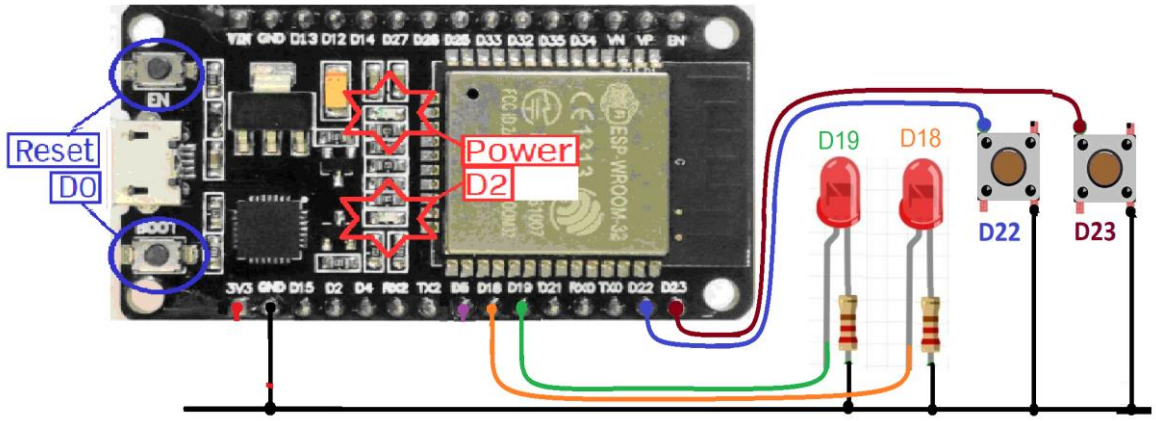
การควบคุมเครื่องจักรอัจฉริยะโดยใช้การสื่อสารระหว่างเครื่องจักรกับเครื่องจักร
M2M - Intelligence Machine Control

