แนวทางการใช้งานอินเทอร์เน็ตของสรรพสิ่งในระบบการผลิต

IoT Approaches to Manufacturing System

ชื่อ-สกุล : B6310646 สุภานัน เรืองสุข

4/4. คำถามท้ายบทเพื่อทดสอบความเข้าใจ

Quiz_201 – Web Control 2 LED

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

< Test Code >

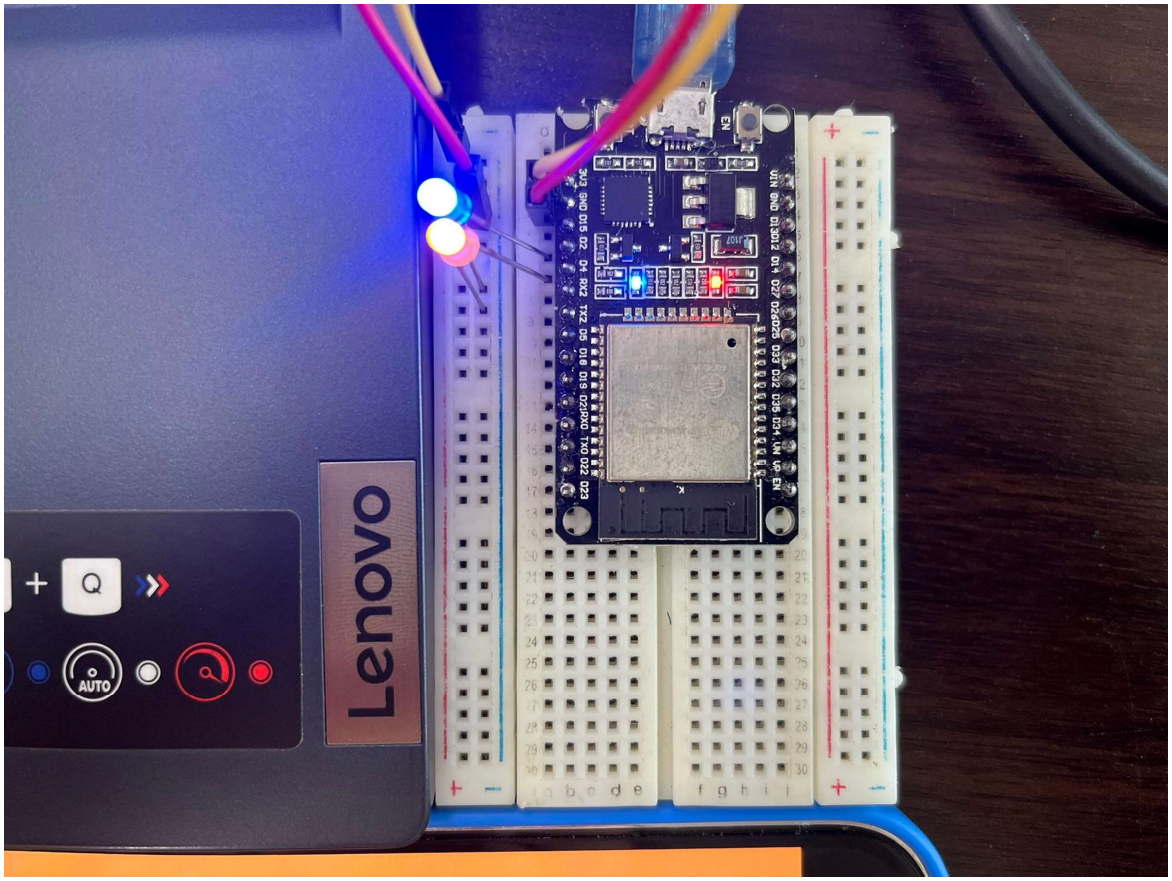
```
#include <WiFi.h>
const char* ssid = "meow";
const char* password = "meowmeow";
int pin2 = 2;
int pin4 = 4;
WiFiServer server(80);
void setup() {
  Serial.begin(115200);
  pinMode(pin2, OUTPUT); // set the LED pin mode
  pinMode(pin4, OUTPUT); // set the LED pin mode
  delay(10);
  Serial.print("\n\nConnecting 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_2 = LOW;
bool LED_Status_4 = LOW;
void loop() {
  digitalWrite(pin2, LED_Status_2);
  digitalWrite(pin4, LED_Status_4);
  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_2 == HIGH)
              client.println("LED On 2");
            else if (LED_Status_2 == LOW)
              client.println("LED Off 2");
            client.println(",");
            if (LED_Status_4 == HIGH)
              client.println("LED On 4");
            else if (LED_Status_4 == LOW)
              client.println("LED Off 4");
            client.println("<p>");
            client.println("<a href=\"/ledon_2\"><button style = \"background-color:
#BF40BF;\">LED On</button></a>");
```

