

การควบคุมเครื่องจักรอัจฉริยะโดยใช้การสื่อสารระหว่างเครื่องจักรกับเครื่องจักร

M2M - Intelligence Machine Control

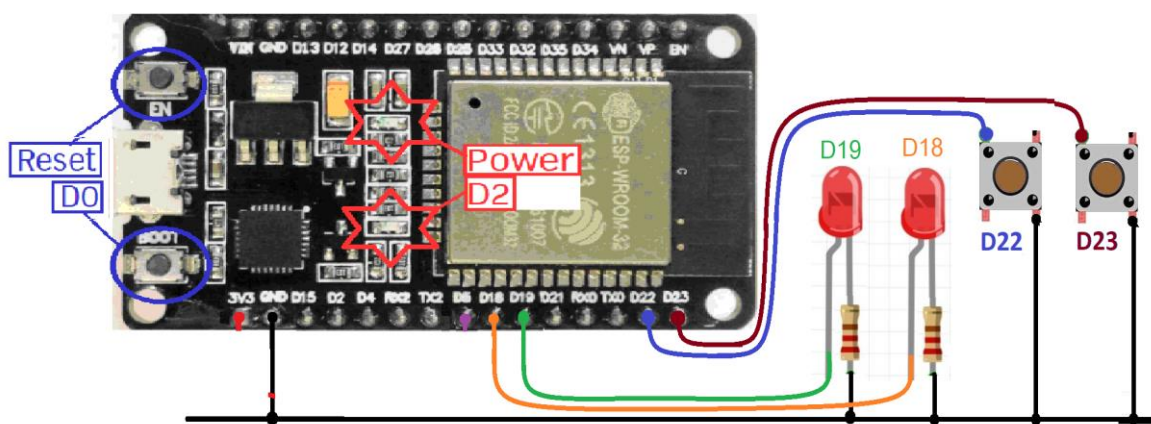
ชื่อ-สกุล : นายปราชญา ธนพิบูลผล

รหัสนักศึกษา : B6323059

6/6 -- คำถามท้ายบทเพื่อทดสอบความเข้าใจ

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}



```
#define pushButton1 22

#define LEDPin1 19

#define pushButton2 23

#define LEDPin2 18

int buttonState1 = 0;

int buttonState2 = 0;

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

  pinMode(pushButton1, INPUT_PULLUP);

  pinMode(LEDPin1, OUTPUT);

  pinMode(pushButton2, INPUT_PULLUP);
```

```
pinMode(LEDPin2, OUTPUT); }

voidloop() {
if (digitalRead(pushButton1) == LOW) {