ชื่อ-สกุล : นายอติชาติ ภู่นิเทศ

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}

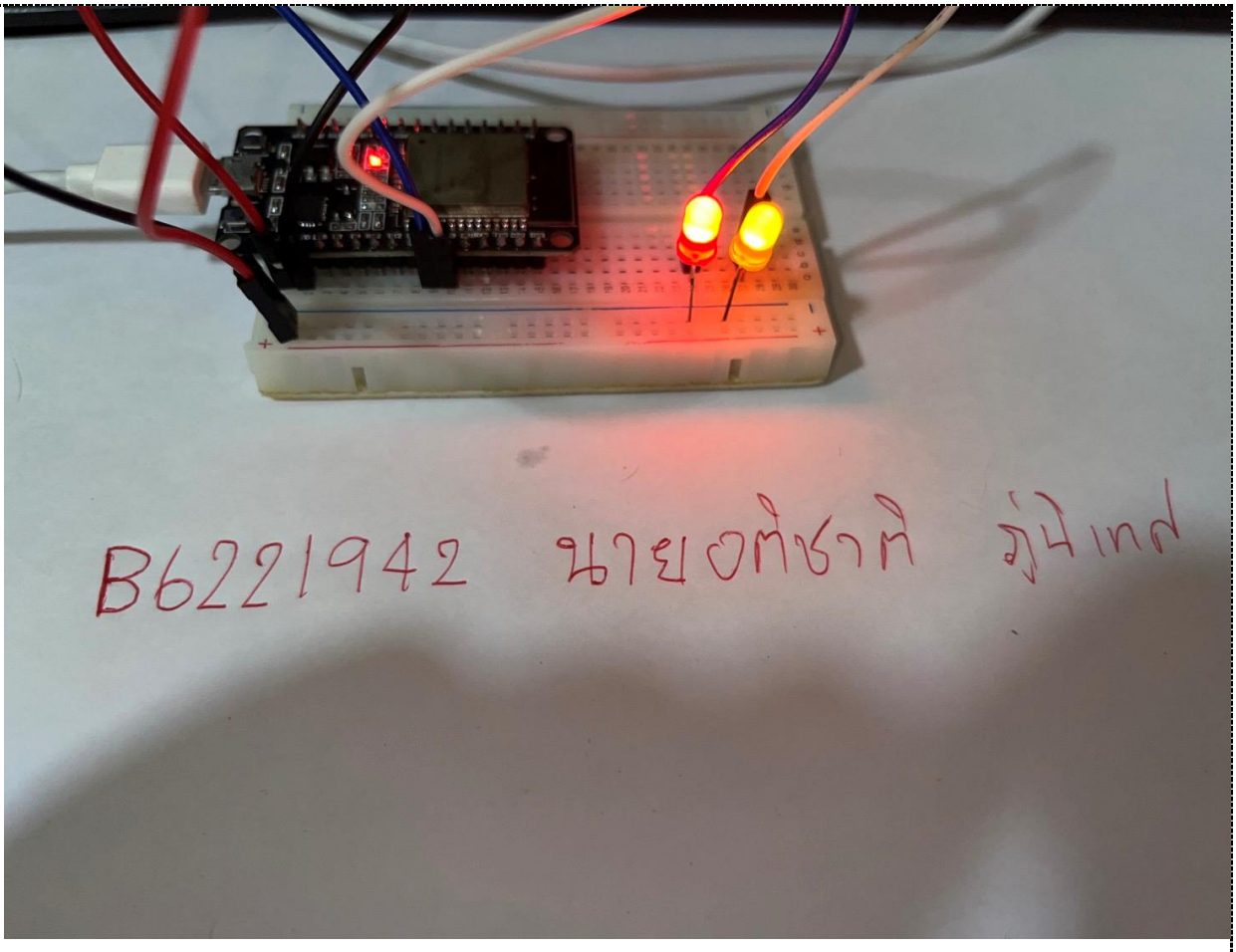


โปรแกรมที่ใช้ทดสอบ

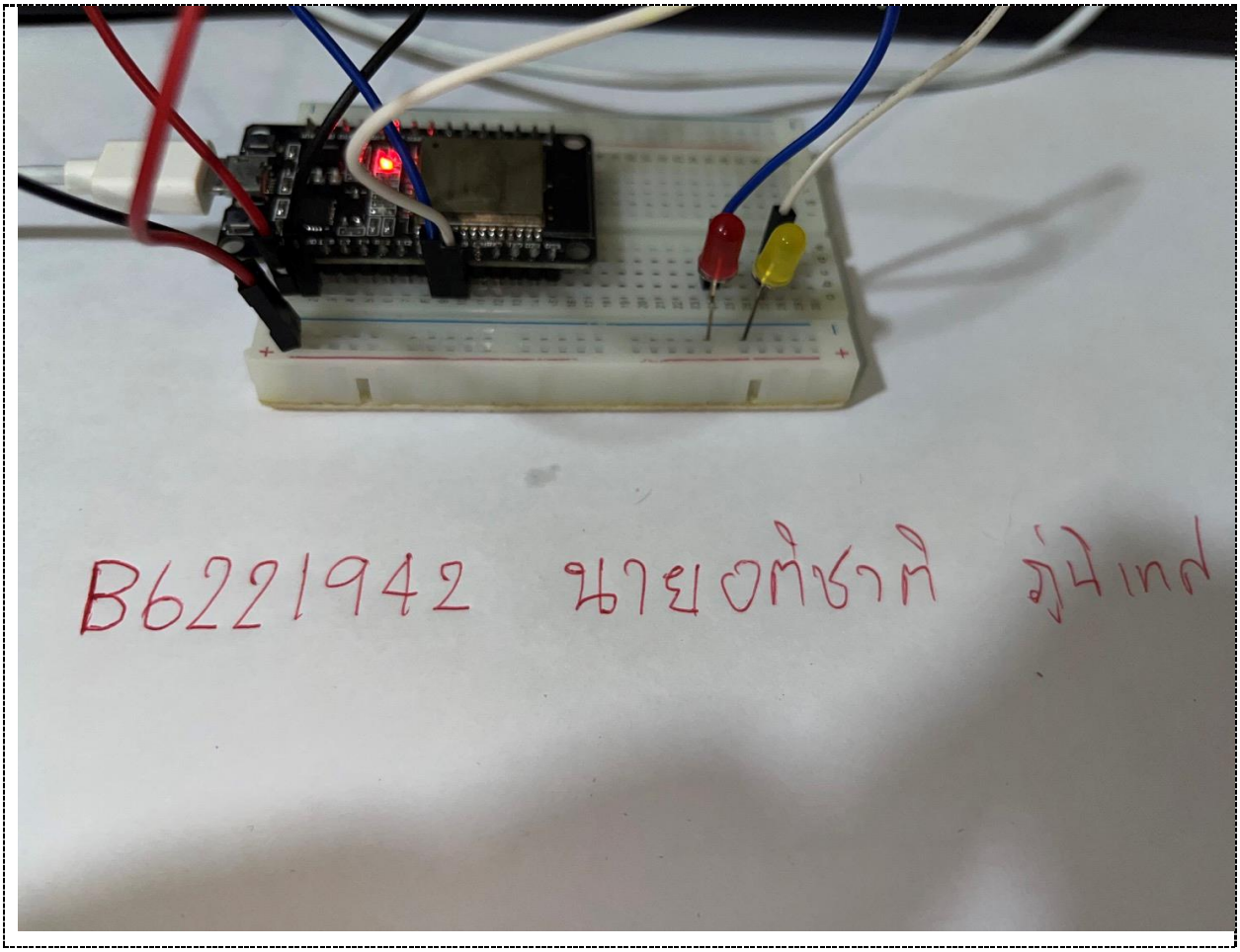
```
#define LED_1 18
#define LED_2 19
#define BUTTON_1 22
#define BUTTON_2 23
int buttonState1 = 0;
int buttonState2 = 0;
void setup()
{
pinMode(BUTTON_1, INPUT_PULLUP);
pinMode(BUTTON_2, INPUT_PULLUP);
pinMode(LED_1, OUTPUT);
pinMode(LED_2, OUTPUT);
Serial.begin(115200);
delay(10);
```

```
}
void loop() {
if (digitalRead(BUTTON_1) == LOW) {
delay(20);
buttonState1 = 1 - buttonState1;
digitalWrite(LED_1, buttonState1);
while (digitalRead(BUTTON_1) == LOW);
delay(20);
}
if (digitalRead(BUTTON_2) == LOW) {
delay(20);
buttonState2 = 1 - buttonState2;
digitalWrite(LED_2, buttonState2);
while (digitalRead(BUTTON_2) == LOW);
delay(20);
}
}
```