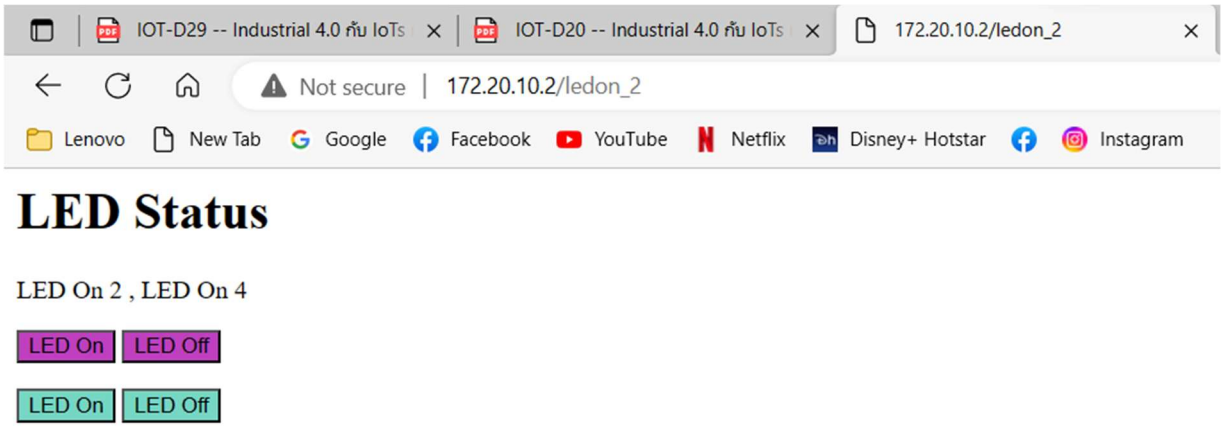
```
        client.println("<a href=\"/ledoff_2\"><button style = \"background-color:
#BF40BF;\">LED Off</button></a>");
        client.println("</p>");


        client.println("<a href=\"/ledon_4\"><button style = \"background-color:
#76D7C4;\">LED On</button></a>");
        client.println("<a href=\"/ledoff_4\"><button style = \"background-color:
#76D7C4;\">LED Off</button></a>");
        client.println("</p>");


        client.println("<body>");
        client.println("<html>");
        break;
       } else {
        currentLine = "";
       }
     } else if (c != '\r') {
      currentLine += c;
     }
     if (currentLine.endsWith("GET /ledon_2")) LED_Status_2 = HIGH;
     if (currentLine.endsWith("GET /ledoff_2")) LED_Status_2 = LOW;
     if (currentLine.endsWith("GET /ledon_4")) LED_Status_4 = HIGH;
     if (currentLine.endsWith("GET /ledoff_4")) LED_Status_4 = LOW;
    }
   }
   client.stop(); // close the connection:
   Serial.println("Client Disconnected.");
  }
}
```

รูปการต่อวงจร – 1

รูปการต่อวงจร – 2



หน้าจอ Web Control



Quiz_202 – Web Control 4 LED and Monitor Humid/Temperature

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

< Test Code >

```
#include <WiFi.h>
const char* ssid = "meow";
const char* password = "meowmeow";
int pin5 = 5;
int pin18 = 18;
int pin19 = 19;
int pin21 = 21;
WiFiServer server(80);
#define DHT22_Pin 15
#include "DHTesp.h"
DHTesp dht;
void setup() {
  Serial.begin(115200);
  dht.setup(DHT22_Pin, DHTesp::DHT22); // Connect DHT sensor to GPIO 15
  pinMode(pin5, OUTPUT); // set the LED pin mode
  pinMode(pin18, OUTPUT); // set the LED pin mode
  pinMode(pin19, OUTPUT); // set the LED pin mode
  pinMode(pin21, OUTPUT); // set the LED pin mode
  delay(10);
  Serial.print("\n\nConnecting to "); Serial.println(ssid);
  WiFi.begin(ssid, password);
```