delay(20);

buttonState1 = 1 -buttonState1;

digitalWrite(LEDPin1, buttonState1);

while (digitalRead(pushButton1) == LOW);

delay(20);}

if (digitalRead(pushButton2) == LOW) {

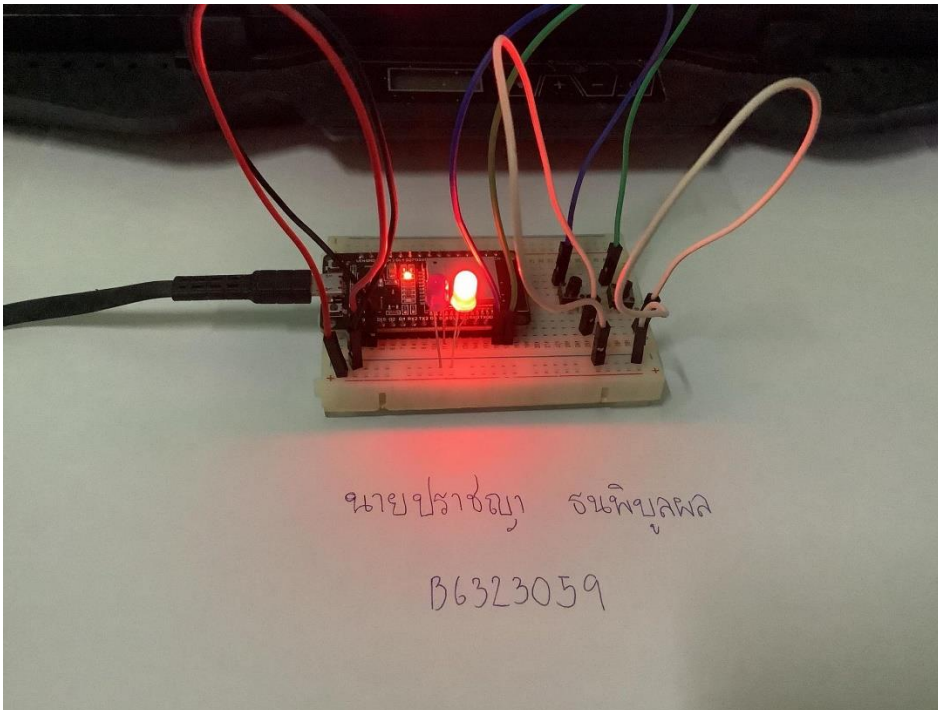
delay(20);

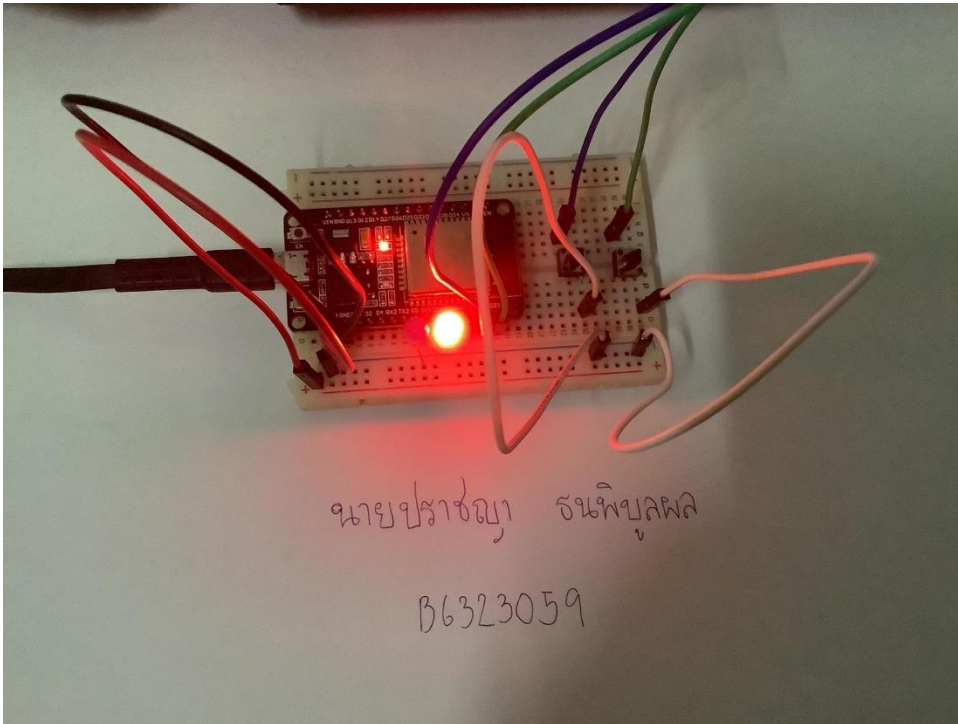
buttonState2 = 1 -buttonState2;

digitalWrite(LEDPin2, buttonState2);

while (digitalRead(pushButton2) == LOW);

delay(20); } }
```





Quiz_102 – Web Control 4 LED and Monitor Humid/Temperature

- เพิ่มเติมจาก Q202 อยากได้ปุ่มสำหรับคุมปิด-เปิด หลอดไฟ LED 4 ดวง
- อยากมีกิด Link ไปที่หน้า FB ของตัวเอง
- https://www.colorhexa.com/008cba?fbclid=IwAR3dIZ_gRgDWmREmnzknLbMxV3pOHY4YIPuLEz8-ZzTOX2VhWxcH2QjLGk

