การใช้งาน ThingsBoard IoTs Platform เพื่อสร้างและจัดการระบบอัฉริยะ

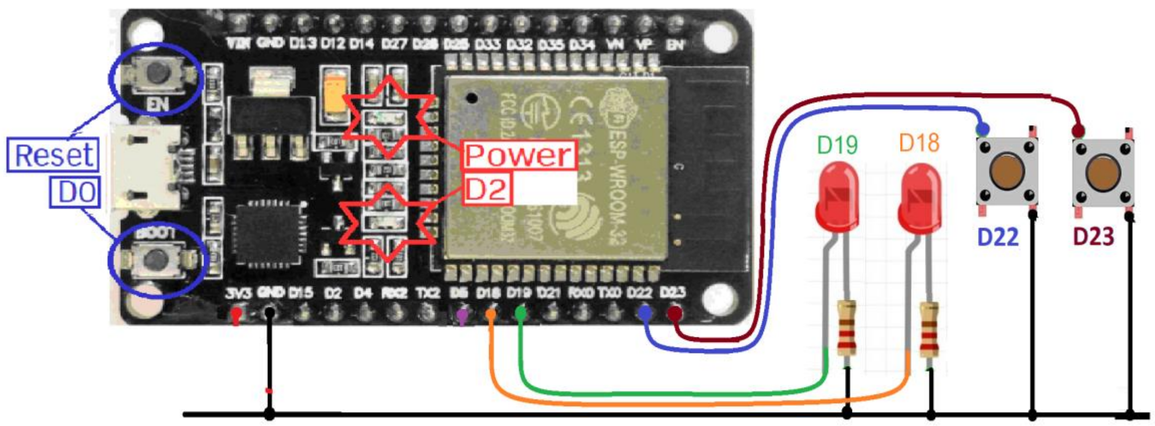
ThingsBoard IoTs Platform for smart system

ชื่อ-สกุล : นายธนพล กาศักดิ์

6/6 -- คำถามท้ายบทเพื่อทดสอบความเข้าใจ

Quiz_101 – กดติด กดดับ 2 ชุด

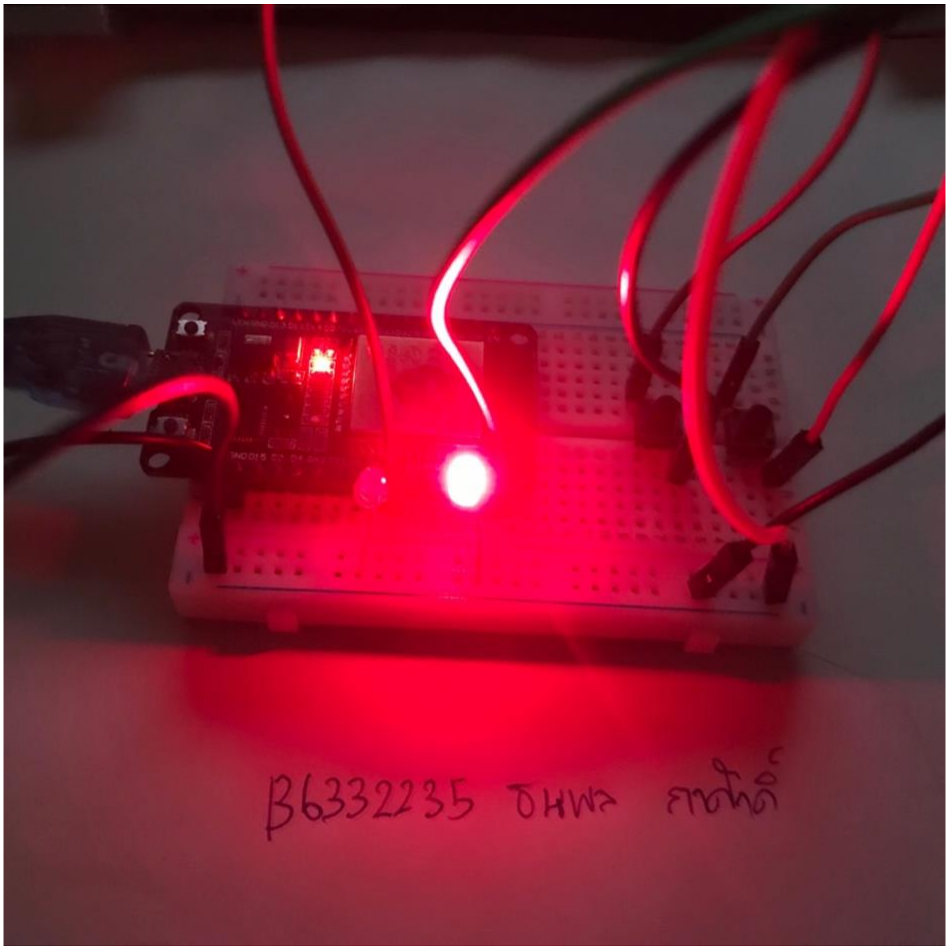
- หากต้องการให้ใช้ 1 สวิตซ์ ควบคุม 1 LED แบบกดติด-กดดับ จำนวน 2 วงจรจะต่อวงจรและเขียนโปรแกรมอย่างไร {SW-D22 -- LED-D19, SW-D23 -- LED-D18}



