

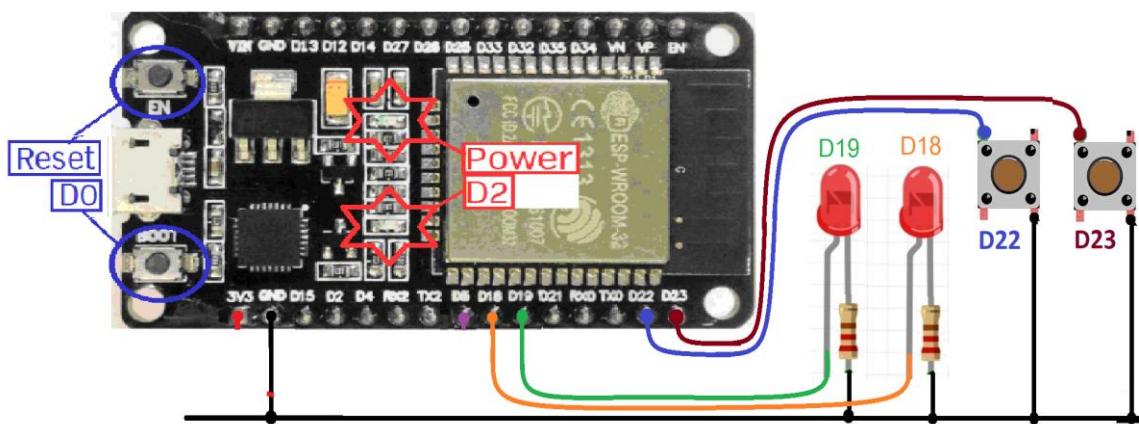
การควบคุมเครื่องจักรอัจฉริยะโดยใช้การสื่อสารระหว่างเครื่องจักรกับเครื่องจักร  
M2M - Intelligence Machine Control

ชื่อ-สกุล : นางสาวณัฐชนยา พ่องกุศล B6226718

6/6 -- คำถ้ามท้ายบทเพื่อทดสอบความเข้าใจ

Quiz\_101 – กดติด กดตับ 2 ชุด

- หากต้องการให้ใช้ 1 สวิตซ์ ควบคุม 1 LED แบบกดติด-กดตับ จำนวน 2 วงจรจะต้องจราจรและเขียนโปรแกรมอย่างไร {SW-D22 -- LED-D19, SW-D23 -- LED-D18}



```
#define pushButton1 22
#define LEDPin1 18
#define pushButton2 23
#define LEDPin2 19

int buttonState1 = 0;
int buttonState2 = 0;

void setup() {
    Serial.begin(115200);
    pinMode(pushButton1, INPUT_PULLUP);
    pinMode(LEDPin1, OUTPUT);
    pinMode(pushButton2, INPUT_PULLUP);
    pinMode(LEDPin2, OUTPUT);
}

void loop() {
```

```

if (digitalRead(pushButton1) == LOW) {
    delay(20);
    buttonState1 = 1 - buttonState1;
    digitalWrite(LEDPin1, buttonState1);
    while (digitalRead(pushButton1) == LOW);
    delay(20);
}

if (digitalRead(pushButton2) == LOW) {
    delay(20);
    buttonState2 = 1 - buttonState2;
    digitalWrite(LEDPin2, buttonState2);
    while (digitalRead(pushButton2) == LOW);
    delay(20);
}

```

D1 | Arduino 1.8.13

File Edit Sketch Tools Help

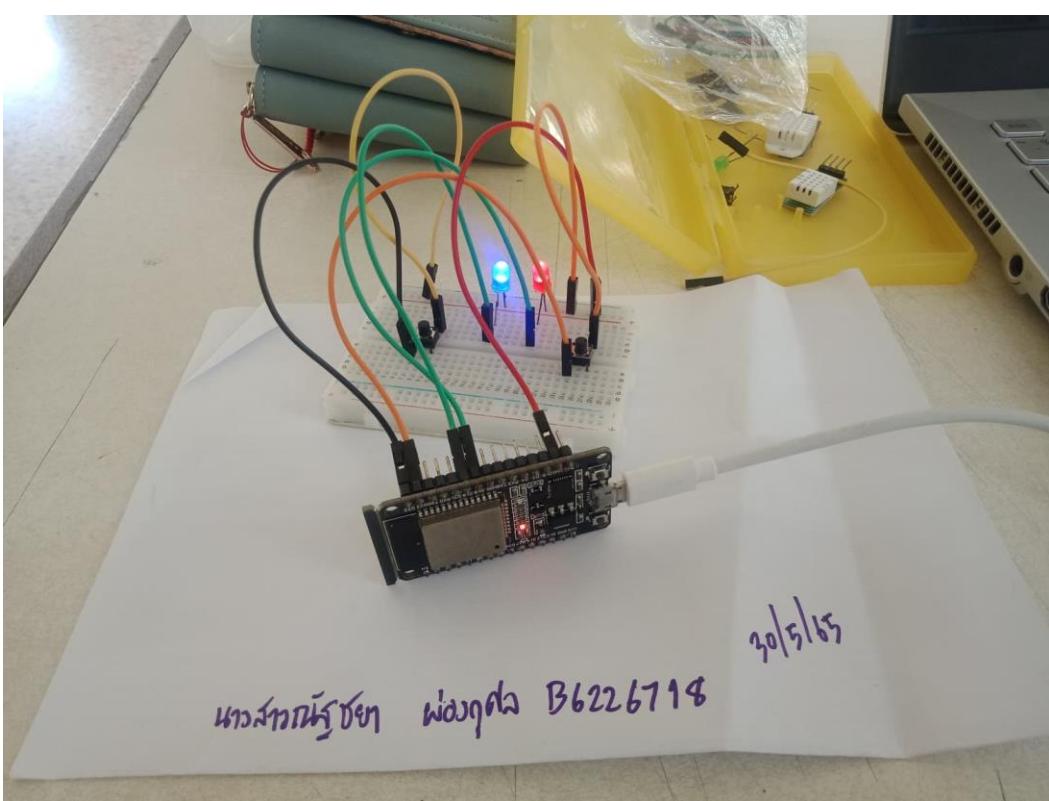
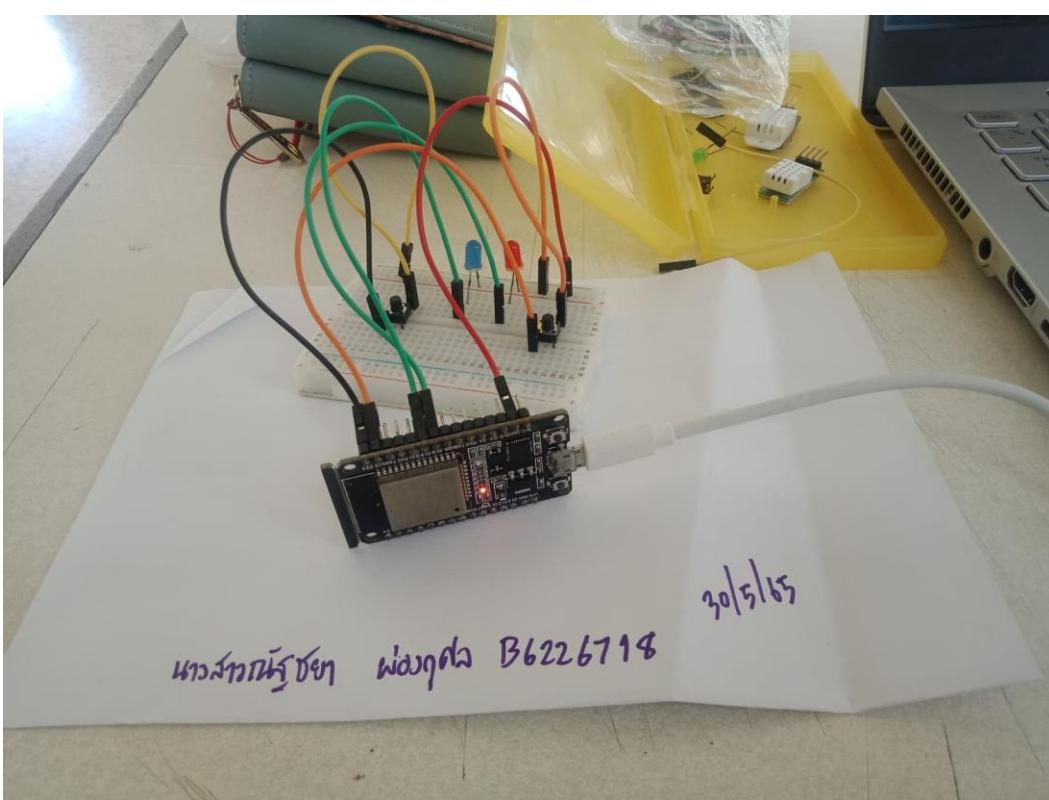
D1

```

1 #define pushButton1 22
2 #define LEDPin1 18
3 #define pushButton2 23
4 #define LEDPin2 19
5 int buttonState1 = 0;
6 int buttonState2 = 0;
7 void setup() {
8     Serial.begin(115200);
9     pinMode(pushButton1, INPUT_PULLUP);
10    pinMode(LEDPin1, OUTPUT);
11    pinMode(pushButton2, INPUT_PULLUP);
12    pinMode(LEDPin2, OUTPUT);
13 }
14 void loop() {
15     if (digitalRead(pushButton1) == LOW) {
16         delay(20);
17         buttonState1 = 1 - buttonState1;
18         digitalWrite(LEDPin1, buttonState1);
19         while (digitalRead(pushButton1) == LOW);
20         delay(20);
21     }
22
23     if (digitalRead(pushButton2) == LOW) {
24         delay(20);
25         buttonState2 = 1 - buttonState2;
26         digitalWrite(LEDPin2, buttonState2);

```

Leaving...  
hard resetting via RTS pin...



**Quiz\_102 – Web Control 4 LED and Monitor Humid/Temperature**

- เพิ่มเติมจาก Q202 อย่างใดดีบุ้งสำหรับคุณปิด-เปิด หลอดไฟ LED 4 ดวง
- อย่างนี้ก็ Link ไปที่หน้า FB ของตัวเอง
- [https://www.colorhexa.com/008cba?fbclid=IwAR3dIZ\\_gRgDWmREmnzuknLbMxV3pOHy4YIPuLEz8-ZzTOX2VhWxcH2QjLGK](https://www.colorhexa.com/008cba?fbclid=IwAR3dIZ_gRgDWmREmnzuknLbMxV3pOHy4YIPuLEz8-ZzTOX2VhWxcH2QjLGK)

← → C ⓘ Not secure | 192.168.43.237

## The ESP-32 Update web page without refresh

LED1 ON
LED2 ON
LED3 ON
LED4 ON

LED1 OFF
LED2 OFF
LED3 OFF
LED4 OFF

State of [LED1, LED2, LED3, LED4] is >> ON, OFF, OFF, ON

DHT-22 sensor : Temp = 28.10 C, Humidity = 43.90 %

[By Wichai Srisuruk](#)

---

```
#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 testLED1 18
#define testLED2 19
#define testLED3 22
#define testLED4 23
//SSID and Password of your WiFi router
const char* ssid = "B6226718";
const char* password = "pungpingjerry";
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/plain", 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(testLED1, HIGH); //Feedback parameter
        ledState1 = "ON";
    }
    if (t_state == "10") {
        digitalWrite(testLED1, LOW); //Feedback parameter
        ledState1 = "OFF";
    }
}
```

```
if (t_state == "21") {  
    digitalWrite(testLED2, HIGH); //Feedback parameter  
    ledState2 = "ON";  
}  
  
if (t_state == "20") {  
    digitalWrite(testLED2, LOW); //Feedback parameter  
    ledState2 = "OFF";  
}  
  
if (t_state == "31") {  
    digitalWrite(testLED3, HIGH); //Feedback parameter  
    ledState3 = "ON";  
}  
  
if (t_state == "30") {  
    digitalWrite(testLED3, LOW); //Feedback parameter  
    ledState3 = "OFF";  
}  
  
if (t_state == "41") {  
    digitalWrite(testLED4, HIGH); //Feedback parameter  
    ledState4 = "ON";  
}  
  
if (t_state == "40") {  
    digitalWrite(testLED4, LOW); //Feedback parameter  
    ledState4 = "OFF";  
}  
  
server.send(200, "text/plain", ledState1 + ", " + ledState2 + ", " + ledState3 + ", " + ledState4);  
//Send web page  
}  
  
void setup(void) {  
    Serial.begin(115200);  
    dht.setup(DHT_Pin, DHTesp::DHT22); // DHT_Pin D4, DHT22
```

```
pinMode(testLED1, OUTPUT);
pinMode(testLED2, OUTPUT);
pinMode(testLED3, OUTPUT);
pinMode(testLED4, 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
}

//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>  
(Updatee: The challenge didn't tell me to do it.) 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); // 2000ms update rate  
  
function getData() {  
  
    var xhttp = new XMLHttpRequest();
```