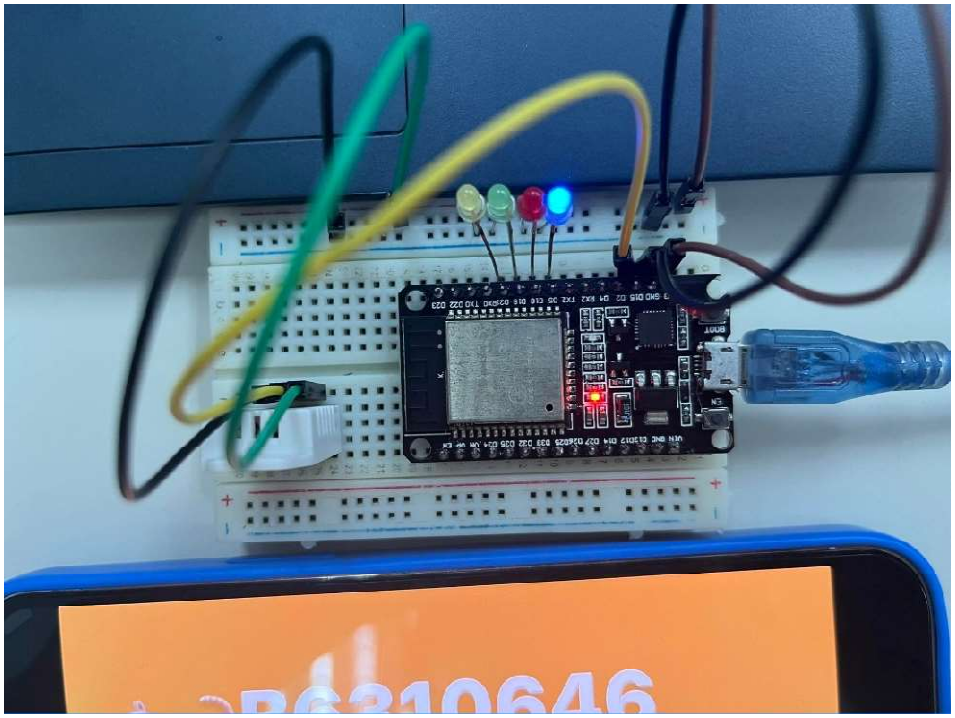
```
  while (WiFi.status() != WL_CONNECTED) {
    delay(500); Serial.print(".");
  }
  Serial.println("");
  Serial.println("WiFi connected."); Serial.println("IP address: ");
  Serial.println(WiFi.localIP()); server.begin();
}
int value = 0;
bool LED_Status_5 = LOW;
bool LED_Status_18 = LOW;
bool LED_Status_19 = LOW;
bool LED_Status_21 = LOW;
void loop() {
  digitalWrite(pin5, LED_Status_5);
  digitalWrite(pin18, LED_Status_18);
  digitalWrite(pin19, LED_Status_19);
  digitalWrite(pin21, LED_Status_21);
  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>The ESP-32 Update web page without refresh</h1>");
            client.println("<p>");
            client.println("<a href=\"/ledon_5\"><button style = \"background-color:
#BF40BF;\">LED1 ON</button></a>");
            client.println("<a href=\"/ledon_18\"><button style = \"background-color:
#BF40BF;\">LED2 ON</button></a>");
```

```
        client.println("<a href=\"/ledon_19\"><button style = \"background-color:
#BF40BF;\">LED3 ON</button></a>");
        client.println("<a href=\"/ledon_21\"><button style = \"background-color:
#BF40BF;\">LED4 ON</button></a>");
        client.println("</p>");


        client.println("<a href=\"/ledoff_5\"><button style = \"background-color:
#76D7C4;\">LED1 OFF</button></a>");
        client.println("<a href=\"/ledoff_18\"><button style = \"background-color:
#76D7C4;\">LED2 OFF</button></a>");
        client.println("<a href=\"/ledoff_19\"><button style = \"background-color:
#76D7C4;\">LED2 OFF</button></a>");
        client.println("<a href=\"/ledoff_21\"><button style = \"background-color:
#76D7C4;\">LED2 OFF</button></a>");
        client.println("</p>");


        client.println("State of[LED1,LED2,LED3,LED4] is >> ");
        if (LED_Status_5 == HIGH)
          client.println("ON");
        else if (LED_Status_5 == LOW)
          client.println("OFF");
        client.println(",");
        if (LED_Status_18 == HIGH)
          client.println("ON");
        else if (LED_Status_18 == LOW)
          client.println("OFF");
        client.println(",");
        if (LED_Status_19 == HIGH)
          client.println("ON");
        else if (LED_Status_19 == LOW)
          client.println("OFF");
        client.println(",");
        if (LED_Status_21 == HIGH)
          client.println("ON");
        else if (LED_Status_21 == LOW)
          client.println("OFF");
        client.println("<p>");


        delay(dht.getMinimumSamplingPeriod());
        float humidity = dht.getHumidity();
```

```
          float temperature = dht.getTemperature();
          client.print("Temperature: ");
          client.print(temperature, 1);
          client.print("C/");
          client.print(dht.toFahrenheit(temperature), 1);
          client.print("F. Humidity: ");
          client.print(humidity, 1);
          client.print("%\n");
          delay(2000);
          client.println("<p>");

          client.println("<br><a
href=\"https://www.facebook.com/miaw.supanan/\">Facebook: Supanan
Rueangsook</a>");

          client.println("<body>");
          client.println("<html>");
          break;
        } else {
          currentLine = "";
        }
      } else if (c != '\r') {
        currentLine += c;
      }
      if (currentLine.endsWith("GET /ledon_5")) LED_Status_5 = HIGH;
      if (currentLine.endsWith("GET /ledoff_5")) LED_Status_5 = LOW;
      if (currentLine.endsWith("GET /ledon_18")) LED_Status_18 = HIGH;
      if (currentLine.endsWith("GET /ledoff_18")) LED_Status_18 = LOW;
      if (currentLine.endsWith("GET /ledon_19")) LED_Status_19 = HIGH;
      if (currentLine.endsWith("GET /ledoff_19")) LED_Status_19 = LOW;
      if (currentLine.endsWith("GET /ledon_21")) LED_Status_21 = HIGH;
      if (currentLine.endsWith("GET /ledoff_21")) LED_Status_21 = LOW;
    }
  }
  client.stop(); // close the connection:
  Serial.println("Client Disconnected.");
 }
}
```

