

การควบคุมเครื่องจักรอัจฉริยะโดยใช้การสื่อสารระหว่างเครื่องจักรกับเครื่องจักร  
M2M - Intelligence Machine Control

ชื่อ-สกุล : นางสาวณัฐชยา ผ่องกุล B6226718

5/5: -- คำถามท้ายบทเพื่อทดสอบความเข้าใจ

Quiz\_301 – Start SCADA [ไม่ต้องทำเพราะไม่มี PLC]

< รูปอุปกรณ์ที่ใช้ทดสอบ ขณะทำการทดสอบ >

< รูปอุปกรณ์ที่ใช้ทดสอบ ขณะทำการทดสอบ >

รายละเอียดการทดสอบ

< โปรแกรมทดสอบ >

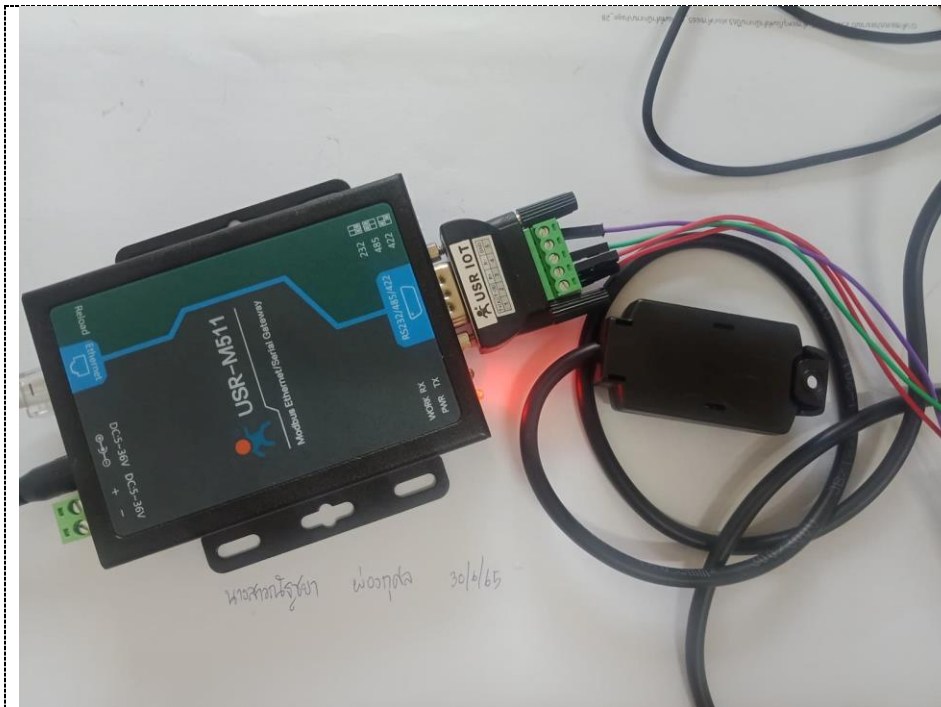
< ผลการทดสอบ >

Quiz\_302 – Modbus TCP Read/Write

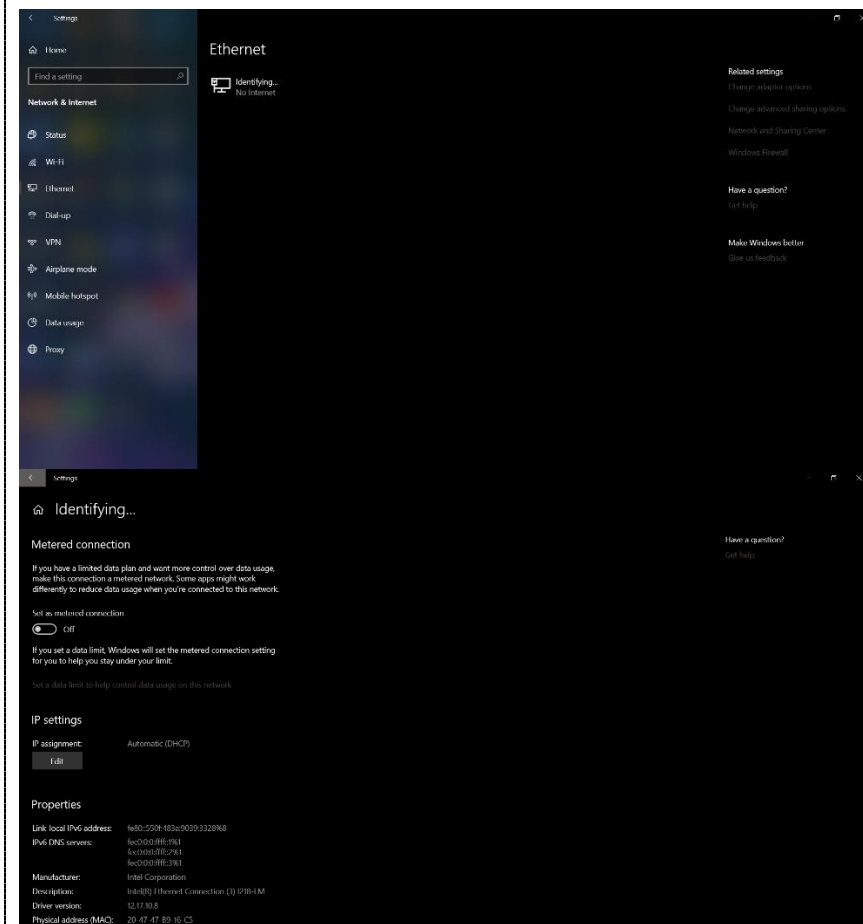
< รูปอุปกรณ์ที่ใช้ทดสอบ ขณะทำการทดสอบ >

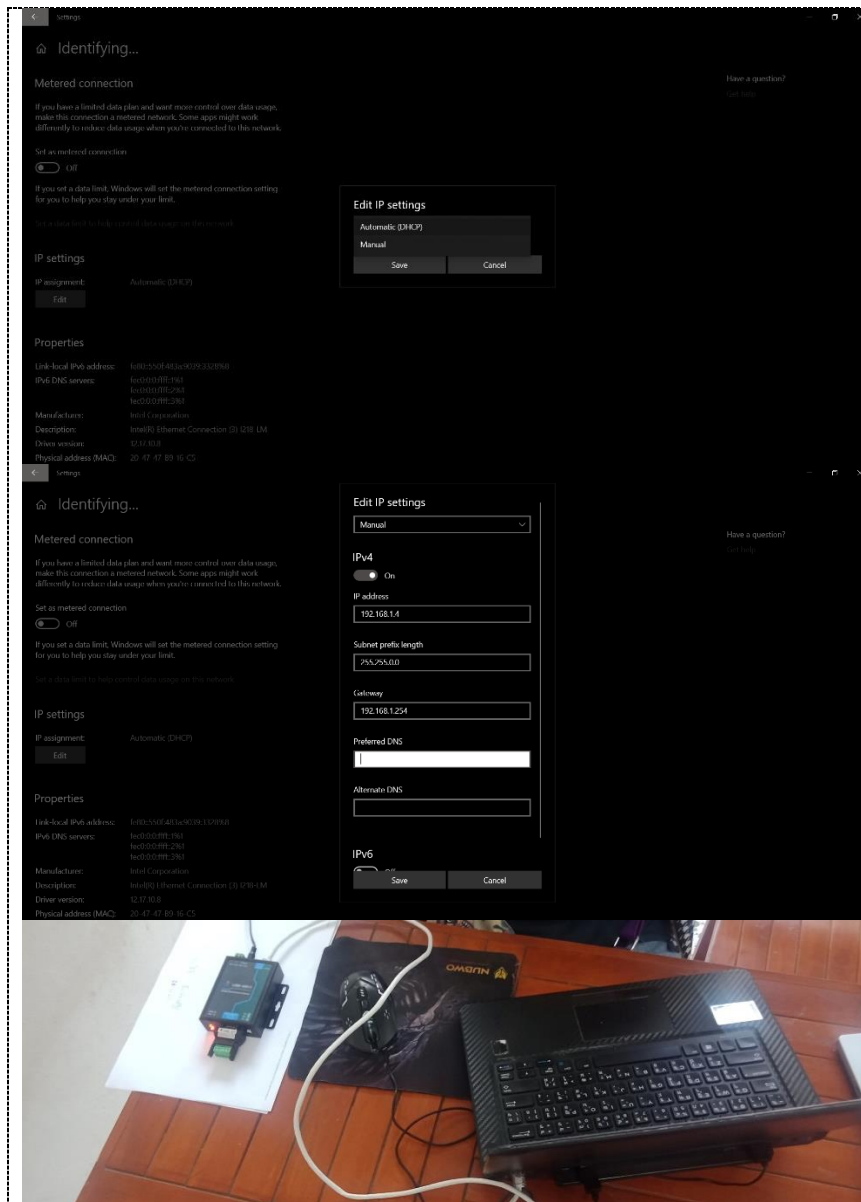


< รูปอุปกรณ์ที่ใช้ทดสอบ ขณะทำการทดสอบ >



## รายละเอียดการทดสอบ





## &lt; โปรแกรมทดสอบ &gt;

The screenshot displays the Modbus Poll software interface. The top window shows the 'Connection Setup' dialog with 'Modbus TCP/IP' selected. The bottom window shows the 'Read/Write Definition' dialog with '03 Read Holding Registers (4x)' selected. The status bar at the bottom shows 'Tx = 7; Err = 0; ID = 1; F = 03; SR = 1000ms'.

