

แนวทางการใช้งานอินเทอร์เน็ตของสรรพสิ่งในระบบการผลิต

IoT Approaches to Manufacturing System

ชื่อ-สกุล : นายสิรภัทร สังข์ทอง b6326319

4/4. คำถามท้ายบทเพื่อทดสอบความเข้าใจ

Quiz_201 – Web Control 2 LED

- อยากได้ปุ่มสำหรับคุมปิด-เปิด หลอดไฟ LED 2 ดวง
- https://www.colorhexa.com/008cba?fbclid=IwAR3dIZ_gRgDWmREmnzukaLbMxV3pOHY4YIPuLEz8-ZzTOX2VhWxcH2QjLGk

←

→

↻

ⓘ

Not secure | 192.168.43.237/led1off

LED Status

LED1-Off , LED2-Off

LED1 On

LED2 On

LED1 Off

LED2 Off

< Test Code >

```

#include <WiFi.h>

const char* ssid = "RATSIRI-HOME 2.4G"; //Your Wifi
const char* password = "0984485615"; //Your Wifi password

int pinTest1 = 22;
int pinTest2 = 23;

WiFiServer server(80);

void setup() {
  Serial.begin(115200);
  pinMode(pinTest1, OUTPUT); // set the LED pin mode
  pinMode(pinTest2, OUTPUT);
  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 LED1_Status, LED2_Status, LED3_Status, LED4_Status = LOW;

void loop() {

  digitalWrite(pinTest1, LED1_Status);

  digitalWrite(pinTest2, LED2_Status);

  WiFiClient client = server.available();

  if (client) {

    Serial.println("New Client.");

    String currentLine = "";

    while (client.connected()) {

      if (client.available()) {

        char c = client.read();

        Serial.write(c);

        if (c == '\n') {

          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 (LED1_Status == HIGH & LED2_Status == HIGH)

            client.println("LED1-On,LED2-On");

        else if (LED1_Status == HIGH & LED2_Status == LOW)

            client.println("LED1-On,LED2-Off");

        else if (LED1_Status == LOW & LED2_Status == HIGH)

            client.println("LED1-Off,LED2-On");

        else

            client.println("LED1-Off,LED2-Off");


        client.println("<p>");
        client.println("<h1>LED Control</h1>");
        client.println("<p>");
        client.println("<a href=\"/led1on\"><button style = \"background-color: #f44336;\">LED1
On</button></a>");

        client.println("<a href=\"/led2on\"><button style = \"background-color: #f44336;\">LED2
On</button></a>");

        client.println("</p>");
        client.println("<a href=\"/led1off\"><button style = \"background-color: #008CBA;\">LED1
Off</button></a>");

        client.println("<a href=\"/led2off\"><button style = \"background-color: #008CBA;\">LED2
Off</button></a>");


        client.println("<body>");
        client.println("<html>");

        break;

    } else {

        currentLine = "";

    }

} else if (c != '\r') {

    currentLine += c;

}

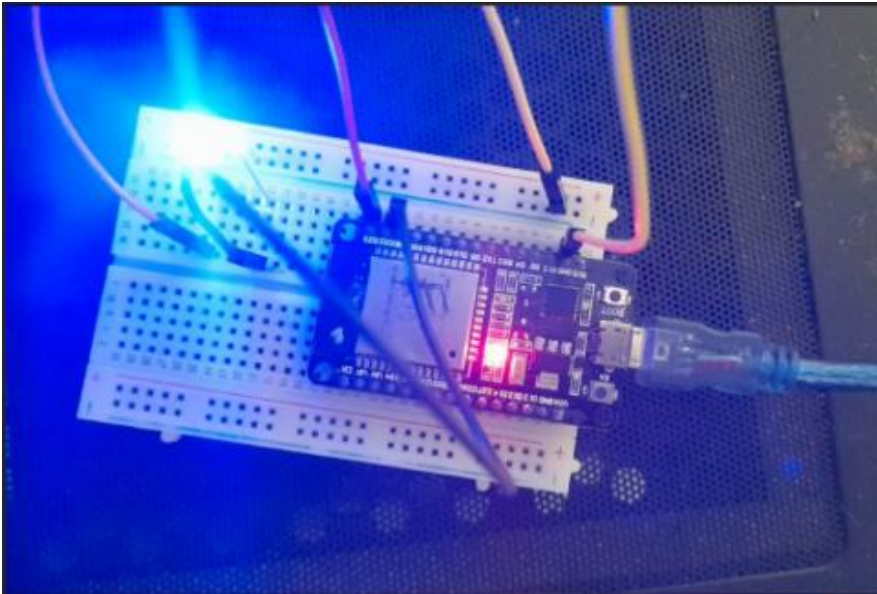
if (currentLine.endsWith("GET /led1on")) LED1_Status = HIGH;

if (currentLine.endsWith("GET /led1off")) LED1_Status = LOW;

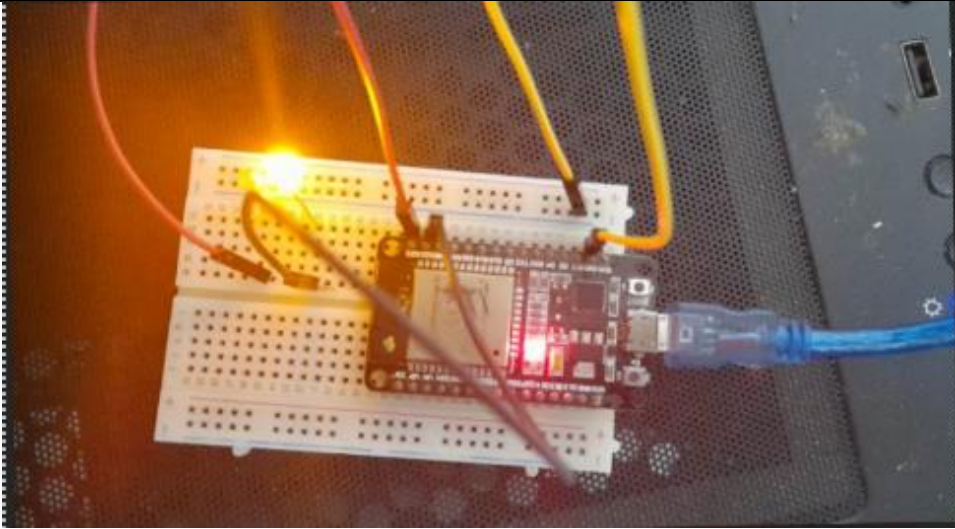
```

```
    if (currentLine.endsWith("GET /led2on")) LED2_Status = HIGH;  
    if (currentLine.endsWith("GET /led2off")) LED2_Status = LOW;  
  }  
}  
client.stop();  
Serial.println("Client Disconnected.");  
}  
}
```

