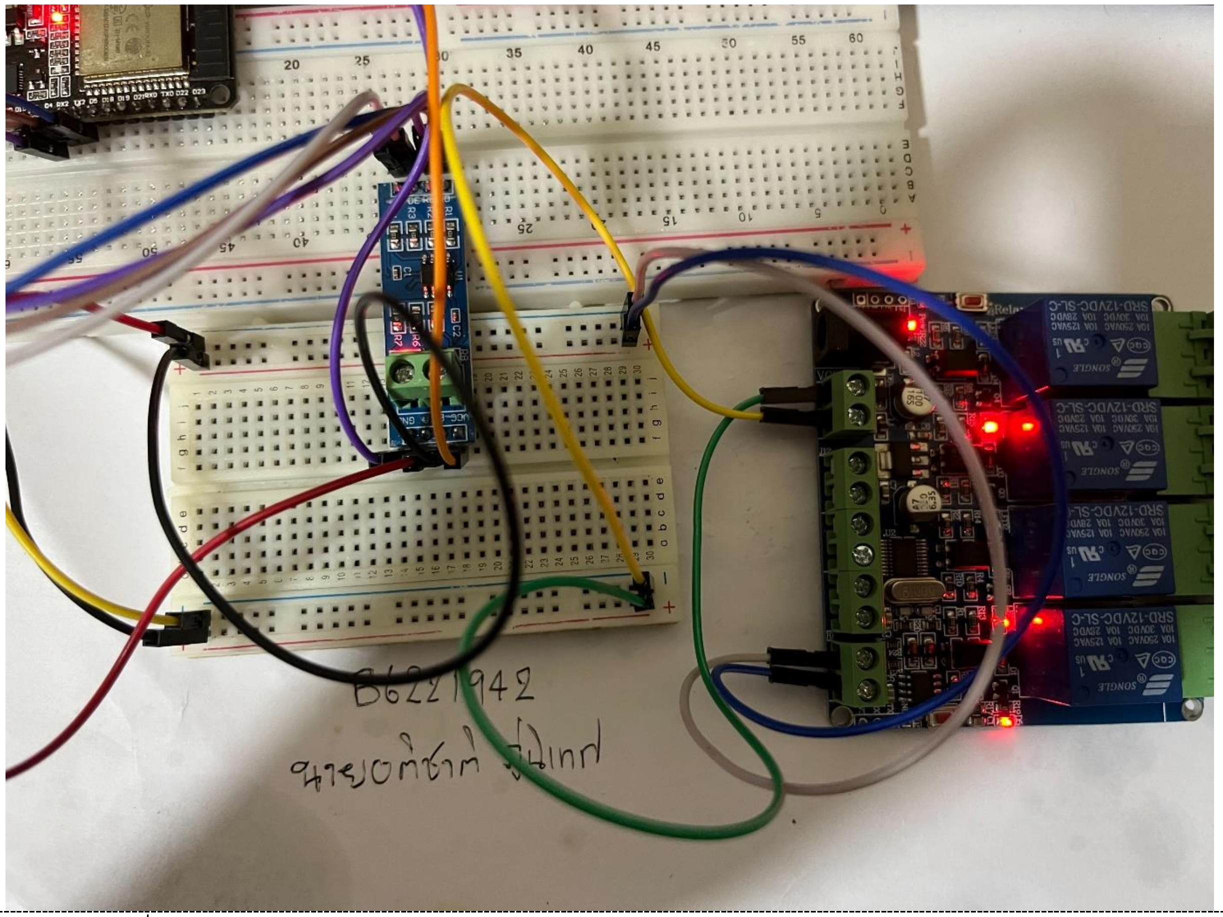
การควบคุมเครื่องจักรอัจฉริยะโดยใช้การสื่อสารระหว่างเครื่องจักรกับเครื่องจักร M2M - Intelligence Machine Control