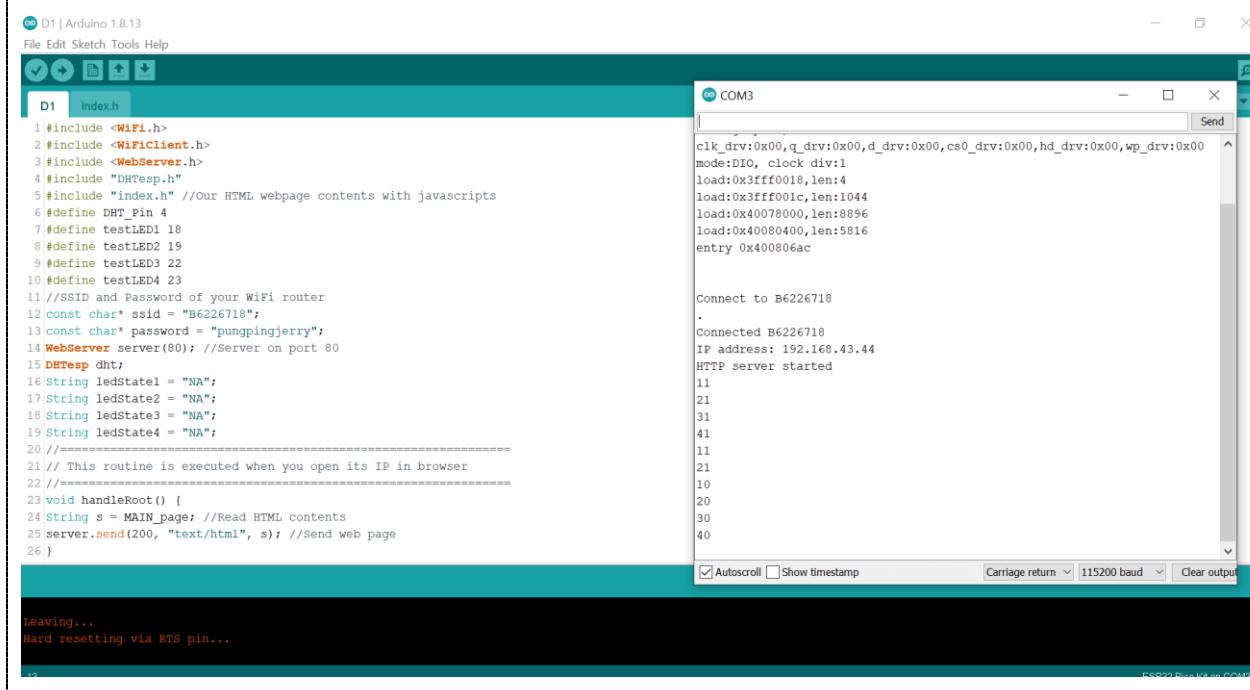
```

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/profile.php?id=100007563972020">Natchaya
Phongkuson</a></body></html>
)=====";

```



The ESP-32 Update web page without refresh

[LED1 ON](#) [LED2 ON](#) [LED3 ON](#) [LED4 ON](#)

[LED1 OFF](#) [LED2 OFF](#) [LED3 OFF](#) [LED4 OFF](#)

State of [LED1, LED2, LED3, LED4] is >> ON, ON, ON, ON

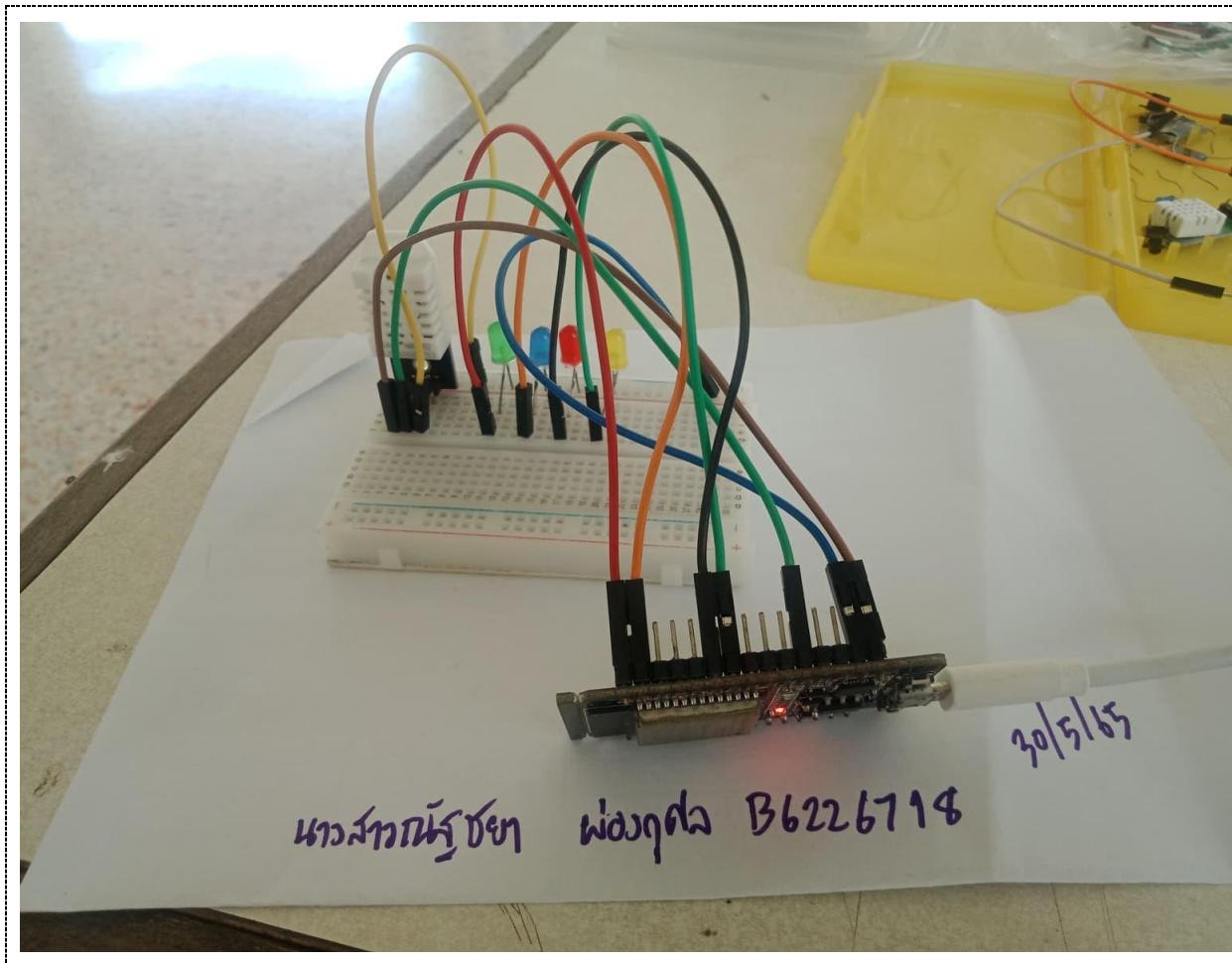
(Update: The challenge didn't tell me to do it.) DHT-22 sensor : Temp = 30.10 C, Humidity = 99.90 %

[Natchaya Phongkuson](#)

รูปการทดสอบ 1

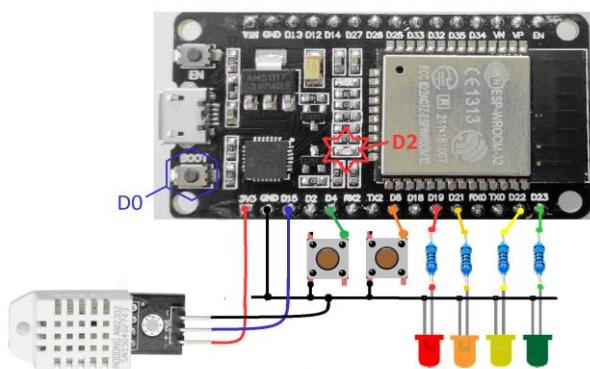
30/5/15

hacangkachan พลังงาน B6226714



**Quiz\_103 – Pub/Sub Data from (DHT22 + 4 LED + 2 Switch)**

- อ่านค่า DHT-22 และส่งไปยัง MQTT Broker ทุกๆ 5 วินาที
- กำหนดให้ใช้ mqtt.eclipse.org เป็น Broker
- ควบคุมการเปิดปิด 4 LED
- รับค่าสวิตซ์กำหนด SW1 แจ้ง Overheat Alarm, SW2 แจ้ง Intruders Alarm



```
#include <WiFi.h>
#include <Wire.h>
#include <PubSubClient.h>
#include "DHTesp.h"

DHTesp dht;

#define testLED1 18
#define testLED2 19
#define testLED3 22
#define testLED4 23

#define DHT22_Pin 4

const char* ssid = "B6226718";
const char* password = "pungpingjerry";
const char* mqtt_server = "test.mosquitto.org";
const char* topic1 = "tptpr";

String ledState1 = "NA";
int pushButton1 = 15;
int pushButton2 = 2;
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(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());
pinMode(testLED1, OUTPUT);
pinMode(testLED2, OUTPUT);
pinMode(testLED3, OUTPUT);
pinMode(testLED4, OUTPUT);
}
void callback(char* topic, byte* payload, unsigned int length)
{ char myPayLoad[50];
Serial.print("Message arrived [");
Serial.print(topic);
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[4] = '\0'; // String less than 4 characters

if ((String)myPayLoad == "ON1") digitalWrite(testLED1, HIGH);
if ((String)myPayLoad == "OFF1") digitalWrite(testLED1, LOW);
if ((String)myPayLoad == "ON2") digitalWrite(testLED2, HIGH);
if ((String)myPayLoad == "OFF2") digitalWrite(testLED2, LOW);
if ((String)myPayLoad == "ON3") digitalWrite(testLED3, HIGH);
if ((String)myPayLoad == "OFF3") digitalWrite(testLED3, LOW);
if ((String)myPayLoad == "ON4") digitalWrite(testLED4, HIGH);
if ((String)myPayLoad == "OFF4") digitalWrite(testLED4, LOW);
}

void reconnect()
{
    while (!client.connected()) // Loop until we're reconnected
    {
        Serial.print("Attempting MQTT connection...");

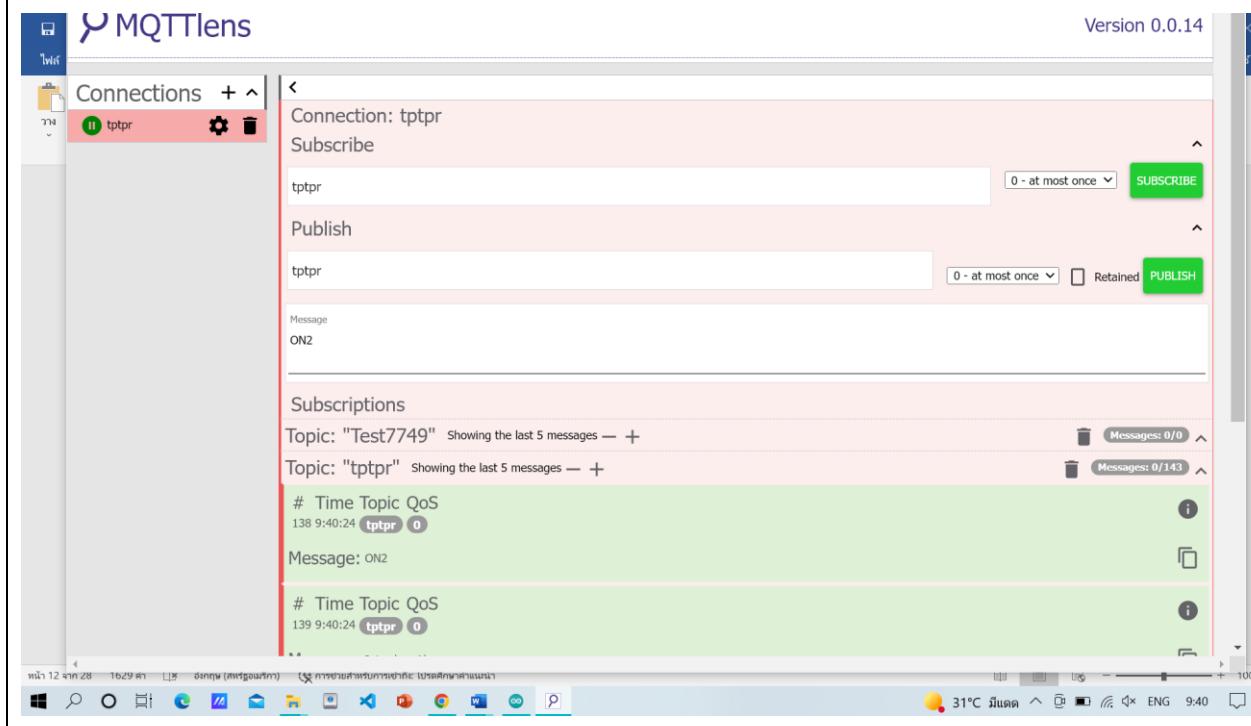
        String clientId = "ESP8266Client-";
        clientId += String(random(0xffff), HEX); // Create a random client ID

        if (client.connect(clientId.c_str())) // Attempt to connect
        {
            Serial.println("connected"); // Once connected, publish an announcement...
            client.publish(topic1, "Hello World Pk007"); // ... and resubscribe
            client.subscribe(topic1);
        }
        else
        {
            Serial.print("failed, rc=");
            Serial.print(client.state());
            Serial.println(" try again in 5 seconds");
            delay(5000);
        }
    }
}

void setup()
```

```
{ Serial.begin(115200);
setup_wifi();
dht.setup(DHT22_Pin, DHTesp::DHT22);
pinMode(pushButton1, INPUT_PULLUP);
pinMode(pushButton2, INPUT_PULLUP);
client.setServer(mqtt_server, 1883);
client.setCallback(callback);
pinMode(testLED1, OUTPUT);
pinMode(testLED2, OUTPUT);
pinMode(testLED3, OUTPUT);
pinMode(testLED4, OUTPUT);
}
void loop()
{
if (!client.connected()) reconnect();
client.loop();
long now = millis();
if (now - lastMsg > 5000)
{ lastMsg = now;
++value;
float h = dht.getHumidity();
float t = dht.getTemperature();
sprintf (msg, "TempC: %.2f C, Humidity: %.2f %%",t,h);
Serial.print("Publish message: ");
Serial.println(msg);
client.publish(topic1, msg);
}
if (digitalRead(pushButton1) == 0) {
sprintf (msg, "Overheat Alarm");
Serial.println(msg);
client.publish(topic1, msg);
```

```
delay(500);  
}  
  
if (digitalRead(pushButton2) == 0) {  
    sprintf (msg, "Intruders Alarm");  
    Serial.println(msg);  
    client.publish(topic1, msg);  
    delay(500);  
}  
}
```



**MQTTlens**

Add a new Connection

Connection Details

Connection name: tptpr

Connection color scheme: Red

Hostname: tcp:// test.mosquitto.org

Port: 1883

Client ID: lens\_pStrGiIZqGSUZSsHOv9vf2dnv

Session: Clean Session

Automatic Connection: Automatic Connection

Keep Alive: 120 seconds

Credentials

Username: Enter username

Password: Enter password

Version 0.0.14

MQTTlens interface showing a connection to test.mosquitto.org.

