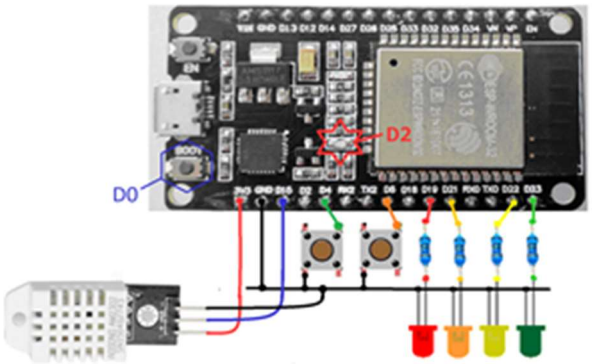
**แนวทางการใช้งานอินเทอร์เน็ตของสรรพสิ่งในระบบการผลิต**

**IoT Approaches to Manufacturing System**

**ชื่อ-สกุล : B6310646 นางสาวสุภานัน เรืองสุข**

**5/5. คำถามท้ายบทเพื่อทดสอบความเข้าใจ**

## Quiz_401 – Ubidots: Monitor DHT22, Monitor Digital Switch and Control 4 LED




```
< Test Code >
#include <WiFi.h>
#include <PubSubClient.h>
#include "DHTesp.h"
const char *My_SSID = "meow";
const char *My_Pass = "meowmeow";
const char *MQTT_Server = "things.ubidots.com";
const char *MQTT_User = "BBFF-Jp2I43cE9YMkzLwE2zipc9X8INLyAk";
const char *MQTT_Pass = "BBFF-Jp2I43cE9YMkzLwE2zipc9X8INLyAk";
const char *PTopic1 = "/v2.0/devices/pk009test";
const char *STopic1 = "/v2.0/devices/pk009test/humid";
const char *STopic2 = "/v2.0/devices/pk009test/tempp";
const char *STopic3 = "/v2.0/devices/pk009test/led1";
const char *STopic4 = "/v2.0/devices/pk009test/led2";
const char *STopic5 = "/v2.0/devices/pk009test/led3";
const char *STopic6 = "/v2.0/devices/pk009test/led4";
const char *STopic7 = "/v2.0/devices/pk009test/sw1";
const char *STopic8 = "/v2.0/devices/pk009test/sw2";
#define MQTT_Port 1883
```

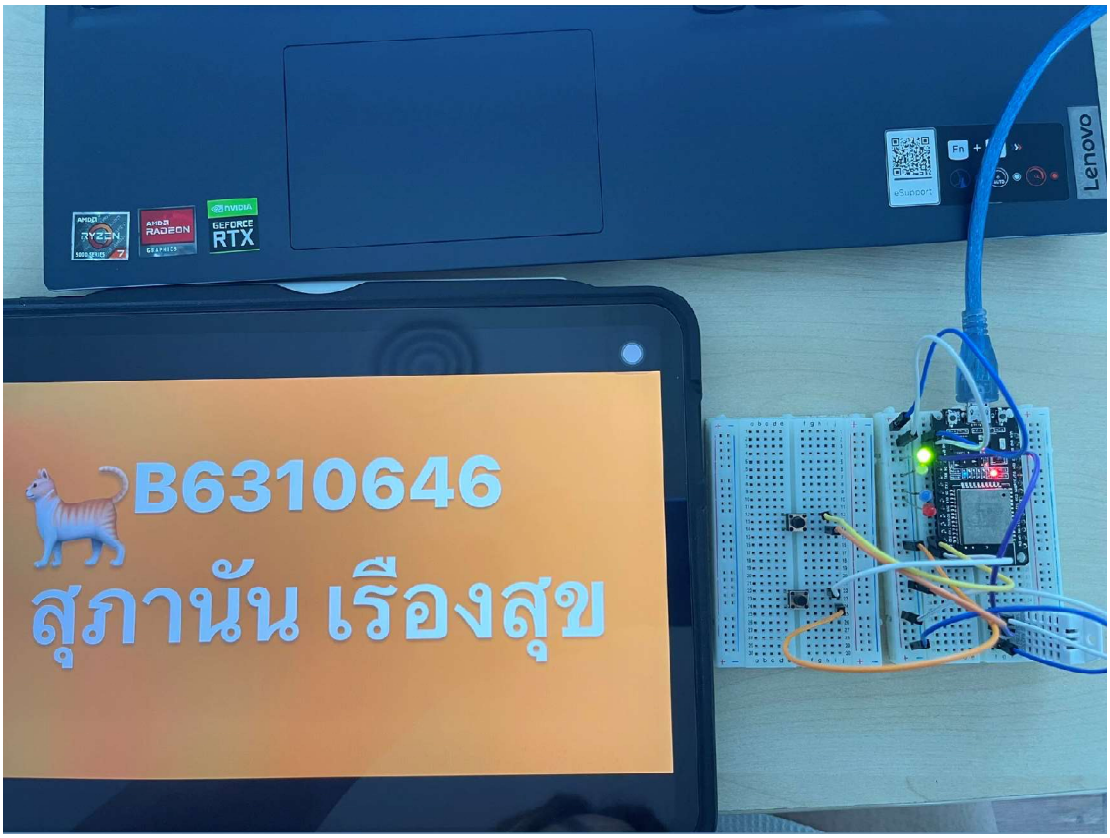
```
#define Test_LED1 2
#define Test_LED2 4
#define Test_LED3 5
#define Test_LED4 18
#define Test_SW1 22
#define Test_SW2 23
#define Pin_DHT22 15
DHTesp dht;
WiFiClient espClient;
PubSubClient client(espClient);
long lastMsg = 0;
char msg[50];
int value = 0;
void Setup_Wifi() {
  delay(10);
  Serial.println();
  Serial.print("Connecting to ");
  Serial.println(My_SSID);
  WiFi.begin(My_SSID, My_Pass);
  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(), MQTT_User, MQTT_Pass)) // Attempt to
connect
```

```
    { Serial.println("connected"); // Once connected, publish an announcement...
      client.subscribe(STopic1);
      client.subscribe(STopic2);
      client.subscribe(STopic3);
      client.subscribe(STopic4);
      client.subscribe(STopic5);
      client.subscribe(STopic6);
      client.subscribe(STopic7);
      client.subscribe(STopic8);
    } 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)
{ Serial.print("Message arrived [");
  Serial.print(topic);
  Serial.print("] ");
  for (int i = 0; i < length; i++)
  { Serial.print((char)payload[i]);
  }
  if (topic[24] == STopic3[24]) {
    Serial.print(" -LED1->> ");
    Serial.print((char)payload[10]);
    if (payload[10] == '1')
      digitalWrite(Test_LED1, HIGH);
    else
      digitalWrite(Test_LED1, LOW);
  }
  else if (topic[24] == STopic4[24]) {
    Serial.print(" -LED2->> ");
    Serial.print((char)payload[10]);
```