ขื่อ-สกุล : นายอติชาติ ภู่นิเทศ

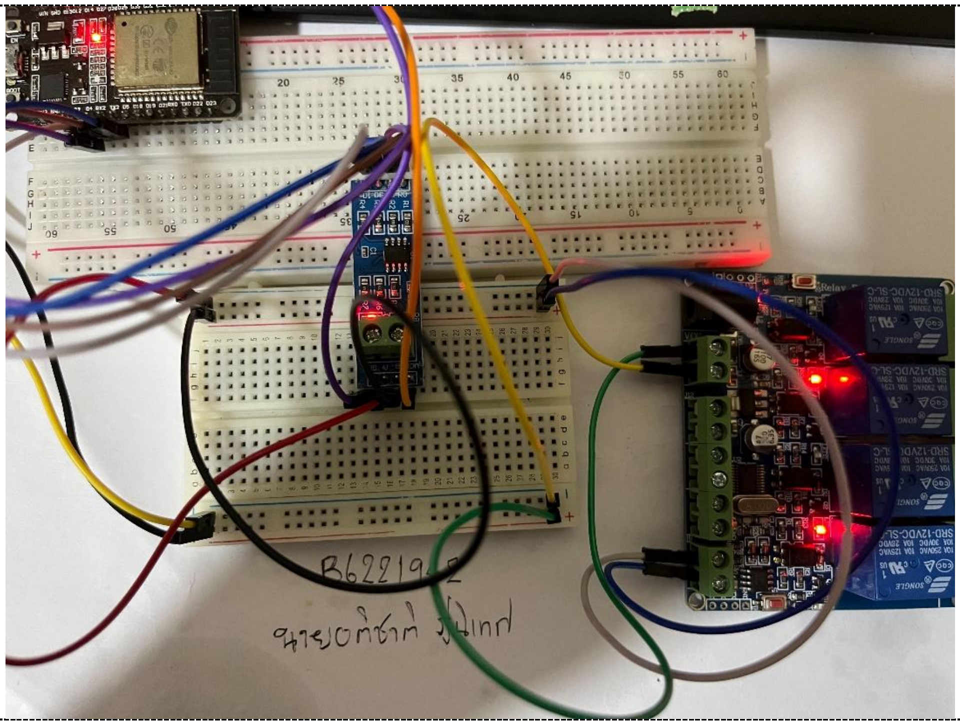
4/4: -- คำถามท้ายบทเพื่อทดสอบความเข้าใจ

Quiz_401 – test Blynk

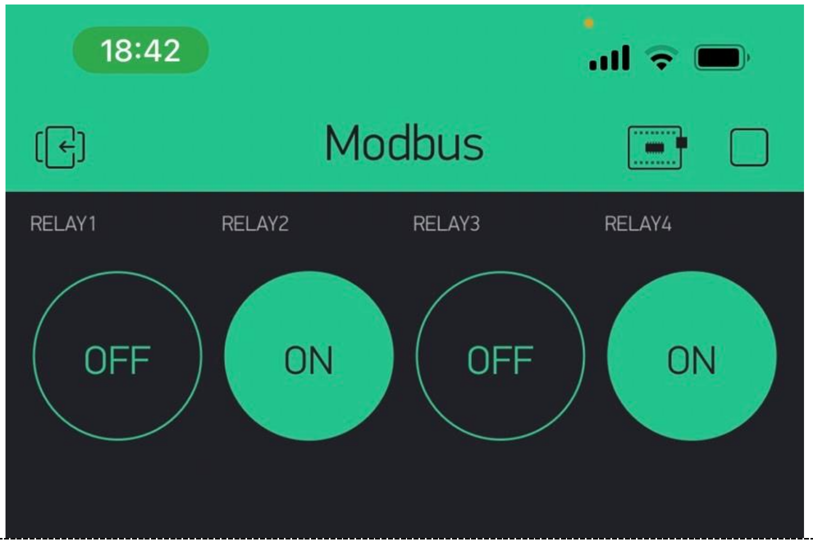
< รูปอุปกรณ์ที่ใช้ทดสอบ ขณะทำการทดสอบ >



< รูปอุปกรณ์ที่ใช้ทดสอบ ขณะทำการทดสอบ >



< รูปหน้าจอ Blynk >



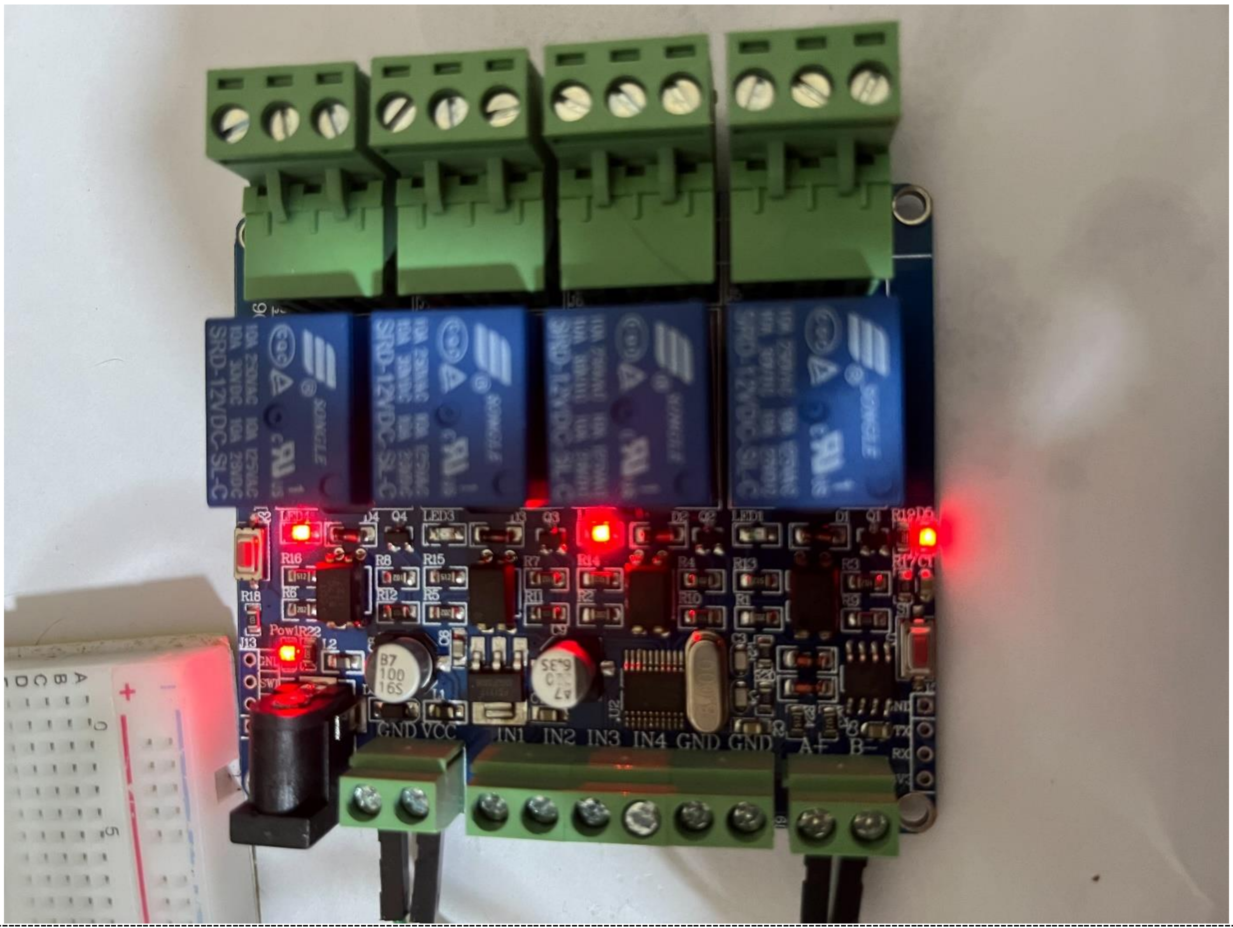
รายยละเอียดการทดสอบ

< โปรแกรมทดสอบ >

```
#define BLYNK_PRINT Serial
#include <WiFi.h>
#include <WiFiClient.h>
#include <BlynkSimpleEsp32.h>
#include "ModbusMaster.h" //https://github.com/4-20ma/ModbusMaster
#define Slave_ID 5
#define MAX485_RE_NEG 4
#define RX_PIN 16
#define TX_PIN 17
ModbusMaster modbus;
```