←

→

↻

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](#)

```

#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 18

#define testLED2 19

#define testLED3 22

#define testLED4 23

//SSID and Password of your WiFi router

const char* ssid = "It's bad day not a bad life";

const char* password = "0641453596";

```

```

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); //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";
}

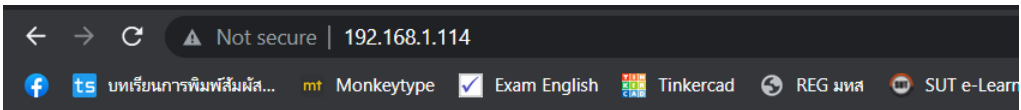
if (t_state == "31") {
    digitalWrite(testLED3, HIGH); //Feedback parameter
    ledState3 = "ON";
}

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/plain", ledState1 + ", " + ledState2 + ", " + ledState3 + ", " + ledState4);  
  
  //Send web page  
  
}  
  
void setup(void) {  
  
  Serial.begin(115200);  
  
  dht.setup(DHT_Pin, DHTesp::DHT22); // DHT_Pin D4, DHT22  
  
  pinMode(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);
```



The ESP-32 Update web page without refresh

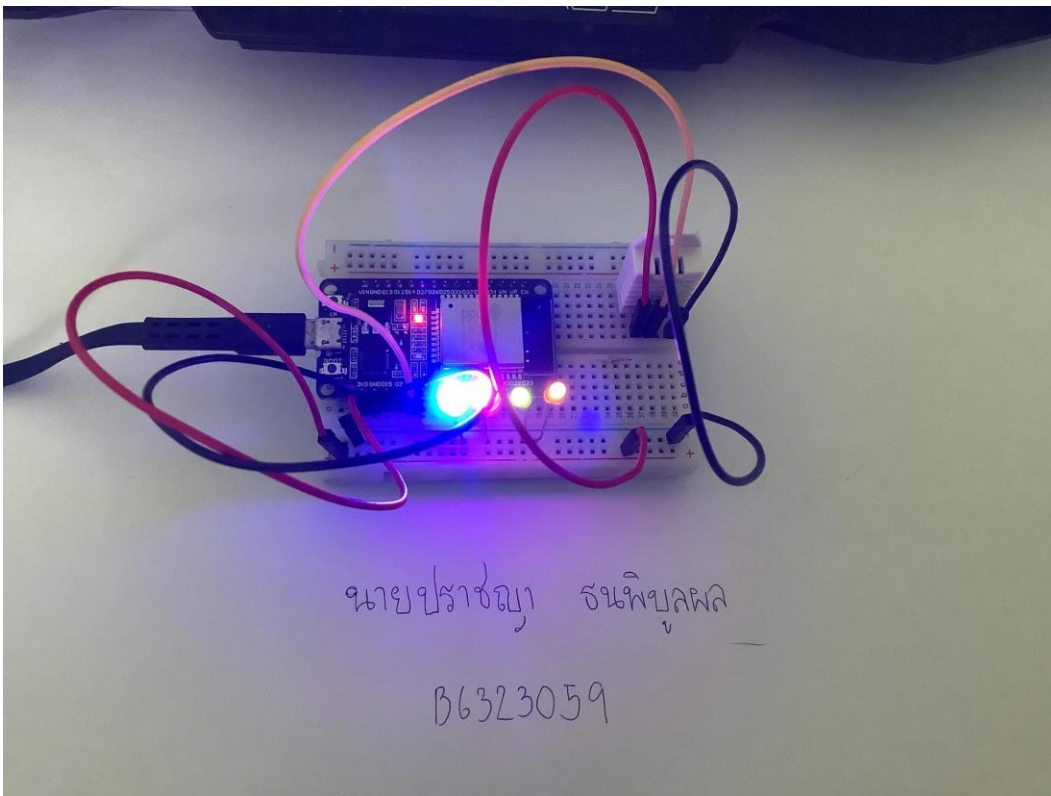
LED1 ON LED2 ON LED3 ON LED4 ON

LED1 OFF LED2 OFF LED3 OFF LED4 OFF

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

DHT-22 sensor : Temp = 36.20 C, Humidity = 1.40 %

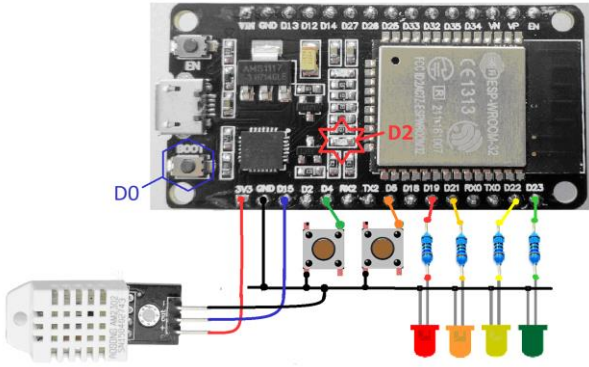

[Benz Pratchaya](#)





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
- ควบคุมการปิดเปิด 4 LED
- รับค่าสวิตช์กำหนด SW1 แจ้ง Overheat Alarm, SW2 แจ้ง Intruders Alarm

	
<pre> #include <WiFi.h> #include <Wire.h> #include <PubSubClient.h> #include "DHTesp.h" DHTesp dht; #define testLED1 19 #define testLED2 21 #define testLED3 22 #define testLED4 23 #define DHT22_Pin 15 const char* ssid = "It's bad day not a bad life"; const char* password = "0641453596"; const char* mqtt_server = "test.mosquitto.org"; const char* topic1 = "test2"; String ledState1 = "NA"; int pushButton1 = 4; </pre>	