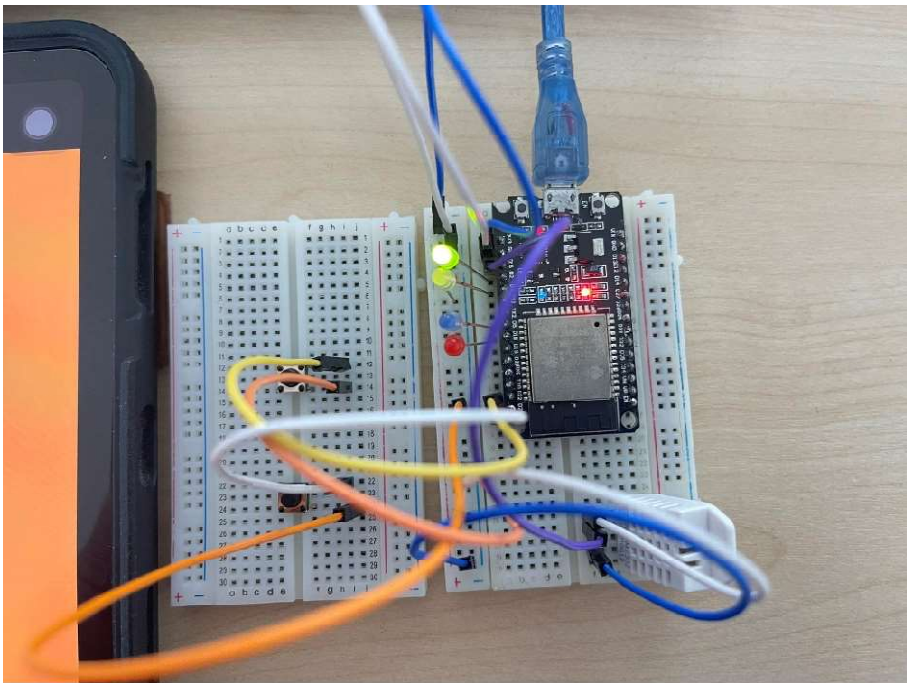
```
  if (payload[10] == '1')
    digitalWrite(Test_LED2, HIGH);
  else
    digitalWrite(Test_LED2, LOW);
 }
 else if (topic[24] == STopic5[24]) {
  Serial.print(" -LED3->> ");
  Serial.print((char)payload[10]);
  if (payload[10] == '1')
    digitalWrite(Test_LED3, HIGH);
  else
    digitalWrite(Test_LED3, LOW);
 }
 else if (topic[24] == STopic6[24]) {
  Serial.print(" -LED4->> ");
  Serial.print((char)payload[10]);
  if (payload[10] == '1')
    digitalWrite(Test_LED4, HIGH);
  else
    digitalWrite(Test_LED4, LOW);
 }

 Serial.println();
}
void setup()
{ pinMode(Test_LED1, OUTPUT);
 pinMode(Test_LED2, OUTPUT);
 pinMode(Test_LED3, OUTPUT);
 pinMode(Test_LED4, OUTPUT);
 pinMode(Test_SW1, INPUT_PULLDOWN);
 pinMode(Test_SW2, INPUT_PULLDOWN);
 dht.setup(Pin_DHT22, DHTesp::DHT22);
 Serial.begin(115200);
 Setup_Wifi();
 client.setServer(MQTT_Server, MQTT_Port);
```

```
  client.setCallback(callback);
}
void loop()
{ if (!client.connected()) reconnect();
  client.loop();
  long now = millis();
  float humidity = dht.getHumidity();
  float temperature = dht.getTemperature();
  int SW1 = 0;
  int SW2 = 0;

  if (digitalRead(Test_SW1) == HIGH) SW1 = 1;
  else SW1 = 0;
  if (digitalRead(Test_SW2) == LOW) SW2 = 1;
  else SW2 = 0;
  if (now - lastMsg >= 5000)
  {
    snprintf (msg, 75, "{ \"humid\" : %.2f, \"tempp\": %.2f, \"sw1\": %d,
\"sw2\": %d }", humidity, temperature, SW1, SW2);
    Serial.print("Publish message: ");
    Serial.println(msg);
    client.publish(PTopic1, msg);
    lastMsg = now;
    delay(1000);
  }

}
```