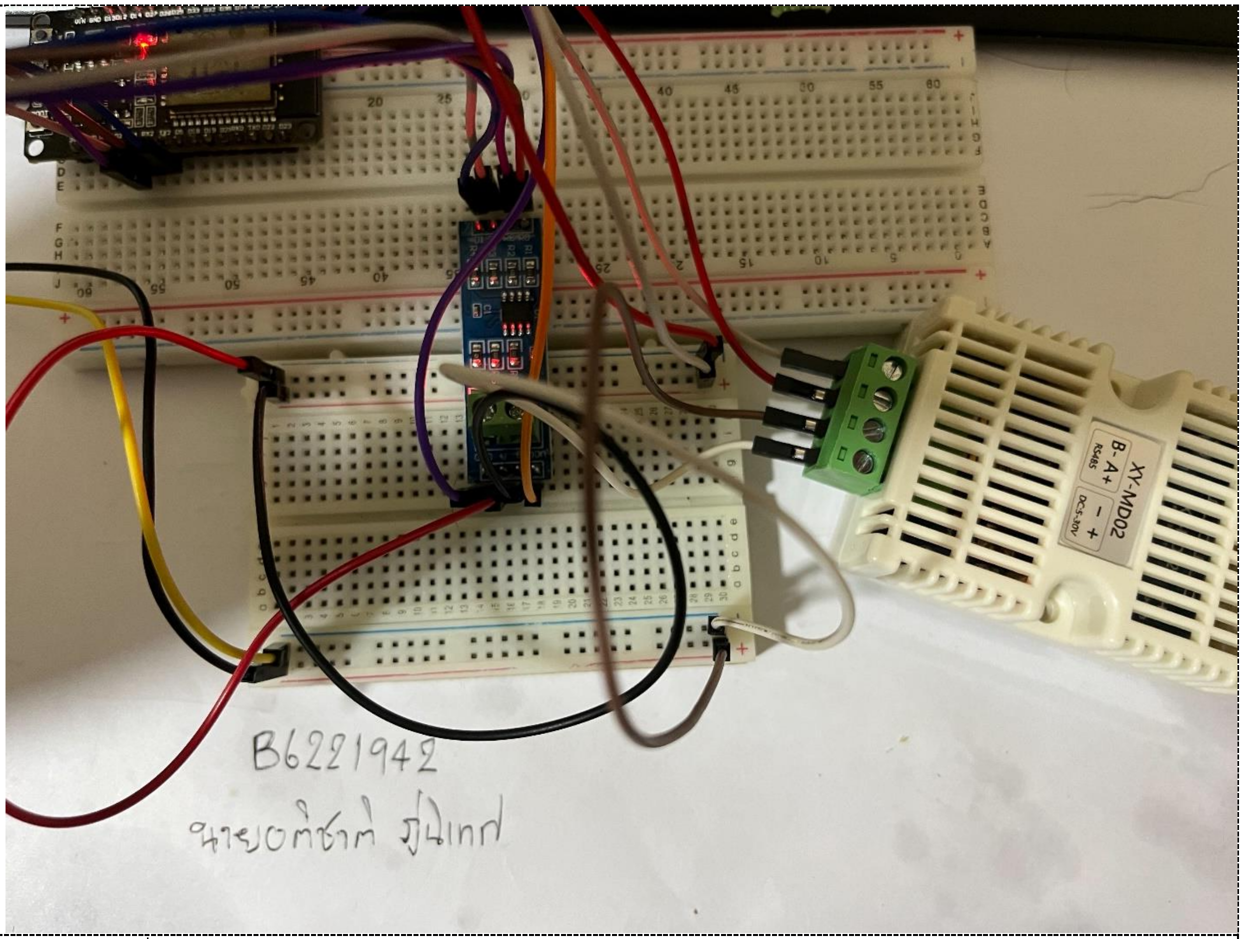
```
char auth[] = "10Dtg7C1TMQgqOjQTNaX0DrSe8uH9YLf"; // Token Key
char ssid[] = "LANTANIDEs-2.4G"; // AP Name
char pass[] = "0887040892"; // Wifi-Password

void preTransmission() {
  digitalWrite(MAX485_RE_NEG, HIGH); //Switch to transmit data
}
void postTransmission() {
  digitalWrite(MAX485_RE_NEG, LOW); //Switch to receive data
}
BLYNK_WRITE (V0) {
  int pinValue = param.asInt();
  if (pinValue == 1) {
    modbus.writeSingleRegister(0, 0x0100);
  }
  else {
    modbus.writeSingleRegister(0, 0x0000);
  }
}
BLYNK_WRITE (V1) {
  int pinValue = param.asInt();
  if (pinValue == 1) {
    modbus.writeSingleRegister(1, 0x0100);
  }
  else {
    modbus.writeSingleRegister(1, 0x0000);
  }
}
BLYNK_WRITE (V2) {
  int pinValue = param.asInt();
  if (pinValue == 1) {
    modbus.writeSingleRegister(2, 0x0100);
  }
  else {
    modbus.writeSingleRegister(2, 0x0000);
  }
}
BLYNK_WRITE (V3) {
  int pinValue = param.asInt();
  if (pinValue == 1) {
    modbus.writeSingleRegister(3, 0x0100);
  }
  else {
    modbus.writeSingleRegister(3, 0x0000);
  }
}
void setup() {
  pinMode(MAX485_RE_NEG, OUTPUT);
  digitalWrite(MAX485_RE_NEG, LOW);
  Serial.begin(115200, SERIAL_8N1);
  Serial2.begin(9600, SERIAL_8N1, RX_PIN, TX_PIN);
  Blynk.begin(auth, ssid, pass);
  modbus.begin(Slave_ID, Serial2);
  modbus.preTransmission(preTransmission);
  modbus.postTransmission(postTransmission);
}
void loop() {
  Blynk.run();

}
```