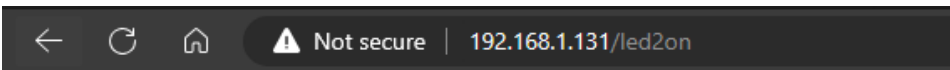
รูปการต่อวงจร – 1



รูปการต่อวงจร – 2



หน้าจอ Web Control



LED Status

LED1-Off,LED2-On

LED Control

LED1 On LED2 On

LED1 Off LED2 Off

Quiz_202 – Web Control 4 LED and Monitor Humid/Temperature

- เพิ่มเติมจาก Q202 อยากได้ปุ่มสำหรับคุมเปิด-ปิด หลอดไฟ LED 4 ดวง
- อยากมีกด Link ไปที่หน้า FB ของตัวเอง

←

→

⌂

Not secure | 192.168.43.237

The ESP-32 Update web page without refresh

LED1 ON

LED2 ON

LED3 ON

LED4 ON

LED1 OFF

LED2 OFF

LED3 OFF

LED4 OFF

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 15
#define testLED1 22
#define testLED2 23
#define testLED3 21
#define testLED4 19

//SSID and Password of your WiFi router
const char* ssid = "RATSIRI-HOME 2.4G";
const char* password = "0984485615";

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(testLED1, HIGH); ledState1 = "ON"; } //Feedback parameter
  if (t_state == "10") { digitalWrite(testLED1, LOW); ledState1 = "OFF"; } //Feedback parameter
  if (t_state == "21") { digitalWrite(testLED2, HIGH); ledState2 = "ON"; } //Feedback parameter
  if (t_state == "20") { digitalWrite(testLED2, LOW); ledState2 = "OFF"; } //Feedback parameter
  if (t_state == "31") { digitalWrite(testLED3, HIGH); ledState3 = "ON"; } //Feedback parameter
  if (t_state == "30") { digitalWrite(testLED3, LOW); ledState3 = "OFF"; } //Feedback parameter
  if (t_state == "41") { digitalWrite(testLED4, HIGH); ledState4 = "ON"; } //Feedback parameter
  if (t_state == "40") { digitalWrite(testLED4, 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::DHT11); // DHT_Pin D23, DHT11

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");