**Connection Setup Dialog:**

- Connection: Modbus TCP/IP
- Serial Settings: Silicon Labs CP210x USB to UART Bridge (COM3), 9600 Baud, 8 Data bits, None Parity, 1 Stop Bit
- Mode: RTU (selected), ASCII
- Response Timeout: 1000 [ms]
- Delay Between Polls: 10 [ms]
- Remote Modbus Server: IP Address or Node Name: 192.168.1.4, Server Port: 502, Connect Timeout: 3000 [ms], IP Version: IPv4 (selected)

**Read/Write Definition Dialog:**

- Slave ID: 1
- Function: 03 Read Holding Registers (4x)
- Address mode: Dec (selected), Hex
- Address: 0, PLC address = 40001
- Quantity: 2
- Scan Rate: 1000 [ms]
- Disable: Read/Write Disabled (checked), Disable on error
- View: Rows: 10 (selected), 20, 50, 100, Fit to Quantity
- Hide Name Columns: (unchecked), PLC Addresses (Base 1): (unchecked)
- Address in Cell: (unchecked), Enron/Daniel Mode: (unchecked)
- Request: RTU: 01 03 00 00 00 02 C4 0B, ASCII: 3A 30 31 30 33 30 30 30 30 30 32 46 41 0D 0A

**Status Bar:** Tx = 7; Err = 0; ID = 1; F = 03; SR = 1000ms

&lt; ผลการทดสอบ &gt;

Mbpoll1

Tx = 7: Err = 0: ID = 1: F = 03: SR = 1000ms

	Alias	00000
0		283
1		533
2		
3		
4		
5		

Communication Traffic

Exit Stop Save

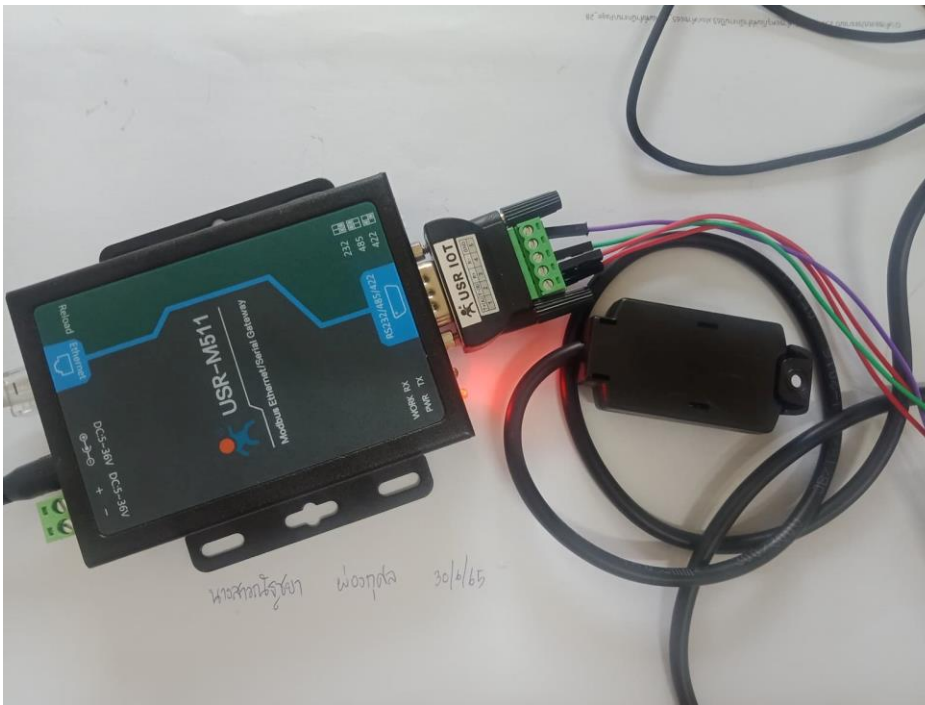
```

000000-Tx:00 7A 00 00 00 06 01 03 00 00 00 02
000001-Rx:00 7A 00 00 00 07 01 03 04 01 1C 02 15
000002-Tx:00 7B 00 00 00 06 01 03 00 00 00 02
000003-Rx:00 7B 00 00 00 07 01 03 04 01 1C 02 15
000004-Tx:00 7C 00 00 00 06 01 03 00 00 00 02
000005-Rx:00 7C 00 00 00 07 01 03 04 01 1C 02 15
000006-Tx:00 7D 00 00 00 06 01 03 00 00 00 02
000007-Rx:00 7D 00 00 00 07 01 03 04 01 1B 02 15
  
```

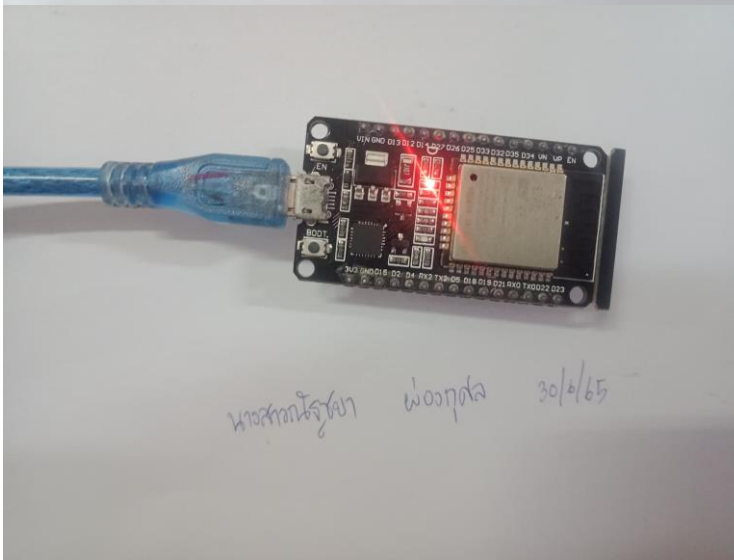
ถ้าสำเร็จจะได้ดังรูป คือจะต้องต่อ Modbus ethernet เข้ากับ modbus sensor และอื่นๆที่อยากทราบค่าด้วย เมื่อเรียบร้อยแล้วให้มาทดสอบใน modbus poll ก็จะได้ตามรูปอาจารย์

### Quiz\_303 – Modbus RTU/ASCII/TCP with IOTs

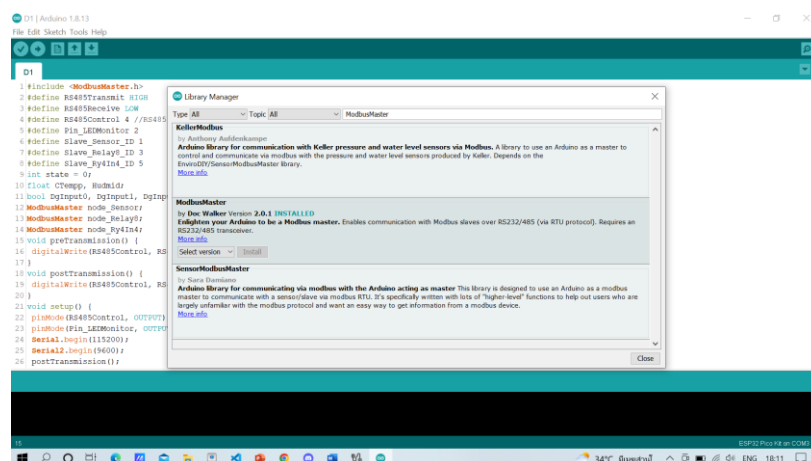
&lt; รูปอุปกรณ์ที่ใช้ทดสอบ ขณะทำการทดสอบ &gt;



&lt; รูปอุปกรณ์ที่ใช้ทดสอบ ขณะทำการทดสอบ &gt;



### รายละเอียดการทดสอบ



< โปรแกรมทดสอบ >

D1 | Arduino 1.8.13

File Edit Sketch Tools Help



D1 §

```

1 // esp32ModbusTCP >> https://github.com/bertmelis/esp32ModbusTCP
2 // AsyncTCP.h >> https://github.com/me-no-dev/AsyncTCP
3 #define BLYNK_PRINT Serial
4 #include <WiFi.h>
5 #include <WiFiClient.h>
6 #include <BlynkSimpleEsp32.h>
7 #include <Arduino.h>
8 #include <esp32ModbusTCP.h>
9 char ssid[] = "B6226718";
10 char pass[] = "pungpingjerry";
11 char auth[] = "WPa2mtFT_6q19RGfh3Nxc8AXs-y1fnoe";
12 bool WiFiConnected = false;
13 int Value_V0, Value_V1;
14 esp32ModbusTCP sunnyboy(1, {192, 168, 1, 4}, 502);
15 enum smaType {
16   ENUM, // enumeration
17   UFIX0, // unsigned 2 Byte, no decimals
18   SFIX0, // signed 4 Byte, no decimals
19 };
20 struct smaData {
21   const char* name;
22   uint16_t address;
23   uint16_t length;
24   smaType type;
25   uint16_t packetId;
26 };