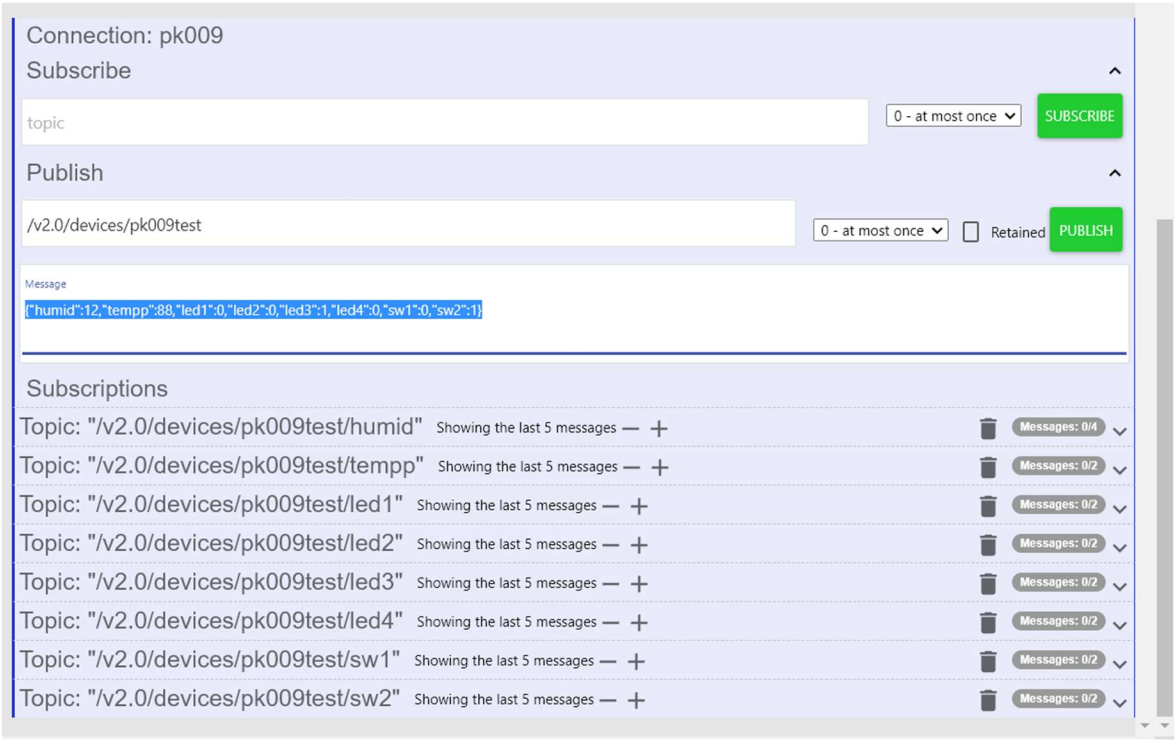
**รูปการต่อวงจร – 1**

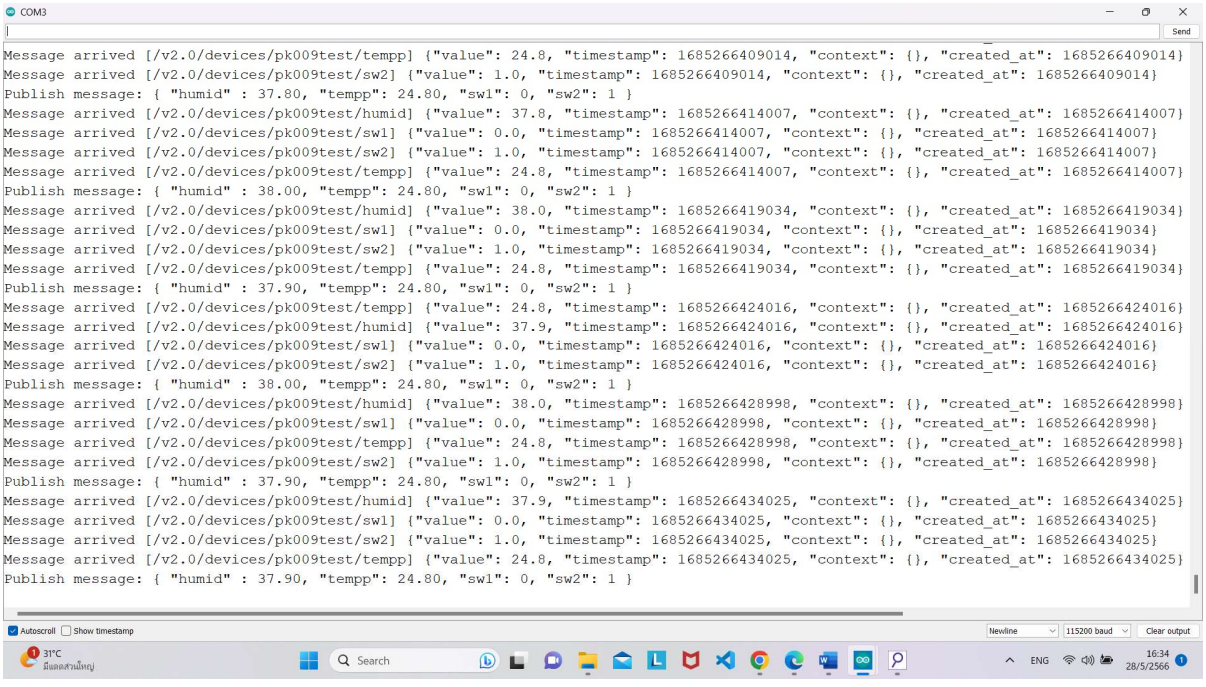


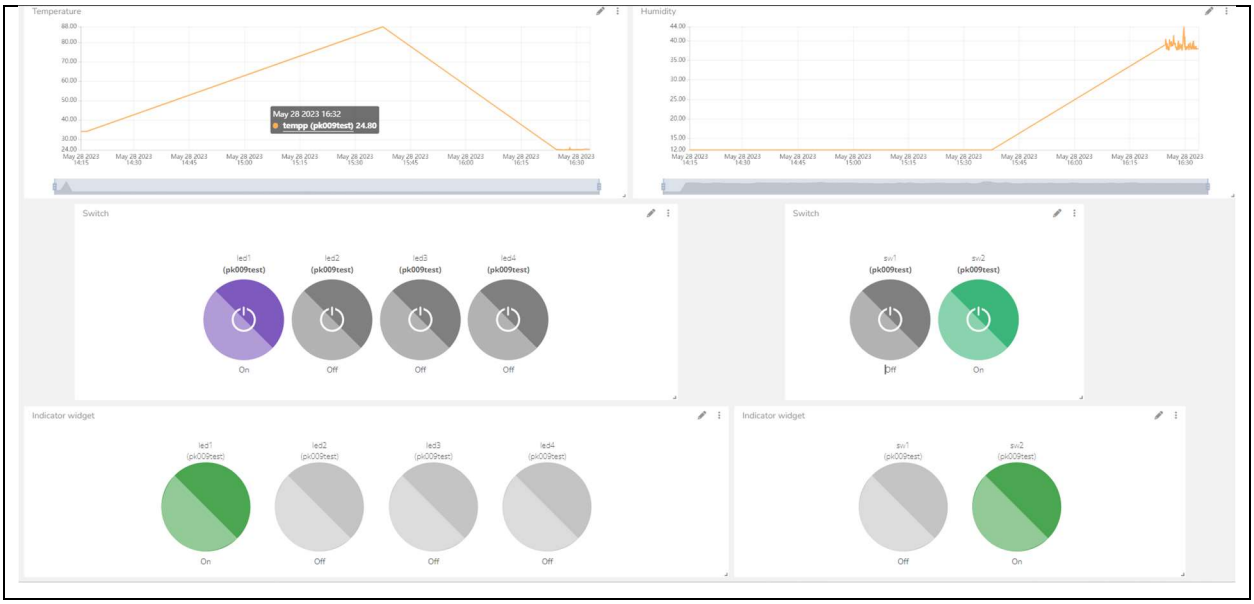

**รูปการต่อวงจร – 2**



## รูปหน้าจอ **Ubidot Dashboard**

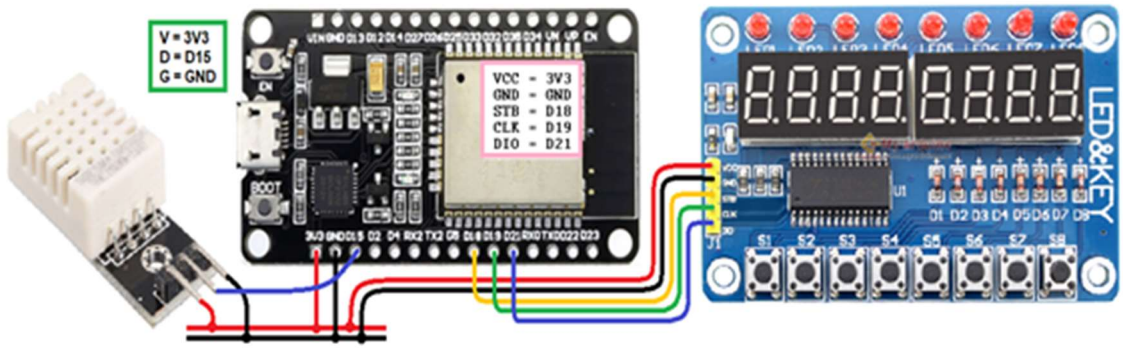



Temperature
May 28 2023 16:32
tempp (pk009test) 24.80
Humidity
Switch
led1 (pk009test)
led2 (pk009test)
led3 (pk009test)
led4 (pk009test)
On
Off
Off
Off
Switch
sw1 (pk009test)
sw2 (pk009test)
Off
On
Indicator widget
led1 (pk009test)
led2 (pk009test)
led3 (pk009test)
led4 (pk009test)
On
Off
Off
Off
Indicator widget