D1 | Arduino 1.8.13

File Edit Sketch Tools Help

Serial Monitor

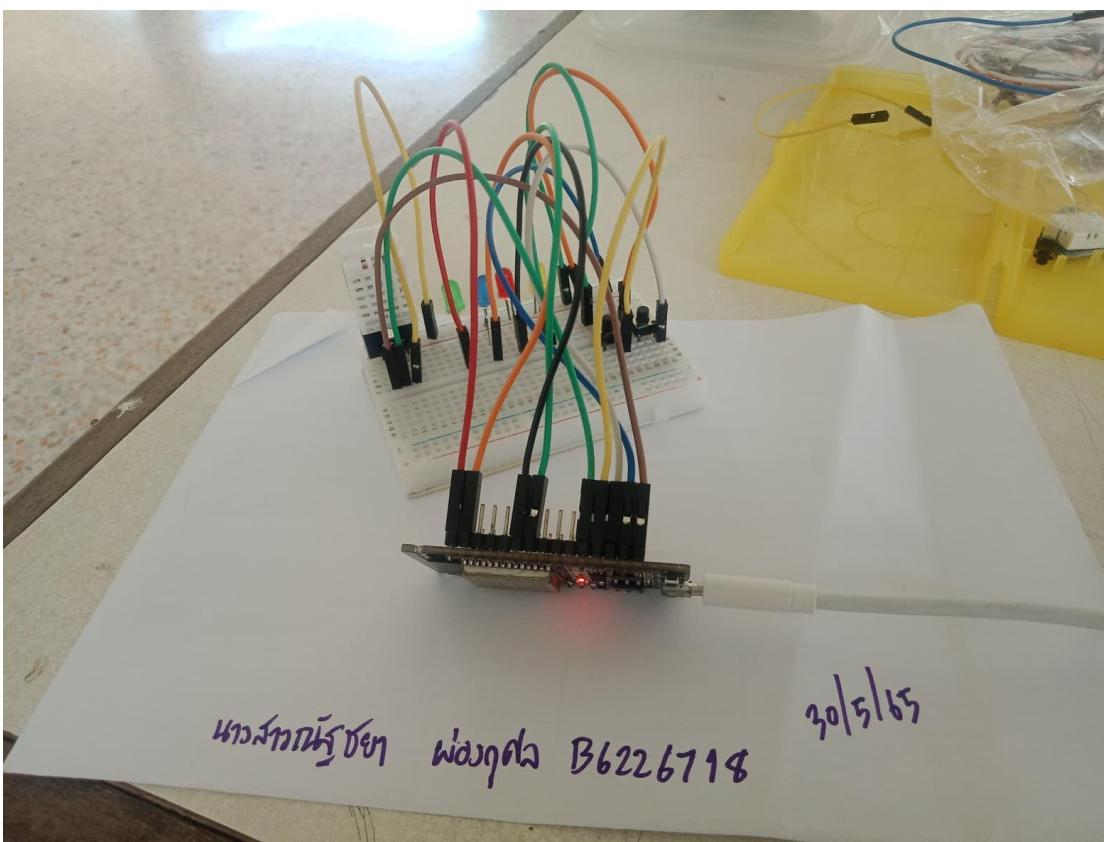
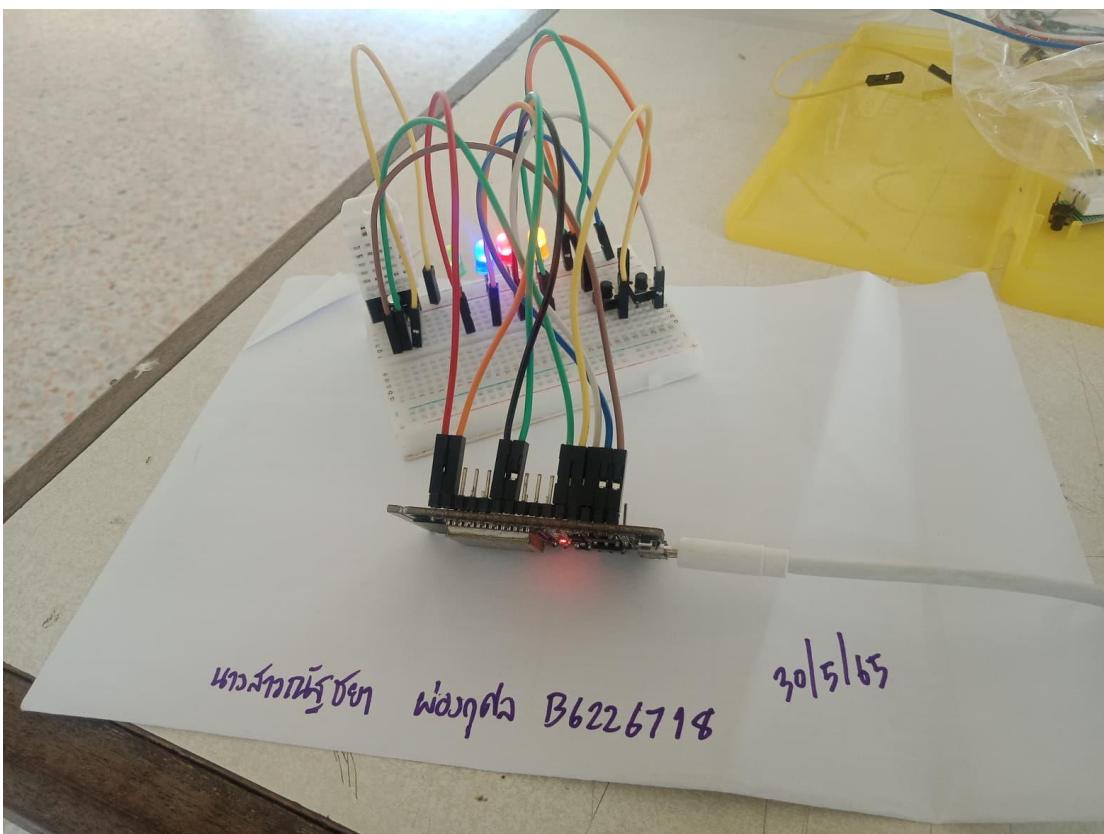
Message arrived [tptpr] Intruders Alarm  
--> Intruders Alarm  
Publish message: TempC: 30.20 C, Humidity: 99.90 %  
Intruders Alarm  
Message arrived [tptpr] Intruders Alarm  
--> Intruders Alarm  
Intruders Alarm  
Message arrived [tptpr] Intruders Alarm  
--> Intruders Alarm  
Intruders Alarm  
Message arrived [tptpr] TempC: 30.20 C, Humidity: 99.90 %  
--> TempC: 30.20 C, Humidity: 99.90 %  
Intruders Alarm  
Message arrived [tptpr] Intruders Alarm  
--> Intruders Alarm  
Intruders Alarm  
Message arrived [tptpr] Intruders Alarm  
--> Intruders Alarm  
Intruders Alarm  
Message arrived [tptpr] Intruders Alarm  
--> Intruders Alarm  
Intruders Alarm  
Message arrived [tptpr] Intruders Alarm  
--> Intruders Alarm  
Intruders Alarm  
Intruders Alarm

Autoscroll  Show timestamp  Carriage return  115200 baud  Clear output

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

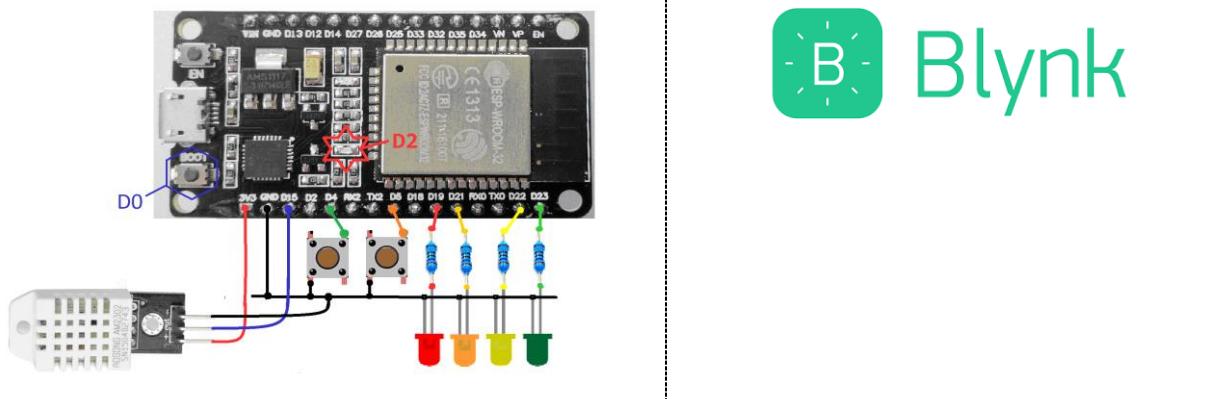
ESP32 Pico Kit on COM3

Arduino IDE interface showing code for D1 and Serial Monitor output.