```

```
// esp32ModbusTCP >> https://github.com/bertmelis/esp32ModbusTCP
```

```
// AsyncTCP.h >> https://github.com/me-no-dev/AsyncTCP
```

```
#define BLYNK_PRINT Serial
```

```
#include <WiFi.h>
```

```
#include <WiFiClient.h>
```

```
#include <BlynkSimpleEsp32.h>
```

```
#include <Arduino.h>
```

```
#include <esp32ModbusTCP.h>
```

```
char ssid[] = "Mue.Home";
```

```
char pass[] = "pk1212312121";
```

```
char auth[] = "YD3FmnLEk5vdhs-BeQlWwrACl8gXNgXK";
```

```
bool WiFiConnected = false;
```

```
int Value_V0, Value_V1;
```

```
esp32ModbusTCP sunnyboy(1, {192, 168, 1, 4}, 502);
```

```
enum smaType {
```

```
  ENUM, // enumeration
```

```
  UFIX0, // unsigned 2 Byte, no decimals
```

```
  SFIX0, // signed 4 Byte, no decimals
```

```
};
```

```
struct smaData {
```

```
  const char* name;
```

```
  uint16_t address;
```

```

uint16_t length;
smaType type;
uint16_t packetId;
};

smaData smaRegisters[] = {
  "Temp", 0, 1, UFIX0, 0,
  "Humid", 1, 1, UFIX0, 0
};

uint8_t numberSmaRegisters = sizeof(smaRegisters) / sizeof(smaRegisters[0]);
uint8_t currentSmaRegister = 0;
uint16_t ResultData[3];

BLYNK_WRITE(V0) {
  int temp = param.asInt();
  if (temp != Value_V0) {
    Value_V0 = temp;
    RelayControl(801 + temp * 10);
  }
}

BLYNK_WRITE(V1) {
  int temp = param.asInt();
  if (temp != Value_V1) {
    Value_V1 = temp;
    RelayControl(802 + temp * 10);
  }
}

void RelayControl(int Code) {
  Serial.println("Code is = " + String(Code));
}

void setup() {
  Serial.begin(115200);
  WiFi.disconnect(true); // delete old config
  sunnyboy.onData([](uint16_t packet, uint8_t slave, esp32Modbus::FunctionCode fc, uint8_t* data,
    uint16_t len) {
    for (uint8_t i = 0; i < numberSmaRegisters; ++i) {
      if (smaRegisters[i].packetId == packet) {

```



```

smaRegisters[i].packetId = 0;
switch (smaRegisters[i].type) {
case ENUM:
case UFIX0: {
uint32_t value = 0; // 2-Byte Data
value = (data[0] << 8) | (data[1]); // 2-Byte Data
Serial.printf("%s: %u\n", smaRegisters[i].name, value);
ResultData[i] = value;
break;
}
case SFIX0: {
int32_t value = 0;
value = (data[0] << 24) | (data[1] << 16) | (data[2] << 8) | (data[3]);
Serial.printf("%s: %i\n", smaRegisters[i].name, value);
break;
}
}
return;
}
}
});

sunnyboy.onError([](uint16_t packet, esp32Modbus::Error e) {
Serial.printf("Error packet %u: %02x\n", packet, e);
});

delay(1000);

WiFi.onEvent([](WiFiEvent_t event, WiFiEventInfo_t info) {
Serial.print("WiFi connected. IP: ");
Serial.println(IPAddress(info.got_ip.ip_info.ip.addr));
WiFiConnected = true;
}, WiFiEvent_t::SYSTEM_EVENT_STA_GOT_IP);

WiFi.onEvent([](WiFiEvent_t event, WiFiEventInfo_t info) {
Serial.print("WiFi lost connection. Reason: ");
Serial.println(info.disconnected.reason);
WiFi.disconnect();
WiFiConnected = false;

```

```

}, WiFiEvent_t::SYSTEM_EVENT_STA_DISCONNECTED);
WiFi.begin(ssid, pass);
Serial.println();
Serial.println("Connecting to WiFi... ");
}
int loopCount = 20;
void loop() {
  if (loopCount < 0 && WiFiConnected) {
    loopCount = 20;
    Serial.print("\nreading registers\n");
    for (uint8_t i = 0; i < numberSmaRegisters; ++i) {
      uint16_t packetId = sunnyboy.readHoldingRegisters(smaRegisters[i].address, smaRegisters[i].length);
      if (packetId > 0) {
        smaRegisters[i].packetId = packetId;
      } else {
        Serial.print("reading error\n");
      }
    }
    delay(5000);
    //Blynk.config(auth);
    float CTempp = ResultData[0] / 10.0;
    float Hudmid = ResultData[1] / 10.0;
    Blynk.virtualWrite(V10, CTempp);
    Blynk.virtualWrite(V11, Hudmid);
    Serial.println("V0=" + String(Value_V0));
    Serial.println("V1=" + String(Value_V1));
    Serial.println("V10=" + String(CTempp, 1));
    Serial.println("V11=" + String(Hudmid, 1));
  }
  Serial.print(String(loopCount--) + ",");
  //Blynk.run();
  delay(500);
}

```

< ผลการทดสอบ >



การควบคุมเครื่องจักรอัจฉริยะโดยใช้การสื่อสารระหว่างเครื่องจักรกับเครื่องจักร  
M2M - Intelligence Machine Control