รูปการต่อวงจร – 1

รูปการต่อวงจร – 2

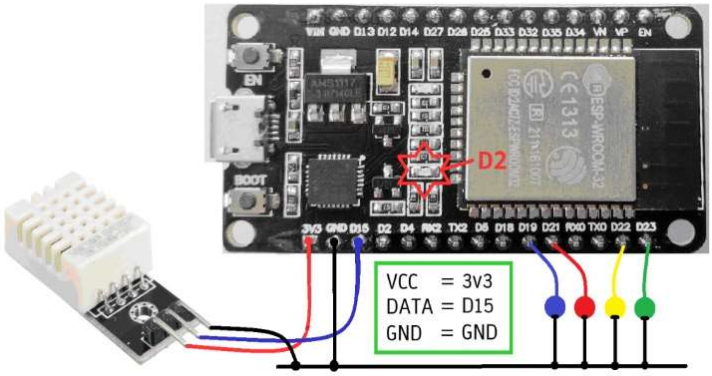


หน้าจอ Web Control

Quiz_203 – Publish

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

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




```
< 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 = "meow";
const char* password = "meowmeow";
```

```
const char* mqtt_server = "test.mosquitto.org";  //MQTT
const char* topic1 = "MS1";
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");
```

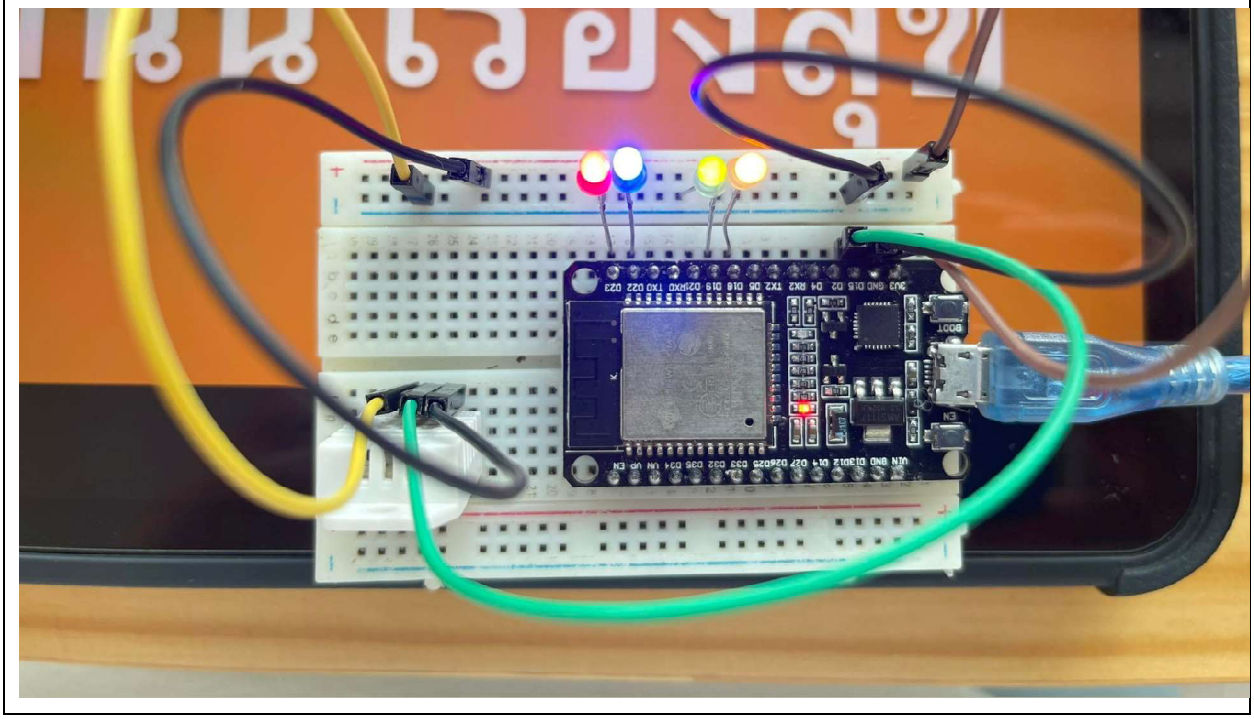
```
    delay(5000);
   }
  }
}