}

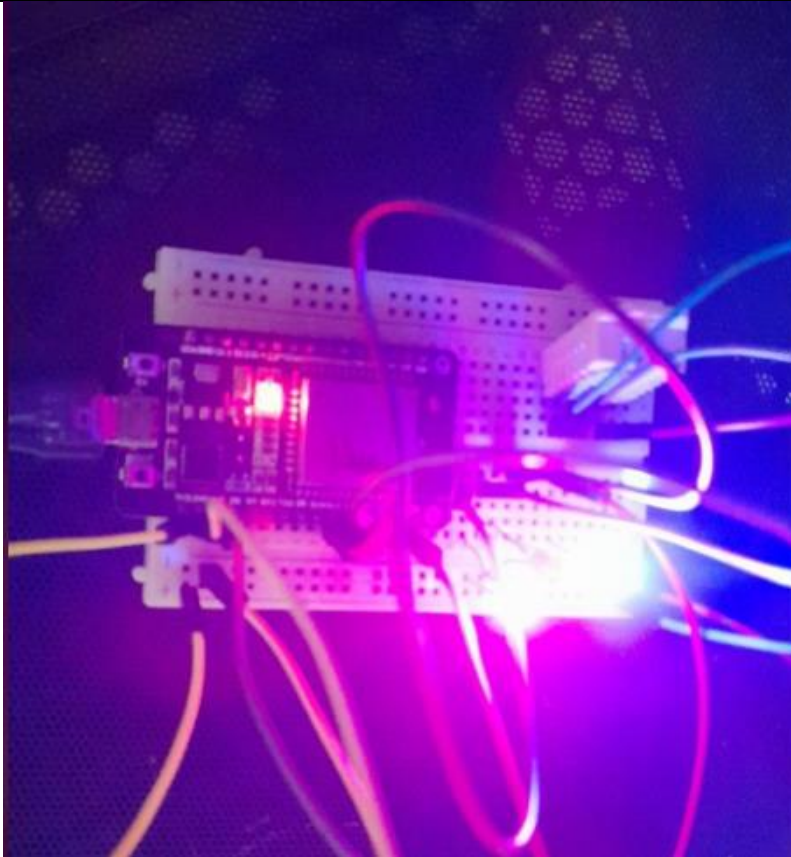
void loop(void) {

  server.handleClient(); //Handle client requests

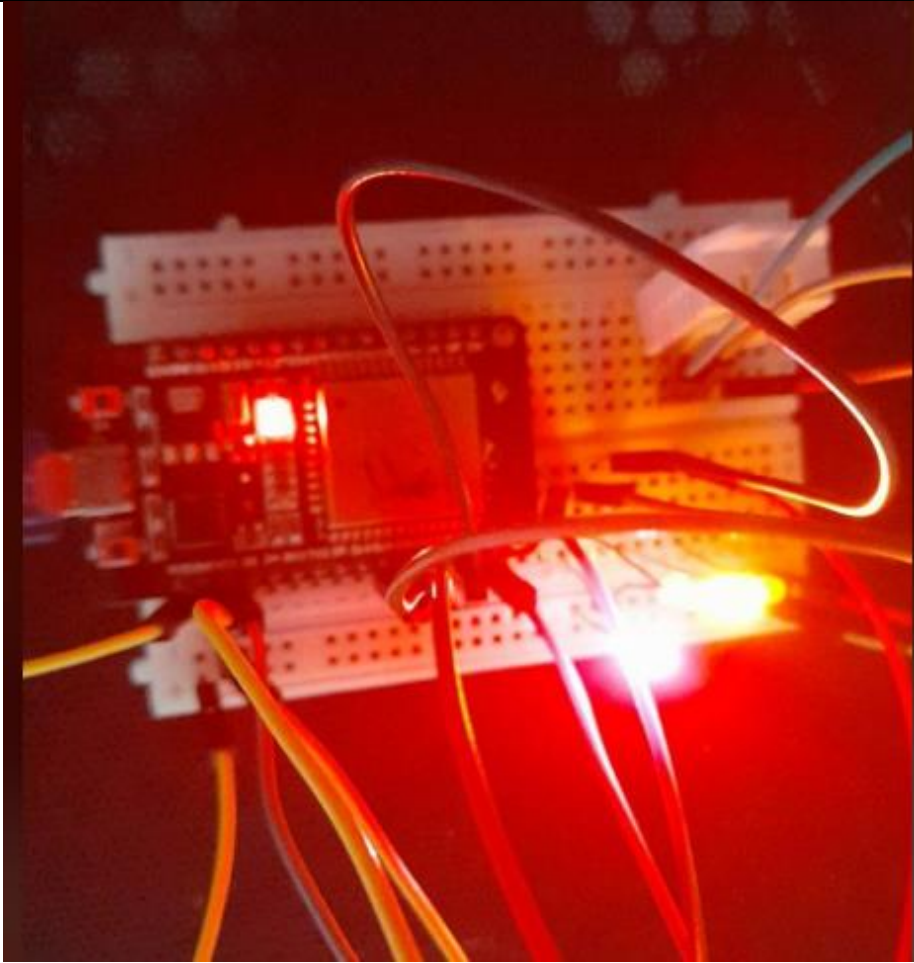
}

```

รูปการต่อวงจร – 1



รูปการต่อวงจร - 2



หน้าจอ Web Control

The ESP-32 Update web page without refresh

LED1 ON LED1 OFF

LED2 ON LED2 OFF

LED3 ON LED3 OFF

LED4 ON LED4 OFF

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

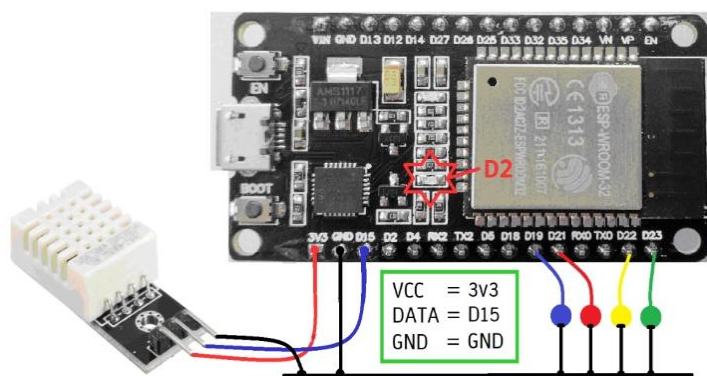
DHT-22 sensor : Temp = 8.50 C, Humidity = 23.30 %

[My Facebook :D :D](#)

Quiz_203 – Publish

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

*■■■(Blink)	หากการอ่านค่าแล้วเป็น null, หรือไม่มีเซ็นเซอร์
■■■■	ช่วงของอุณหภูมิ $(-\infty, 24)$
■■■■	ช่วงของอุณหภูมิ $[24,26)$
■■■■	ช่วงของอุณหภูมิ $[26,28)$
■■■■	ช่วงของอุณหภูมิ $[28,30)$
****(Blink)	ช่วงของอุณหภูมิ $[30,\infty)$



< Test Code >