รูปการทดสอบ 1

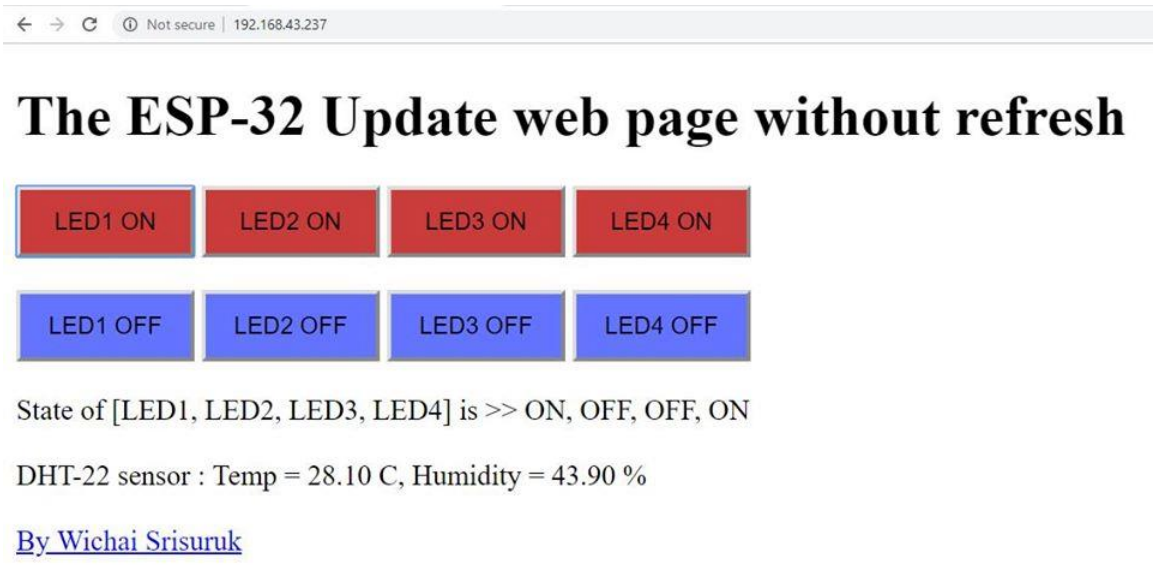


รูปการทดสอบ 2


B6221942 นายอัชชา ภูนิเทศ

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



โปรแกรมที่ใช้ทดสอบ

ไฟล์ **Q3.ino**

```
#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 LED1 18
#define LED2 19
#define LED3 22
#define LED4 23
//SSID and Password of your WiFi router
const char* ssid = "LANTANIDEs-2.4G";
const char* password = "0887040892";
```

```
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/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(LED1, HIGH); ledState1 = "ON"; } //Feedback parameter
if (t_state == "10") { digitalWrite(LED1, LOW); ledState1 = "OFF";} //Feedback parameter
if (t_state == "21") { digitalWrite(LED2, HIGH); ledState2 = "ON"; } //Feedback parameter
if (t_state == "20") { digitalWrite(LED2, LOW); ledState2 = "OFF";} //Feedback parameter
if (t_state == "31") { digitalWrite(LED3, HIGH); ledState3 = "ON"; } //Feedback parameter
if (t_state == "30") { digitalWrite(LED3, LOW); ledState3 = "OFF";} //Feedback parameter
if (t_state == "41") { digitalWrite(LED4, HIGH); ledState4 = "ON"; } //Feedback parameter
if (t_state == "40") { digitalWrite(LED4, LOW); ledState4 = "OFF";} //Feedback parameter
```

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

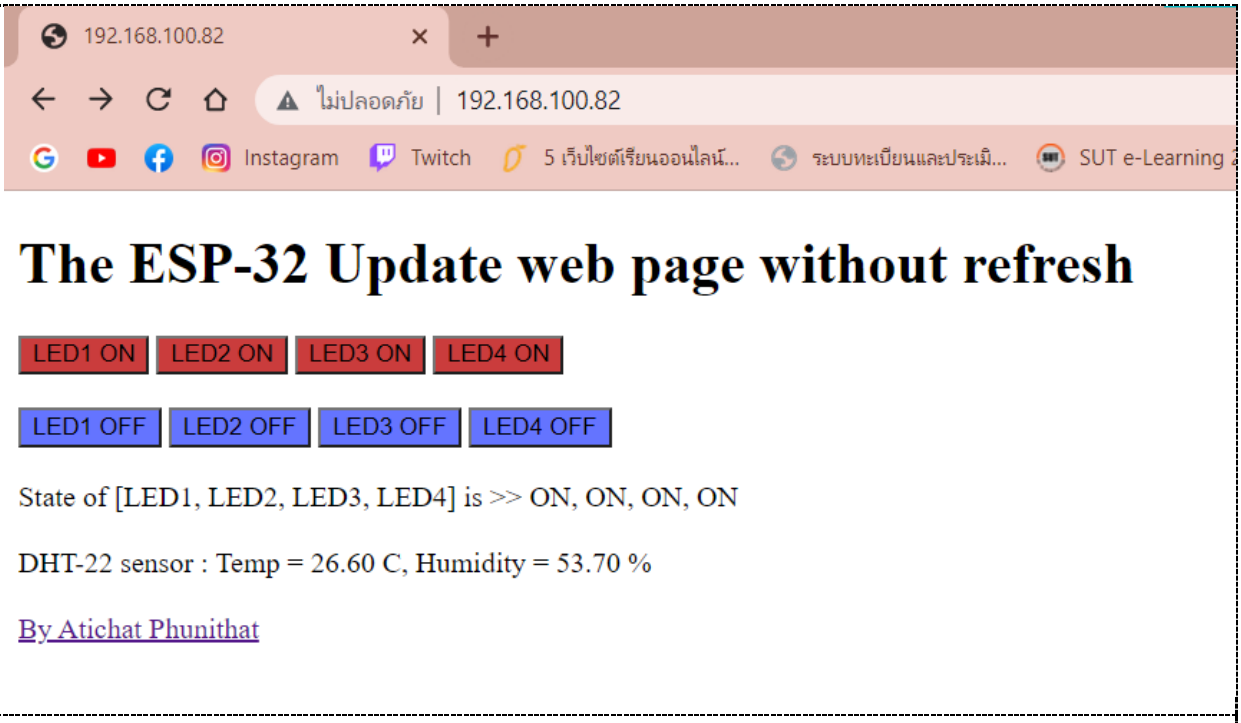
ไฟล์ **index.h**

```
const char MAIN_page[] PROGMEM = R"=====(
<!DOCTYPE html>
<html>
<body>
```

