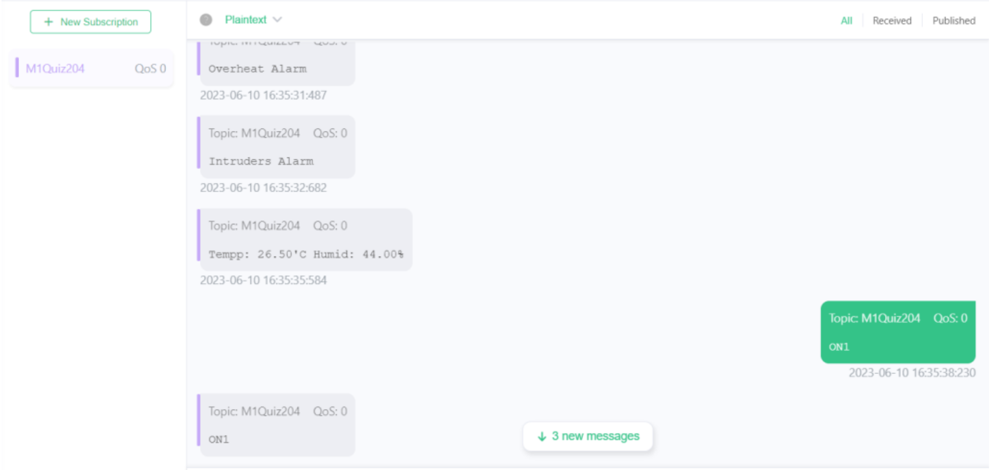
รูปการต่อวงจร – 1



รูปการต่อวงจร – 2



หน้าจอ MQTT Lens



```
Overheat Alarm
```
2023-06-10 16:35:31:487

Topic: M1Quiz204 QoS: 0
```
Intruders Alarm
```
2023-06-10 16:35:32:682

Topic: M1Quiz204 QoS: 0
```
Tempp: 26.50'C Humid: 44.00%
```
2023-06-10 16:35:35:584

Topic: M1Quiz204 QoS: 0
```
ON1
```
2023-06-10 16:35:38:230

Topic: M1Quiz204 QoS: 0
```
ON1
```