```
#include <WiFi.h>
#include <PubSubClient.h>
#include "DHTesp.h"
#define Pin_DHT22 15

const char* ssid = "RATSIRI-HOME 2.4G"; //Your Wifi
const char* password = "0984485615"; //Your Wifi password
const char* mqtt_server = "test.mosquitto.org";
const char* topic1 = "Test";

DHTesp dht;
WiFiClient espClient;
PubSubClient client(espClient);

long lastMsg = 0;
char msg[50];
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 LED_state(float tempp)
{ if (tempp < 24)
    { digitalWrite(LED1, HIGH);
      digitalWrite(LED2, LOW);
      digitalWrite(LED3, LOW);
      digitalWrite(LED4, LOW); }
  else if (tempp >= 24 && tempp < 26)
  { digitalWrite(LED1, HIGH);
    digitalWrite(LED2, HIGH);
    digitalWrite(LED3, LOW);
    digitalWrite(LED4, LOW); }
  else if (tempp >= 26 && tempp < 28)
  { digitalWrite(LED1, HIGH);
    digitalWrite(LED2, HIGH);
    digitalWrite(LED3, HIGH);
    digitalWrite(LED4, LOW); }
```

```

else if (tempp >= 28 && tempp < 30)
{
    digitalWrite(LED1, HIGH);

    digitalWrite(LED2, HIGH);

    digitalWrite(LED3, HIGH);

    digitalWrite(LED4, HIGH); }

else if (tempp >= 30)
{
    digitalWrite(LED1, HIGH);digitalWrite(LED2, HIGH);

    digitalWrite(LED3, HIGH);digitalWrite(LED4, HIGH);delay(250);

    digitalWrite(LED1, LOW); digitalWrite(LED2, LOW);

    digitalWrite(LED3, LOW); digitalWrite(LED4, LOW);delay(250);}

else
{
    digitalWrite(LED1, HIGH);delay(250);

    digitalWrite(LED1, LOW);delay(250);

    digitalWrite(LED2, LOW);

    digitalWrite(LED3, LOW);

    digitalWrite(LED4, LOW);}
}

void reconnect()
{
    while (!client.connected()) // Loop until we're reconnected

    {
        Serial.println("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 EIEI"); // ... and resubscribe

            client.subscribe(topic1);
        } else
        {
            Serial.print("failed, rc=");

            Serial.print(client.state());

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

            delay(5000);
        }
    }
}

```

```

}
}
}

void setup()
{
  Serial.begin(115200);

  pinMode(LED1, OUTPUT);
  pinMode(LED2, OUTPUT);
  pinMode(LED3, OUTPUT);
  pinMode(LED4, OUTPUT);

  dht.setup(Pin_DHT22, DHTesp::DHT22);

  setup_wifi();

  client.setServer(mqtt_server, 1883);
}

void loop()
{
  if (!client.connected()) reconnect();

  client.loop();

  long now = millis();

  float humid = dht.getHumidity();

  float temp = dht.getTemperature();

  LED_state(temp);

  if (now - lastMsg > 5000)
  {
    lastMsg = now;

    snprintf (msg, 75, "Temp: %.2f'C Humid: %.2f%%",temp,humid);