< ผลการทดสอบ >

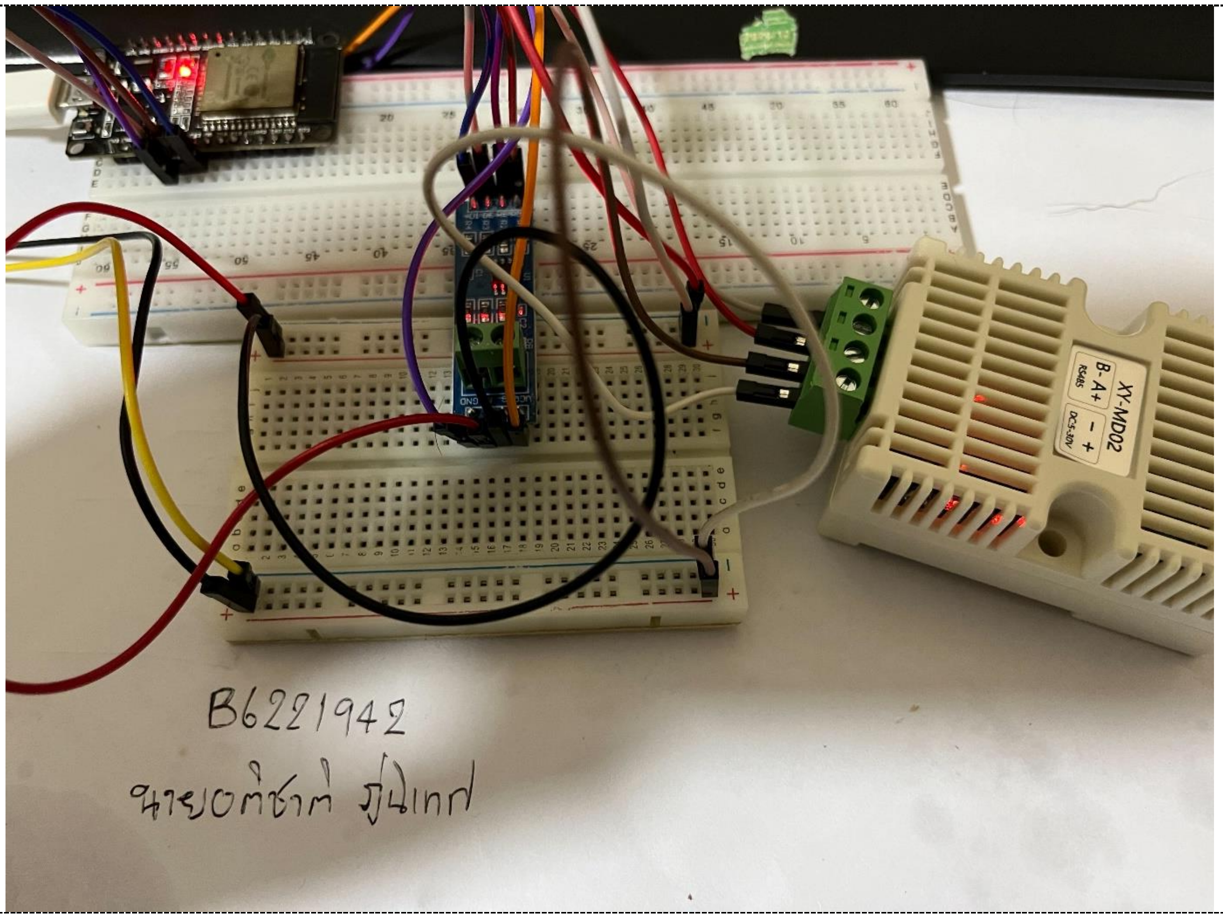


Quiz_402 – test Ubidot with ESP32

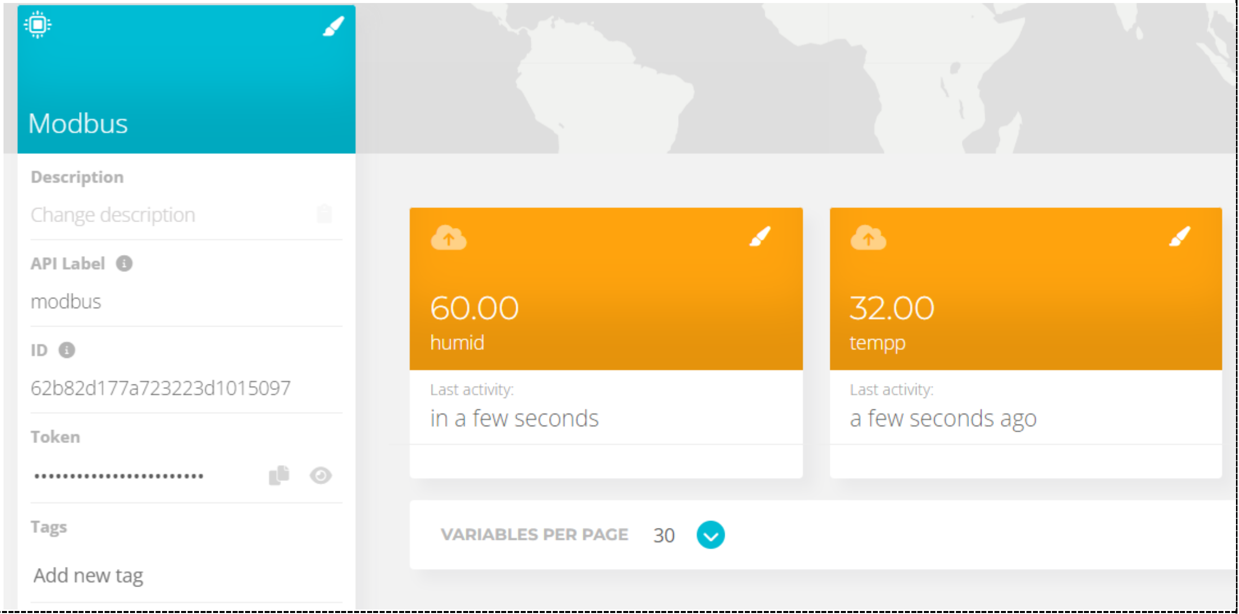
< รูปอุปกรณ์ที่ใช้ทดสอบ ขณะทำการทดสอบ >



< รูปอุปกรณ์ที่ใช้ทดสอบ ขณะทำการทดสอบ >



< รูปหน้าจอ Ubidots >



Description
Change description
API Label
modbus
ID
62b82d177a723223d1015097
Token
••••••••••••••••••••••••
Tags
Add new tag