sw1 (pk009test)
sw2 (pk009test)
Off
On

## Quiz_402 – Ubidots: Monitor DHT22 with TM1638 Display and LINE Alert

- ส่งข้อมูลอุณหภูมิไปยัง Ubidots
- หากอุณหภูมิที่อ่านได้เกิน 28'C ให้แจ้งเตือนผ่าน LINE และบอกด้วยว่าอุณหภูมิเท่าใด
- แสดงอุณหภูมิที่ 7_Segment Display TM1638 Board




**< Test Code >**

```
#include <WiFi.h>
#include <PubSubClient.h>
#include <HTTPClient.h>
#include <TM1638plus.h>
#include "DHTesp.h"
#include <TridentTD_LineNotify.h>
const char *My_SSID = "meow";
const char *My_Pass = "meowmeow";
const char *MQTT_Server = "things.ubidots.com";
const char *MQTT_User = "BBFF-Jp2I43cE9YMkzLwE2zipc9X8lNLyAk";
const char *MQTT_Pass = "BBFF-Jp2I43cE9YMkzLwE2zipc9X8lNLyAk";

#define LINE_TOKEN "vvdHAVniW3I2lqonRVuKxiJIBzoveYwfuRbHeRENjHJ"

const char *PTopic1 = "/v2.0/devices/pk009test";
const char *STopic1 = "/v2.0/devices/pk009test/humid";
const char *STopic2 = "/v2.0/devices/pk009test/tempp";
#define Brd_STB 18 // strobe = GPIO connected to strobe line of module
#define Brd_CLK 19 // clock = GPIO connected to clock line of module
#define Brd_DIO 5 // data = GPIO connected to data line of module
```