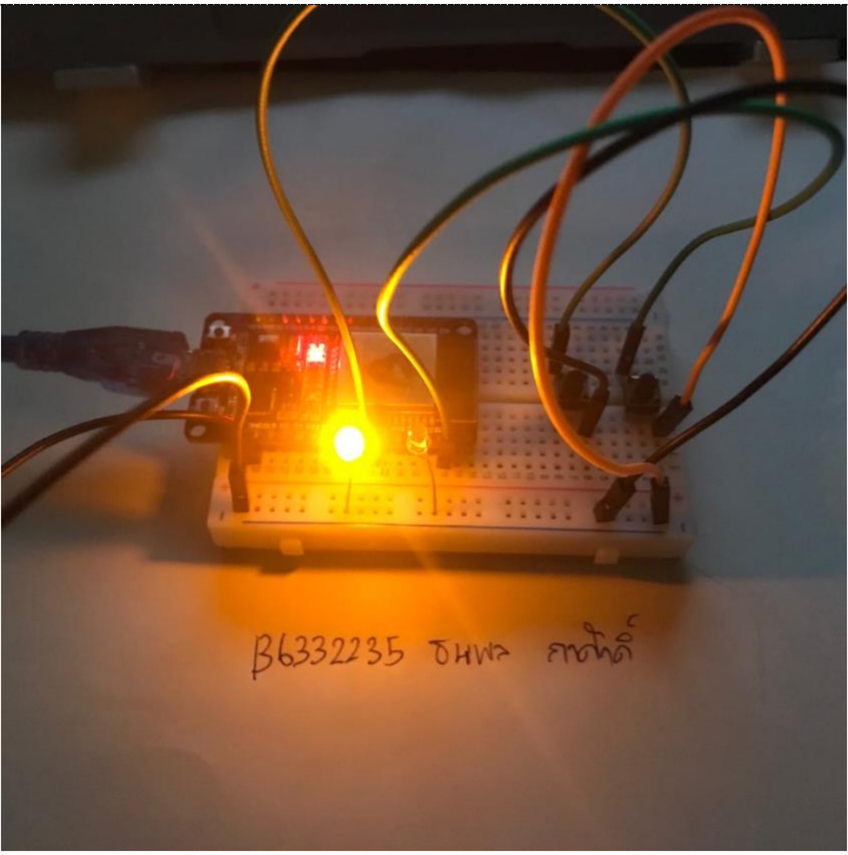
โปรแกรมที่ใช้ทดสอบ

```
#define pushButton1 23
#define pushButton2 19
#define LEDPin1 22
#define LEDPin2 18
int buttonState = 0;
void setup() {
  Serial.begin(115200);
  pinMode(pushButton1, INPUT_PULLUP);
  pinMode(pushButton2, INPUT_PULLUP);
  pinMode(LEDPin1, OUTPUT);
  pinMode(LEDPin2, OUTPUT);
}
void loop() {
  if (digitalRead(pushButton1) == LOW) {
    delay(20);
    buttonState = 1 - buttonState;
    digitalWrite(LEDPin1, buttonState);
    while (digitalRead(pushButton1) == LOW);
    delay(20);
  }else if (digitalRead(pushButton2) == LOW) {
    delay(20);
    buttonState = 1 - buttonState;
    digitalWrite(LEDPin2, buttonState);
    while (digitalRead(pushButton2) == LOW);
    delay(20);
  }
}
```