```
int pushButton2 = 5;

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());

  pinMode(testLED1, OUTPUT);

  pinMode(testLED2, OUTPUT);

  pinMode(testLED3, OUTPUT);

  pinMode(testLED4, OUTPUT);

}

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 less than 4 characters

if ((String)myPayload == "ON1") digitalWrite(testLED1, HIGH);
if ((String)myPayload == "OFF1") digitalWrite(testLED1, LOW);
if ((String)myPayload == "ON2") digitalWrite(testLED2, HIGH);
if ((String)myPayload == "OFF2") digitalWrite(testLED2, LOW);
if ((String)myPayload == "ON3") digitalWrite(testLED3, HIGH);
if ((String)myPayload == "OFF3") digitalWrite(testLED3, LOW);
if ((String)myPayload == "ON4") digitalWrite(testLED4, HIGH);
if ((String)myPayload == "OFF4") digitalWrite(testLED4, LOW);
}

void reconnect()
{
  while (!client.connected()) // Loop until we're reconnected
  {
    Serial.print("Attempting MQTT connection...");

    String clientId = "ESP8266Client-";

    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 setup()

{ Serial.begin(115200);

  setup_wifi();

  dht.setup(DHT22_Pin, DHTesp::DHT22);

  pinMode(pushButton1, INPUT_PULLUP);

  pinMode(pushButton2, INPUT_PULLUP);

  client.setServer(mqtt_server, 1883);

  client.setCallback(callback);

  pinMode(testLED1, OUTPUT);

  pinMode(testLED2, OUTPUT);

  pinMode(testLED3, OUTPUT);

  pinMode(testLED4, OUTPUT);

}

void loop()

{

  if (!client.connected()) reconnect();

  client.loop();

  long now = millis();
```

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

    ++value;

    float h = dht.getHumidity();

    float t = dht.getTemperature();

    sprintf (msg, "TempC: %.2f C, Humidity: %.2f %%", t, h);