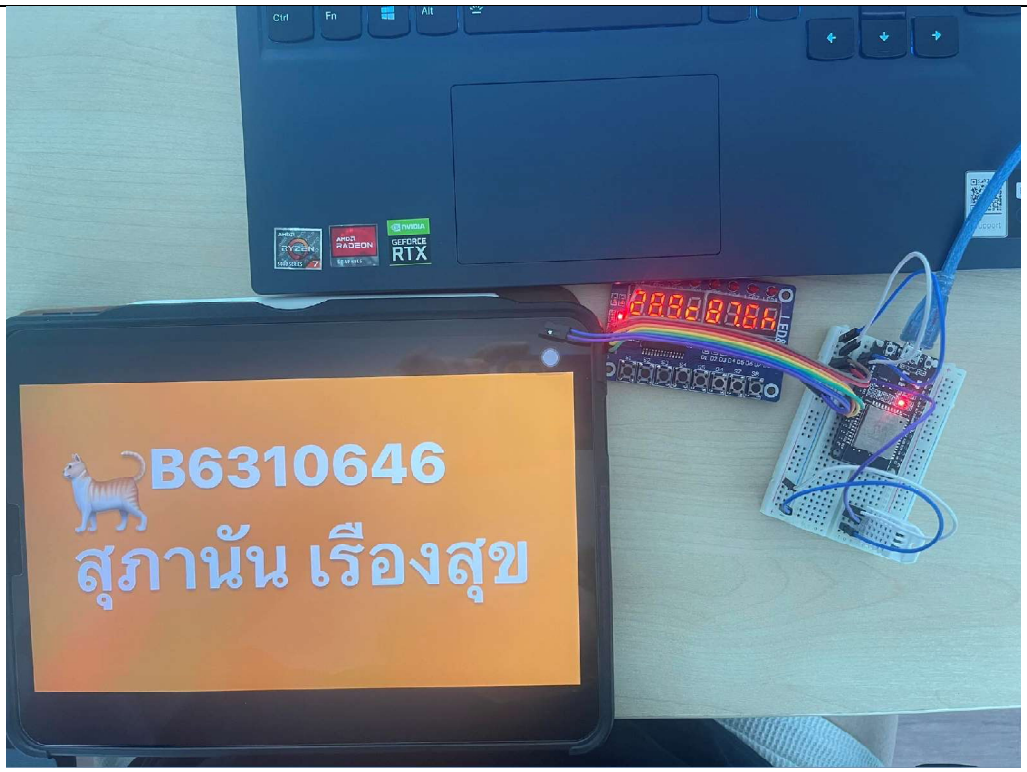
```
bool high_freq = true; //default false,, If using a high freq CPU > ~100 MHZ set to true.
TM1638plus tm(Brd_STB, Brd_CLK , Brd_DIO, high_freq);
#define MQTT_Port 1883
#define Pin_DHT22 15

DHTesp dht;
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(My_SSID);
  WiFi.begin(My_SSID, My_Pass);
  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(), MQTT_User, MQTT_Pass)) // Attempt to connect
    { Serial.println("connected"); // Once connected, publish an announcement...
      client.subscribe(STopic1);
```

```
    client.subscribe(STopic2);
   } else
   { Serial.print("failed, rc=");
    Serial.print(client.state());
    Serial.println(" try again in 5 seconds");
    delay(5000);
   }
  }
}
void setup()
{
 tm.displayBegin();
 dht.setup(Pin_DHT22, DHTesp::DHT22);
 Serial.begin(115200);
 Setup_Wifi();
 client.setServer(MQTT_Server, MQTT_Port);

 Serial.println(LINE.getVersion());
 Serial.println(WiFi.localIP());
 LINE.setToken(LINE_TOKEN);
}
void loop()
{ if (!client.connected()) reconnect();
 client.loop();
 long now = millis();
 if (now - lastMsg > 5000)
 { lastMsg = now;
  float humidity = dht.getHumidity();
  float temperature = dht.getTemperature();

  snprintf (msg, 75, "{ \"humid\" : %.2f, \"tempp\": %.2f}", humidity,
temperature);
  Serial.print("Publish message: ");
  Serial.println(msg);
  client.publish(PTopic1, msg);
```

```
  Serial.println();
  Serial.print("\nTemperature('C) = ");
  Serial.print(temperature, 1);
  Serial.print("\tHumidity(%) = ");
  Serial.print(humidity, 1);
  Serial.println();



  if (WiFi.status() == WL_CONNECTED) {
   HTTPClient http;
   http.addHeader("Content-Type", "application/x-www-form-urlencoded");
   Serial.print("HTTP Response code: ");
   http.end();
  }
  /// if temp > 28 C send notifications >> line
  if (temperature > 28) {
   Serial.println(LINE.getVersion());
   Serial.println(WiFi.localIP());
   LINE.setToken(LINE_TOKEN);
   LINE.notify("Temperature Overheat");
   LINE.notify("Temperature");
   LINE.notify(temperature);
   LINE.notify("Humidity");
   LINE.notify(humidity);


  }
  /*Display */
  int t = int(temperature * 100);
  int Tempp2 = (int)temperature / 10; int Tempp1 = (int)temperature % 10; int
Tempp0 = (int)(temperature * 10) % 10;
  int Humi2 = (int)humidity / 10; int Humi1 = (int)humidity % 10; int Humi0 =
(int)(humidity * 10) % 10;
  tm.displayHex(0, Tempp2);
  tm.displayASCIIwDot(1, Tempp1 + '0'); // turn on dot
  tm.displayHex(2, Tempp0);
```