ชื่อ-สกุล : นางสาวณัฐชยา ผ่องกุล B6226718

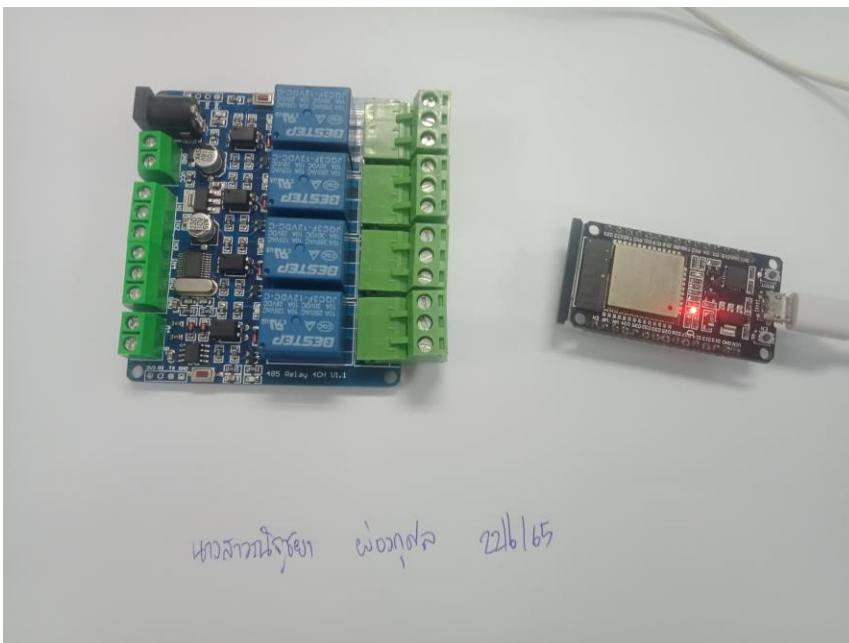
4/4: -- คำถามท้ายบทเพื่อทดสอบความเข้าใจ

Quiz\_401 – test Blynk

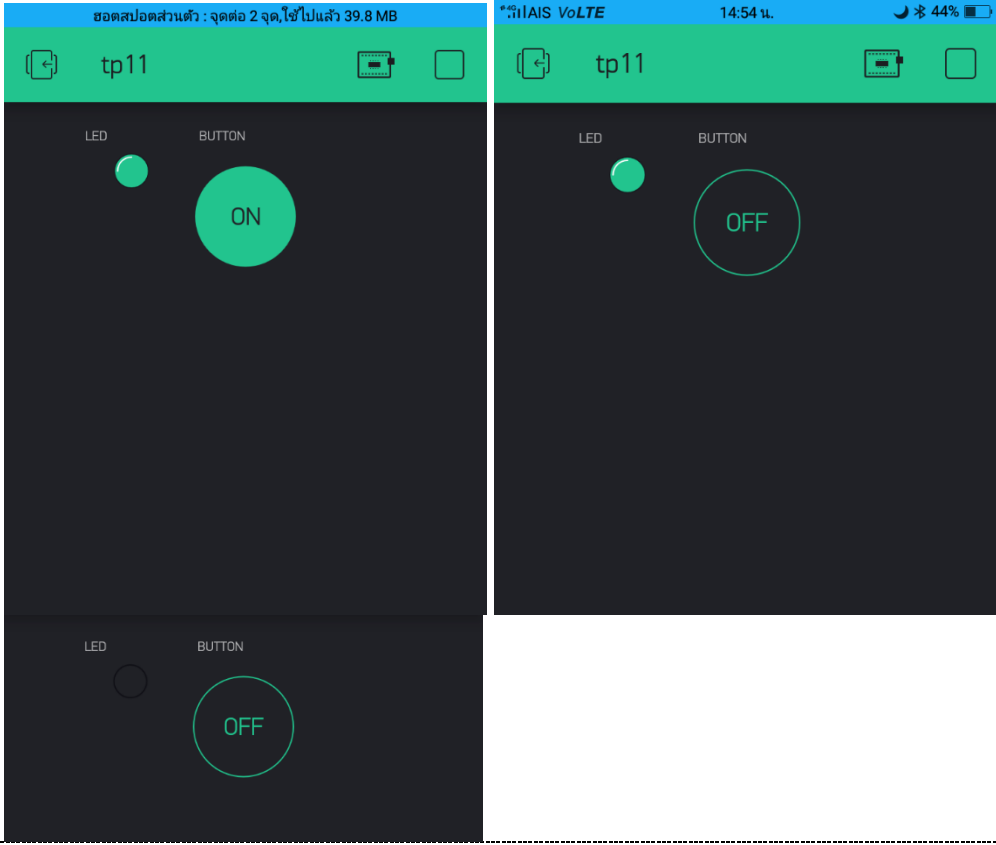
< รูปอุปกรณ์ที่ใช้ทดสอบ ขณะทำการทดสอบ >



< รูปอุปกรณ์ที่ใช้ทดสอบ ขณะทำการทดสอบ >



< หน้าจอ Blynk >



รายละเอียดการทดสอบ

```
#define BLYNK_PRINT Serial
#include <WiFi.h>
#include <WiFiClient.h>
#include <BlynkSimpleEsp32.h>
char auth[] = "WPa2mtFT_6qi9RGfh3Nxc8AXs-y1fnoe";
char ssid[] = "B6226718";
char pass[] = "pungpingjerry";
#define testLED 2
#define testSW 0
WidgetLED LED_V4(V4);
BLYNK_WRITE(V0) {
  int Value_V0 = param.asInt();
  digitalWrite(testLED, Value_V0);
}
void setup() {
  Serial.begin(115200);
```

```

pinMode(testLED, OUTPUT);
pinMode(testSW, INPUT_PULLUP);
Blynk.begin(auth, ssid, pass);
}

int loopCount = 10;
void loop() {
  Blynk.run();
  if (loopCount < 0) {
    loopCount = 20;
    //int stsTestSW = digitalRead(testSW);
    int stsTestSW = random(2);
    Serial.println("stsTestSW = " + String(stsTestSW));
    if (stsTestSW == 0)
      LED_V4.off();
    else
      LED_V4.on();
  }
  delay(100);
  loopCount--;
}

```

#### < โปรแกรมทดสอบ >

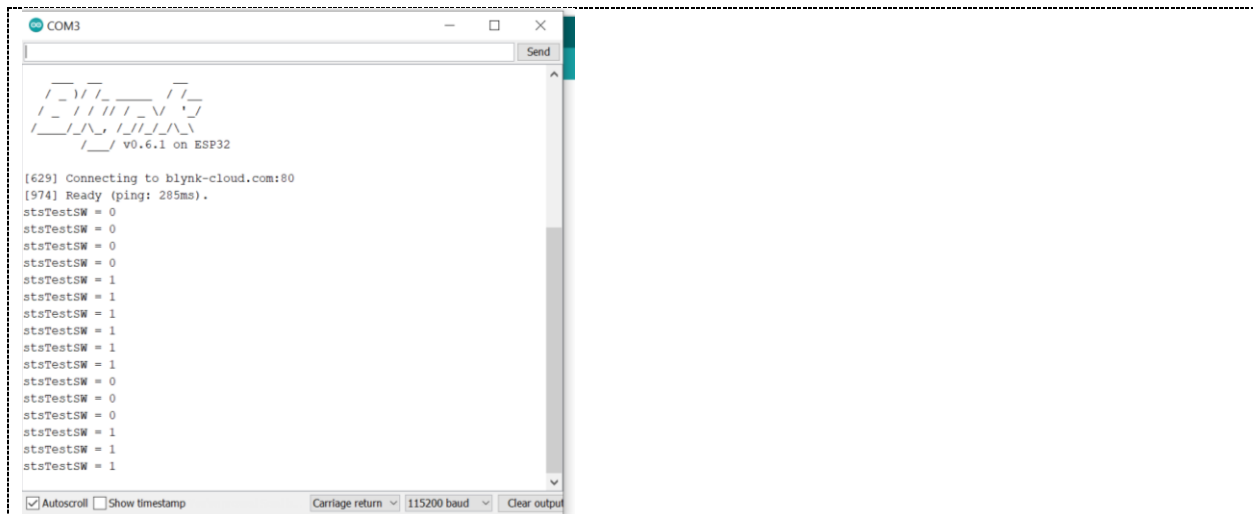


```

1 #define BLYNK_PRINT Serial
2 #include <WiFi.h>
3 #include <WiFiClient.h>
4 #include <BlynkSimpleEsp32.h>
5 char auth[] = "WPa2mtFT_6qi9RGfh3Nxc8AXs-yifnoe";
6 char ssid[] = "B6226718";
7 char pass[] = "pungpingjerry";
8 #define testLED 2
9 #define testSW 0
10 #define LED_V4 V4;
11 BLYNK_WRITE(V0) {
12   int Value_V0 = param.asInt();
13   digitalWrite(testLED, Value_V0);
14 }
15 void setup() {
16   Serial.begin(115200);
17   pinMode(testLED, OUTPUT);
18   pinMode(testSW, INPUT_PULLUP);
19   Blynk.begin(auth, ssid, pass);
20 }
21 int loopCount = 10;
22 void loop() {
23   Blynk.run();
24   if (loopCount < 0) {
25     loopCount = 20;
26     //int stsTestSW = digitalRead(testSW);

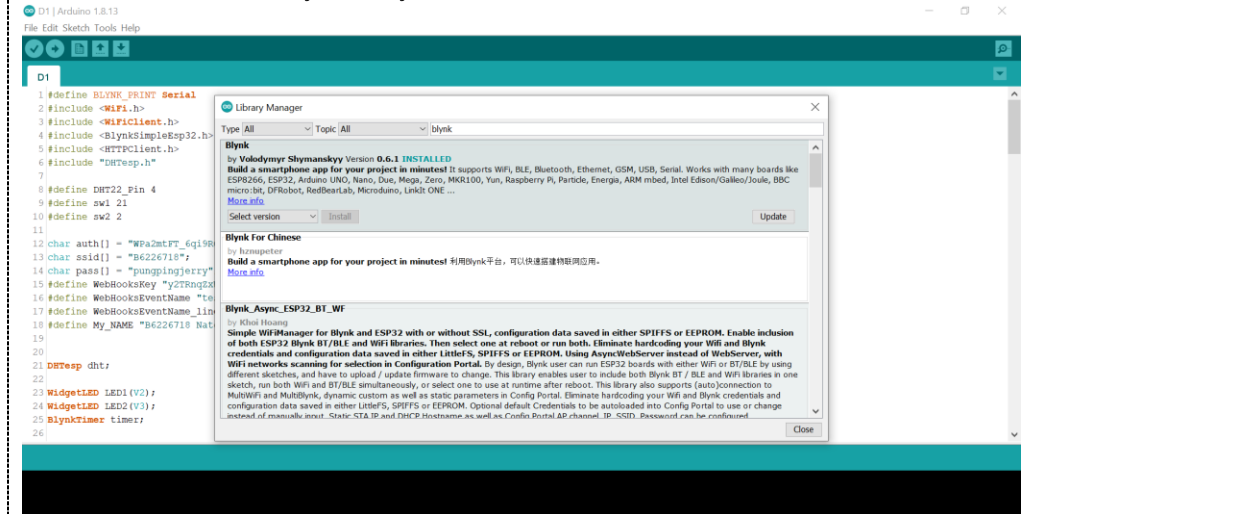
```