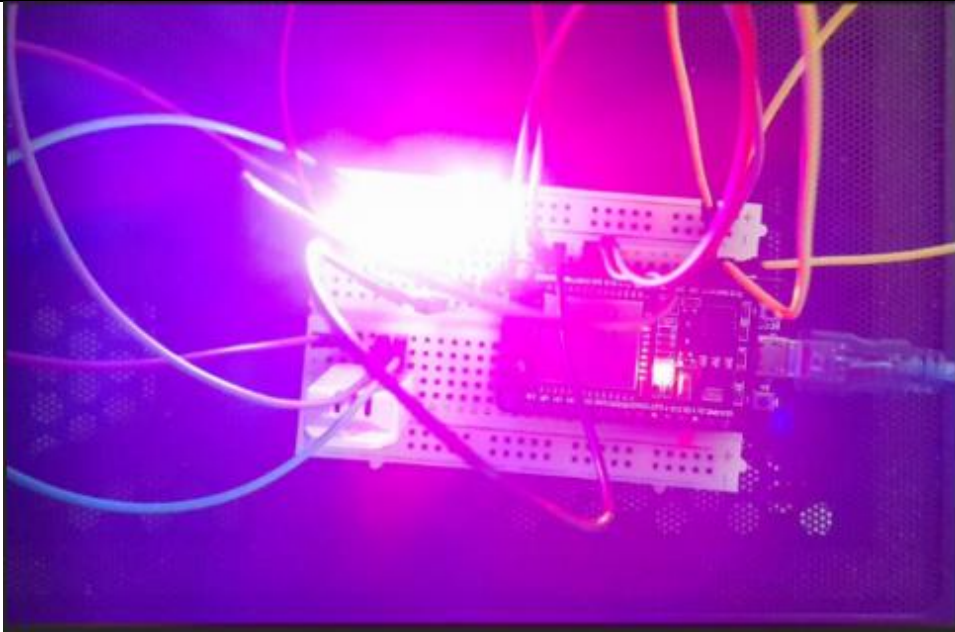
    Serial.print("Publish message: ");

    Serial.println(msg);

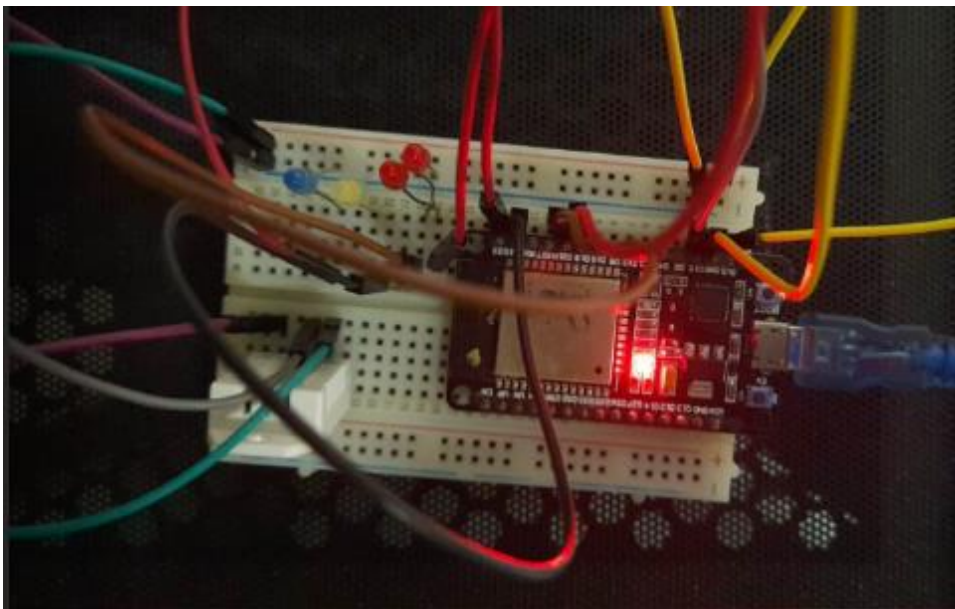
    client.publish(topic1, msg);
  }
}

```

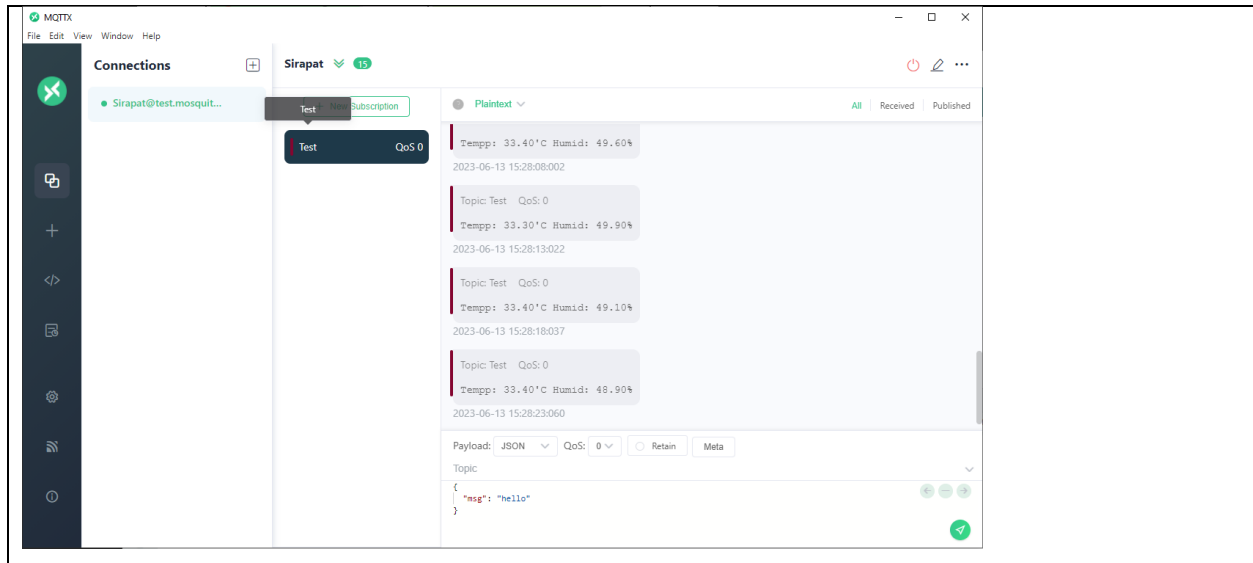