60.00
humid
Last activity:
in a few seconds

32.00
tempp
Last activity:
a few seconds ago

VARIABLES PER PAGE 30

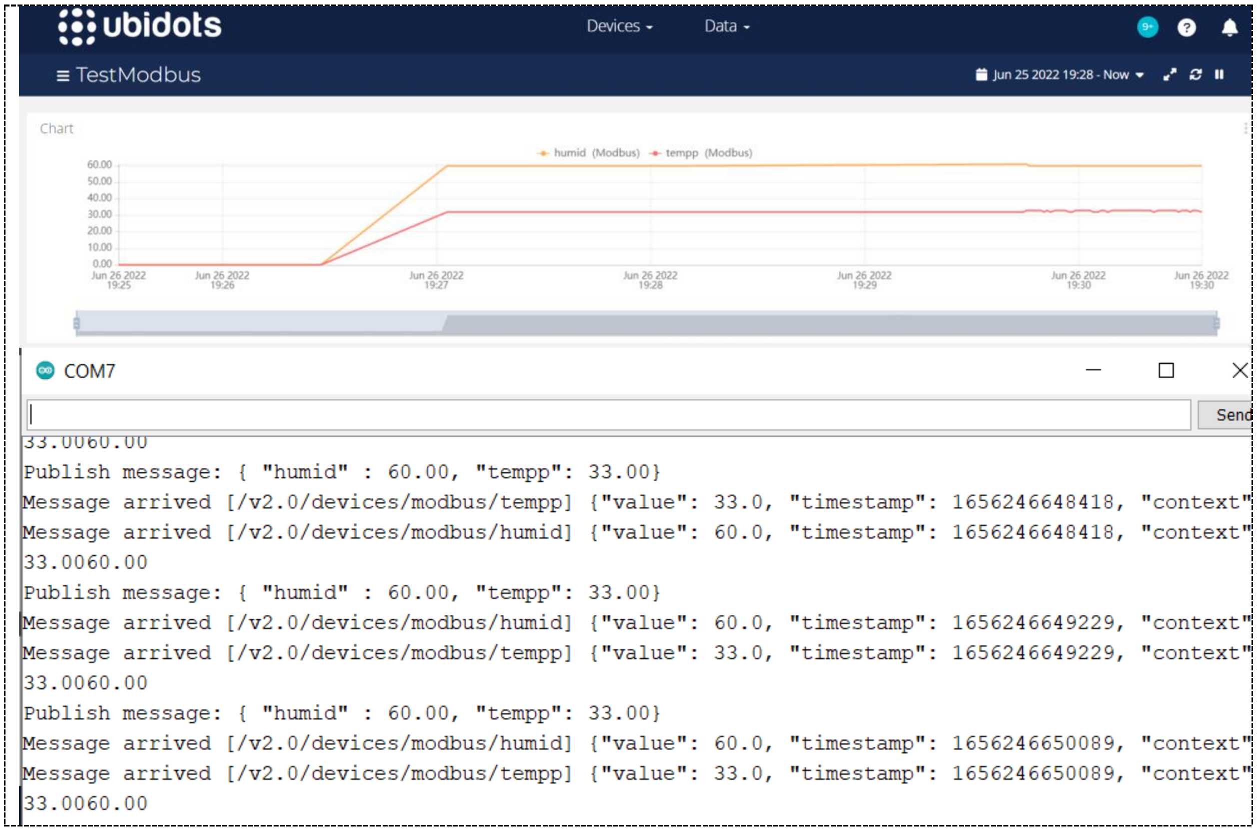
รายยละเอียดการทดสอบ

< โปรแกรมทดสอบ >

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

```
#include <PubSubClient.h>
#include "ModbusMaster.h" //https://github.com/4-20ma/ModbusMaster
#define Slave_ID 1
#define MAX485_RE_NEG 4
#define RX_PIN 16
#define TX_PIN 17
ModbusMaster modbus;

const char *My_SSID = "LANTANIDEs-2.4G";
const char *My_Pass = "0887040892";
const char *MQTT_Server = "things.ubidots.com";
const char *MQTT_User = "BBFF-oV2r6ePhvvQl4VX0b10BrF273YKUuE";
const char *MQTT_Pass = "BBFF-oV2r6ePhvvQl4VX0b10BrF273YKUuE";
const char *PTopic1 = "/v2.0/devices/modbus";
const char *STopic1 = "/v2.0/devices/modbus/humid";
const char *STopic2 = "/v2.0/devices/modbus/tempp";
#define MQTT_Port 1883
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);
    } 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]);
  }
  Serial.println();
}
void preTransmission() {
  digitalWrite(MAX485_RE_NEG, HIGH); //Switch to transmit data
}
void postTransmission() {
  digitalWrite(MAX485_RE_NEG, LOW); //Switch to receive data
}
void setup()
{ Serial.begin(115200);
  pinMode(MAX485_RE_NEG, OUTPUT);
  digitalWrite(MAX485_RE_NEG, LOW);
  Serial2.begin(9600, SERIAL_8N1, RX_PIN, TX_PIN);
  modbus.begin(Slave_ID, Serial2);
  modbus.preTransmission(preTransmission);
  modbus.postTransmission(postTransmission);
Setup_Wifi();
client.setServer(MQTT_Server, MQTT_Port);
client.setCallback(callback);
}

long lastMillis = 0;
void loop() {
  if (!client.connected()) reconnect();
  client.loop();
  long currentMillis = millis();
  if (currentMillis - lastMillis > 1000) {
    uint8_t result = modbus.readInputRegisters(1, 2);
    double res1 = modbus.getResponseBuffer(0) / 10;
    float xTempp = float(res1);
    double res2 = modbus.getResponseBuffer(1) / 10;
    float xHumid = float(res2);
    Serial.println(String(xTempp)+String(xHumid));
    snprintf (msg, 75, "{ \"humid\" : %5.2f, \"tempp\": %5.2f}", xHumid, xTempp);
    Serial.print("Publish message: ");
    Serial.println(msg);
    client.publish(PTopic1, msg);
    lastMillis = currentMillis;
  }
}
```