    Serial.print("Publish message: ");

    Serial.println(msg);

    client.publish(topic1, msg);
}

if (digitalRead(pushButton1) == 0) {

    sprintf (msg, "Overheat Alarm");

    Serial.println(msg);

    client.publish(topic1, msg);

    delay(500);

}

if (digitalRead(pushButton2) == 0) {

    sprintf (msg, "Intruders Alarm");

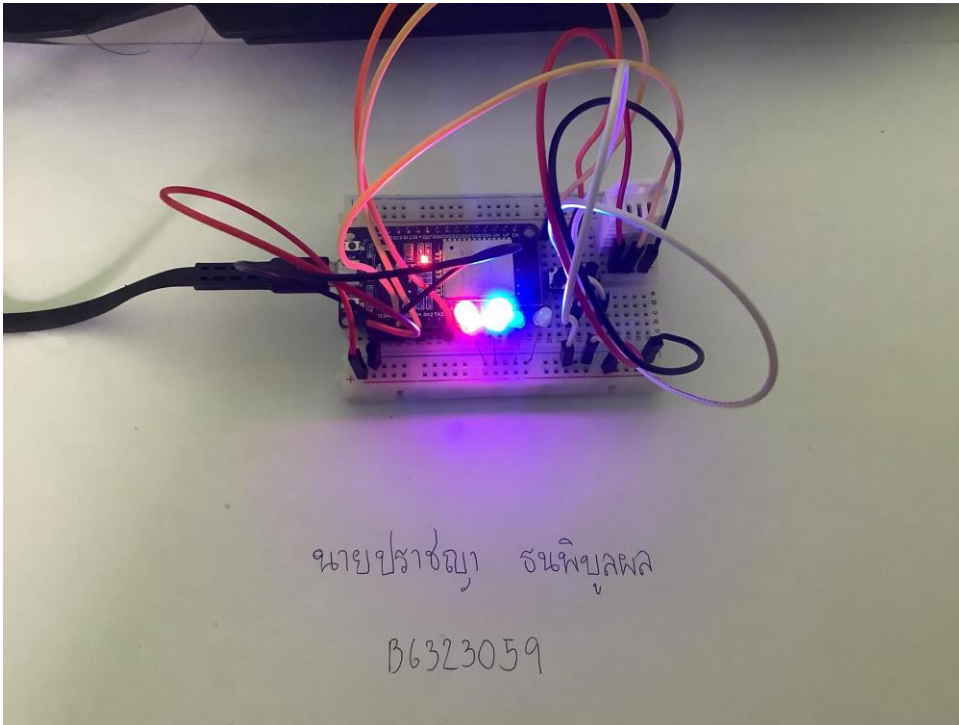
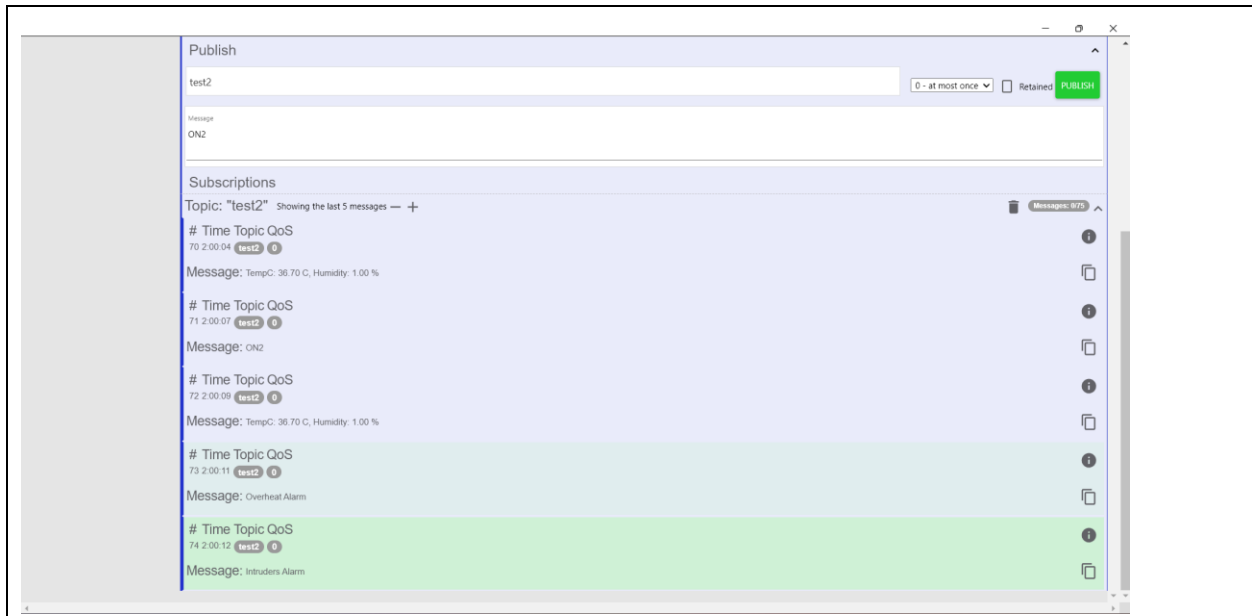
    Serial.println(msg);