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++) {
```

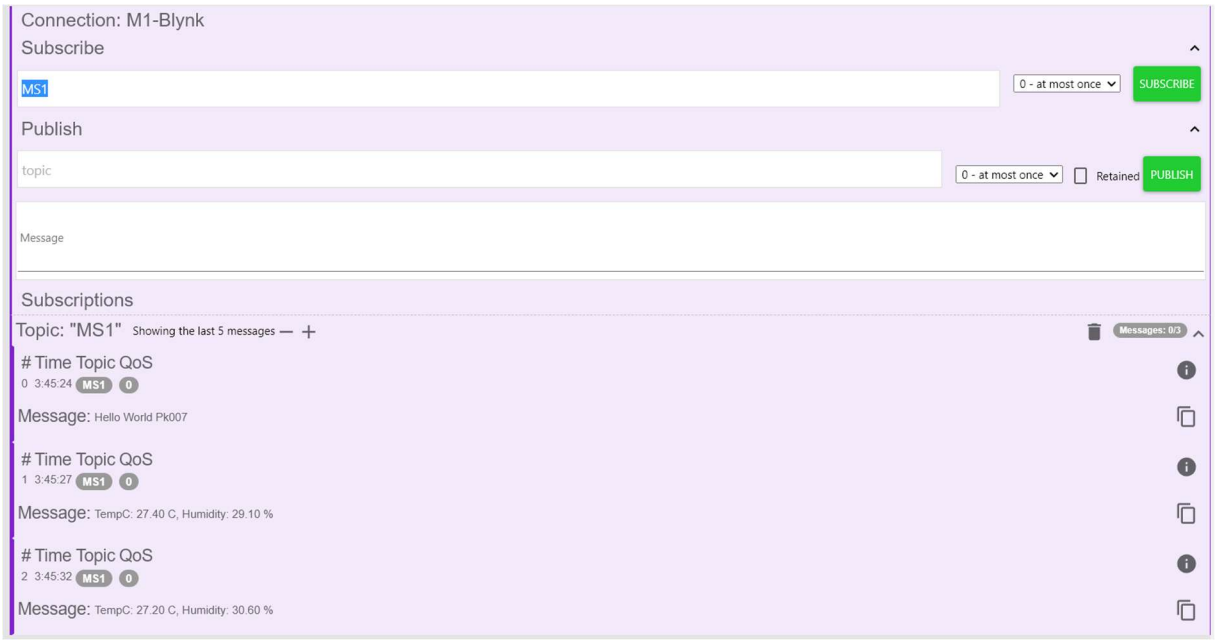
```
    pinMode(LED_PinArray[i], OUTPUT);
  }
}
void loop()
{
  if (!client.connected()) reconnect();
  client.loop();
  long now = millis();
  if (now - lastMsg > 5000)
  { lastMsg = now;
    ++value;
    //float t = s.readTempC();
    //float h = s.readHumidity();
    delay(dht.getMinimumSamplingPeriod());
    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

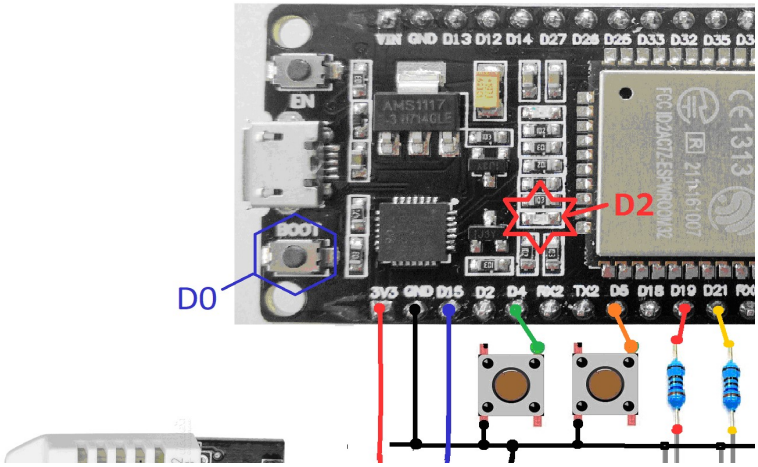


หน้าจอ **MQTT Lens**



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



< Test Code >