### Quiz\_104 – Blynk and LINE from (DHT22 + 4 LED + 2 Switch)

- ควบคุมการปิดเปิด 4 LED
- อ่านค่า DHT-22 และส่งไปยัง Blynk ทุกๆ 5 วินาที
- บันทึกค่าไปยัง Google Sheet
- หากอุณหภูมิเกิน 28°C ให้แจ้งไปยัง LINE
- รับค่าสวิตซ์กำหนด SW1 และ Overheat Alarm, SW2 และ Intruders Alarm ไปยัง LINE



```
#define BLYNK_PRINT Serial

#include <WiFi.h>
#include <WiFiClient.h>
#include <BlynkSimpleEsp32.h>
#include <HTTPClient.h>
#include "DHTesp.h"

#define DHT22_Pin 4
#define sw1 15
#define sw2 2

char auth[] = "WPa2mtFT_6qi9RGfh3Nxc8AXs-y1fnoe";
char ssid[] = "BOOK";
char pass[] = "book1017";
#define WebHooksKey "y2TRnqZxW1XjxELy7-rO-"
#define WebHooksEventName "test_GSheet"
```

```
#define WebHooksEventName_line "Test_Key"
#define My_NAME "B6226718 Natchaya Phongkuson"

DHTesp dht;

WidgetLED LED1(V2);
WidgetLED LED2(V3);
BlynkTimer timer;

void myTimerEvent() {
    float humidity = dht.getHumidity();
    float temperature = dht.getTemperature();
    Blynk.virtualWrite(V0, temperature);
    Blynk.virtualWrite(V1, humidity);
    if (digitalRead(sw1)) LED1.on();
    else LED1.off();
    if (digitalRead(sw2)) LED2.on();
    else LED2.off();
    Serial.print(" Temp('c) >> "); Serial.print(temperature, 1);
    Serial.print(", Humidity(%) >> "); Serial.println(humidity, 1);
}

void setup() {
    Serial.begin(115200);
    dht.setup(DHT22_Pin, DHTesp::DHT22); // Connect DHT sensor to GPIO 15
    pinMode(sw1, INPUT_PULLDOWN);
    pinMode(sw2, INPUT_PULLDOWN);
```

```
Blynk.begin(auth, ssid, pass);
timer.setInterval(1000L, myTimerEvent);
WiFi.begin(ssid, pass);
Serial.println("Connecting");
while (WiFi.status() != WL_CONNECTED) {
delay(500);
Serial.print(".");
}
}

void loop(){
delay(5000);
float humidity = dht.getHumidity();
float temperature = dht.getTemperature();

String serverName = "http://maker.ifttt.com/trigger/" +
String(WebHooksEventName) + "/with/key/" + String(WebHooksKey);
String httpRequestData = "value1=" + String(My_NAME) + "&value2=" +
String(temperature) + "&value3=" +
String(humidity);
if (WiFi.status() == WL_CONNECTED) {
HTTPClient http;
http.begin(serverName);
http.addHeader("Content-Type", "application/x-www-form-urlencoded");
int httpResponseCode = http.POST(httpRequestData);
Serial.print("HTTP Response code: ");
Serial.println(httpResponseCode);
http.end();
}
```

```
if (httpResponseCode == 200)
Serial.println("[Google sheet] --> Successfully sent");
else
Serial.println("[Google sheet] --> Failed!");
}
else {
Serial.println("WiFi Disconnected");
}
/// if temp > 28 C send notifications >> line
if (temperature > 28) {
String serverName = "http://maker.ifttt.com/trigger/" +
String(WebHooksEventName_line) + "/with/key/" + String(WebHooksKey);
String httpRequestData = "value1=" + String(temperature);
Serial.println();
Serial.println("Server Name >> " + serverName);
Serial.println("json httpRequestData >> " + httpRequestData);
if (WiFi.status() == WL_CONNECTED) {
HTTPClient http;
http.begin(serverName);
http.addHeader("Content-Type", "application/x-www-form-urlencoded");
int httpResponseCode = http.POST(httpRequestData);
Serial.print("HTTP Response code: ");
Serial.println(httpResponseCode);
http.end();
if (httpResponseCode == 200)
Serial.println("[Line] --> Successfully sent");
else
Serial.println("[Line] --> Failed!");
```

```
}

else {

Serial.println("WiFi Disconnected");

}

}

if (digitalRead(sw1) == LOW) {

String serverName = "http://maker.ifttt.com/trigger/" +
String(WebHooksEventName_line) + "/with/key/" + String(WebHooksKey);
String httpRequestData = "value3=" + String("Intruders Alarm");
Serial.println();
Serial.println("Server Name >> " + serverName);
Serial.println("json httpRequestData >> " + httpRequestData);
if (WiFi.status() == WL_CONNECTED) {

HTTPClient http;
http.begin(serverName);
http.addHeader("Content-Type", "application/x-www-form-urlencoded");
int httpResponseCode = http.POST(httpRequestData);
Serial.print("HTTP Response code: ");
Serial.println(httpResponseCode);
http.end();
if (httpResponseCode == 200)
Serial.println("[Line] --> Successfully sent");
else
Serial.println("[Line] --> Failed!");
}
else {
Serial.println("WiFi Disconnected");
```

```
        }

    }

    if (digitalRead(sw2) == LOW) {

        String serverName = "http://maker.ifttt.com/trigger/" +
            String(WebHooksEventName_line) + "/with/key/" + String(WebHooksKey);

        String httpRequestData = "value3=" + String("Overheat Alarm");

        Serial.println();

        Serial.println("Server Name >> " + serverName);
        Serial.println("json httpRequestData >> " + httpRequestData);

        if (WiFi.status() == WL_CONNECTED) {

            HttpClient http;

            http.begin(serverName);

            http.addHeader("Content-Type", "application/x-www-form-urlencoded");

            int httpResponseCode = http.POST(httpRequestData);

            Serial.print("HTTP Response code: ");

            Serial.println(httpResponseCode);

            http.end();

            if (httpResponseCode == 200)
                Serial.println("[Line] --> Successfully sent");
            else
                Serial.println("[Line] --> Failed!");
        }
        else {
            Serial.println("WiFi Disconnected");
        }
    }

    Blynk.run();
}
```

```
timer.run(); // running timer every 250ms
```

}

The smartphone screen shows a mobile application interface. At the top, it displays "4G AIS VoLTE", the time "10:14 น.", and a battery level of "61%". The main interface has a green header with the text "tp11". Below the header, there are four large circular buttons labeled "OFF", one of which is highlighted in green and labeled "ON". Below these buttons are two small circular status indicators labeled "LED" and "TEMPP". To the right of the status indicators is a large circular gauge with a scale from 0 to 1023, currently showing a value of 31.5. Further to the right is a digital display showing "HUMID" and the value "99.9".

The screenshot shows a Google Sheets document titled "Intruders Alarm". The sheet contains a table with the following data:

	A	B	C	D	E	F	G
279	May 30, 2022 at test_GSheet	B6226718 Natch	30.5	99.9			
280	May 30, 2022 at test_GSheet	B6226718 Natch	30.5	99.9			
281	May 30, 2022 at test_GSheet	B6226718 Natch	30.6	99.9			
282	May 30, 2022 at test_GSheet	B6226718 Natch	30.6	99.9			
283	May 30, 2022 at test_GSheet	B6226718 Natch	30.6	99.9			
284	May 30, 2022 at test_GSheet	B6226718 Natch	30.6	99.9			
285	May 30, 2022 at test_GSheet	B6226718 Natch	30.6	99.9			
286	May 30, 2022 at test_GSheet	B6226718 Natch	30.6	99.9			
287	May 30, 2022 at test_GSheet	B6226718 Natch	30.6	99.9			
288	May 30, 2022 at test_GSheet	B6226718 Natch	30.6	99.9			
289	May 30, 2022 at test_GSheet	B6226718 Natch	30.6	99.9			
290	May 30, 2022 at test_GSheet	B6226718 Natch	30.6	99.9			
291	May 30, 2022 at test_GSheet	B6226718 Natch	30.6	99.9			
292	May 30, 2022 at test_GSheet	B6226718 Natch	30.7	99.9			
293	May 30, 2022 at test_GSheet	B6226718 Natch	30.7	99.9			
294	May 30, 2022 at test_GSheet	B6226718 Natch	30.8	99.9			
295							
296							
297							

Below the table, a red text message in Thai reads: "ครั้นนี้จะให้การกดปุ่มไปแสดงใน serial monitor ไม่ได้ส่งไปในไลน์ และแสดง tempp ด้วย gauge".

The screenshot shows an IFTTT applet titled "If Maker Event 'test\_GSheet', then Add row to tungpar7401@hotmail.com's Google Drive spreadsheet". The title includes icons for a maker event and a Google Sheets document. Below the title, it says "Connected". A blue banner at the bottom displays a series of recent triggers, each with a bell icon and the text "IFTTT: อุณหภูมิ: ปั๊มกด:" followed by a timestamp (e.g., 09:47 น., 10:05 น.).

If Maker Event  
"test\_GSheet", then  
Add row to  
tungpar7401@hotmail.com's  
Google Drive  
spreadsheet

Connected

IFTTT: อุณหภูมิ: ปั๊มกด: 09:47 น.

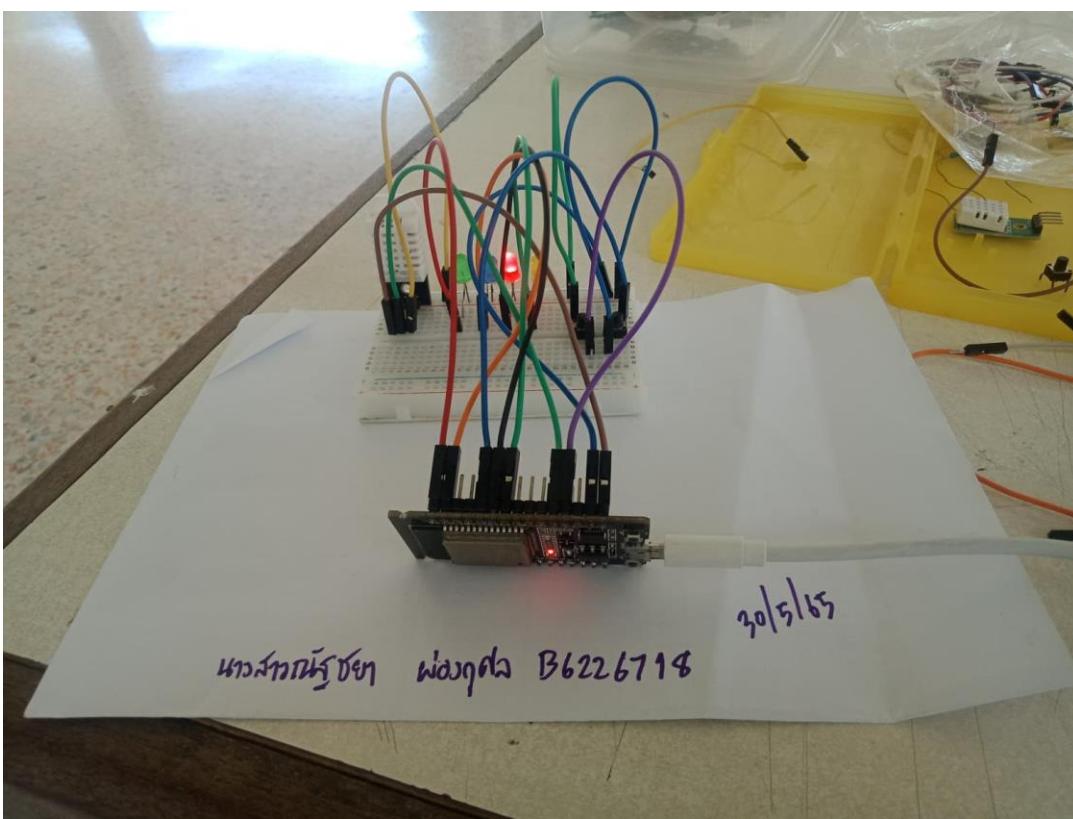
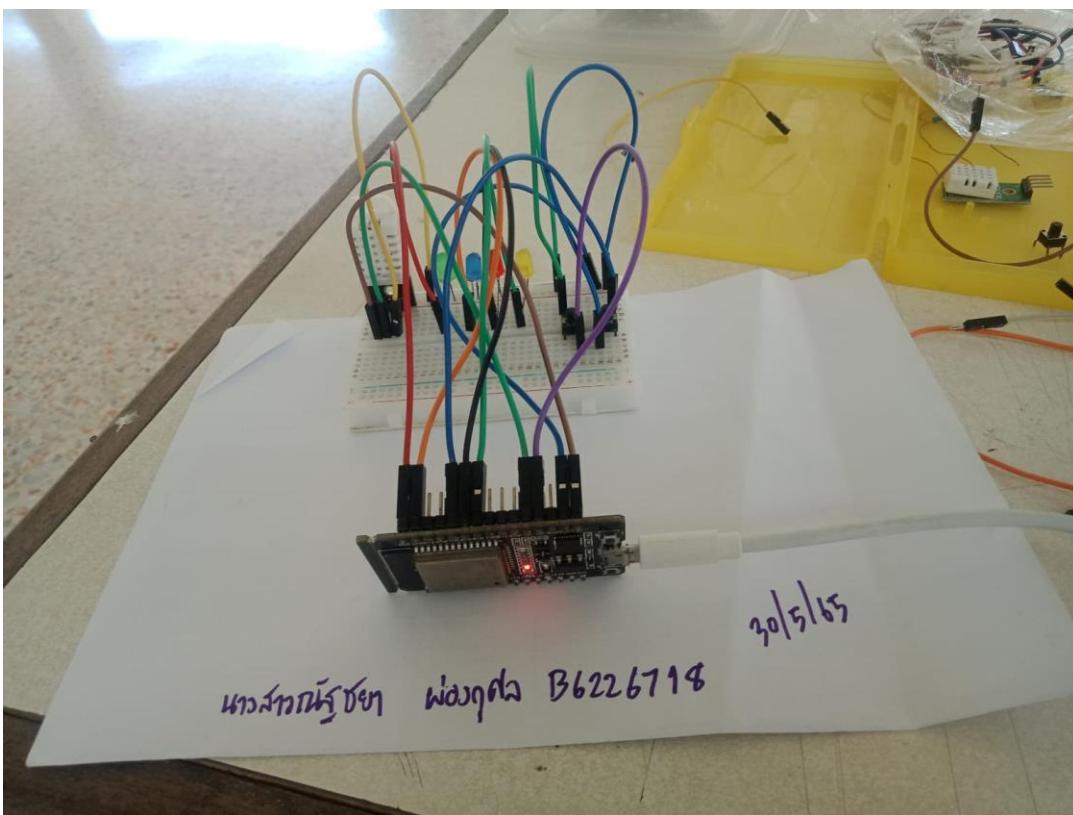
IFTTT: อุณหภูมิ: 30.90 ปั๊มกด: 10:05 น.

IFTTT: อุณหภูมิ: 30.90 ปั๊มกด: 10:05 น.

IFTTT: อุณหภูมิ: 31.00 ปั๊มกด: 10:05 น.

IFTTT: อุณหภูมิ: 30.90 ปั๊มกด: 10:05 น.

IFTTT: อุณหภูมิ: 31.00 ปั๊มกด: 10:05 น.