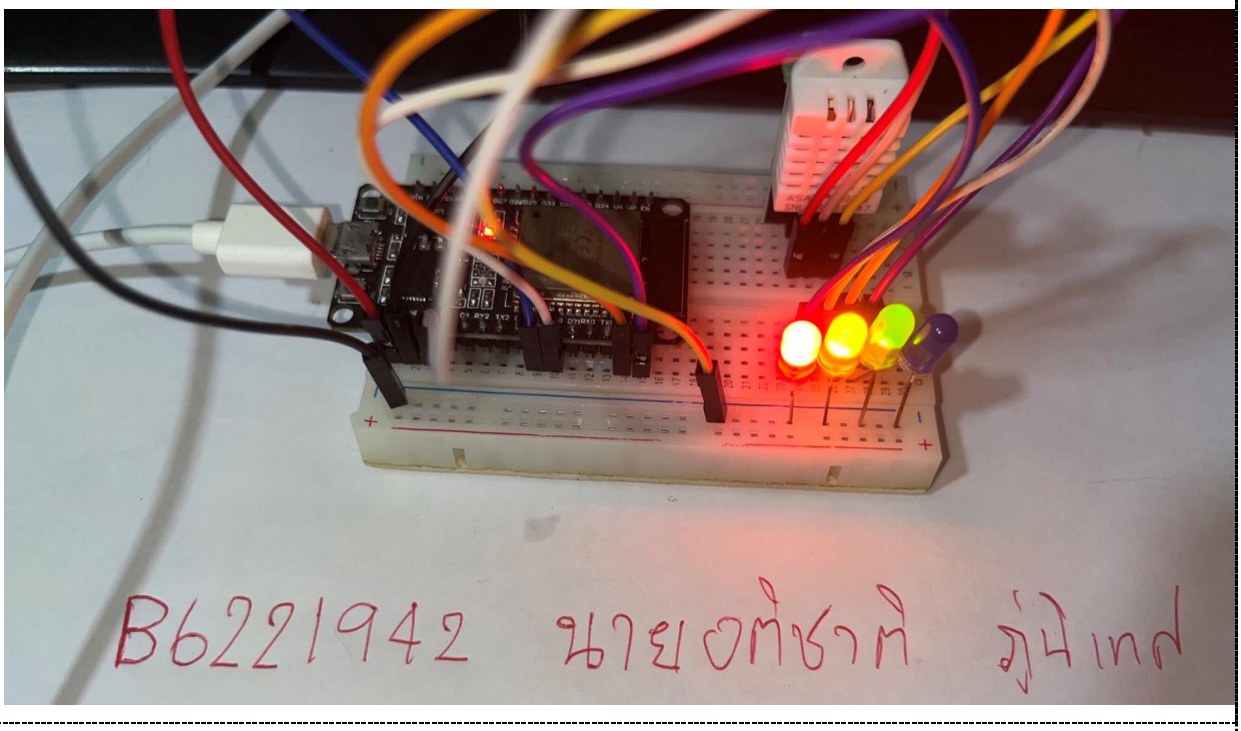
```
<div id="demo">
<h1>The ESP-32 Update web page without refresh</h1>
<button type="button" onclick="sendData(11)" style="background: rgb(202, 60, 60);">LED1
ON</button>
<button type="button" onclick="sendData(21)" style="background: rgb(202, 60, 60);">LED2
ON</button>
<button type="button" onclick="sendData(31)" style="background: rgb(202, 60, 60);">LED3
ON</button>
<button type="button" onclick="sendData(41)" style="background: rgb(202, 60, 60);">LED4
ON</button><br><br>
<button type="button" onclick="sendData(10)" style="background:
rgb(100,116,255);">LED1
OFF</button>
<button type="button" onclick="sendData(20)" style="background:
rgb(100,116,255);">LED2
OFF</button>
<button type="button" onclick="sendData(30)" style="background:
rgb(100,116,255);">LED3
OFF</button>
<button type="button" onclick="sendData(40)" style="background:
rgb(100,116,255);">LED4
OFF</button><br><br>
State of [LED1, LED2, LED3, LED4] is >> <span id="LEDState">NA</span><br>
</div>
<div>
<br>DHT-22 sensor : <span id="ADCValue">0</span><br>
</div>
<script>
function sendData(led) {
var xhttp = new XMLHttpRequest();
xhttp.onreadystatechange = function() {
if (this.readyState == 4 && this.status == 200) {
document.getElementById("LEDState").innerHTML =
this.responseText;
```

```
}
};
xhttp.open("GET", "setLED?LEDstate="+led, true);
xhttp.send();
}
setInterval(function() {
// Call a function repetatively with 2 Second interval
getData();
}, 2000); //2000mSeconds update rate
function getData() {
var xhttp = new XMLHttpRequest();
xhttp.onreadystatechange = function() {
if (this.readyState == 4 && this.status == 200) {
document.getElementById("ADCValue").innerHTML =
this.responseText;
}
};
xhttp.open("GET", "readADC", true);
xhttp.send();
}
</script>
<br><a href="https://www.facebook.com/P.Atichat.14">By Atichat Phunithat</a>
</body>
</html>
)=====";
```