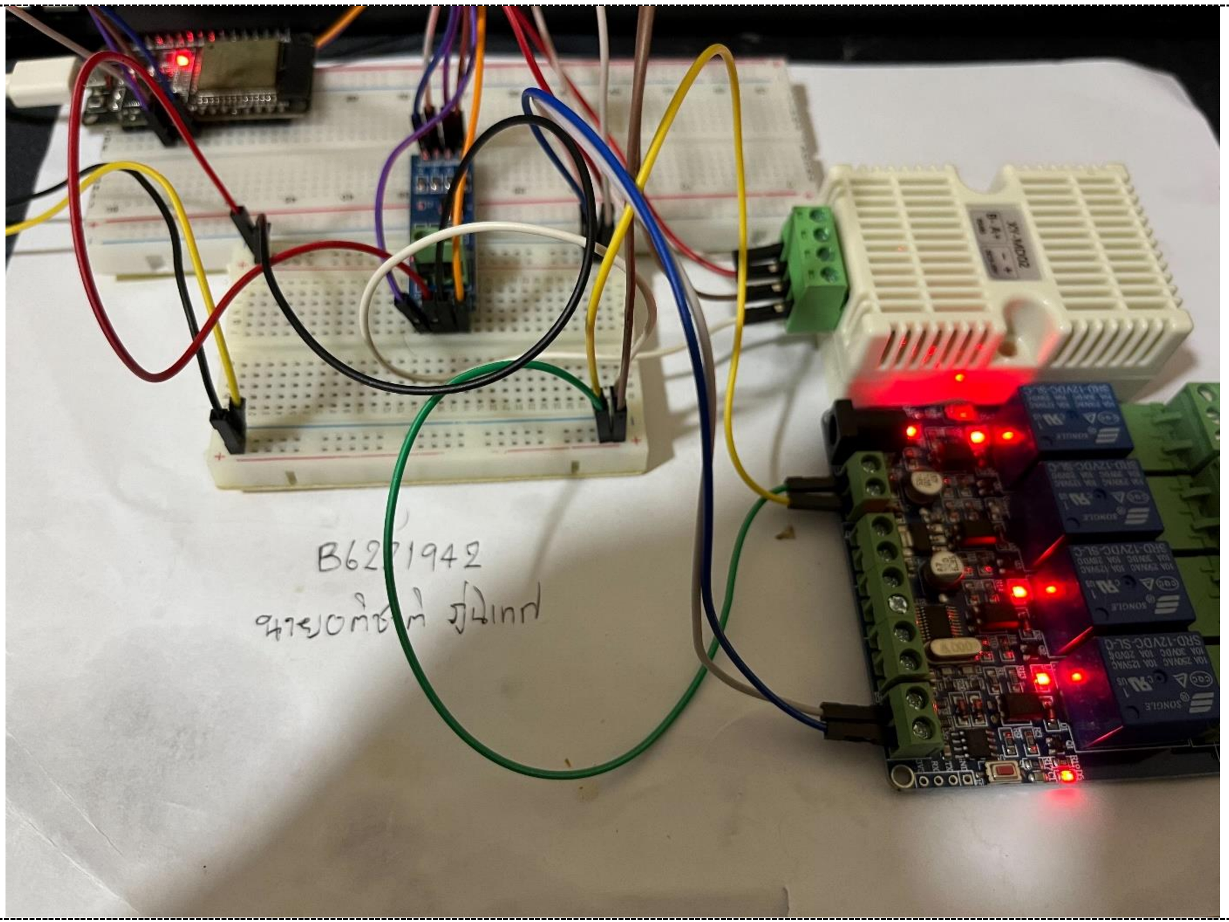
< ผลการทดสอบ >



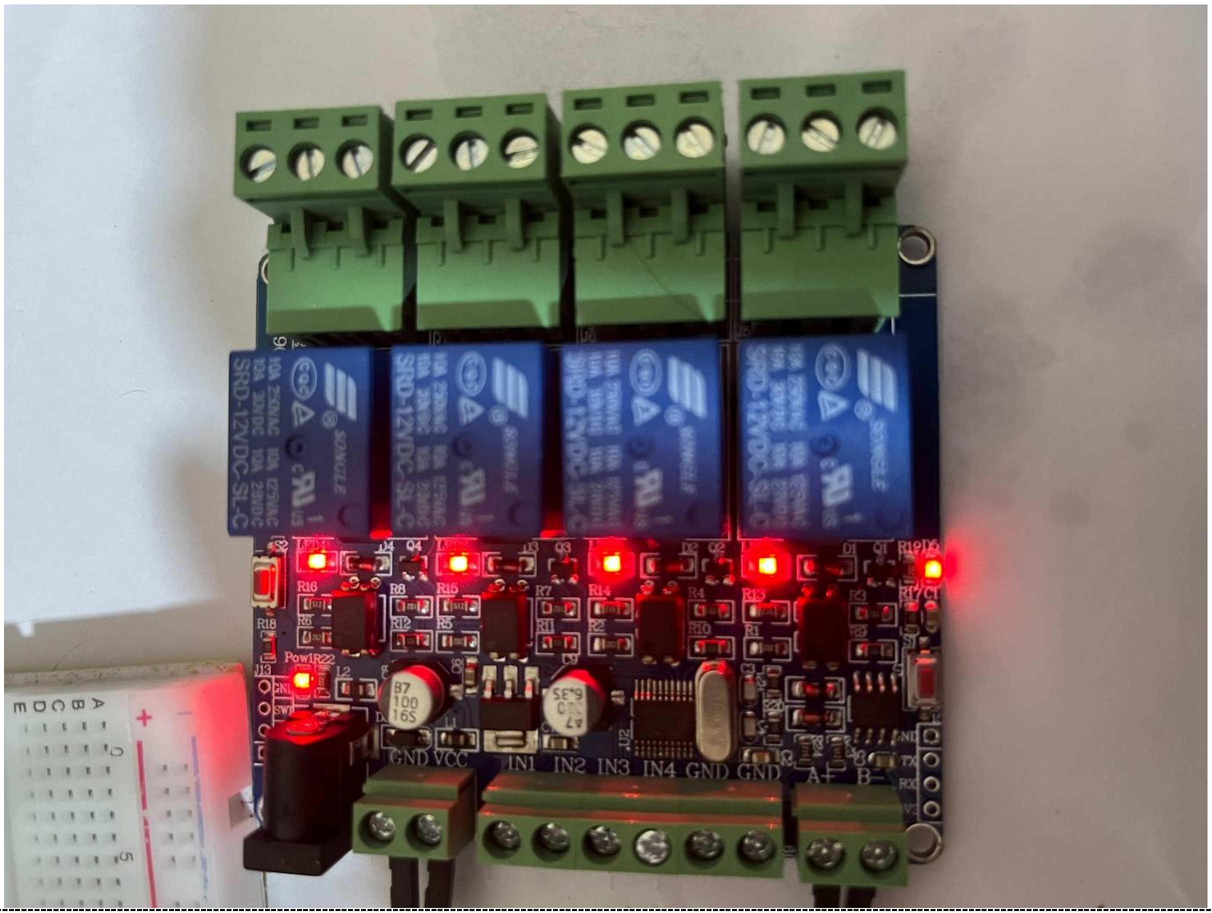
```
33.0060.00
Publish message: { "humid" : 60.00, "tempp": 33.00}
Message arrived [/v2.0/devices/modbus/tempp] {"value": 33.0, "timestamp": 1656246648418, "context"
Message arrived [/v2.0/devices/modbus/humid] {"value": 60.0, "timestamp": 1656246648418, "context"
33.0060.00
Publish message: { "humid" : 60.00, "tempp": 33.00}
Message arrived [/v2.0/devices/modbus/humid] {"value": 60.0, "timestamp": 1656246649229, "context"
Message arrived [/v2.0/devices/modbus/tempp] {"value": 33.0, "timestamp": 1656246649229, "context"
33.0060.00
Publish message: { "humid" : 60.00, "tempp": 33.00}
Message arrived [/v2.0/devices/modbus/humid] {"value": 60.0, "timestamp": 1656246650089, "context"
Message arrived [/v2.0/devices/modbus/tempp] {"value": 33.0, "timestamp": 1656246650089, "context"
33.0060.00
```

