

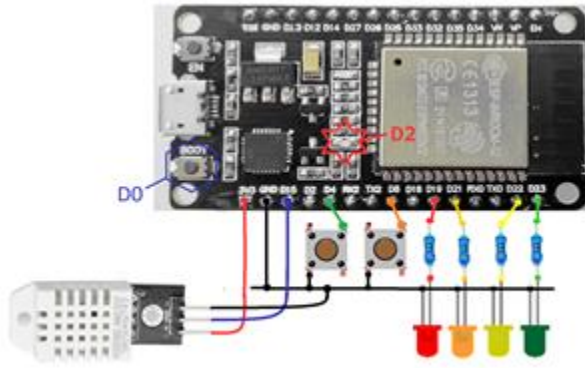
## แนวทางการใช้งานอินเทอร์เน็ตของสรรพสิ่งในระบบการผลิต

## IoT Approaches to Manufacturing System

ชื่อ-สกุล : ญัฐพงศ์ โต๊ะแอ รหัสนักศึกษา : B6310158

## 5/5. คำถามท้ายบทเพื่อทดสอบความเข้าใจ

## Quiz\_401 – Ubidots: Monitor DHT22, Monitor Digital Switch and Control 4 LED



&lt; Test Code &gt;

```

1 #include <WiFi.h>
2 #include <PubSubClient.h>
3 #include "DHTesp.h"
4 const char *My_SSID = "jjjj";
5 const char *My_Pass = "0846894722";
6 const char *MQTT_Server = "things.ubidots.com";
7 const char *MQTT_User = "BBFF-MYZ5ORgUkVAHftitciOOhQJfYLIDf1";
8 const char *MQTT_Pass = "BBFF-MYZ5ORgUkVAHftitciOOhQJfYLIDf1";
9 const char *PTopic1 = "/v2.0/devices/quiz401";
10 const char *STopic1 = "/v2.0/devices/quiz401/humid";
11 const char *STopic2 = "/v2.0/devices/quiz401/tempp";
12 const char *STopic3 = "/v2.0/devices/quiz401/led1";
13 const char *STopic4 = "/v2.0/devices/quiz401/led2";
14 const char *STopic5 = "/v2.0/devices/quiz401/led3";
15 const char *STopic6 = "/v2.0/devices/quiz401/led4";
16 const char *STopic7 = "/v2.0/devices/quiz401/sw1";
17 const char *STopic8 = "/v2.0/devices/quiz401/sw2";
18 #define MQTT_Port 1883
19 #define Test_LED1 2
20 #define Test_LED2 4
21 #define Test_LED3 5
22 #define Test_LED4 18
23 #define Test_SW1 22
24 #define Test_SW2 23
25 #define Pin_DHT22 15
26 DHTesp dht;
27 WiFiClient espClient;
28 PubSubClient client(espClient);
29 long lastMsg = 0;
30 char msg[50];
31 int value = 0;
32 void setup_wifi() {
33   delay(10);
34   Serial.println();
35   Serial.print("Connecting to ");

```

```

#include <WiFi.h>
#include <PubSubClient.h>
#include "DHTesp.h"
const char *My_SSID = "jjjj";
const char *My_Pass = "0846894722";
const char *MQTT_Server = "things.ubidots.com";
const char *MQTT_User = "BBFF-MYZ5ORgUkVAHftitciOOhQJfYLIDf1";
const char *MQTT_Pass = "BBFF-MYZ5ORgUkVAHftitciOOhQJfYLIDf1";
const char *PTopic1 = "/v2.0/devices/quiz401";
const char *STopic1 = "/v2.0/devices/quiz401/humid";
const char *STopic2 = "/v2.0/devices/quiz401/tempp";
const char *STopic3 = "/v2.0/devices/quiz401/led1";
const char *STopic4 = "/v2.0/devices/quiz401/led2";
const char *STopic5 = "/v2.0/devices/quiz401/led3";
const char *STopic6 = "/v2.0/devices/quiz401/led4";
const char *STopic7 = "/v2.0/devices/quiz401/sw1";

```

```

const char *STopic8 = "/v2.0/devices/quiz401/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);
}
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);
}
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);
}
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;