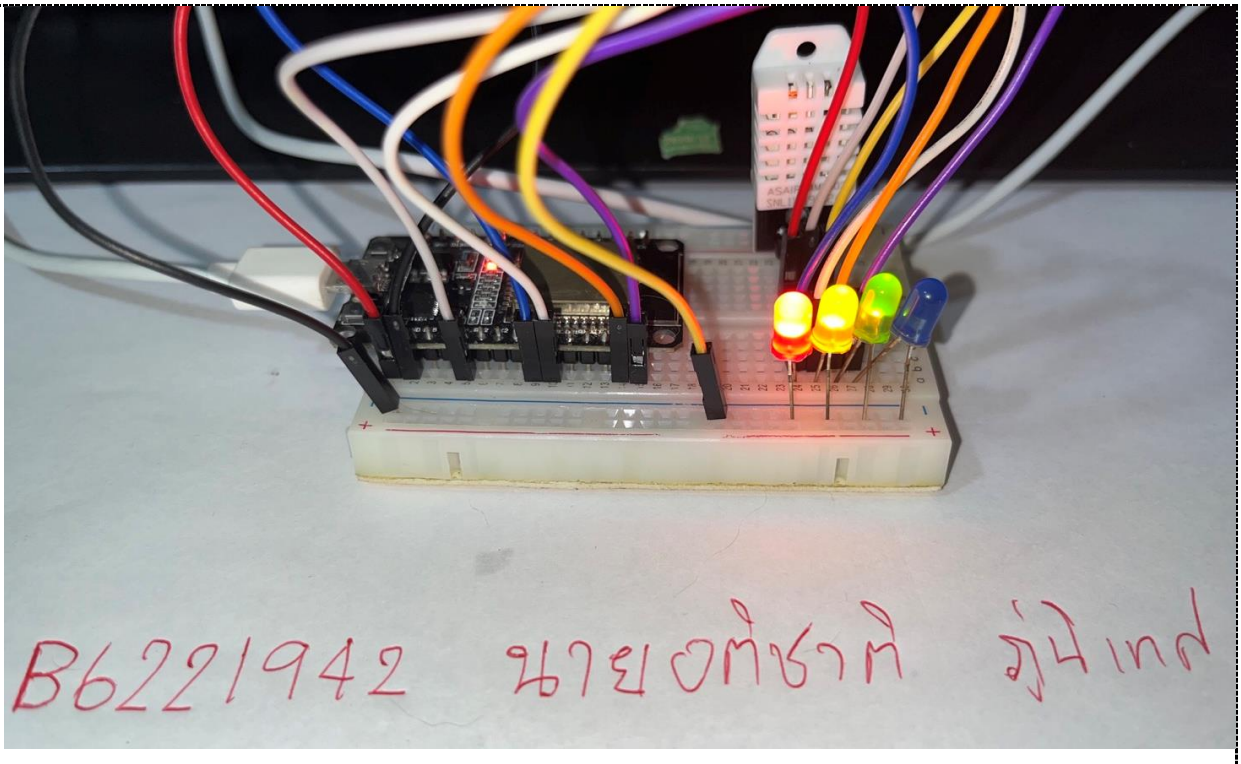
รูปถ่ายหน้า Web Broswer



รูปการทดสอบ 1

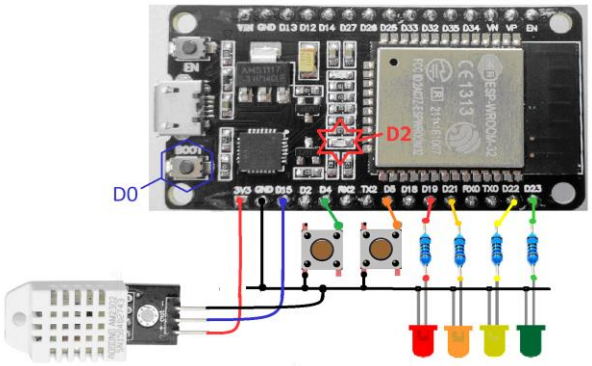


รูปการทดสอบ 2


B6221942 นายอัชชา ภู่ม

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

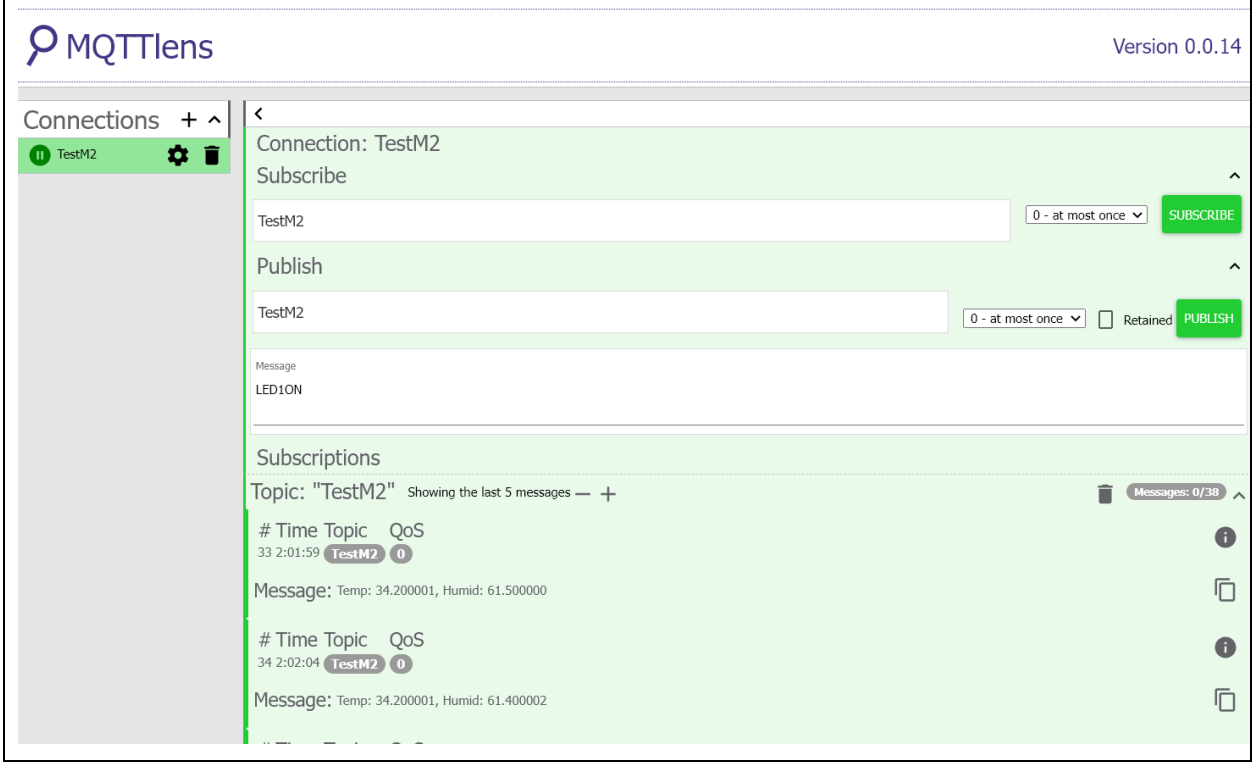
```
#include <WiFi.h>
#include <PubSubClient.h>
#include "DHTesp.h"
#define DHT22_Pin 15
DHTesp dht;
const char* ssid = "LANTANIDEs-2.4G";
const char* password = "0887040892";
const char* mqtt_server = "broker.mqttdashboard.com";
const char* topic1 = "TestM2";
WiFiClient espClient;
PubSubClient client(espClient);
long lastMsg = 0;
char msg[50];
int LED1 = 18;
int LED2 = 19;
int LED3 = 22;
int LED4 = 23;
int Button1 = 4;
```

```
int Button2 = 5;
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 ATP"); // ... 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 [");
```