Quiz_403 – Modbus RTU/ASCII/TCP with Ubidots IoTs Platform

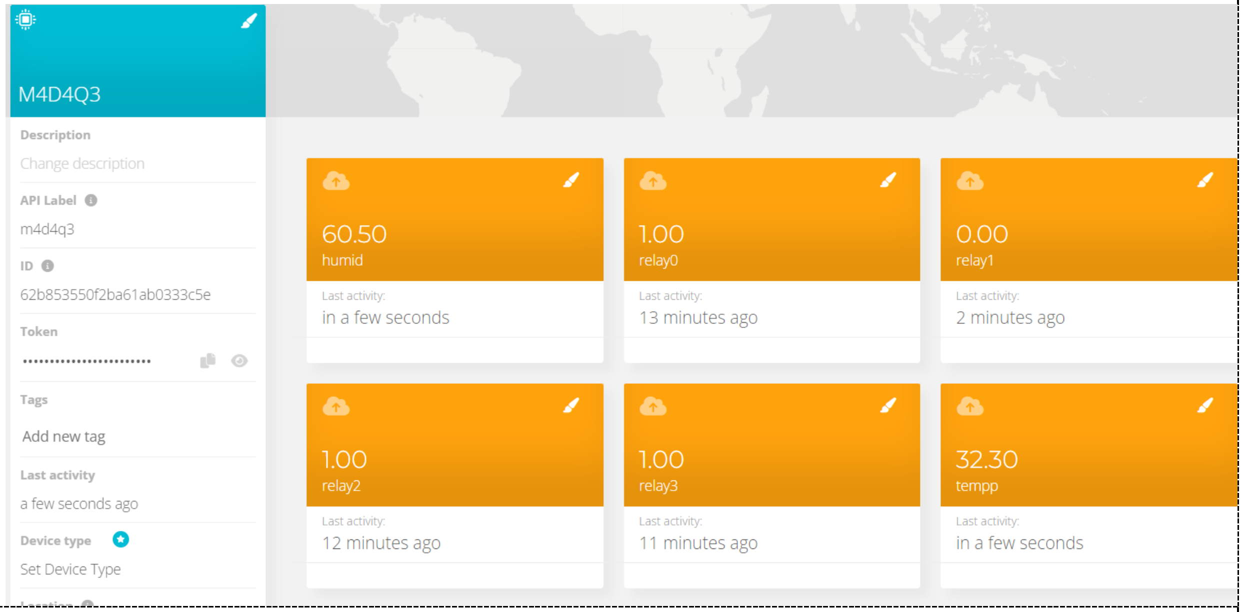
< รูปอุปกรณ์ที่ใช้ทดสอบ ขณะทำการทดสอบ >


B6271942
นายอภิชาติ ภูมิแกก

< รูปอุปกรณ์ที่ใช้ทดสอบ ขณะทำการทดสอบ >



< รูปหน้าจอ Ubidots >



M4D4Q3

Description
Change description

API Label
m4d4q3

ID
62b853550f2ba61ab0333c5e

Token
••••••••••••••••••••••••

Tags
Add new tag

Last activity
a few seconds ago

Device type
Set Device Type

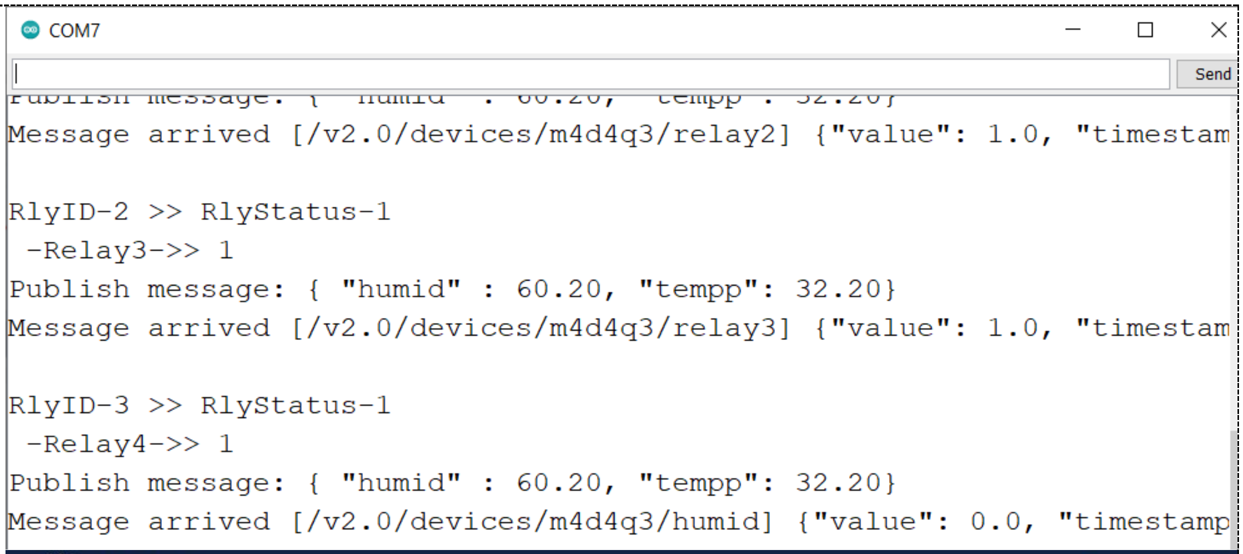
Value	Variable	Last activity
60.50	humid	in a few seconds
1.00	relay0	13 minutes ago
0.00	relay1	2 minutes ago
1.00	relay2	12 minutes ago
1.00	relay3	11 minutes ago
32.30	tempp	in a few seconds