รูปการทดสอบ 1

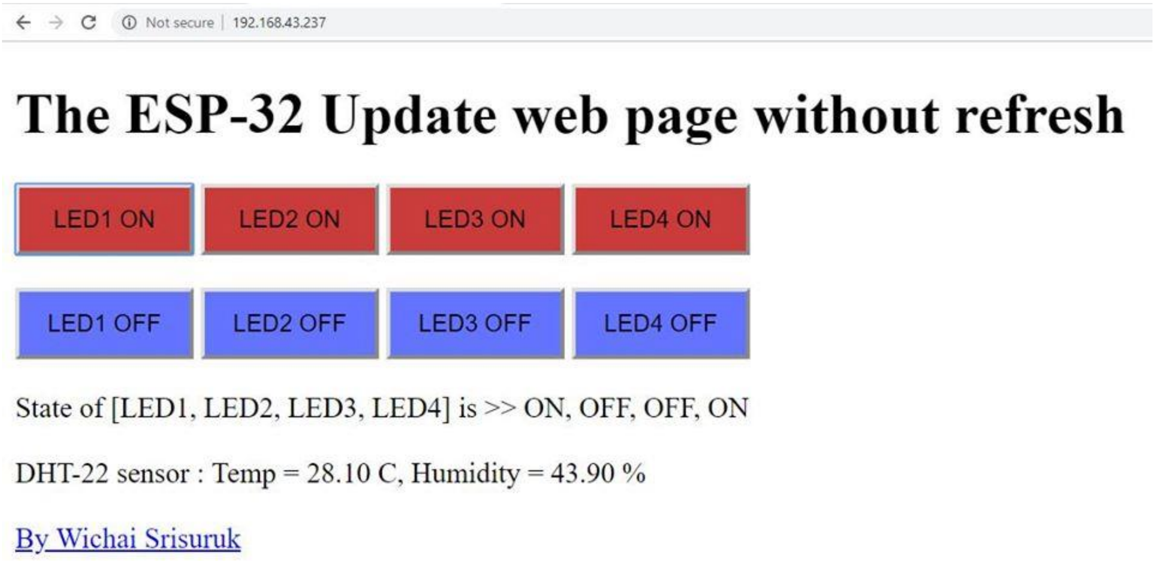


รูปการทดสอบ 2


B6332235 ธนพล ภาษโด์

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=IwAR3dIZ_gRgDWmREmnzuknLbMxV3pOHy4YIPuLEz8-ZzTOX2VhWxcH2QjLGk



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 >

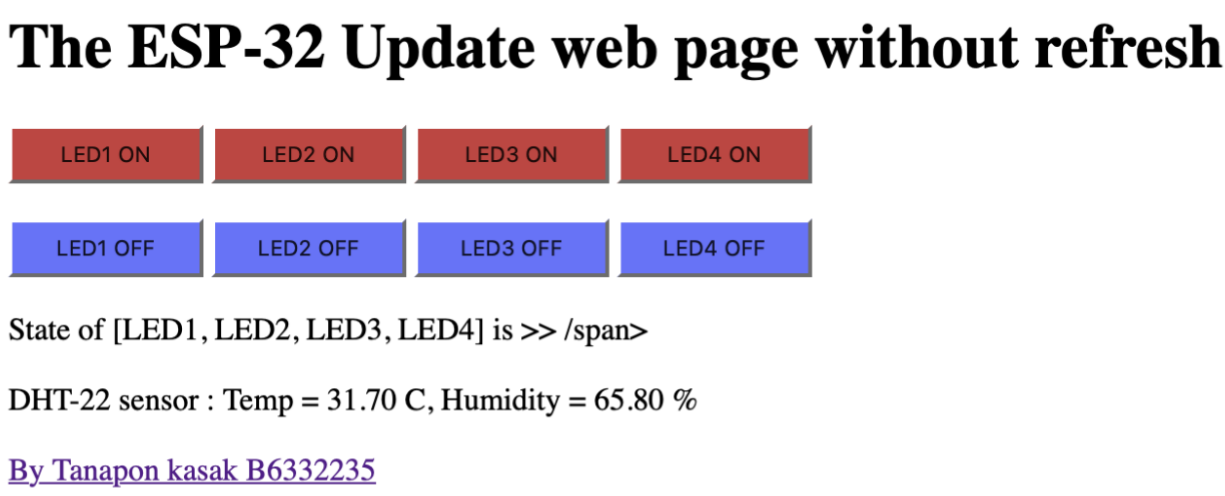
```
#include <WiFi.h>

#include <WiFiClient.h>

#include <WebServer.h>

#include "DHTesp.h"

#include "index.h" //Our HTML webpage contents with javascripts

#define DHT_Pin 4



#define testLED1 18

#define testLED2 19

#define testLED3 22

#define testLED4 23
```

```
//SSID and Password of your WiFi router
const char* ssid = "iPhoneOhm";
const char* password = "2444666668888888";
WebServer server(80); //Server on port 80
DHTesp dht;
String ledState1 = "OFF";
String ledState2 = "OFF";
String ledState3 = "OFF";
String ledState4 = "OFF";
//============================================================
// 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/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 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 loop(void) {

  server.handleClient(); //Handle client requests

}
```