#### < ผลการทดสอบ >



อื่นๆ

### 1. ตรวจสอบให้แน่ใจว่ามี Blynk Library เรียบร้อยแล้ว

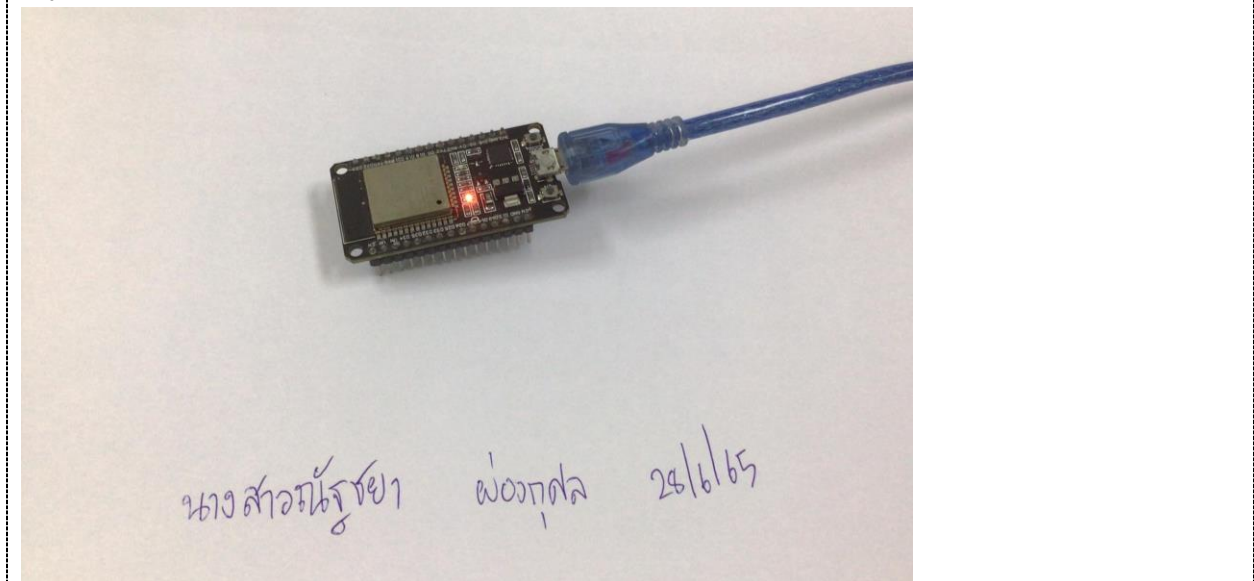


## Quiz\_402 – test Ubidot with ESP32

< รูปอุปกรณ์ที่ใช้ทดสอบ ขณะทำการทดสอบ >

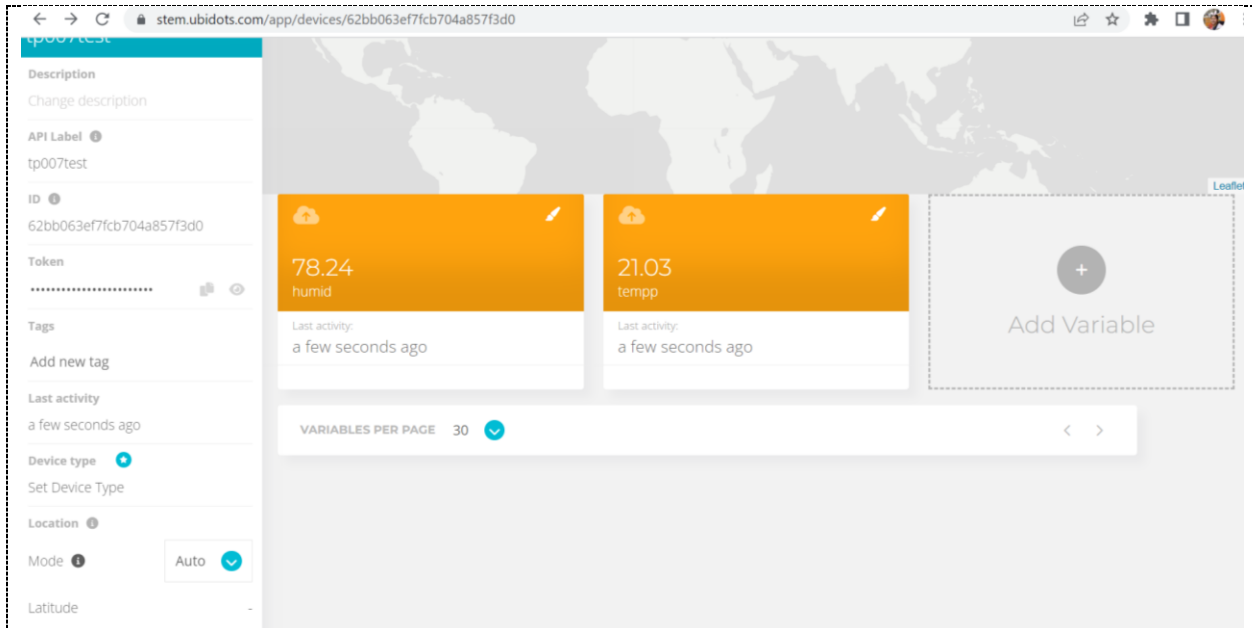


< รูปอุปกรณ์ที่ใช้ทดสอบ ขณะทำการทดสอบ >



< รูปหน้าจอ Ubidots >





### รายละเอียดการทดสอบ

```
COM3
Send
Message arrived [/v2.0/devices/tp007test/humid] {"value": 63.54, "timestamp": 1466122257}
Message arrived [/v2.0/devices/tp007test/temp] {"value": 21.7, "timestamp": 1466122257}
ets Jun  8 2016 00:22:57

rst:0x1 (POWERON_RESET),boot:0x13 (SPI_FAST_FLASH_BOOT)
configsip: 0, SPIWP:0xee
clk_drv:0x00,q_drv:0x00,d_drv:0x00,cs0_drv:0x00,hd_drv:0x00,wp_drv:0x00
mode:DIO, clock div:1
load:0x3fff0018,len:4
load:0x3fff001c,len:1044
load:0x40078000,len:8896
load:0x40080400,len:5816
entry 0x400806ac

Connecting to B6226718
WiFi connected
IP address:
192.168.43.113
Attempting MQTT connection...connected
Message arrived [/v2.0/devices/tp007test/humid] {"value": 63.54, "timestamp": 1466122257}
Message arrived [/v2.0/devices/tp007test/temp] {"value": 21.7, "timestamp": 1466122257}
Publish message: { "humid" : 65.79, "temp": 22.85}
```

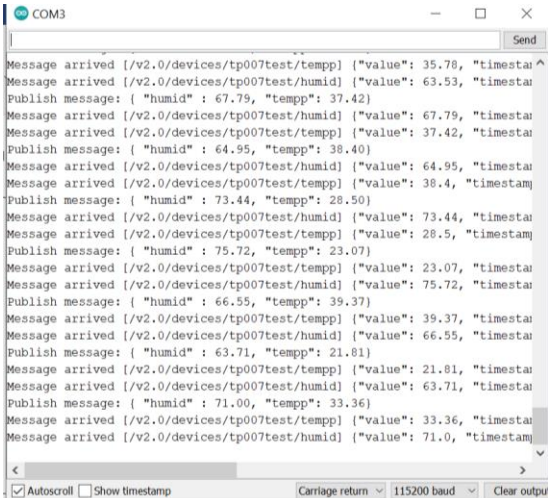
< โปรแกรมทดสอบ >

```

D1 | Arduino 1.8.13
File Edit Sketch Tools Help
D1
37 } else
38 { Serial.print("failed, rc=");
39 Serial.print(client.state());
40 Serial.println(" try again in 5 seconds");
41 delay(5000);
42 }
43 }
44 }
45 void callback(char* topic, byte* payload, unsigned int length)
46 { Serial.print("Message arrived [");
47 Serial.print(topic);
48 Serial.print("] ");
49 for (int i = 0; i < length; i++)
50 { Serial.print((char)payload[i]);
51 }
52 Serial.println();
53 }
54 void setup()
55 { Serial.begin(115200);
56 Setup_Wifi();
57 client.setServer(MQTT_Server, MQTT_Port);
58 client.setCallback(callback);
59 }
60 void loop()
61 { if (!client.connected()) reconnect();
62 client.loop();

```

### < ผลการทดสอบ >



The screenshot shows a serial monitor window titled "COM3" with a "Send" button. It displays a series of MQTT messages received from a device. The messages are formatted as JSON objects containing "value" and "timestamp" fields. The topics are either "/v2.0/devices/tp007test/tempp" or "/v2.0/devices/tp007test/humid". The messages alternate between these two topics, showing temperature and humidity readings respectively. At the bottom of the window, there are checkboxes for "Autoscroll" and "Show timestamp", and a dropdown menu for "Carriage return" set to "115200 baud".