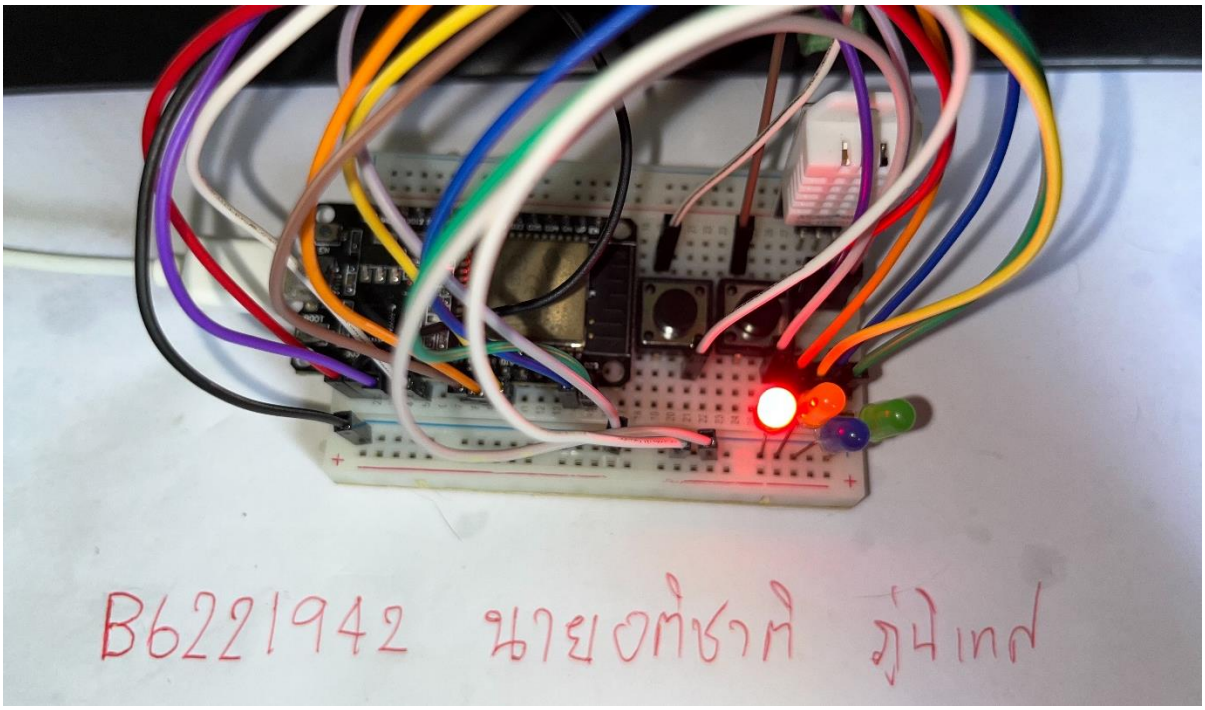
```
Serial.print(topic1);
Serial.print("] ");
for (int i = 0; i < length; i++){
Serial.print((char)payload[i]);
myPayLoad[i] = payload[i];
myPayLoad[i + 1] = '\0'; // End of String
}
Serial.print("\n ---> "); Serial.println(myPayLoad);
myPayLoad[8] = '\0'; // String lessthan 4 Charector
if ((String)myPayLoad == "LED1ON") digitalWrite(LED1, HIGH);
if ((String)myPayLoad == "LED1OFF") digitalWrite(LED1, LOW);
if ((String)myPayLoad == "LED2ON") digitalWrite(LED2, HIGH);
if ((String)myPayLoad == "LED2OFF") digitalWrite(LED2, LOW);
if ((String)myPayLoad == "LED3ON") digitalWrite(LED3, HIGH);
if ((String)myPayLoad == "LED3OFF") digitalWrite(LED3, LOW);
if ((String)myPayLoad == "LED4ON") digitalWrite(LED4, HIGH);
if ((String)myPayLoad == "LED4OFF") digitalWrite(LED4, LOW);
}
void setup(){
Serial.begin(115200);
pinMode(LED1, OUTPUT);
pinMode(LED2, OUTPUT);
pinMode(LED3, OUTPUT);
pinMode(LED4, OUTPUT);
pinMode(Button1, INPUT_PULLUP);
pinMode(Button2, INPUT_PULLUP);
dht.setup(DHT22_Pin, DHTesp::DHT22);
setup_wifi();
client.setServer(mqtt_server, 1883);
client.setCallback(callback);
}
void loop(){
float temperature = dht.getTemperature();
```

```
float humidity = dht.getHumidity();
if (!client.connected()) reconnect();
client.loop();
if(digitalRead(Button1) == 0){
client.publish(topic1, "Overheat");
delay(1000);
}
if(digitalRead(Button2) == 0){
client.publish(topic1, "Intruders");
delay(1000);
}
sprintf (msg, "Temp: %f, Humid: %f", temperature,humidity);
Serial.print("Publish message: ");
Serial.println(msg);
client.publish(topic1, msg);
delay(5000);
}
```