รูปการต่อวงจร – 1



รูปการต่อวงจร - 2

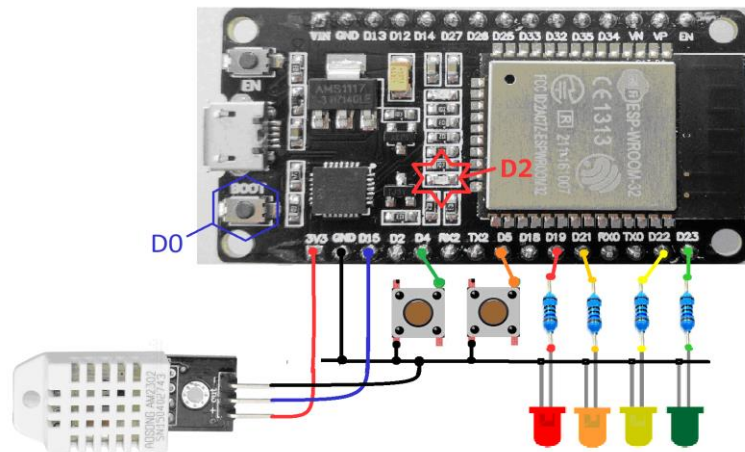


หน้าจอ MQTT Lens



Quiz_204 – Publish and Subscribe

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



< Test Code >