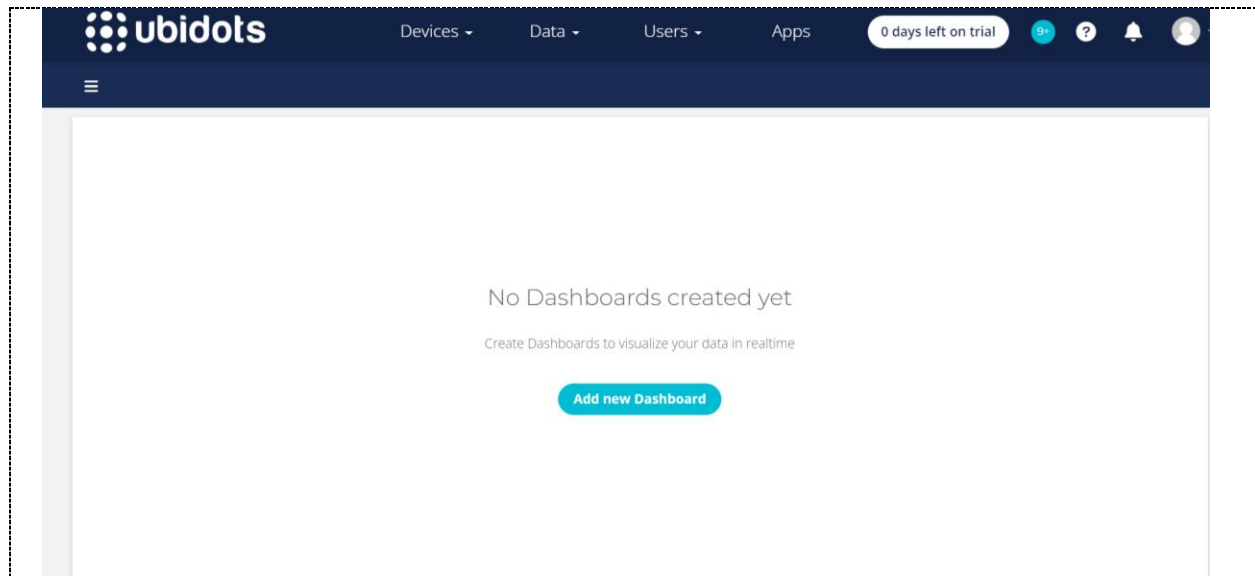
```

Message arrived [/v2.0/devices/tp007test/tempp] {"value": 35.78, "timestamp": 35.78}
Message arrived [/v2.0/devices/tp007test/humid] {"value": 63.53, "timestamp": 63.53}
Publish message: { "humid" : 67.79, "temp": 37.42}
Message arrived [/v2.0/devices/tp007test/humid] {"value": 67.79, "timestamp": 67.79}
Message arrived [/v2.0/devices/tp007test/tempp] {"value": 37.42, "timestamp": 37.42}
Publish message: { "humid" : 64.95, "temp": 38.40}
Message arrived [/v2.0/devices/tp007test/humid] {"value": 64.95, "timestamp": 64.95}
Message arrived [/v2.0/devices/tp007test/tempp] {"value": 38.4, "timestamp": 38.4}
Publish message: { "humid" : 73.44, "temp": 28.50}
Message arrived [/v2.0/devices/tp007test/humid] {"value": 73.44, "timestamp": 73.44}
Message arrived [/v2.0/devices/tp007test/tempp] {"value": 28.5, "timestamp": 28.5}
Publish message: { "humid" : 75.72, "temp": 23.07}
Message arrived [/v2.0/devices/tp007test/tempp] {"value": 23.07, "timestamp": 23.07}
Message arrived [/v2.0/devices/tp007test/humid] {"value": 75.72, "timestamp": 75.72}
Publish message: { "humid" : 66.55, "temp": 39.37}
Message arrived [/v2.0/devices/tp007test/tempp] {"value": 39.37, "timestamp": 39.37}
Message arrived [/v2.0/devices/tp007test/humid] {"value": 66.55, "timestamp": 66.55}
Publish message: { "humid" : 63.71, "temp": 21.81}
Message arrived [/v2.0/devices/tp007test/tempp] {"value": 21.81, "timestamp": 21.81}
Message arrived [/v2.0/devices/tp007test/humid] {"value": 63.71, "timestamp": 63.71}
Publish message: { "humid" : 71.00, "temp": 33.36}
Message arrived [/v2.0/devices/tp007test/tempp] {"value": 33.36, "timestamp": 33.36}
Message arrived [/v2.0/devices/tp007test/humid] {"value": 71.0, "timestamp": 71.0}

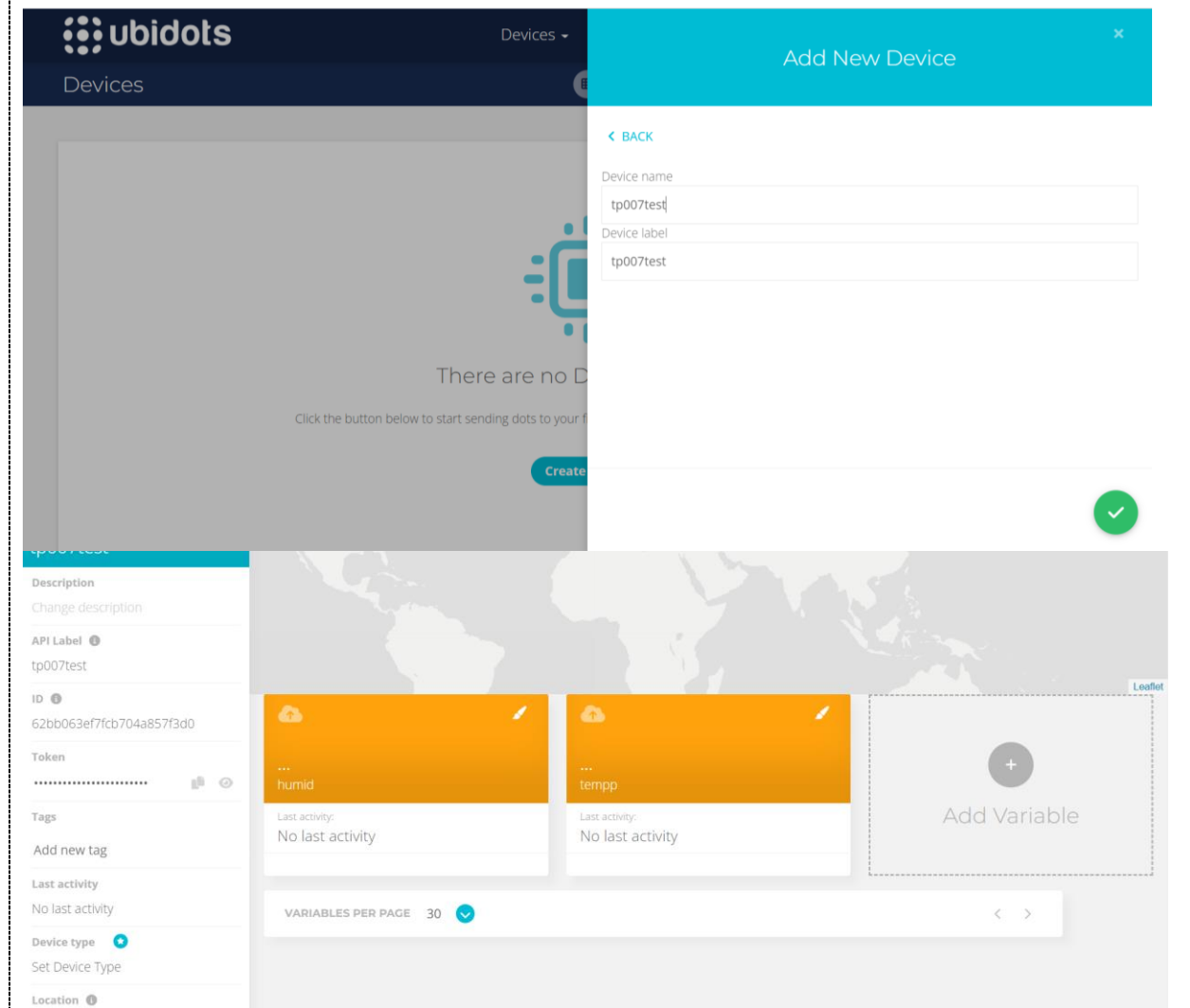
```

อื่นๆ

### 1. Signed Up (or Signed In)



## 2. Create Device and Variable



The screenshot displays two windows. The top window is the Arduino IDE, showing a sketch for an ESP8266 module. The sketch includes MQTT client code for sending and receiving data. The bottom window is a web browser showing the Ubidots IoT platform interface. The interface includes a sidebar with the Ubidots logo and navigation links. The main content area shows a world map and a list of devices. The selected device, 'tp007test', is shown with its details, including the API label, ID, token, and tags. The device's last activity is shown as 'No last activity' for both 'humid' and 'tempp' variables. A dropdown menu is open on the right side of the browser window, showing options like 'Username: natchayap', 'My Profile', 'API Credentials', 'Docs', and 'Log out'.

```

37 } else
38 { Serial.print("failed, rc=");
39 Serial.print(client.state());
40 Serial.println(" try again in 5 seconds");
41 delay(5000);
42 }
43 }
44 }
45 void callback(char* topic, byte* payload, unsigned int length)
46 { Serial.print("Message arrived [");
47 Serial.print(topic);
48 Serial.print("] ");
49 for (int i = 0; i < length; i++)
50 { Serial.print((char)payload[i]);
51 }
52 Serial.println();
53 }
54 void setup()
55 { Serial.begin(115200);
56 Setup_Wifi();
57 client.setServer(MQTT_Server, MQTT_Port);
58 client.setCallback(callback);
59 }
60 void loop()
61 { if (!client.connected()) reconnect();
62 client.loop();
  
```

COM3

```

Message arrived [/v2.0/devices/tp007test/humid] {"value": 63.54, "timesta
Message arrived [/v2.0/devices/tp007test/tempp] {"value": 21.7, "timesta
ets Jun  8 2016 00:22:57

rst:0x1 (POWERON_RESET),boot:0x13 (SPI_FAST_FLASH_BOOT)
configsip: 0, SPIWP:0xee
clk_drv:0x00,q_drv:0x00,d_drv:0x00,cs0_drv:0x00,hd_drv:0x00,wp_drv:0x00
mode:DIO, clock div:1
load:0x3fff0018,len:4
load:0x3fff001c,len:1044
load:0x40078000,len:8896
load:0x40080400,len:5816
entry 0x400806ac

Connecting to B6226718
.
WiFi connected
IP address:
192.168.43.113
Attempting MQTT connection...connected
Message arrived [/v2.0/devices/tp007test/humid] {"value": 63.54, "timesta
Message arrived [/v2.0/devices/tp007test/tempp] {"value": 21.7, "timesta
Publish message: { "humid" : 65.79, "temp": 22.85}
  
```

Leaving...  
Hard resetting via RTS pin...

ubidots

Devices Data

← Devices

tp007test

Description  
Change description

API Label  
tp007test

ID  
62bb063ef7cb704a857f3d0

Token  
.....