รายยละเอียดการทดสอบ

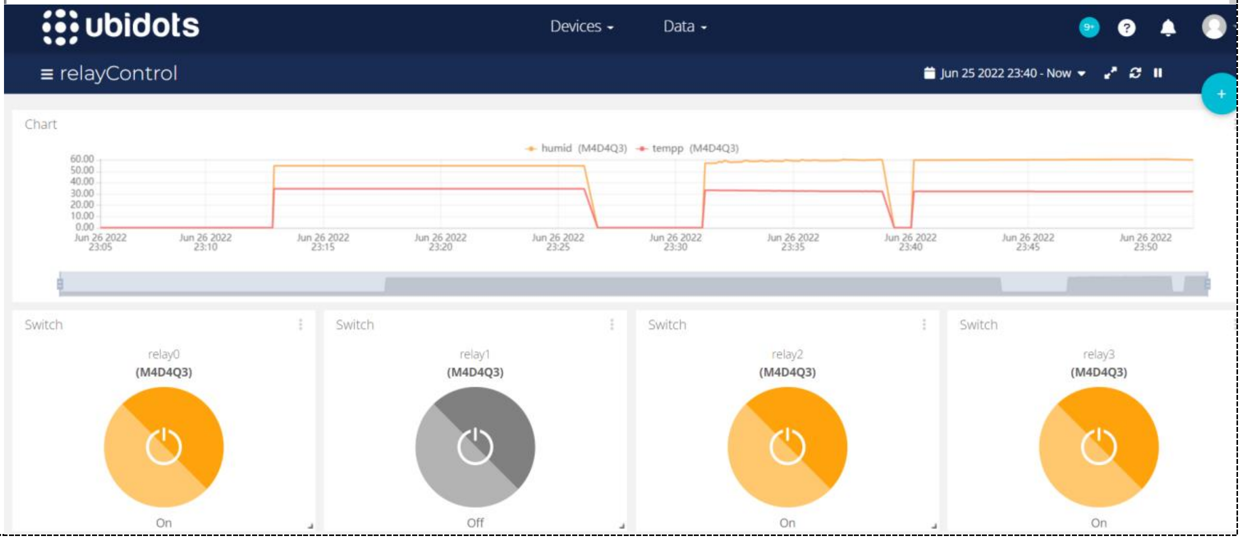
< โปรแกรมทดสอบ >

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

#include <ModbusMaster.h>
#define RS485Transmit HIGH
#define RS485Receive LOW
#define RS485Control 4 //RS485 Direction control
#define Pin_LEDMonitor 2
#define Slave_Sensor_ID 1
#define Slave_Ry4In4_ID 5
int state = 0;
float CTempp, Hudmid;
bool DgInput0, DgInput1, DgInput2, DgInput3;
ModbusMaster node_Sensor;
ModbusMaster node_Ry4In4;

const char *My_SSID = "LANTANIDEs-2.4G";
const char *My_Pass = "0887040892";
const char *MQTT_Server = "things.ubidots.com";
const char *MQTT_User = "BBFF-PEInYY9W5g8JhAQQ9UsegwmHK8mlOf";
const char *MQTT_Pass = "BBFF-PEInYY9W5g8JhAQQ9UsegwmHK8mlOf";
const char *PTopic1 = "/v2.0/devices/m4d4q3";
const char *STopic1 = "/v2.0/devices/m4d4q3/humid";
const char *STopic2 = "/v2.0/devices/m4d4q3/tempp";
const char *STopic3 = "/v2.0/devices/m4d4q3/relay0";
const char *STopic4 = "/v2.0/devices/m4d4q3/relay1";
const char *STopic5 = "/v2.0/devices/m4d4q3/relay2";
const char *STopic6 = "/v2.0/devices/m4d4q3/relay3";
#define MQTT_Port 1883
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);
      client.subscribe(STopic5);
      client.subscribe(STopic6);
    } 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]);
  }
  Serial.println();
  int RlyID = (int)topic[26] - 0x30; // '0'
  int RlySts = (int)payload[10] - 0x30; // '0'
  Serial.println("\nRlyID-" + (String)(RlyID) + " >> RlyStatus-" + (String)(RlySts));
  node_Ry4In4.writeSingleCoil(RlyID, RlySts);
  if (topic[26] == STopic3[26]) {
    Serial.print(" -Relay1->> ");
    Serial.println((char)payload[10]);
  }
  if (topic[26] == STopic4[26]) {
    Serial.print(" -Relay2->> ");
    Serial.println((char)payload[10]);
  }
  if (topic[26] == STopic5[26]) {
    Serial.print(" -Relay3->> ");
    Serial.println((char)payload[10]);
  }
  if (topic[26] == STopic6[26]) {
    Serial.print(" -Relay4->> ");
    Serial.println((char)payload[10]);
  }
}

void preTransmission() {
  digitalWrite(RS485Control, RS485Transmit);
}
void postTransmission() {
  digitalWrite(RS485Control, RS485Receive);
}
void setup() {
  pinMode(RS485Control, OUTPUT);
  pinMode(Pin_LEDMonitor, OUTPUT);
  Serial.begin(115200);
  Serial2.begin(9600);
  postTransmission();
  node_Sensor.begin(Slave_Sensor_ID, Serial2); // Modbus slave ID=1
  node_Sensor.preTransmission(preTransmission);
  node_Sensor.postTransmission(postTransmission);
  node_Ry4In4.begin(Slave_Ry4In4_ID, Serial2); // Modbus slave ID=5
  node_Ry4In4.preTransmission(preTransmission);
  node_Ry4In4.postTransmission(postTransmission);
  Setup_Wifi();
  client.setServer(MQTT_Server, MQTT_Port);
  client.setCallback(callback);
}
void ReadTemperature(void) {
  uint8_t result;
  // Toggle the coil at address (Manual Load Control)
  result = node_Sensor.writeSingleCoil(Slave_Sensor_ID, state);
  state = !state;
  // Read 2 registers starting at 0x0000)
  result = node_Sensor.readInputRegisters(0x0001, 2); // From=0, nByte=2
  if (result == node_Sensor.ku8MBSuccess) {
    CTempp = node_Sensor.getResponseBuffer(0x00) / 10.0f;
    Hudmid = node_Sensor.getResponseBuffer(0x01) / 10.0f;
  }
}

void RelayControl(int inputCase) {
  int rnMode = inputCase / 10;
  int nRelay = inputCase % 10;
  if (rnMode == 41) node_Ry4In4.writeSingleRegister(nRelay, 0x0100); // On RelayX
  if (rnMode == 40) node_Ry4In4.writeSingleRegister(nRelay, 0x0000); // Off RelayX
}
void loop() {
  if (!client.connected()) reconnect();
  client.loop();
  ReadTemperature();
  snprintf (msg, 75, "{ \"humid\" : %5.2f, \"tempp\": %5.2f}", Hudmid, CTempp);
  client.publish(PTopic1, msg);
  Serial.print("Publish message: ");
  Serial.println(msg);
  if (Serial.available() > 0) {
    int DataInput = Serial.parseInt();
    Serial.print("\n >> XYZ > X={8,4}Board Name, Y={1,0}On,Off, Z={0-8}RlyID >> ");
    Serial.println(DataInput);
    RelayControl(DataInput);
  }
  delay(2000);
}
```

< ผลการทดสอบ >



```
Publish message: { "humid" : 60.20, "tempp": 32.20}
Message arrived [/v2.0/devices/m4d4q3/relay2] {"value": 1.0, "timestam

RlyID-2 >> RlyStatus-1
 -Relay3->> 1
Publish message: { "humid" : 60.20, "tempp": 32.20}
Message arrived [/v2.0/devices/m4d4q3/relay3] {"value": 1.0, "timestam

RlyID-3 >> RlyStatus-1
 -Relay4->> 1
Publish message: { "humid" : 60.20, "tempp": 32.20}
Message arrived [/v2.0/devices/m4d4q3/humid] {"value": 0.0, "timestamp
```



Quiz_404 – Application

< อธิบายแนวคิด การนำไปใช้เกี่ยวกับงานที่รับผิดชอบ >