รูปถ่ายหน้า Web Broswer



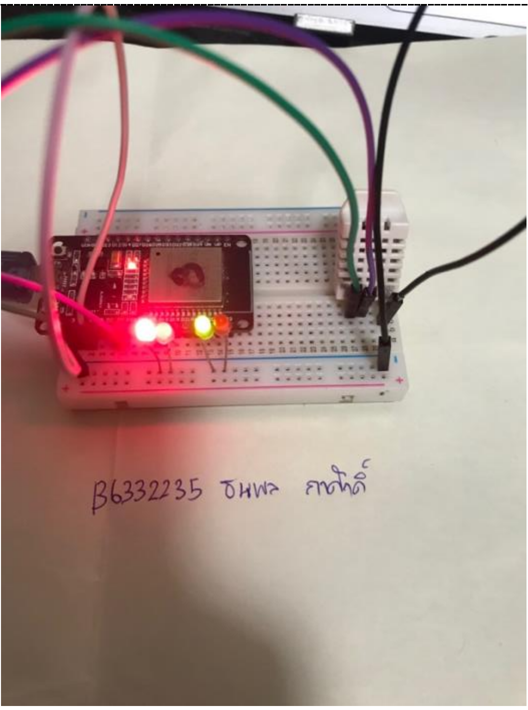
The ESP-32 Update web page without refresh

State of [LED1, LED2, LED3, LED4] is >> /span>

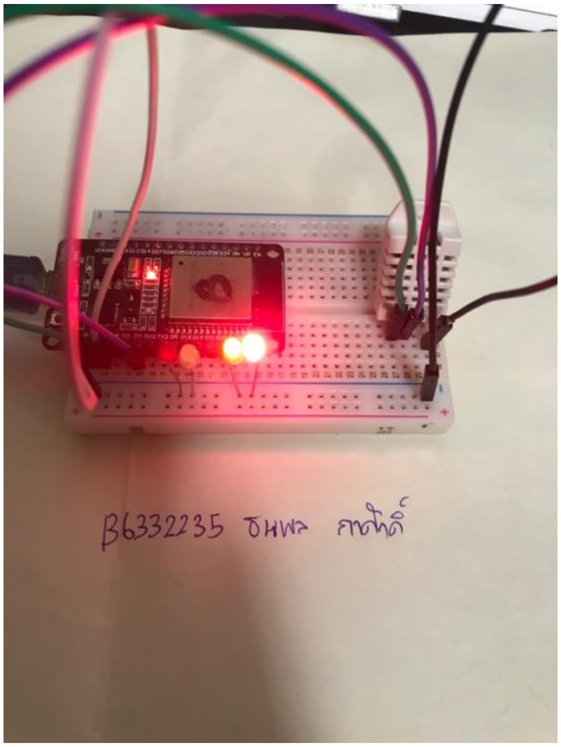
DHT-22 sensor : Temp = 31.70 C, Humidity = 65.80 %