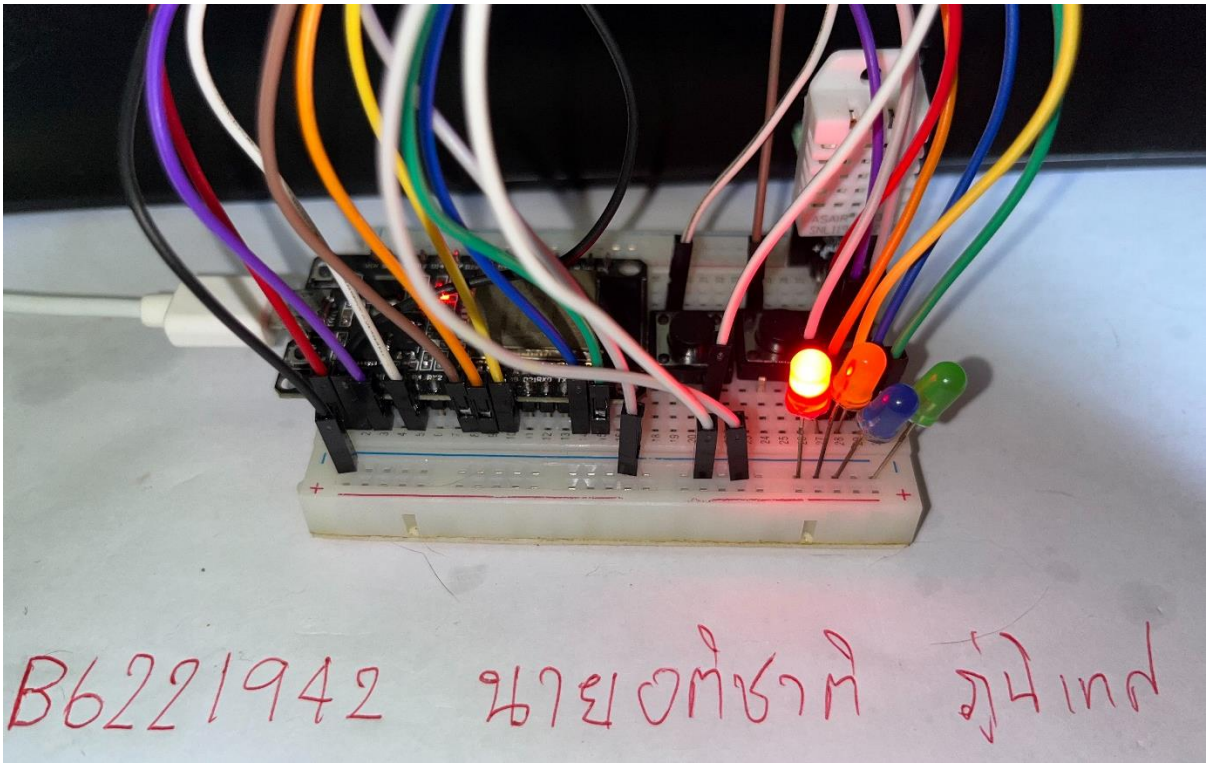
รูปหน้าจอ MQTT Lens



รูปการต่อวงจร – 1

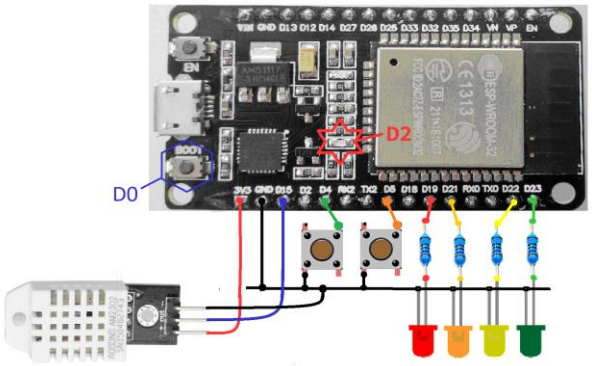


รูปการต่อวงจร – 2



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



โปรแกรมที่ใช้ทดสอบ

```
#include <ssl_client.h>
#include <TridentTD_LineNotify.h>
#include <WiFi.h>
#include <WiFiClientSecure.h>
#include "DHTesp.h"
#include <BlynkSimpleEsp32.h>
#define BLYNK_PRINT Serial
#define SSID "LANTANIDEs-2.4G"
#define PASSWORD "0887040892"
#define LINE_TOKEN "deYsdW8uCn5arxMSkY9pjzttU0rg0IPnvgYwslV1CZn"
#define DHT22_Pin 15


char auth[] = "dtj70oSSPd_42bJBSwfbNJ9-pw2bek3E";
int bnt1 = 4;
int bnt2 = 5;
String GAS_ID = "AKfycbypmjbOTQ9NYdRH0qWIJMZuiNQJQdAP0VFY5lSE2txJaiQaNXZen0Z-6EdbZL4RyFKJ";
String GAS_Sheet = "Sensor_Data";
```

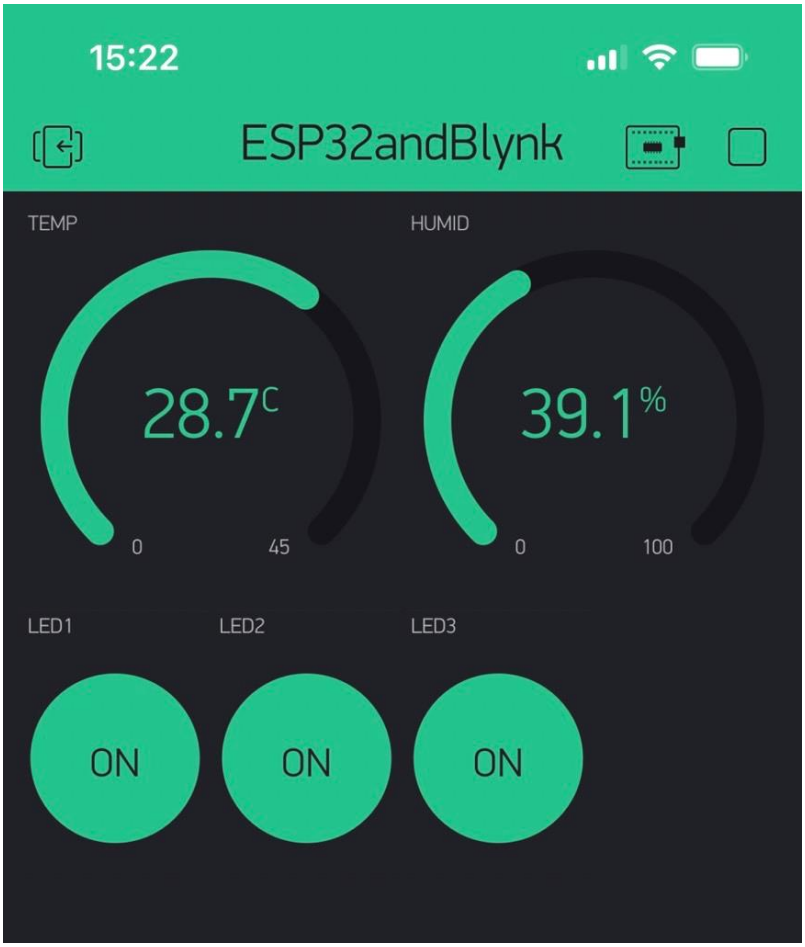
```
String t;
const char* host = "script.google.com";
const int httpsPort = 443;
long now = millis();
long lastMeasure = 0;
float temperature = 00.00, humidity = 00.00;

BlynkTimer timer;
WiFiClientSecure client;
DHTesp dht;

void myTimerEvent()
{ temperature = dht.getTemperature();
  humidity = dht.getHumidity();
  Blynk.virtualWrite(V10, temperature);
  Blynk.virtualWrite(V11, humidity);
  Serial.print(" Temp('C) >> "); Serial.print(temperature, 1);
  Serial.print(", Humidity(%) >> "); Serial.println(humidity, 1);
}

void setup() {
  Serial.begin(115200); Serial.println();
  Serial.println(LINE.getVersion());

  client.setInsecure();
  // กำหนด Line Token
  LINE.setToken(LINE_TOKEN);
  dht.setup(DHT22_Pin, DHTesp::DHT22);
  pinMode(bnt1, INPUT_PULLUP);
  pinMode(bnt2, INPUT_PULLUP);
  Blynk.begin(auth, SSID, PASSWORD);
  timer.setInterval(1000L, myTimerEvent);
}
```

```
void loop() {
  Blynk.run();
  timer.run();

  if(digitalRead(bnt1) == LOW){
    LINE.notify("Overheat Alarm");
    Serial.print("BNT1 ON");
  }
  if(digitalRead(bnt2) == LOW){
    LINE.notify("Intruders Alarm");
    Serial.print("BNT2 ON");
  }
  delay(5000);
  now = millis();
  if (now - lastMeasure > 5000) {
    lastMeasure = now;
    float humidity = dht.getHumidity();
    float temperature = dht.getTemperature();
    Serial.print("Temperature: " + String(temperature) + "C/");
    Serial.print("Humidity: " + String(humidity) + "%");

    sendData(temperature, humidity);

    if (temperature > 28.00) {
      String msg = "อุณหภูมิ: " + String(temperature) + " องศา";
      LINE.notify(msg);
    }
  }
}

void sendData(float SValue1, float SValue2) {
  Serial.println("==========");
```