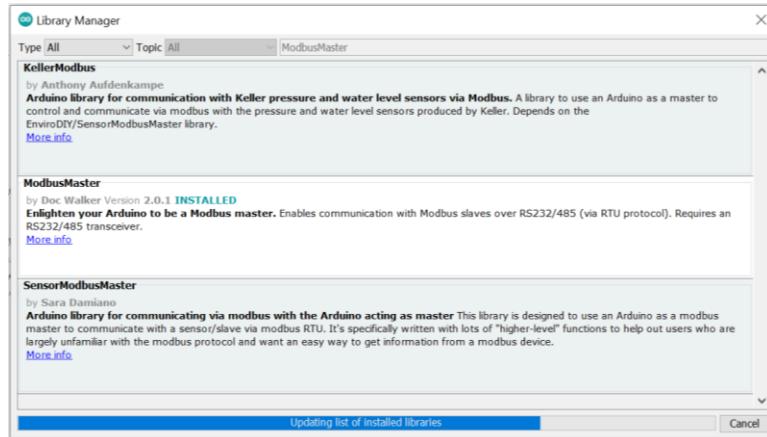
การควบคุมเครื่องจักรอัตโนมัติโดยใช้การสื่อสารระหว่างเครื่องจักรกับเครื่องจักร  
M2M - Intelligence Machine Control