```
#include <WiFi.h>
#include <PubSubClient.h>
#include "DHTesp.h"
#define Pin_DHT22 15
const char* ssid = "meow"; //Your Wifi
const char* password = "meowmeow"; //Your Wifi password
const char* mqtt_server = "test.mosquitto.org";
const char* topic1 = "Flow1";
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);
```

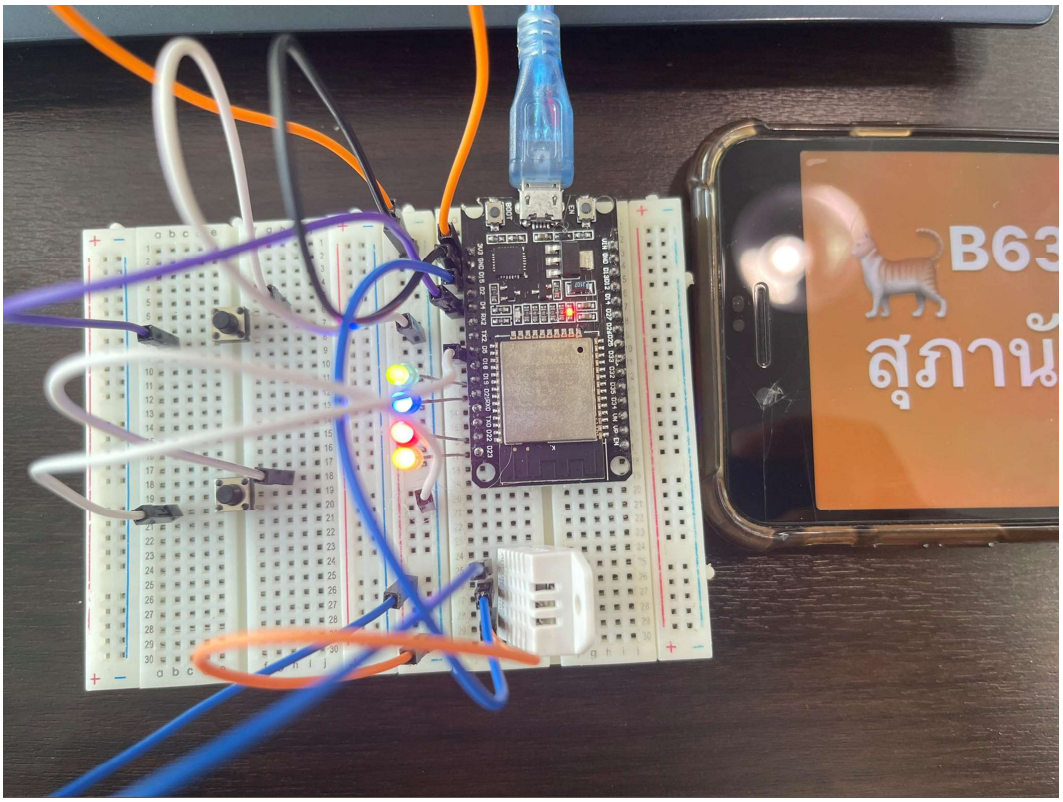
```
Serial.print("] ");
for (int i = 0; i < length; i++)
{ Serial.print((char)payload[i]);
  myPayLoad[i] = payload[i];
  myPayLoad[i + 1] = '\0'; // End of String
}
Serial.print("\n ---> "); Serial.println(myPayLoad);
myPayLoad[4] = '\0'; // String lessthan 4 Charector
if ((String)myPayLoad == "ON")
{ 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, "Alert!!! The temperature is too high.");