    client.publish(topic1, msg);

    delay(500);

}

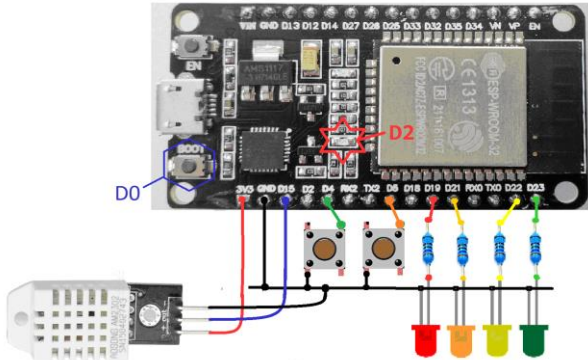

}
```





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

	
<pre>#define BLYNK_PRINT Serial #define BLYNK_TEMPLATE_ID "TMPL6vwqBIOjf" #define BLYNK_TEMPLATE_NAME "Test" #define BLYNK_AUTH_TOKEN "oX3Ov2l9WArgv_IURrUoFKI11QUS3B-B" #include <WiFi.h> #include <WiFiClient.h> #include <BlynkSimpleEsp32.h> #include "DHTesp.h" #define DHT22_Pin 15 #define sw1 18 #define sw2 19 char ssid[] = "It's bad day not a bad life"; char pass[] = "0641453596"; DHTesp dht; //boolean btnState = false;</pre>	

```
WidgetLED LED1(V2);

WidgetLED LED2(V3);

BlynkTimer timer;

void setup() {

  Serial.begin(115200);

  dht.setup(DHT22_Pin, DHTesp::DHT22); // Connect DHT sensor to GPIO 15

  pinMode(sw1, INPUT_PULLDOWN);

  pinMode(sw2, INPUT_PULLDOWN);

  Blynk.begin(BLYNK_AUTH_TOKEN, ssid, pass);

  timer.setInterval(1000L, myTimerEvent);

}

void myTimerEvent() {

  float humidity = dht.getHumidity();

  float temperature = dht.getTemperature();

  Blynk.virtualWrite(V0, temperature);

  Blynk.virtualWrite(V1, humidity);

  if (digitalRead(sw1)) LED1.on();

  else LED1.off();

  if (digitalRead(sw2)) LED2.on();

  else LED2.off();

  Serial.print(" Temp('C) >> "); Serial.print(temperature, 1);

  Serial.print(", Humidity(%) >> "); Serial.println(humidity, 1);

}

void loop()

{ Blynk.run();

  timer.run(); // running timer every 250ms}
```