ชื่อ-สกุล : นางสาวณัฐชนยา พองกุศล B6226718

4/4: -- คำถ้าท้ายบทเพื่อทดสอบความเข้าใจ

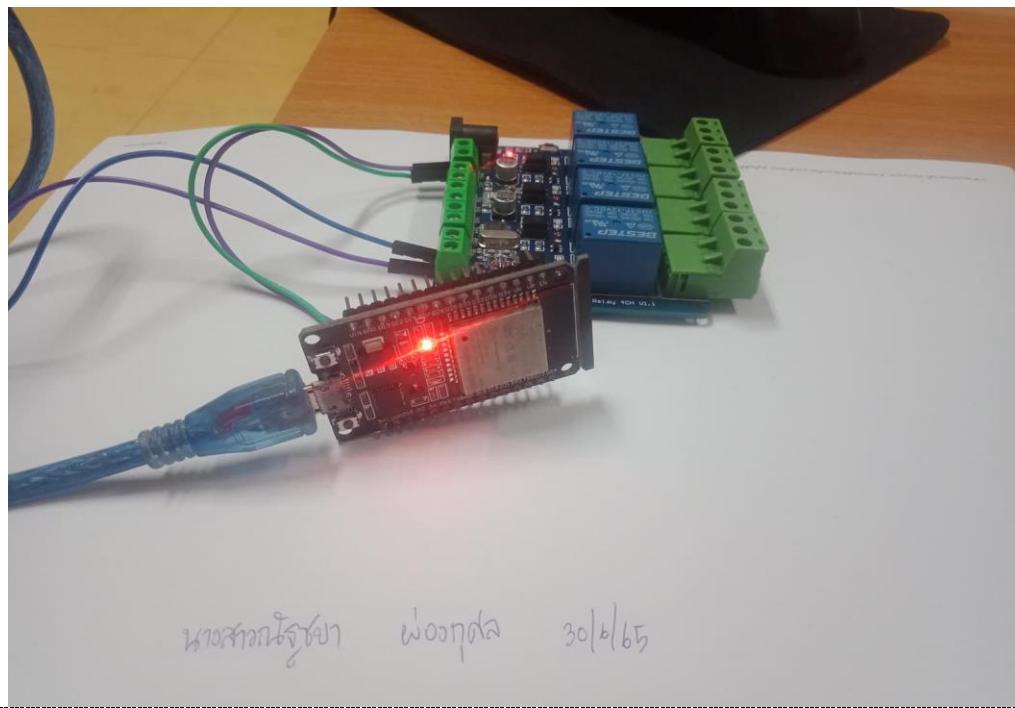
**ແນມ\_ກາຮ setup**

Install -> ถ้าจะทำผ่าน Arduino

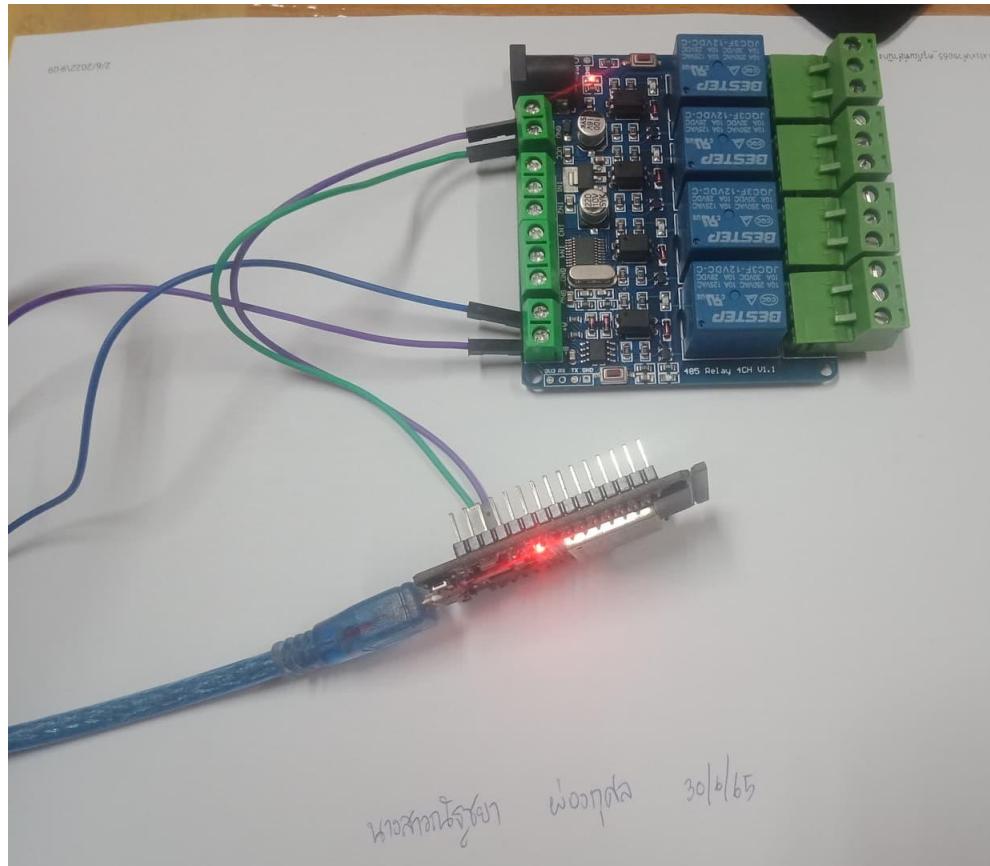


**Quiz\_201 – Read Modbus RTU**

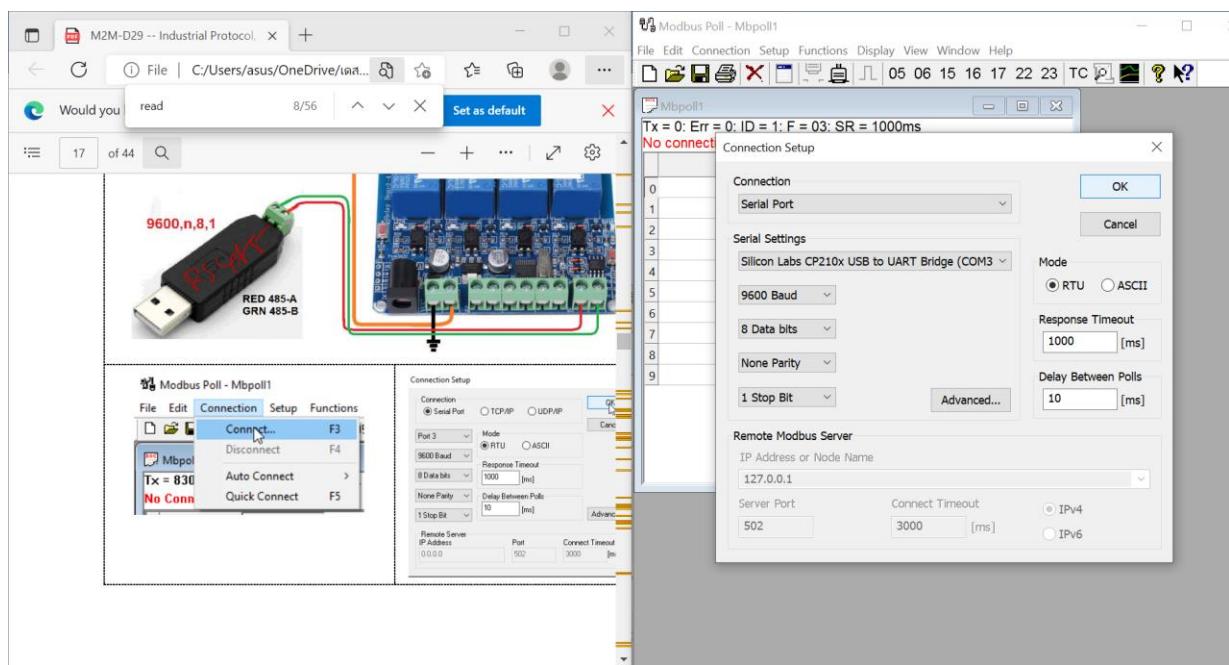
< รูปอุปกรณ์ที่ใช้ทดสอบ ขณะทำการทดสอบ >

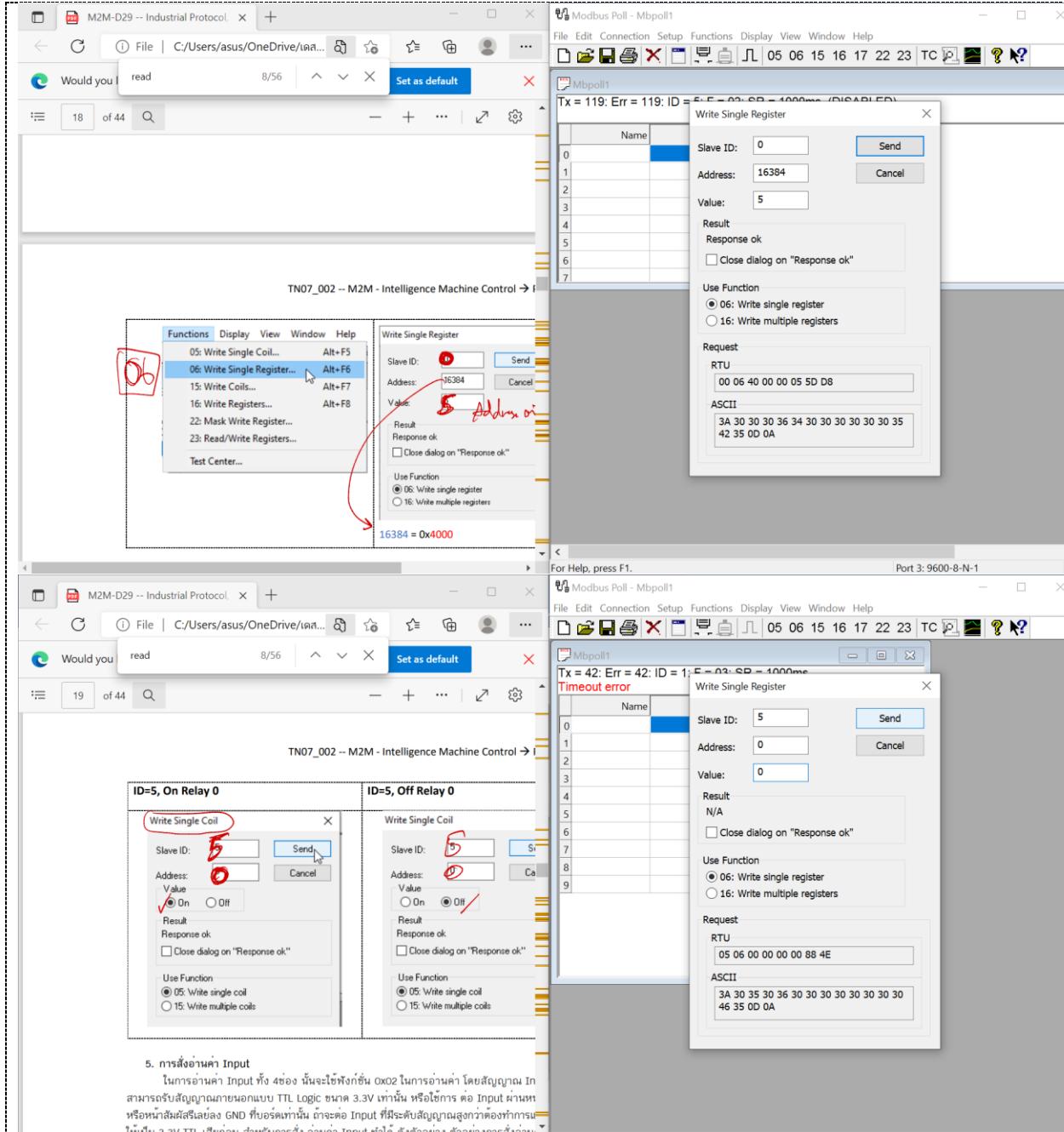


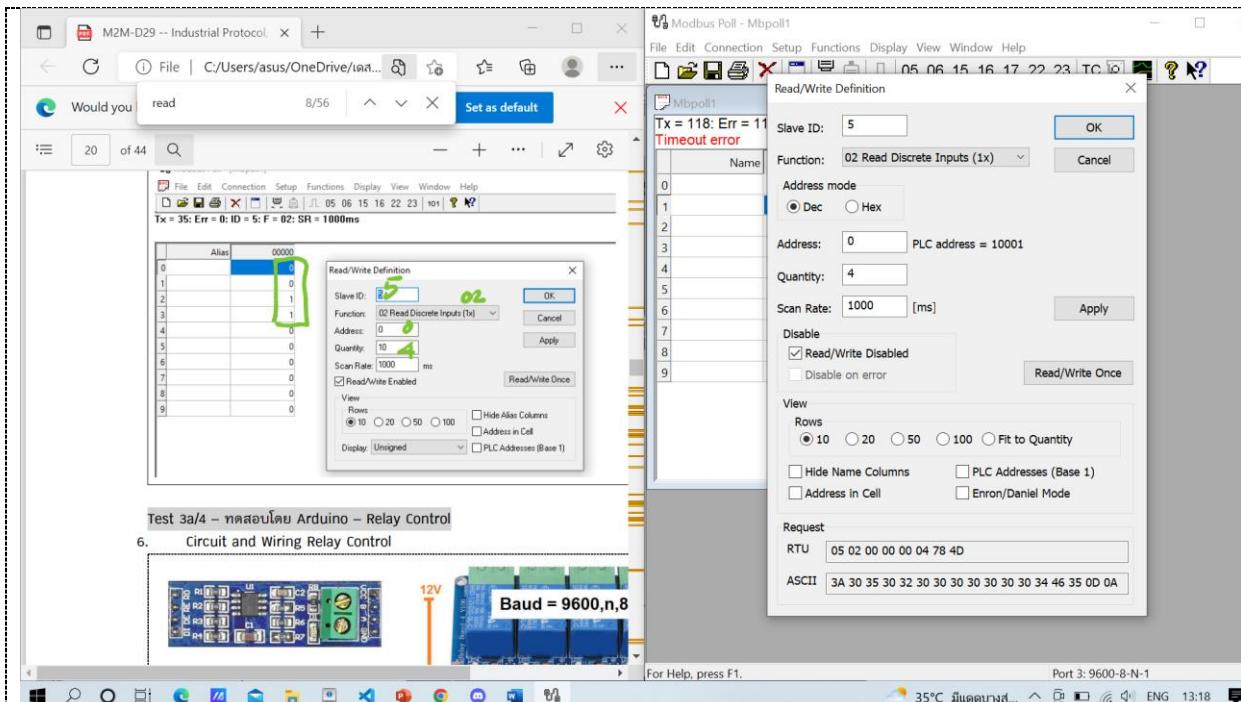
< รูปอุปกรณ์ที่ใช้ทดสอบ ขณะทำการทดสอบ >



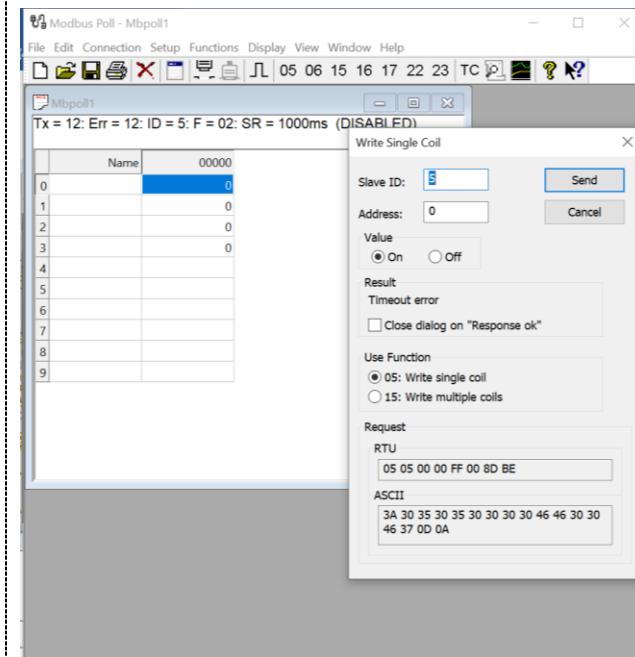
< โปรแกรมทดสอบ >