Tags  
Add new tag

Last activity

humid

Last activity:  
No last activity

tempp

Last activity:  
No last activity

Add Variable

Username:  
natchayap

My Profile

API Credentials

Docs

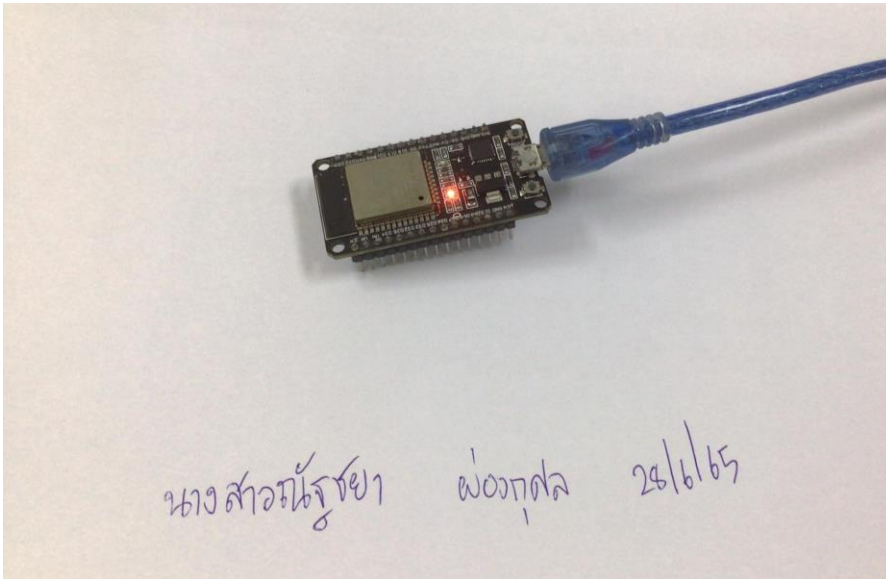
Log out

### Quiz\_403 – Modbus RTU/ASCII/TCP with Ubidots IoTs Platform

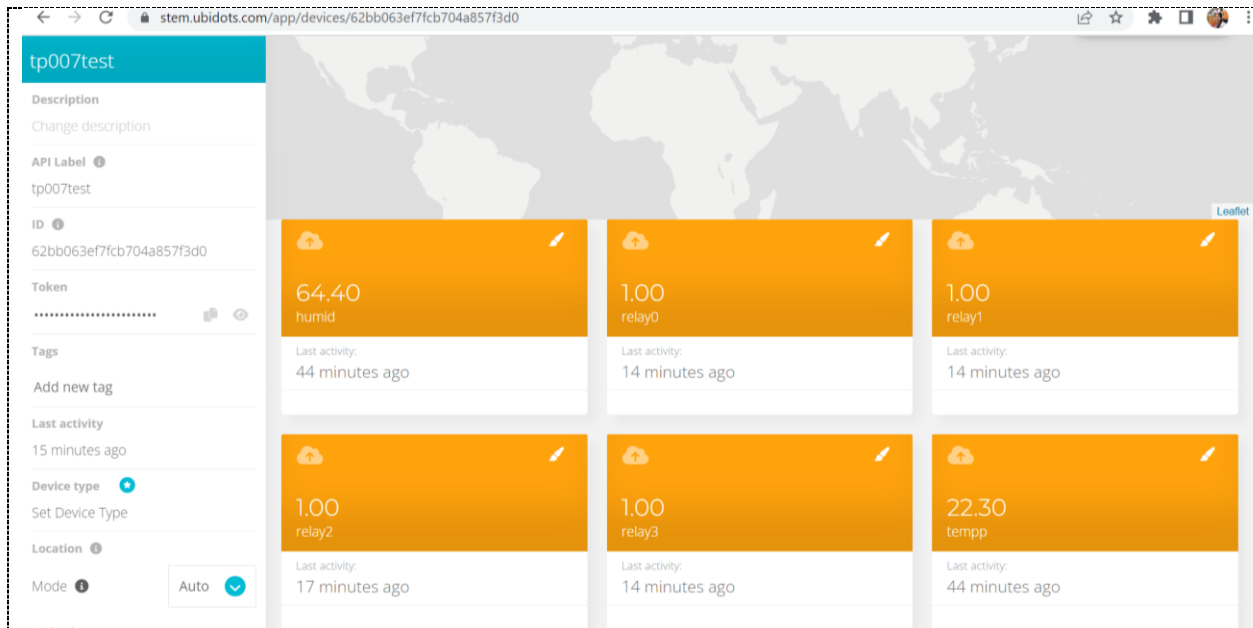
< รูปอุปกรณ์ที่ใช้ทดสอบ ขณะทำการทดสอบ >



< รูปอุปกรณ์ที่ใช้ทดสอบ ขณะทำการทดสอบ >



< รูปหน้าจอ Ubidots >



รายละเอียดการทดสอบ

```
#include <WiFi.h>
#include <PubSubClient.h>
#include <ModbusMaster.h>
#define MAX485_Monitor 2
#define MAX485_Ctrl 5 // Pin Ctrl 1=Tx and 0=Rx_NEG
#define MAX485_Rx 16 // Pin RXD2 16
#define MAX485_Tx 17 // Pin TXD2 17
#define Slave_ID 5 // Slave ID
ModbusMaster node; // instantiate ModbusMaster object
const char *My_SSID = "B6226718";
const char *My_Pass = "pungpingjerry";
const char *MQTT_Server = "things.ubidots.com";
const char *MQTT_User = "BBFF-dZWJxFOz4nw2XpLm18D4tsNLUi3M1G";
const char *MQTT_Pass = "BBFF-dZWJxFOz4nw2XpLm18D4tsNLUi3M1G";
const char *PTopic1 = "/v2.0/devices/tp007test";
const char *STopic1 = "/v2.0/devices/tp007test/relay0";
const char *STopic2 = "/v2.0/devices/tp007test/relay1";
const char *STopic3 = "/v2.0/devices/tp007test/relay2";
const char *STopic4 = "/v2.0/devices/tp007test/relay3";
#define MQTT_Port 1883
#define testLED 2
```

```

int stsLED = 0;
WiFiClient espClient;
PubSubClient client(espClient);
long lastMsg = 0;
char msg[50];
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);
    } 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]);
  }

  int RlyID = (int)topic[29] - 0x30; // '0'
  int RlySts = (int)payload[10] - 0x30; // '0'
  Serial.println("\nRlyID-" + (String)(RlyID) + " >> RlyStatus-" + (String)(RlySts));
  node.writeSingleCoil(RlyID, RlySts);
}

void preTransmission() {
  digitalWrite(MAX485_Monitor, 1);
  digitalWrite(MAX485_Ctrl, 1);
}

void postTransmission() {
  digitalWrite(MAX485_Monitor, 0);
  digitalWrite(MAX485_Ctrl, 0);
}

void setup()
{
  pinMode(testLED, OUTPUT);
  pinMode(MAX485_Monitor, OUTPUT);
  pinMode(MAX485_Ctrl, OUTPUT);
  postTransmission(); // Init in receive mode
  Serial.begin(115200);
  Serial2.begin(9600, SERIAL_8N1, MAX485_Rx, MAX485_Tx);
  node.begin(Slave_ID, Serial2); // Modbus slave ID Setting
  // Callbacks allow us to configure the RS485 transceiver correctly
  node.preTransmission(preTransmission);
  node.postTransmission(postTransmission);
  Setup_Wifi();
  client.setServer(MQTT_Server, MQTT_Port);
  client.setCallback(callback);
}

```



```

}

void loop()
{ if (!client.connected()) reconnect();
  client.loop();
  long now = millis();
  if (now - lastMsg > 5000)
  { lastMsg = now;
    digitalWrite(testLED, stsLED);
    stsLED = 1 - stsLED;
  }
}

```

< โปรแกรมทดสอบ >

The screenshot displays the Ubidots web application interface for a device named 'tp007test'. The interface includes a sidebar with device details and a main area showing a grid of sensor data points. The sidebar details include the device description, API label, ID, token, tags, last activity, device type, location, and mode. The main area shows a grid of six data points, each with a value, a unit, and a last activity timestamp.