      Serial.print("Publish message: ");
      Serial.println(msg);
      client.publish(topic1, msg);
      while (digitalRead(SW1) == 0);
      delay(100);
    }
    if (digitalRead(SW2) == 0)
    { client.loop();
```

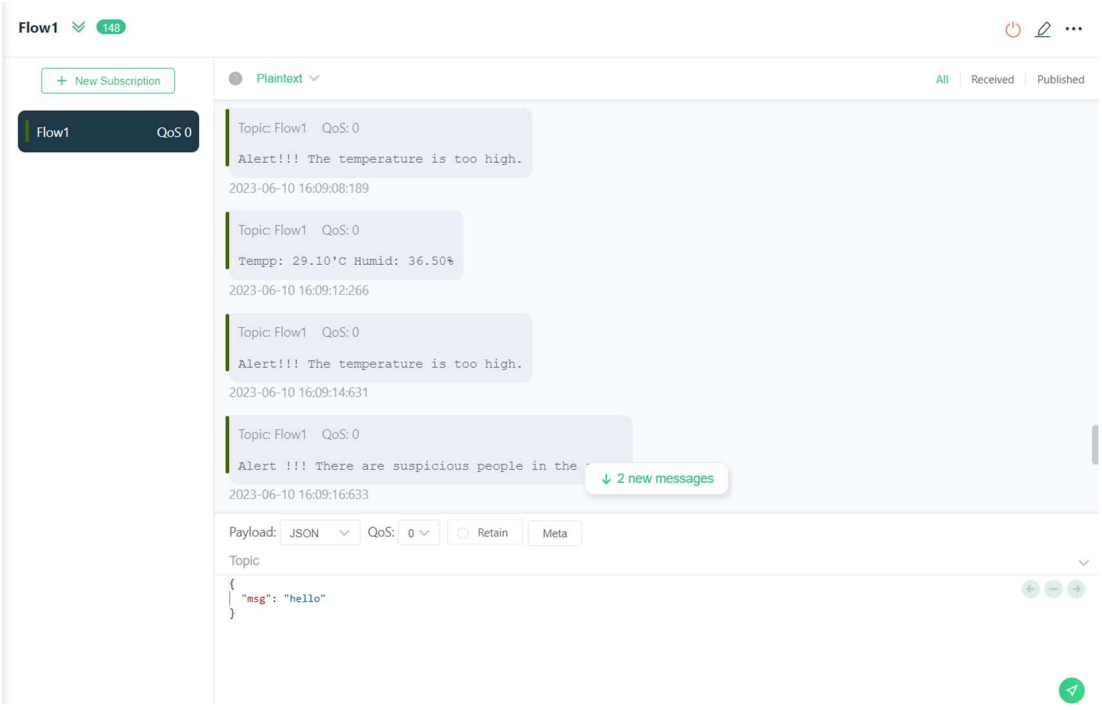
```
    snprintf (msg, 75, "Alert !!! There are suspicious people in the area.");
    Serial.print("Publish message: ");
    Serial.println(msg);
    client.publish(topic1, msg);
    while (digitalRead(SW2) == 0);
    delay(100);
   }
 }
 long now = millis();
 if (now - lastMsg > 5000)
 { lastMsg = now;
  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);
 }
}
```

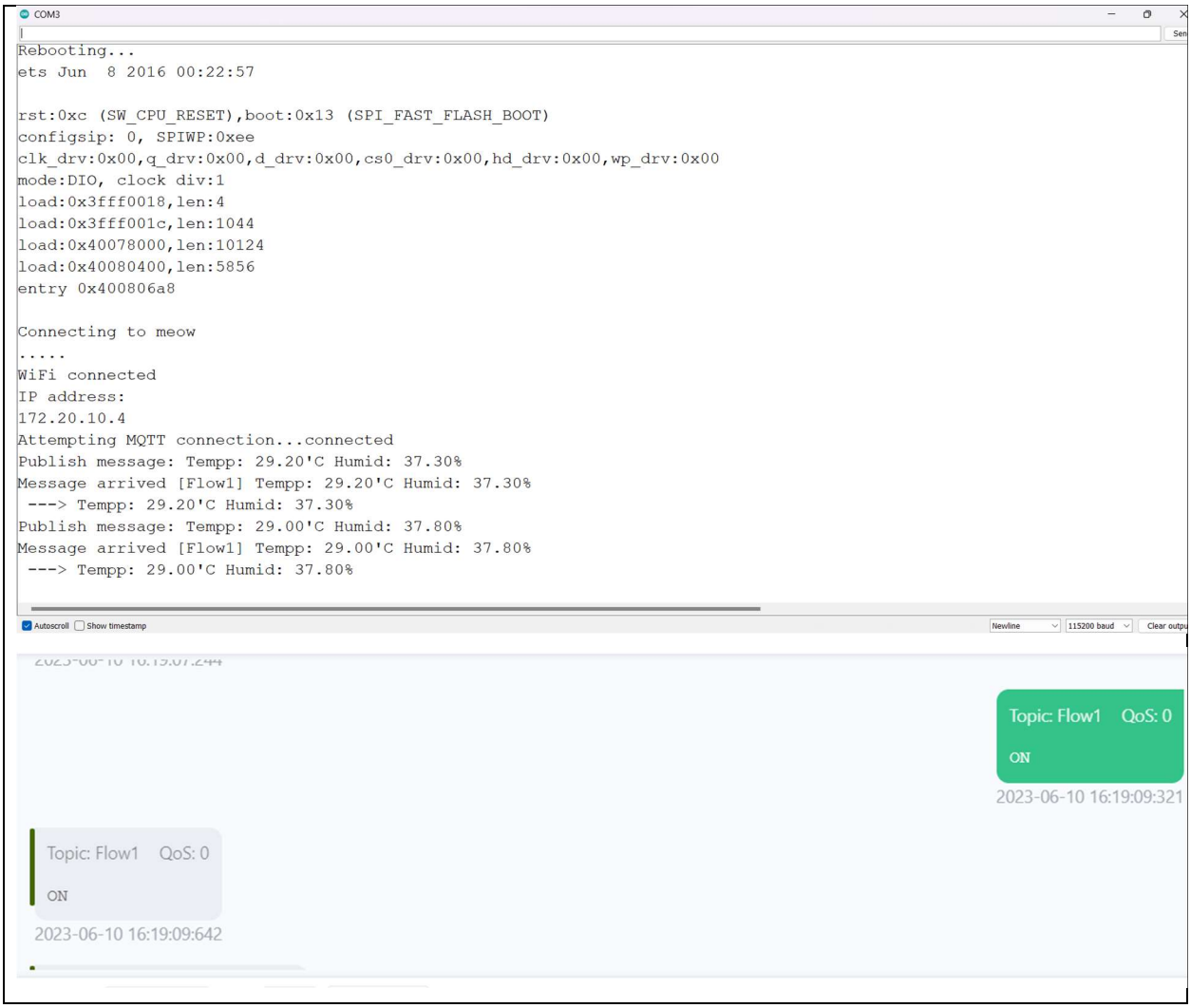
รูปการต่อวงจร – 1

รูปการต่อวงจร – 2



หน้าจอ MQTT x




COM3
Rebooting...
ets Jun 8 2016 00:22:57
rst:0xc (SW_CPU_RESET),boot:0x13 (SPI_FAST_FLASH_BOOT)
configsip: 0, SPIWP:0xee
clk_drv:0x00,q_drv:0x00,d_drv:0x00,cs0_drv:0x00,hd_drv:0x00,wp_drv:0x00
mode:DIO, clock div:1
load:0x3fff0018,len:4
load:0x3fff001c,len:1044
load:0x40078000,len:10124
load:0x40080400,len:5856
entry 0x400806a8
Connecting to meow
.....
WiFi connected
IP address:
172.20.10.4
Attempting MQTT connection...connected
Publish message: Tempp: 29.20'C Humid: 37.30%
Message arrived [Flow1] Tempp: 29.20'C Humid: 37.30%
---> Tempp: 29.20'C Humid: 37.30%
Publish message: Tempp: 29.00'C Humid: 37.80%
Message arrived [Flow1] Tempp: 29.00'C Humid: 37.80%
---> Tempp: 29.00'C Humid: 37.80%
Autoscroll
Show timestamp
Newline
115200 baud
Clear output
Topic: Flow1 QoS: 0
ON
2023-06-10 16:19:09:321
Topic: Flow1 QoS: 0
ON
2023-06-10 16:19:09:642