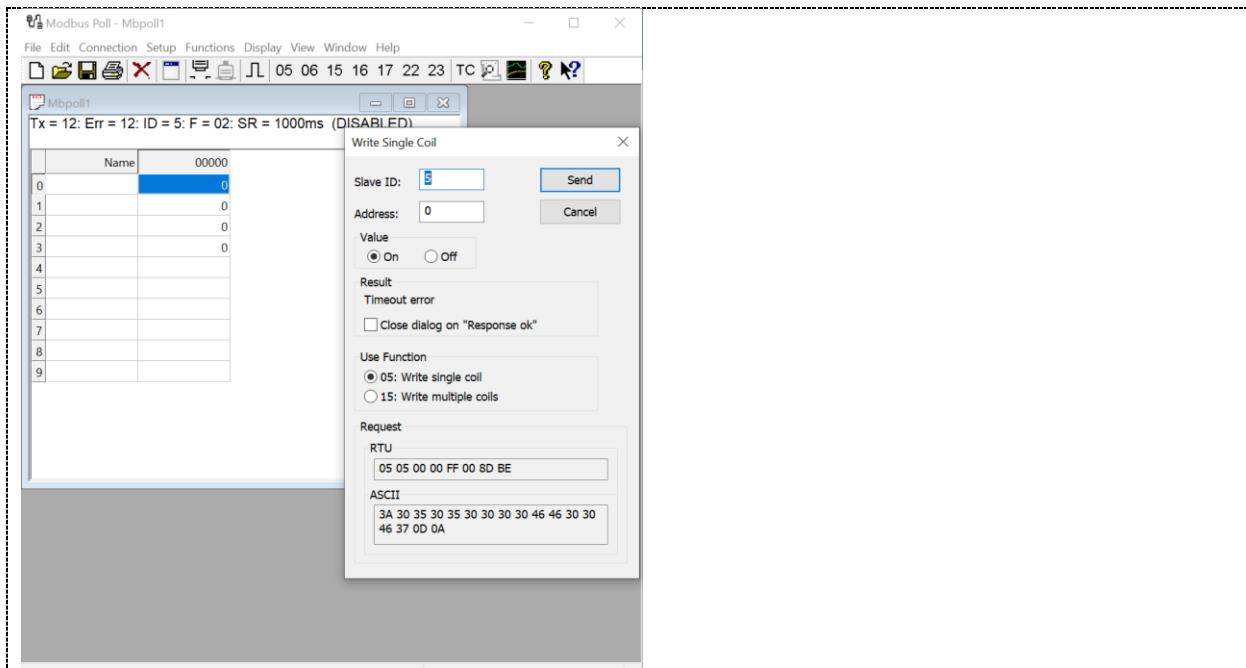






&lt; ผลการทดสอบ &gt;

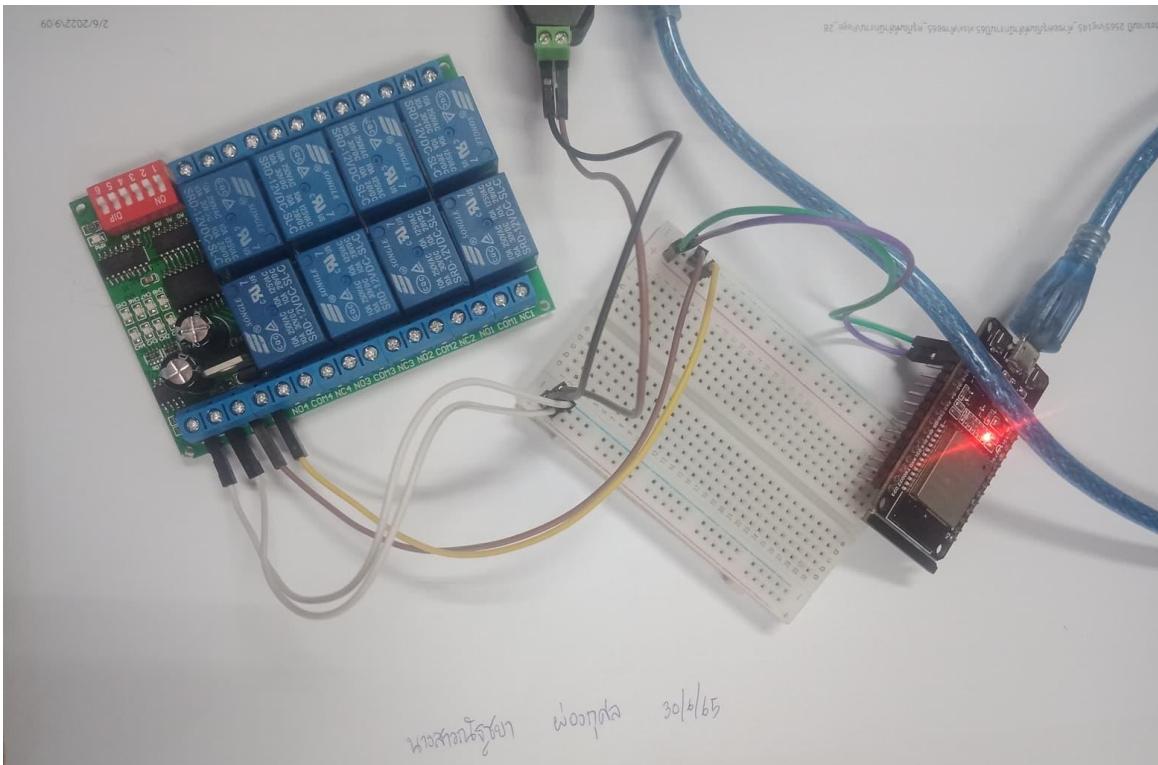




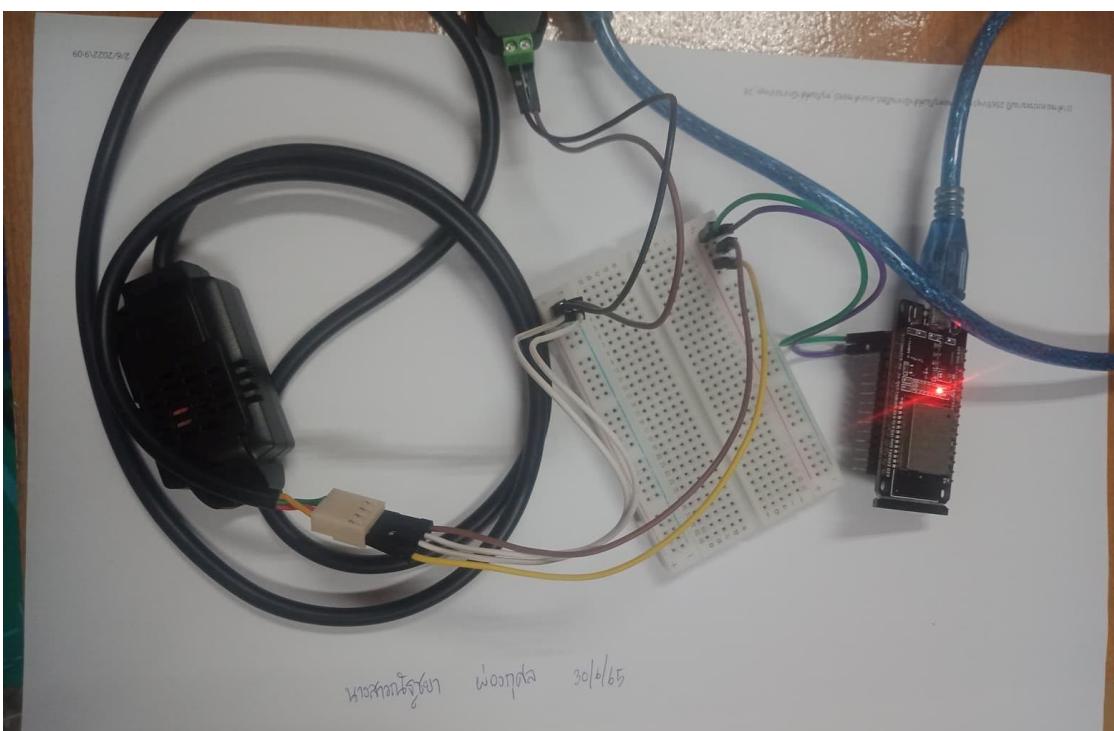
**Quiz\_202 – Write Modbus RTU**

< รูปอุปกรณ์ที่ใช้ทดสอบ ขณะทำการทดสอบ >

ของ Modbus Relay

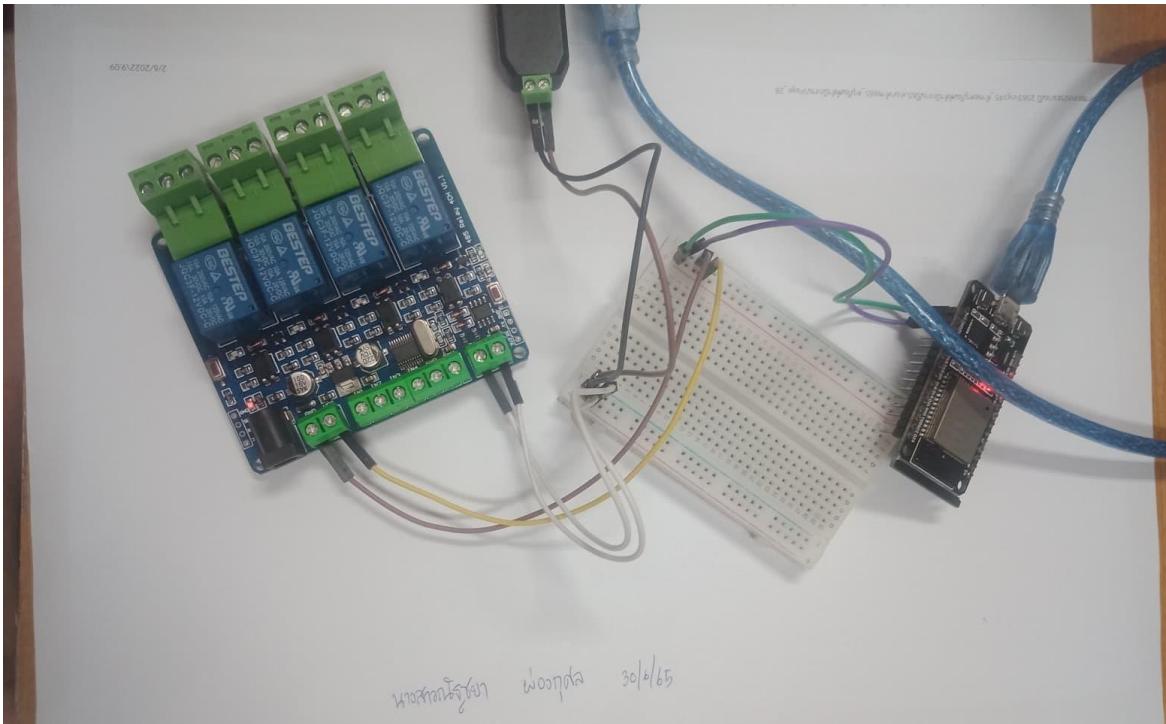


ของ Modbus Senser



&lt; รูปอุปกรณ์ที่ใช้ทดสอบ ขณะทำการทดสอบ &gt;

ของ modbus #3



&lt; โปรแกรมทดสอบ &gt;

ของ modbus RTU SENSER

Windows Taskbar:

- M2M-D29 -- Industrial
- M2M-D42.pdf
- File | C:/Users/asus/OneDrive/เอกสาร...
- Set as default
- 28 of 44
- Search
- ...