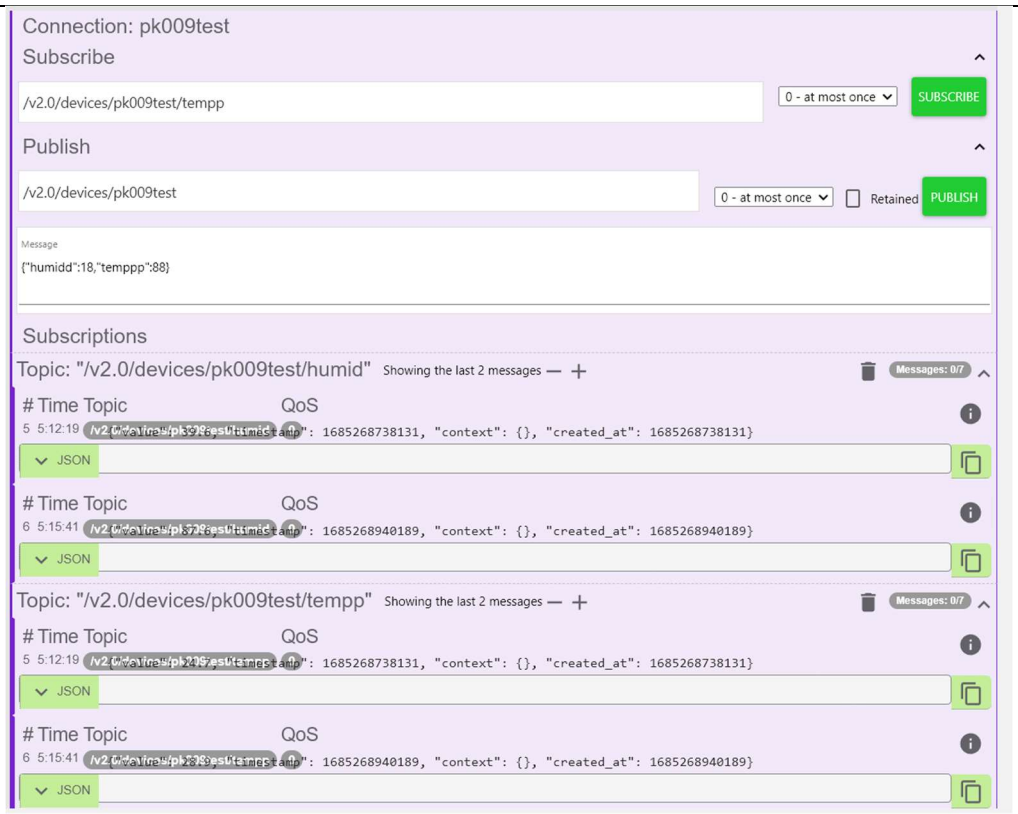
```
    tm.display7Seg(3, B01011000); // Code=tgfedcba
    tm.displayHex(4, Humi2);
    tm.displayASCIIwDot(5, Humi1 + '0'); // turn on dot
    tm.displayHex(6, Humi0);
    tm.display7Seg(7, B01110100); // Code=tgfedcba
    delay(2000);

    int WaitTime = 60;
    Serial.print(" >> Wait for next time --> ");
    for (int i = WaitTime; i >= 0; i -= 5) {
      Serial.print(",");
      Serial.print(i);
      delay(5000);
    }
  }
}
```