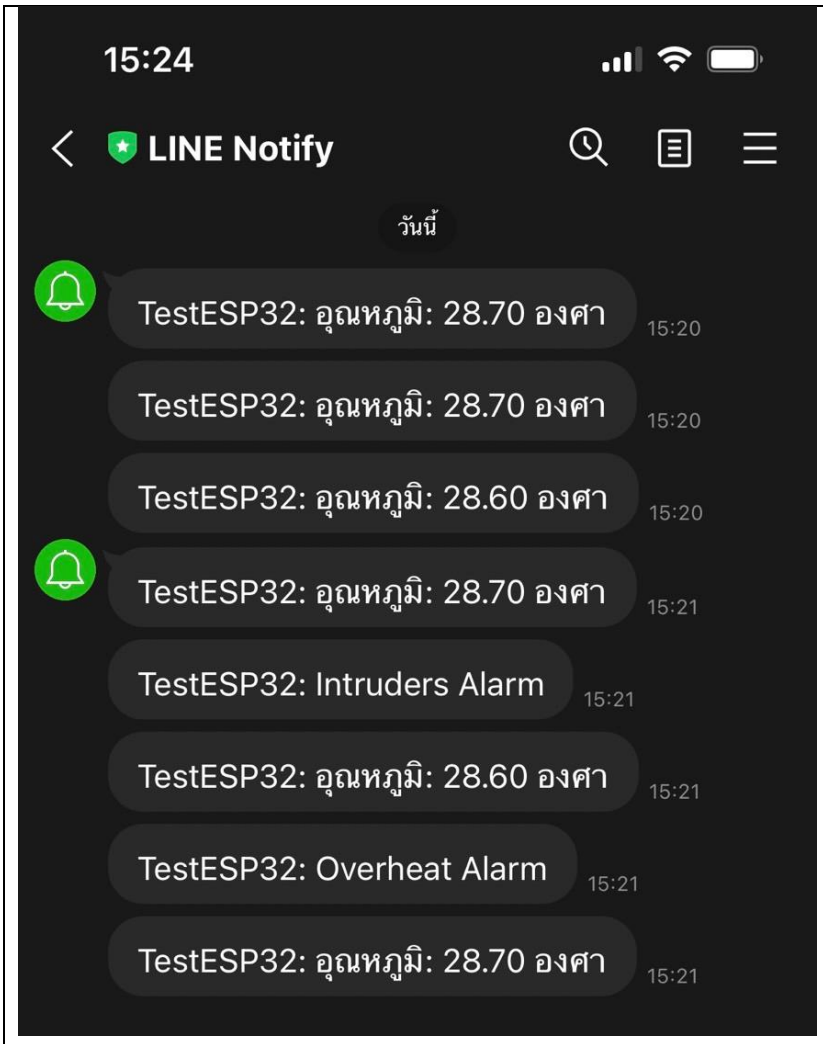
```
Serial.print("connecting to "); Serial.println(host);
//---- Connect to Google host
if (!client.connect(host, httpsPort)) {
Serial.println("connection failed");
return;
}

//---- Post Data
String url;
url += "/macros/s/" + GAS_ID + "/exec?";
url += "id=" + String(GAS_Sheet);
url += "&Sensor1=" + String(SValue1, 2);
url += "&Sensor2=" + 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();
}
```

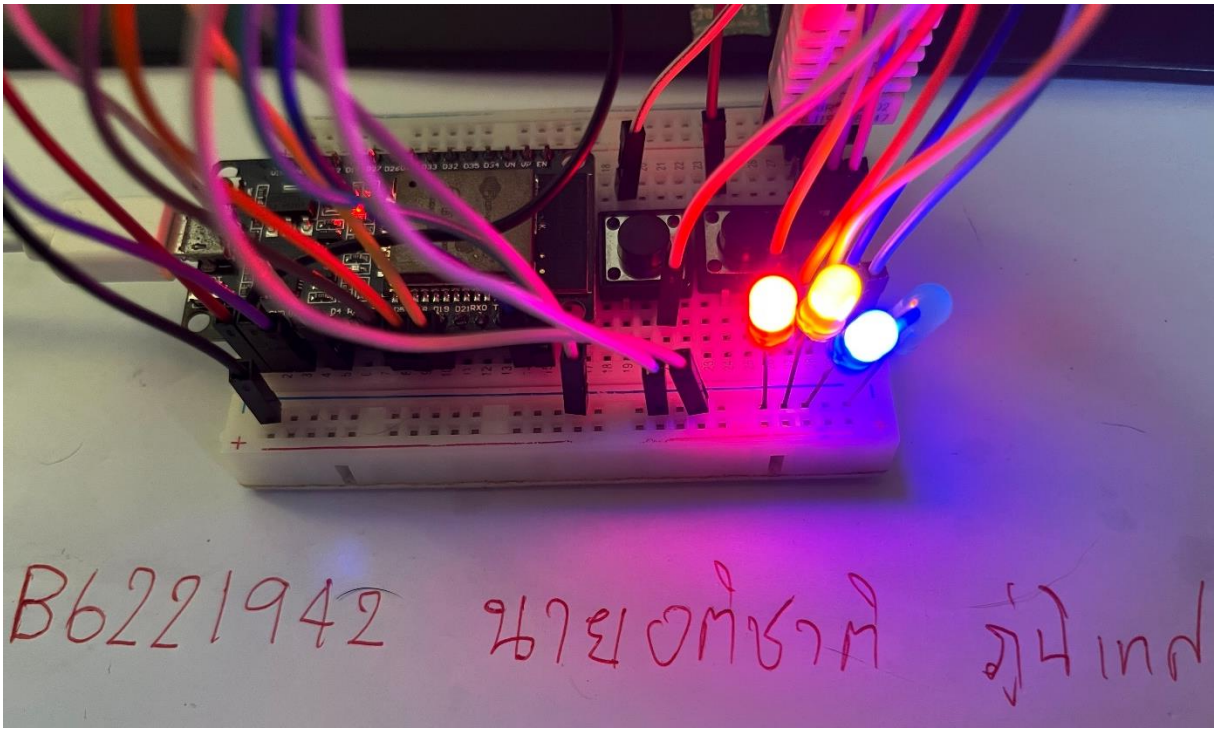
รูปหน้าจอ Blynk



รูปหน้าจอ LINE


15:24
LINE Notify
วันนี้
TestESP32: อุณหภูมิ: 28.70 องศา
15:20
TestESP32: อุณหภูมิ: 28.70 องศา
15:20
TestESP32: อุณหภูมิ: 28.60 องศา
15:20
TestESP32: อุณหภูมิ: 28.70 องศา
15:21
TestESP32: Intruders Alarm
15:21
TestESP32: อุณหภูมิ: 28.60 องศา
15:21
TestESP32: Overheat Alarm
15:21
TestESP32: อุณหภูมิ: 28.70 องศา
15:21

รูปการต่อวงจร – 1



รูปการต่อวงจร – 2