By Tanapon kasak B6332235

รูปการทดสอบ 1

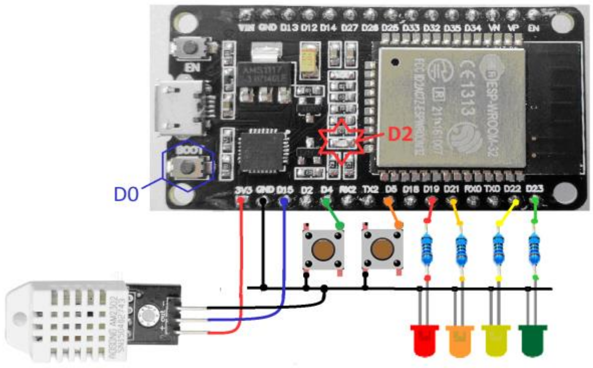


รูปการทดสอบ 2



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
- รับค่าสวิตซ์กำหนด SW1 แจ้ง Overheat Alarm, SW2 แจ้ง Intruders Alarm



โปรแกรมที่ใช้ทดสอบ

```
#include <WiFi.h>
#include <PubSubClient.h>
#include "DHTesp.h"
#define Pin_DHT22 15const char* ssid = "xxxx"; //Your Wifi
const char* password = "1234"; //Your Wifi password
const char* mqtt_server = "test.mosquitto.org";
const char* topic1 = "myHome1421";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) == LOW)
    { delay(100);
      snprintf (msg, 75, "Alert!!! The temperature is too high.");
      Serial.print("Publish message: ");
      Serial.println(msg);
      client.publish(topic1, msg);
      while (digitalRead(SW1) == LOW);
        delay(100);}
    if (digitalRead(SW2) == LOW)
```

```
{ delay(100);
 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) == LOW);
  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); }
```

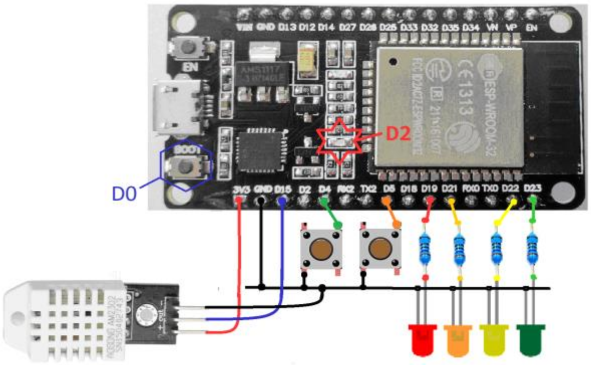
รูปหน้าจอ MQTT Lens

รูปการต่อวงจร – 1

รูปการต่อวงจร – 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
- รับค่าสวิตซ์กำหนด SW1 แจ้ง Overheat Alarm, SW2 แจ้ง Intruders Alarm ไปยัง LINE



Blynk

โปรแกรมที่ใช้ทดสอบ

```
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.toDateString() + ", " + 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');
}
```