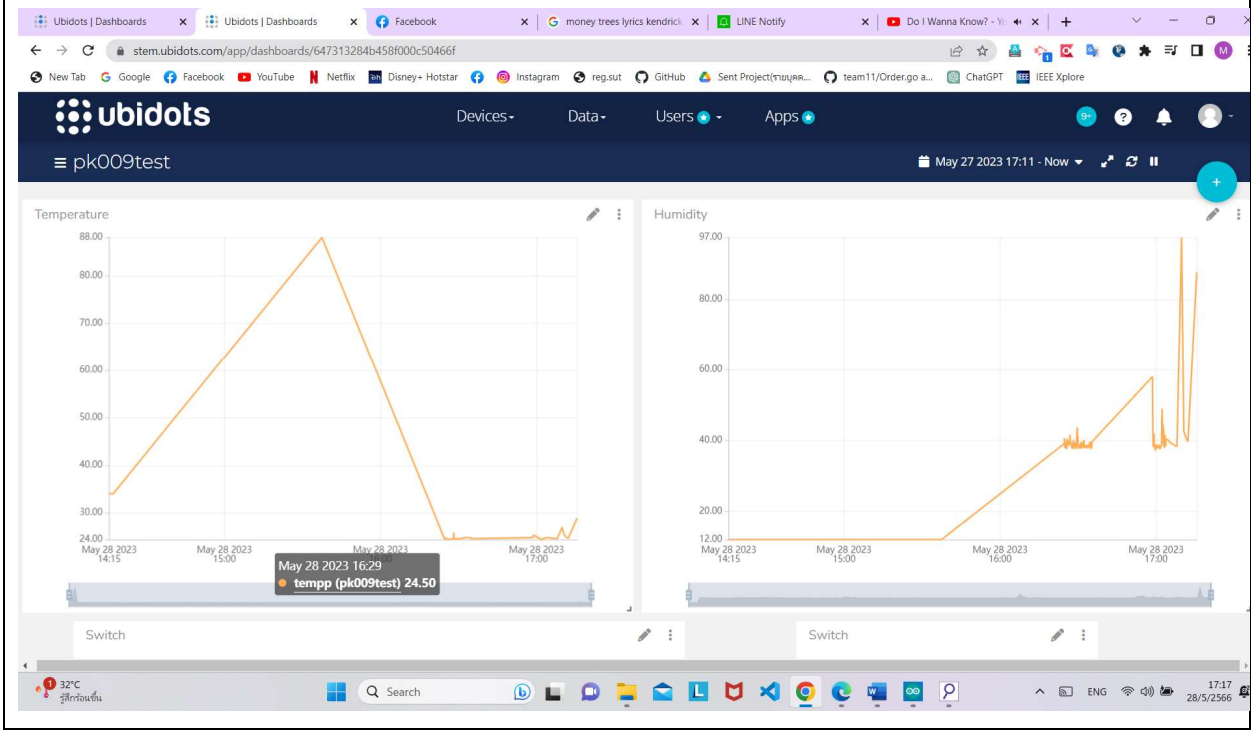
**รูปการต่อวงจร – 1**




**รูปการต่อวงจร – 2**


**รูปหน้าจอ Ubidot Dashboard**


Connection: pk009test
Subscribe
/v2.0/devices/pk009test/tempp
0 - at most once
SUBSCRIBE
Publish
/v2.0/devices/pk009test
0 - at most once
Retained
PUBLISH
Message
{"humidd":18,"temppp":88}
Subscriptions
Topic: "/v2.0/devices/pk009test/humid"
Showing the last 2 messages
Messages: 0/7
# Time Topic QoS
5 5:12:19 /v2.0/devices/pk009test/humid {"value": 18, "timestamp": 1685268738131, "context": {}, "created_at": 1685268738131}
JSON
# Time Topic QoS
6 5:15:41 /v2.0/devices/pk009test/humid {"value": 18, "timestamp": 1685268940189, "context": {}, "created_at": 1685268940189}
JSON
Topic: "/v2.0/devices/pk009test/tempp"
Showing the last 2 messages
Messages: 0/7
# Time Topic QoS
5 5:12:19 /v2.0/devices/pk009test/tempp {"value": 24.5, "timestamp": 1685268738131, "context": {}, "created_at": 1685268738131}
JSON
# Time Topic QoS
6 5:15:41 /v2.0/devices/pk009test/tempp {"value": 24.5, "timestamp": 1685268940189, "context": {}, "created_at": 1685268940189}
JSON


ubidots
Devices
Data
Users
Apps
pk009test
May 27 2023 17:11 - Now
Temperature
Humidity
May 28 2023 16:29
tempp (pk009test) 24.50
Switch
Switch

```
COM3
172.20.10.4
[TridentTD_LineNotify] Version 3.03
172.20.10.4
Attempting MQTT connection...connected
Publish message: { "humid" : 39.60, "tempp": 24.70}


Temperature('C) = 24.7  Humidity(%) = 39.6
HTTP Response code:  >> Wait for next time --> ,60,55,50,45,40,35,30,25,20,15,10,5,0Publish message: { "humid" : 40.80, "tempp": 24.60}


Temperature('C) = 24.6  Humidity(%) = 40.8
HTTP Response code:  >> Wait for next time --> ,60,55,50,45,40,35,30,25,20,15,10,5,0Publish message: { "humid" : 42.90, "tempp": 25.80}


Temperature('C) = 25.8  Humidity(%) = 42.9
HTTP Response code:  >> Wait for next time --> ,60,55,50,45,40,35,30,25,20,15,10,5,0Attempting MQTT connection...connected
Publish message: { "humid" : 87.60, "tempp": 28.90}


Temperature('C) = 28.9  Humidity(%) = 87.6
HTTP Response code: [TridentTD_LineNotify] Version 3.03
172.20.10.4
 >> Wait for next time --> ,60,55,50,45,40,35,30,25,20,15,10,5,0Publish message: { "humid" : 87.00, "tempp": 30.40}


Temperature('C) = 30.4  Humidity(%) = 87.0
HTTP Response code: [TridentTD_LineNotify] Version 3.03
172.20.10.4
 >> Wait for next time --> ,60,55
Autoscroll  Show timestamp   Newline  115200 baud  Clear output
```

## รูปหน้าจอ **LINE** ผลการทดสอบ