```
#include <WiFi.h>
#include <PubSubClient.h>
#include "DHTesp.h"
#define Pin_DHT22 15

const char* ssid = "RATSIRI-HOME 2.4G"; //Your Wifi
const char* password = "0984485615"; //Your Wifi password
const char* mqtt_server = "test.mosquitto.org";
const char* topic1 = "Test";

DHTesp dht;
WiFiClient espClient;
PubSubClient client(espClient);

long lastMsg = 0;
char msg[50];
int Counter = 0;
```

```

int SW1 = 2;

int SW2 = 4;

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, "sadsadsadsad"); // ... 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[4] = '\0'; // String less than 4 characters

    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 .");
      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 Come");

  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 temp = dht.getTemperature();

  snprintf (msg, 75, "Temp: %.2f'C Humid: %.2f%",temp,humid);

  Serial.print("Publish message: ");

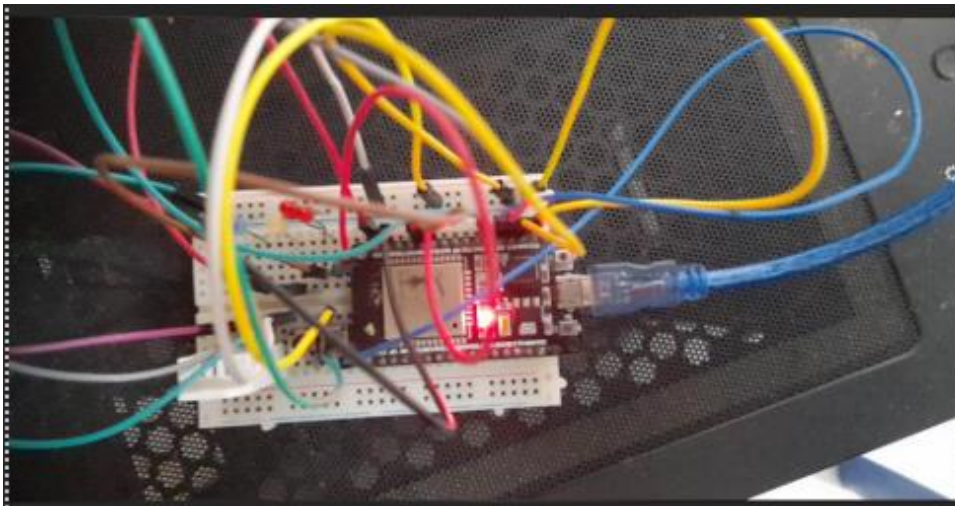
  Serial.println(msg);

  client.publish(topic1, msg); }

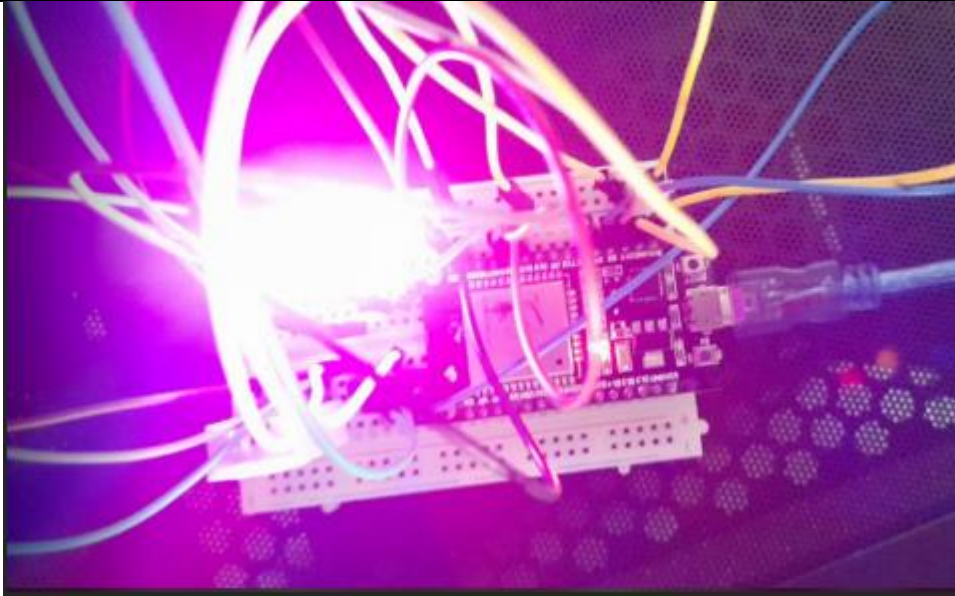
}

```

รูปการต่อวงจร – 1



รูปการต่อวงจร – 2



หน้าจอ MQTT Lens

The screenshot displays the MQTTX application window. On the left, the 'Connections' panel shows a single connection named 'Sirapat@test.mosquit...'. The main area is titled 'Sirapat' and shows a subscription to the 'Test' topic with 'QoS 0'. A message history list on the right shows three received messages:

Topic	QoS	Message	Timestamp
Topic: Test	QoS: 0	ON1	2023-06-13 15:45:306
Topic: Test	QoS: 0	sadsadsadsad	2023-06-13 15:46:00:185
Topic: Test	QoS: 0	Overheat .	2023-06-13 15:46:00:602

At the bottom, the 'Payload' section shows the selected message 'ON1' with its 'QoS' set to 0 and 'Retain' checked.