```

```

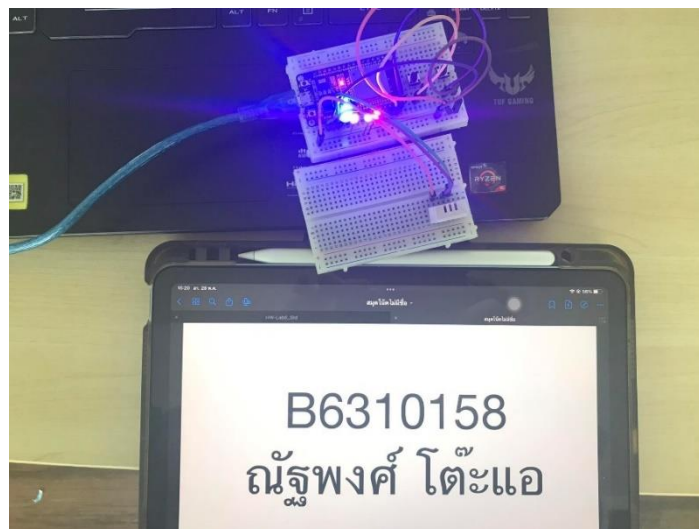
else SW2 = 0;
if (now - lastMsg >= 5000)
{
  snprintf (msg, 75, "{ \"humid\" : %.2f, \"tempp\" : %.2f, \"sw1\" : %d, \"sw2\" : %d
} ", humidity, temperature, SW1, SW2);
  Serial.print("Publish message: ");
  Serial.println(msg);
  client.publish(PTopic1, msg);
  lastMsg = now;
  delay(1000);
}
}