Code arduno

```
#define BLYNK_TEMPLATE_ID "TMPL9_tNVE9p"
#define BLYNK_DEVICE_NAME "new"
#define LINE_TOKEN "ZScxjc1PFsyJQRMxxxxxxx"
#define WebHooksKey "caqQ1V4_W1TzQltMcpkid7"
#define WebHooksEvent1 "Temp.now"
#define My_NAME "My_RooM"#define BLYNK_PRINT Serial
#include <WiFi.h>
#include <WiFiClient.h>
#include <BlynkSimpleEsp32.h>
#include <TridentTD_LineNotify.h>
#include <HTTPClient.h>
#include "DHTesp.h"
#include "HandOnSheets.h"WidgetLED blynk_LED1(V4);
WidgetLED blynk_LED2(V5);
BlynkTimer timer;
DHTesp dht;
long now = millis();
long lastMeasure = 0;char auth[] = "Blynk_auth_key"; //form blynk dashboard
char ssid[] = "Your_WiFi";
char pass[] = "Your_Password";const int btnPin1 = 4;
const int btnPin2 = 5;
const int ledPin1 = 19;
const int ledPin2 = 21;
const int ledPin3 = 22;
const int ledPin4 = 23;
const int pinDHT_22 = 15;boolean btn1State, btn2State;
float temperature = dht.getTemperature();
float humidity = dht.getHumidity();void DHTTimerEvent()
```

```
{ temperature = dht.getTemperature();
  humidity = dht.getHumidity();
  Blynk.virtualWrite(V6, temperature);
  Blynk.virtualWrite(V7, humidity);
}BLYNK_WRITE(V0)
{ int ledState1 = param.asInt();
  digitalWrite(ledPin1, ledState1);
}
BLYNK_WRITE(V1)
{ int ledState2 = param.asInt();
  digitalWrite(ledPin2, ledState2);
}
BLYNK_WRITE(V2)
{ int ledState3 = param.asInt();
  digitalWrite(ledPin3, ledState3);
}
BLYNK_WRITE(V3)
{ int ledState4 = param.asInt();
  digitalWrite(ledPin4, ledState4);
}void SWTimerEvent()
{ if (digitalRead(btnPin1) == LOW)
  { btn1State = LOW;
    blynk_LED1.on();
  } else {
    btn1State = HIGH;
    blynk_LED1.off();
  }

  if (digitalRead(btnPin2) == LOW)
  { btn2State = LOW;
    blynk_LED2.on();
  } else {
    btn2State = HIGH;
    blynk_LED2.off();
  }
}void setup()
{ Serial.begin(115200);
```

```
  Blynk.begin(auth, ssid, pass);
  WiFi.begin(ssid, pass);
  LINE.setToken(LINE_TOKEN);
  pinMode(btnPin1, INPUT_PULLUP);
  pinMode(btnPin2, INPUT_PULLUP);
  pinMode(ledPin1, OUTPUT); pinMode(ledPin2, OUTPUT);
  pinMode(ledPin3, OUTPUT); pinMode(ledPin4, OUTPUT);
  dht.setup(pinDHT_22, DHTesp::DHT22);
  timer.setInterval(250L, SWTimerEvent);
  timer.setInterval(1000L, DHTTimerEvent);
  client.setInsecure();
}
void loop()
{ Blynk.run();
  timer.run();
  now = millis();
  if (now - lastMeasure > 10 * 1000) {
    Serial.print(" Temp('C) >> "); Serial.print(temperature, 1);
    Serial.print(", Humidity(%) >> "); Serial.println(humidity, 1);
    lastMeasure = now;
    sendData(temperature, humidity);
  }
  if (temperature > 36) {
    LINE.notify("อุณหภูมิเกินกำหนด");
    LINE.notify("อุณหภูมิ " + String(temperature) + " °C");
    delay(60 * 1000);
  }

}
```

.h

```
String t;
const char* host = "script.google.com";
const int httpsPort = 443;
String GAS_ID = "AKfycbx76UkbodS19ec6d_dvMgyQtn7_SZjwAQO_D-PgmdmkoSdrRPKysAci04juSPboK4PW";
String GAS_Sheet = "Sensor_Data";
```

```
WiFiClientSecure client;void sendData(float SValue1, float SValue2) {
 Serial.println("==========");
 Serial.print("connecting to "); Serial.println(host);
 //---- Connect to Google host
 if (!client.connect(host, httpsPort)) {
  Serial.println("connection failed");
  return;
 }String url;
 url += "/macros/s/"+ GAS_ID + "/exec?";
 url += "id=" + String(GAS_Sheet);
 url += "&temp=" + String(SValue1,2);
 url += "&humi=" + String(SValue2,2);
 Serial.print("requesting URL: "); Serial.println(url);client.print(String("GET ") + url + " HTTP/1.1\r\n" +
 "Host: " + host + "\r\n" +
 "User-Agent: BuildFailureDetectorESP8266\r\n" +
 "Connection: close\r\n\r\n");
 Serial.println("request sent");
 //---- Wait Echo
 while (client.connected()) {
  String line = client.readStringUntil('\n');
  if (line == "\r") {
    Serial.println("headers received");
    break;
  }
 }
 String line = client.readStringUntil('\n');
 if (line.startsWith("{\"state\":\"success\"")) {
   Serial.println("ESP-32/Arduino CI successfull!");
 } else {
   Serial.println("ESP-32/Arduino CI has failed");
 }Serial.print("reply was : ");
 Serial.println(line);
 Serial.println("closing connection");
 Serial.println("==========");
 Serial.println();

}
```

รูปหน้าจอ Blynk
รูปหน้าจอ LINE
รูปการต่อวงจร – 1
รูปการต่อวงจร – 2