Device Name	Description	API Label	ID	Token	Tags	Last activity	Device type	Location	Mode
tp007test	Change description	tp007test	62bb063ef7fcb704a857f3d0	.....	Add new tag	15 minutes ago	Set Device Type	Set Device Type	Auto

Sensor	Value	Unit	Last activity
humid	64.40		44 minutes ago
relay0	1.00		14 minutes ago
relay1	1.00		14 minutes ago
relay2	1.00		17 minutes ago
relay3	1.00		14 minutes ago
tempp	22.30		44 minutes ago

< ผลการทดสอบ >

COM3

Send

Connecting to B6226718  
.  
WiFi connected  
IP address:  
192.168.43.113  
Attempting MQTT connection...connected  
ets Jun 8 2016 00:22:57  
  
rst:0x1 (POWERON\_RESET),boot:0x13 (SPI\_FAST\_FLASH\_BOOT)  
configsip: 0, SPIWP:0xee  
clk\_drv:0x00,q\_drv:0x00,d\_drv:0x00,cs0\_drv:0x00,hd\_drv:0x00,wp\_drv:0x00  
mode:DIO, clock div:1  
load:0x3fff0018,len:4  
load:0x3fff001c,len:1044  
load:0x40078000,len:8896  
load:0x40080400,len:5816  
entry 0x400806ac  
  
Connecting to B6226718  
.  
WiFi connected  
IP address:  
192.168.43.113  
Attempting MQTT connection...connected

☒ Autoscroll ☐ Show timestamp Carriage return 115200 baud Clear output

ubidots

Devices Data

007Dashboard

Jun 27 2022 22:12 - Now

Indicator widget

relay0 (tp007test)

On

Indicator widget

relay1 (tp007test)

On

Indicator widget

relay2 (tp007test)

On

Indicator widget

relay3 (tp007test)

On

Switch

relay0 (tp007test)

On

Switch

relay1 (tp007test)

On

Switch

relay2 (tp007test)

On

Switch

relay3 (tp007test)

On

ubidots

Devices Data

007Dashboard

Add new widget

Pie

Rose chart

Scatter

Slider

Switch

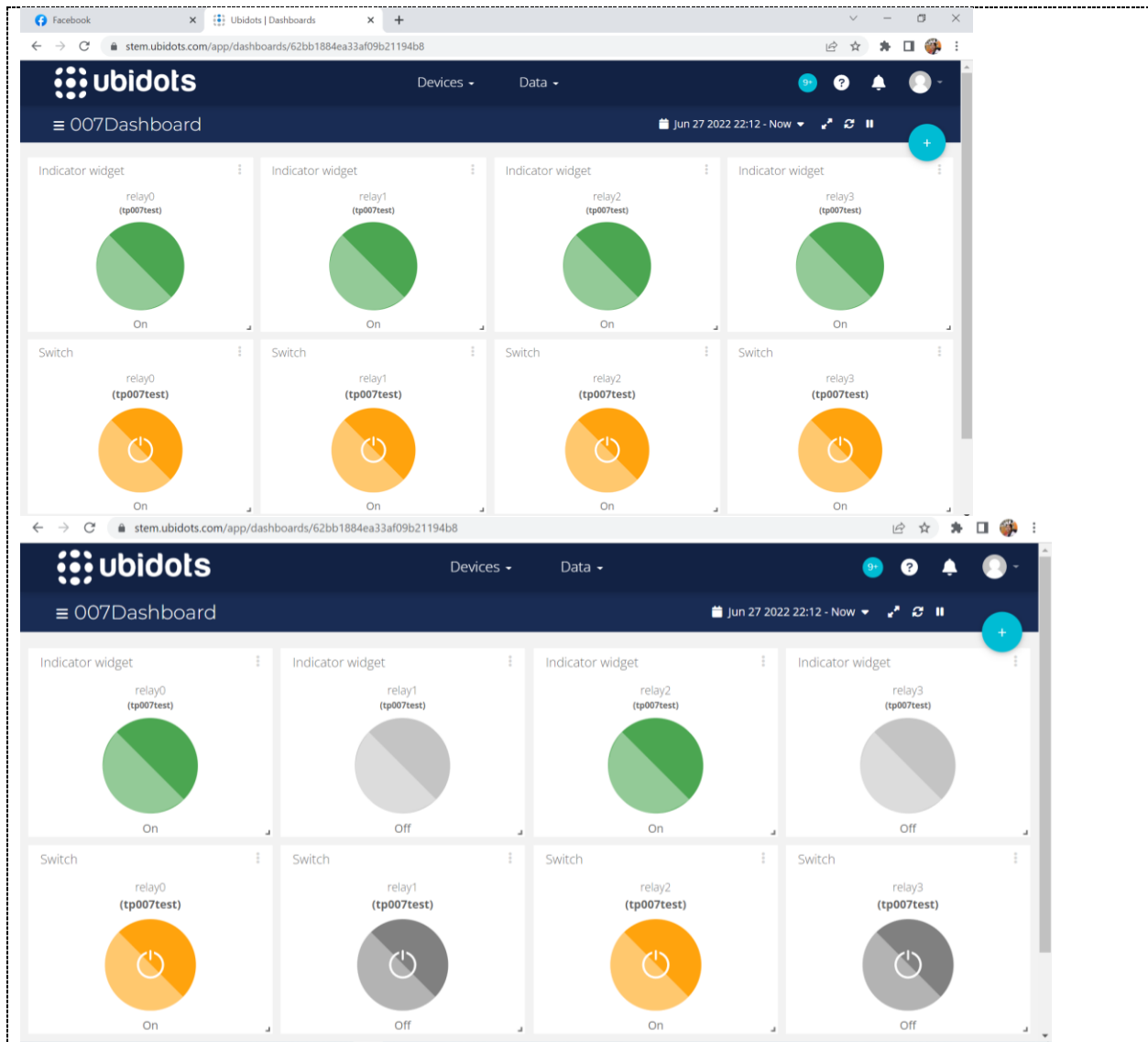
Tank

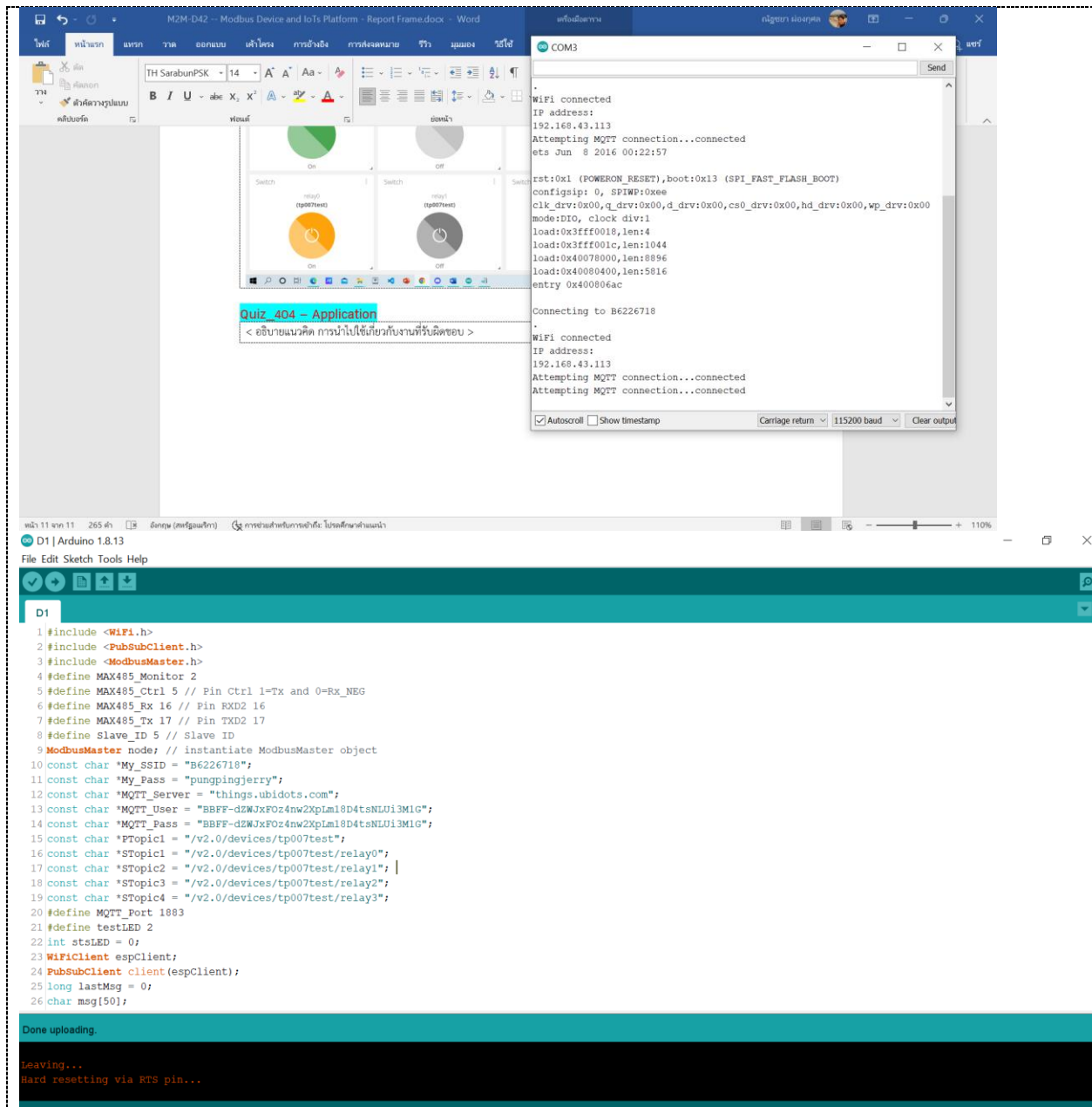
Text

Thermometer

Values table

Variables table





### Quiz\_404 – Application

ยังไม่มีการทริบผิดชอบ แต่สามารถคิดไอเดียเพื่อนำไปต่อยอดได้ คือการออกแบบควบคุมวงจรต่างๆบน dashboard รวมถึงสามารถควบคุมการแสดงผลบนอุปกรณ์จริงผ่าน ubidot ได้