```

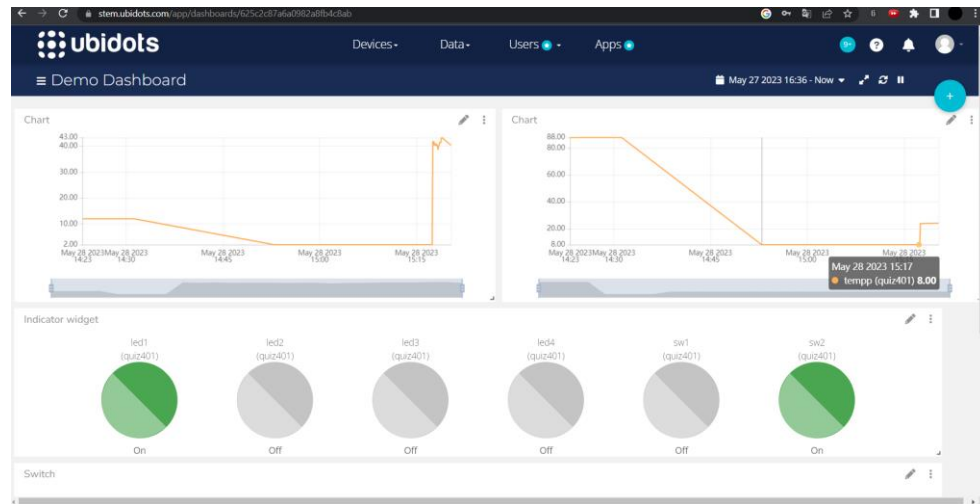
รูปการต่อวงจร – 1



รูปการต่อวงจร – 2

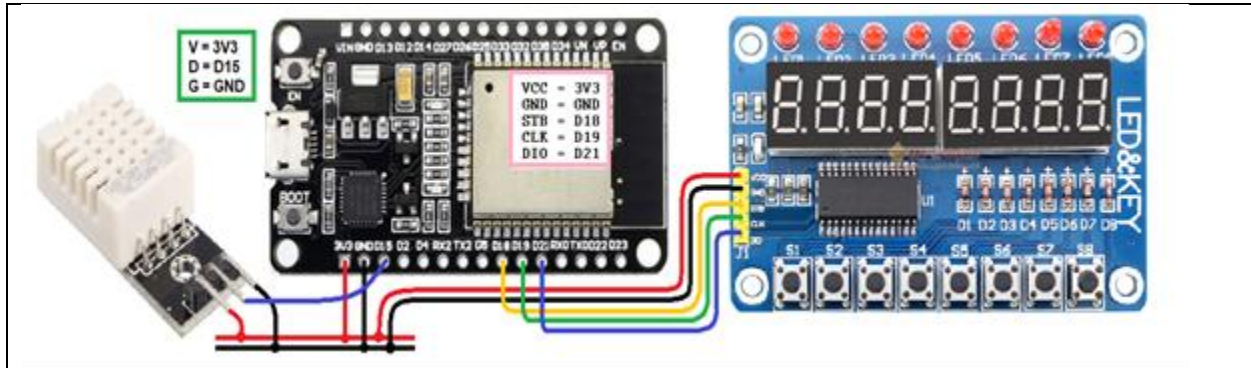


## หน้าจอ Ubidot Dashboard



## 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



## &lt; Test Code &gt;

```

1 #include <WiFi.h>
2 #include <PubSubClient.h>
3 #include <HTTPClient.h>
4 #include <TM1638plus.h>
5 #include "DHTesp.h"
6 #include <TridentTD_LineNotify.h>
7 const char *My_SSID = "jjjj";
8 const char *My_Pass = "0846894722";
9 const char *MQTT_Server = "things.ubidots.com";
10 const char *MQTT_User = "BBFF-MYZ5ORgUkVAHftitciOOhQJfYLIDf1";
11 const char *MQTT_Pass = "BBFF-MYZ5ORgUkVAHftitciOOhQJfYLIDf1";
12
13 #define LINE_TOKEN "ACvQcYKf1jBky1m9saYYqKjGha8W42YvVc4wzSPNCgM"
14
15 const char *PTopic1 = "/v2.0/devices/gg007test";
16 const char *STopic1 = "/v2.0/devices/gg007test/humid";
17 const char *STopic2 = "/v2.0/devices/gg007test/tempp";
18 #define Brd_STB 18 // strobe = GPIO connected to strobe line of module
19 #define Brd_CLK 19 // clock = GPIO connected to clock line of module
20 #define Brd_DIO 5 // data = GPIO connected to data line of module
21 bool high_freq = true; // default false, If using a high freq CPU > ~100 MHz set to true.
22 #include <TM1638plus.h> // Brd_STB, Brd_CLK, Brd_DIO, high_freq;
23 #define MQTT_Port 1883
24 #define Pin_DHT22 4
25
26 DHTesp dht;
27 WiFiClient espClient;

```

```

#include <WiFi.h>
#include <PubSubClient.h>
#include <HTTPClient.h>
#include <TM1638plus.h>
#include "DHTesp.h"
#include <TridentTD_LineNotify.h>
const char *My_SSID = "jjjj";
const char *My_Pass = "0846894722";
const char *MQTT_Server = "things.ubidots.com";
const char *MQTT_User = "BBFF-MYZ5ORgUkVAHftitciOOhQJfYLIDf1";
const char *MQTT_Pass = "BBFF-MYZ5ORgUkVAHftitciOOhQJfYLIDf1";
#define LINE_TOKEN "ACvQcYKf1jBky1m9saYYqKjGha8W42YvVc4wzSPNCgM"
const char *PTopic1 = "/v2.0/devices/gg007test";
const char *STopic1 = "/v2.0/devices/gg007test/humid";
const char *STopic2 = "/v2.0/devices/gg007test/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("อุณหภูมิ เกิน 28 องศา");
      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);
```

```
  }
}
}
```

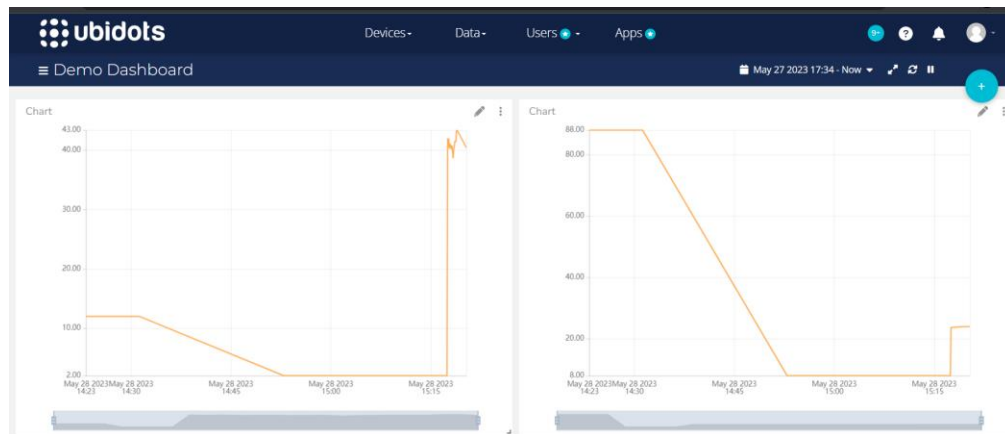
รูปการต่อวงจร - 1



รูปการต่อวงจร - 2



## รูปหน้าจอ Ubidot Dashboard



## รูปหน้าจอ LINE ผลการทดสอบ