Modbus Poll - Mbpoll1

File Edit Connection Setup Functions Display View Window Help

05 06 15 16 17 22 23 | TC | P | Q | ? | K | N |

Would you like to write? 38/90

Read ID#1

Read/Write Definition

Slave ID: 1 Function: 03 Read Holding Registers (4x) Address: 0 Quantity: 2 Scan Rate: 1000 ms  Read/Write Enabled  Read/Write Disabled  Disable on error  Read/Write Once

View

Rows: 10 20 50 100 Fit to Quantity  Hide Name Columns  PLC Addresses (Base 1)  Address in Cell  Enron/Daniel Mode

Request

RTU: 01 03 00 00 00 02 C4 0B

ASCII: 3A 30 31 30 33 30 30 30 30 30 30 30 32 46 41 0D 0A

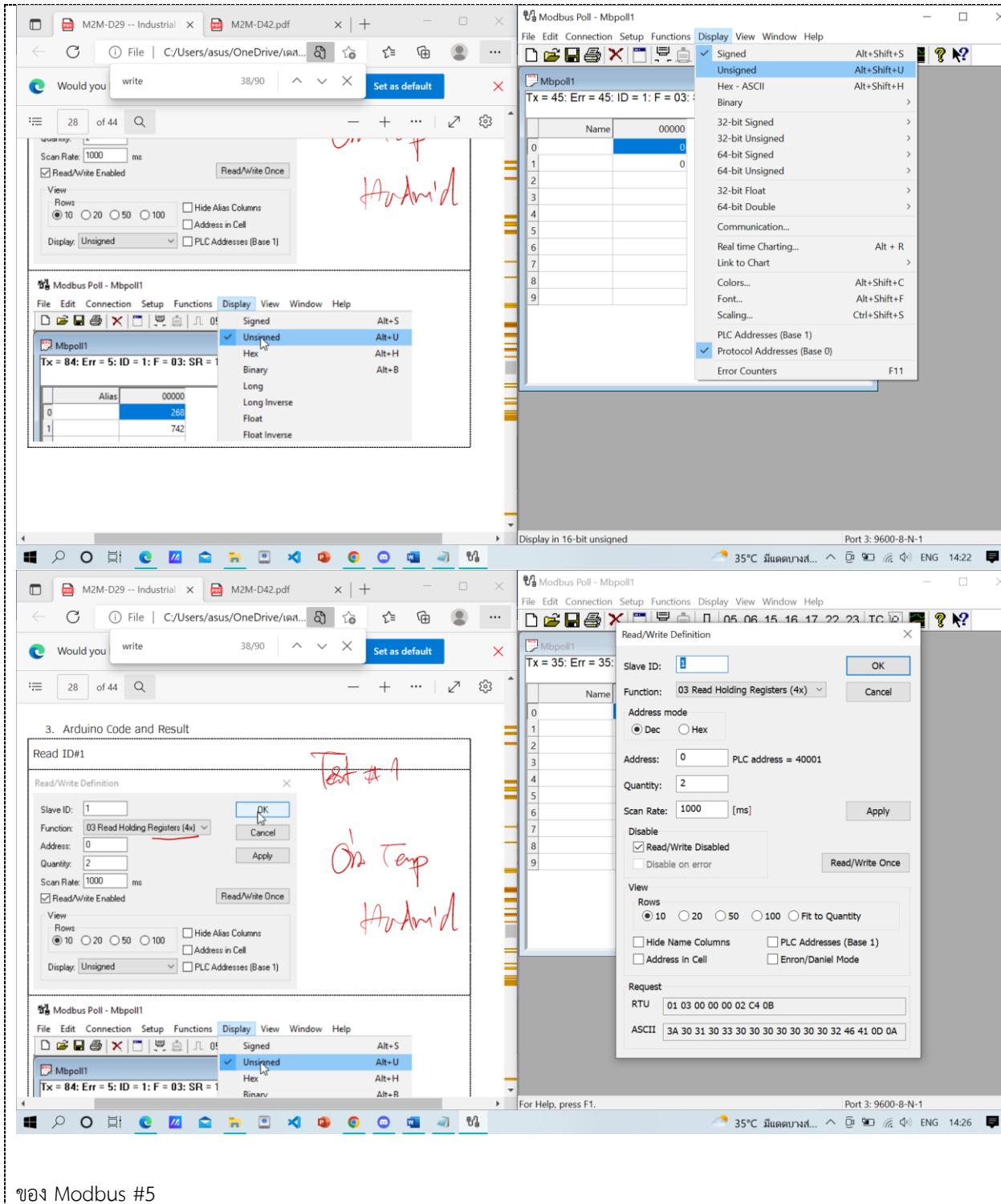
For Help, press F1. Port 3: 9600-8-N-1

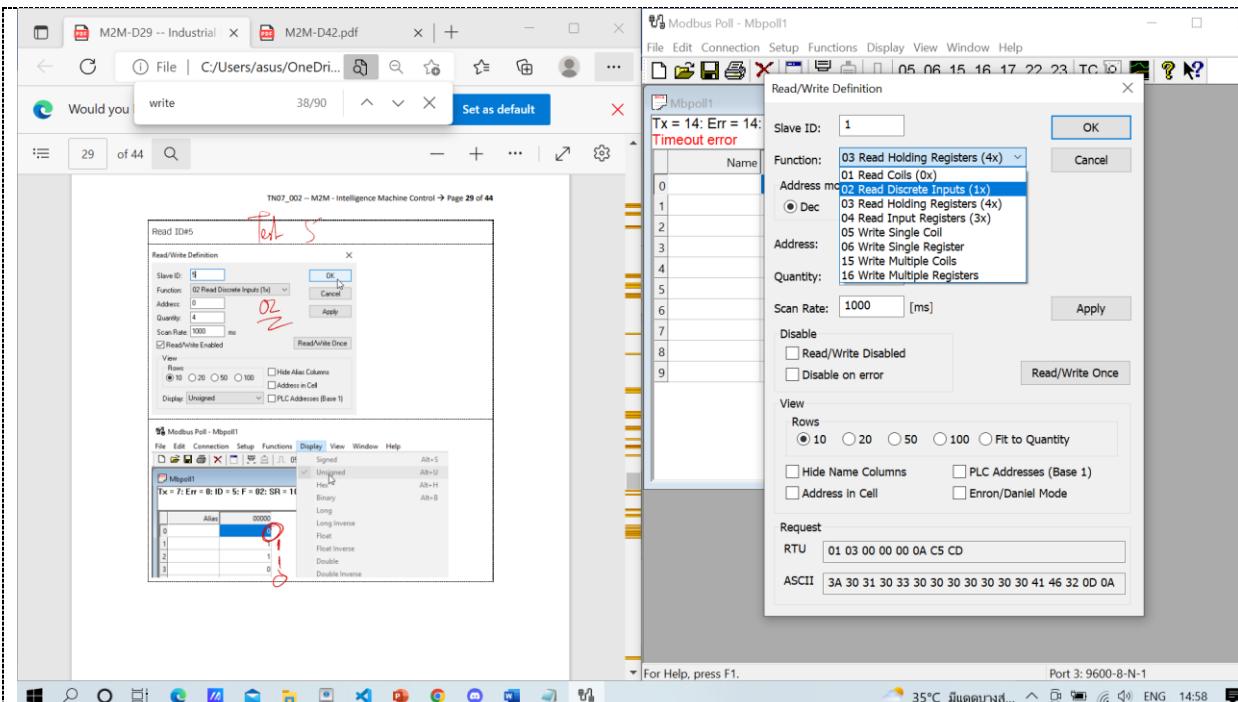
35°C ฝั่งด้านหลัง... ENG 14:22

*Test #1*

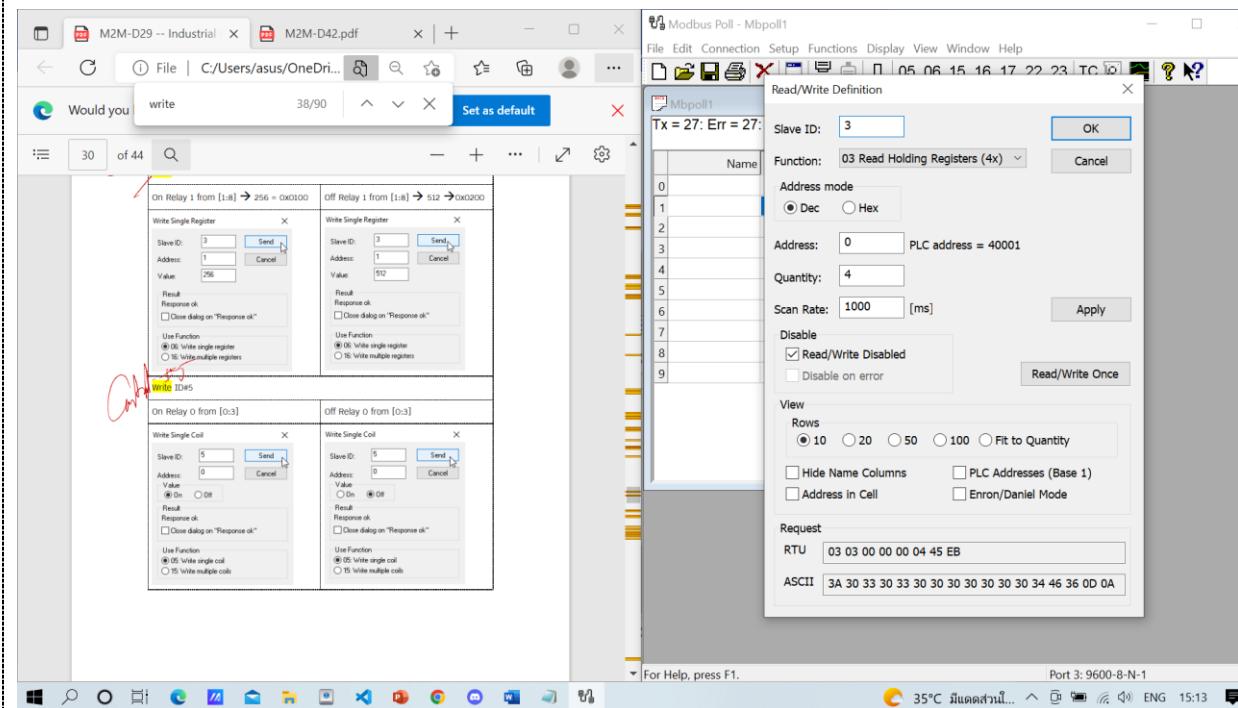
*On Temp*

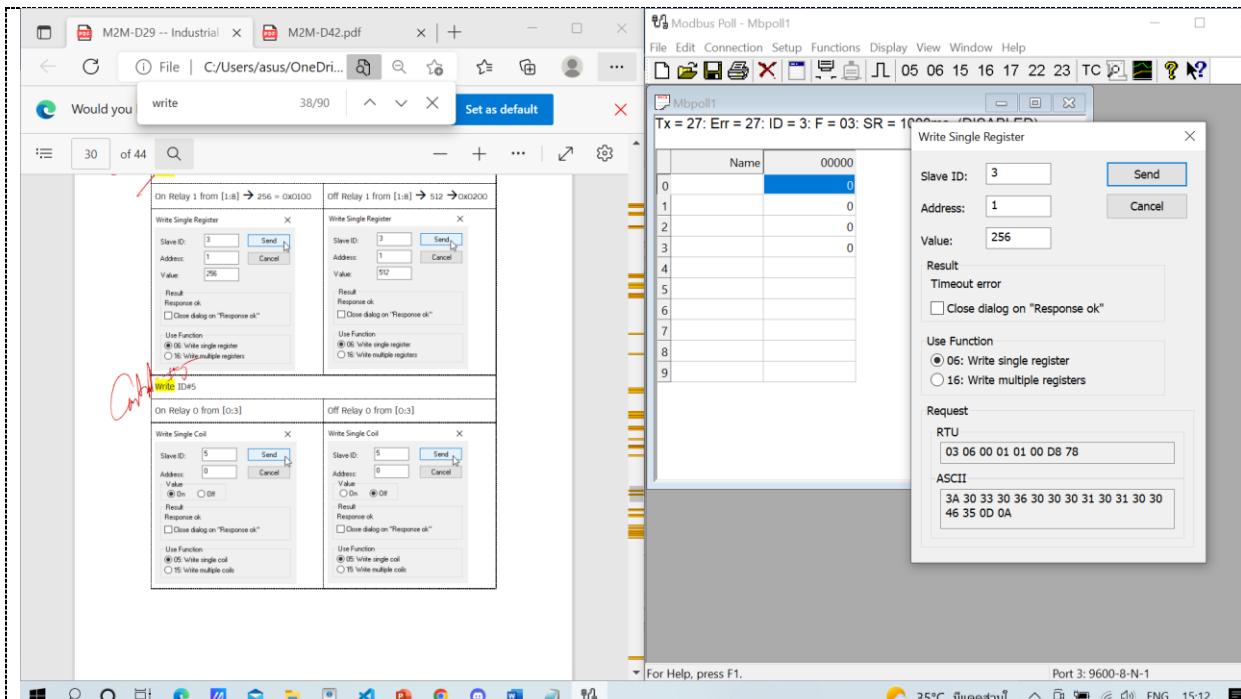
*Hamid*





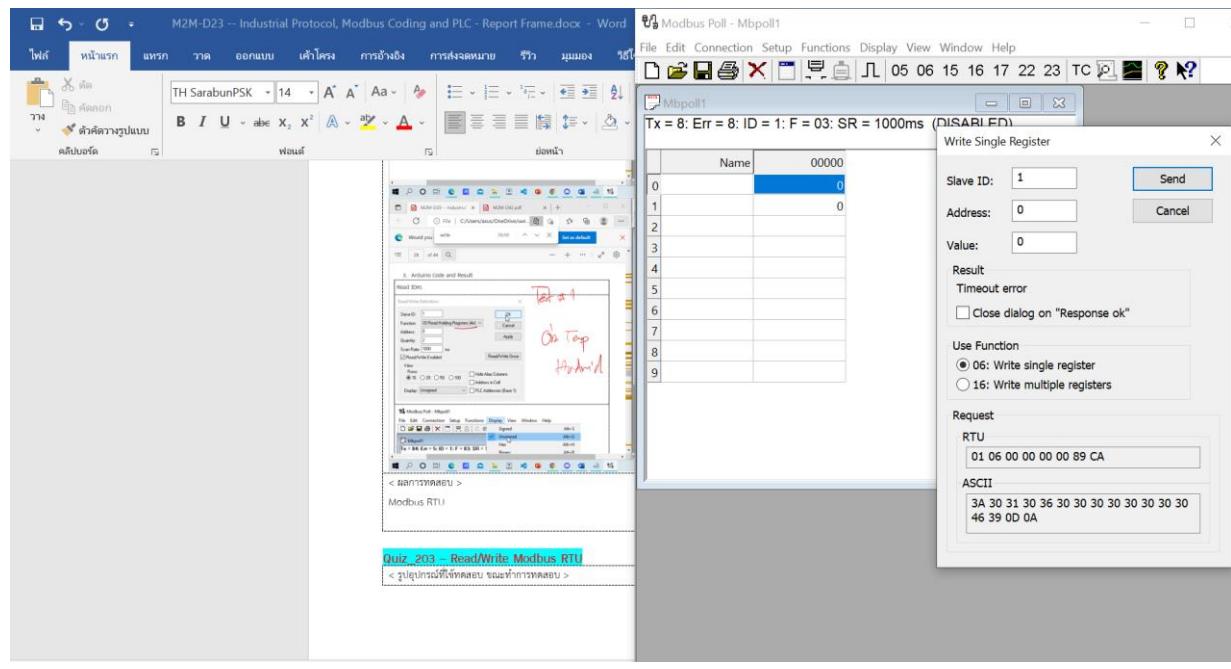
## ข้อง 1 modbus RTU RELAY



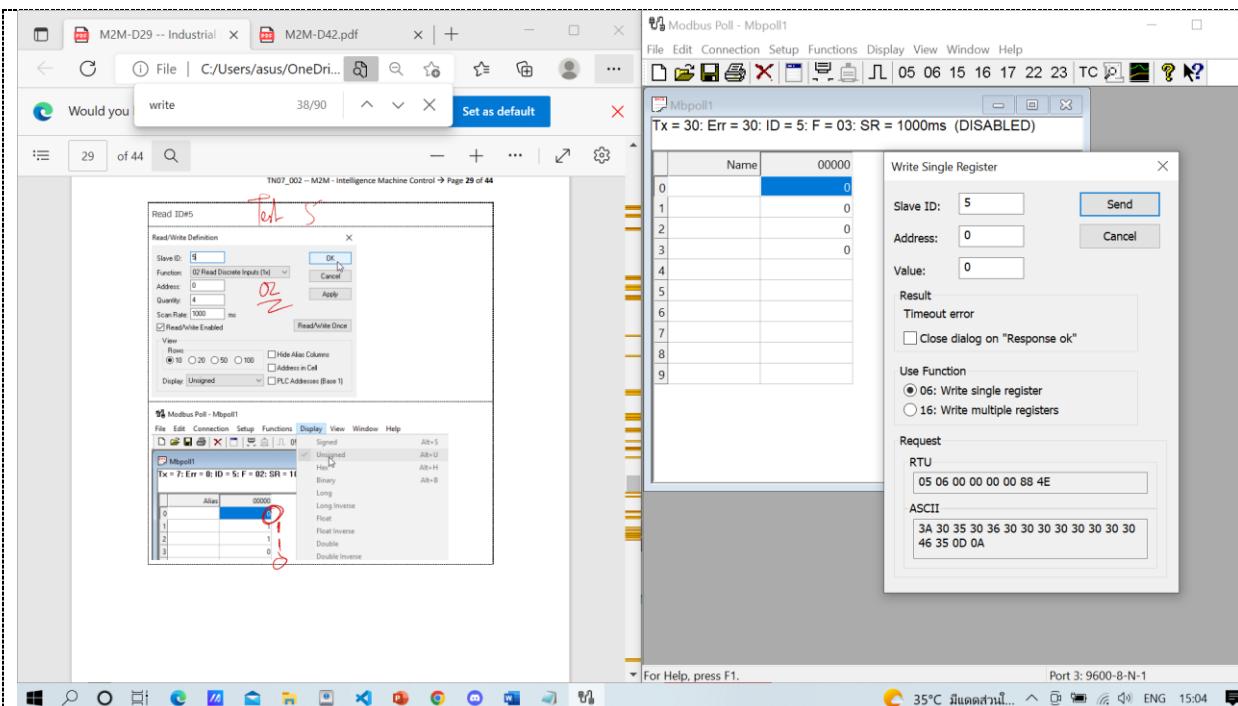


&lt; ผลการทดสอบ &gt;

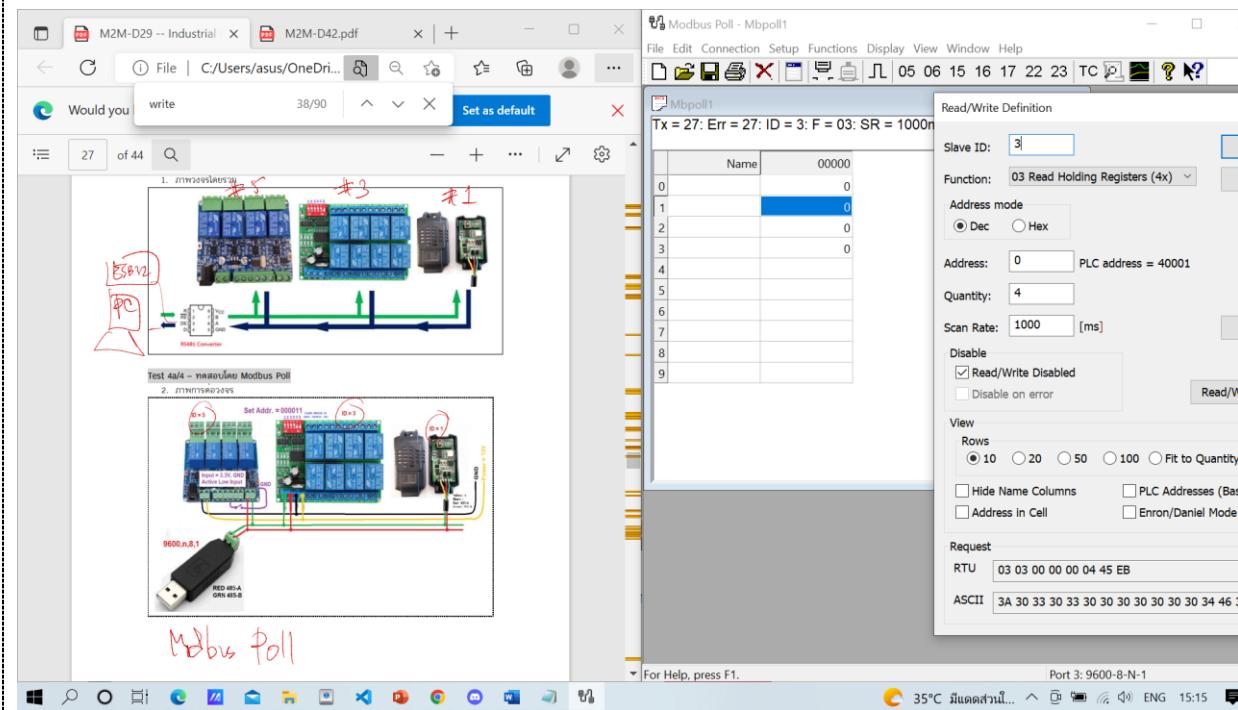
## Modbus RTU



ข้อจง modbus #5

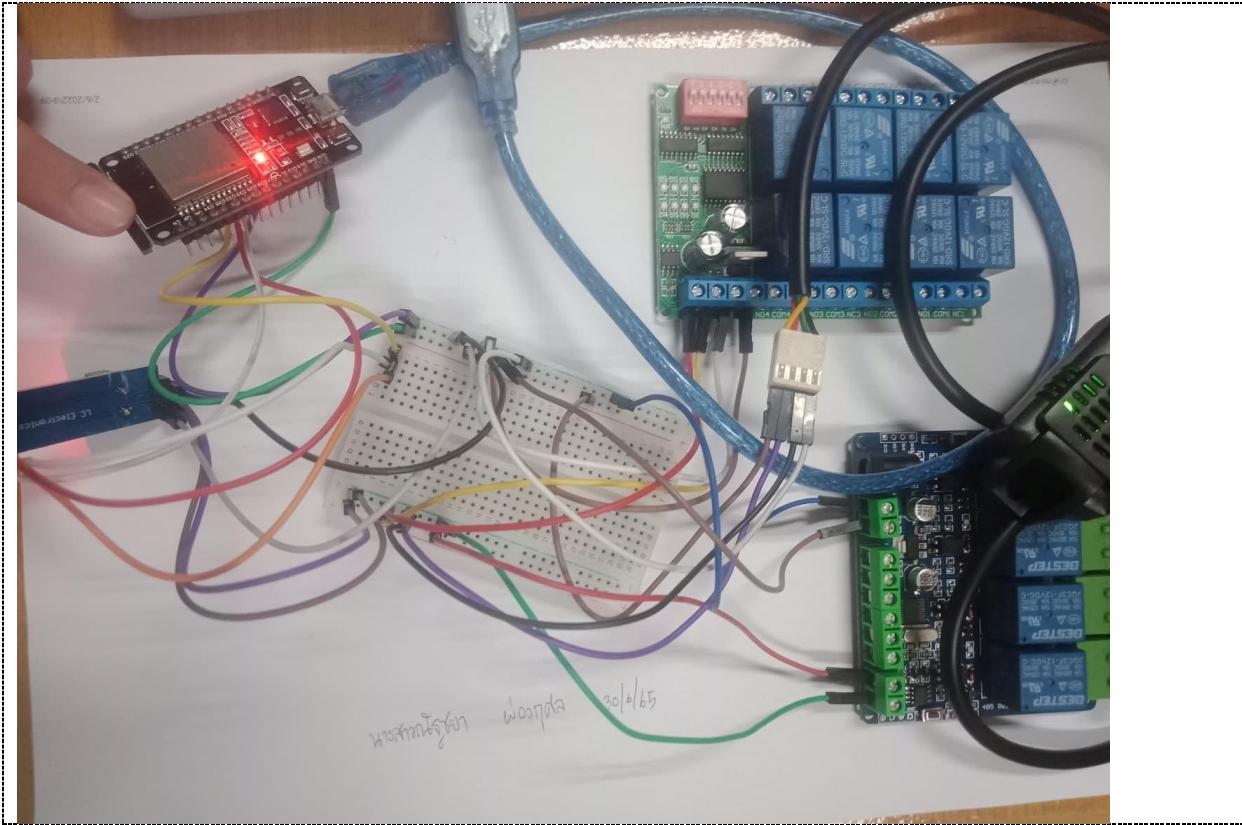


### - ของ Modbus Relay

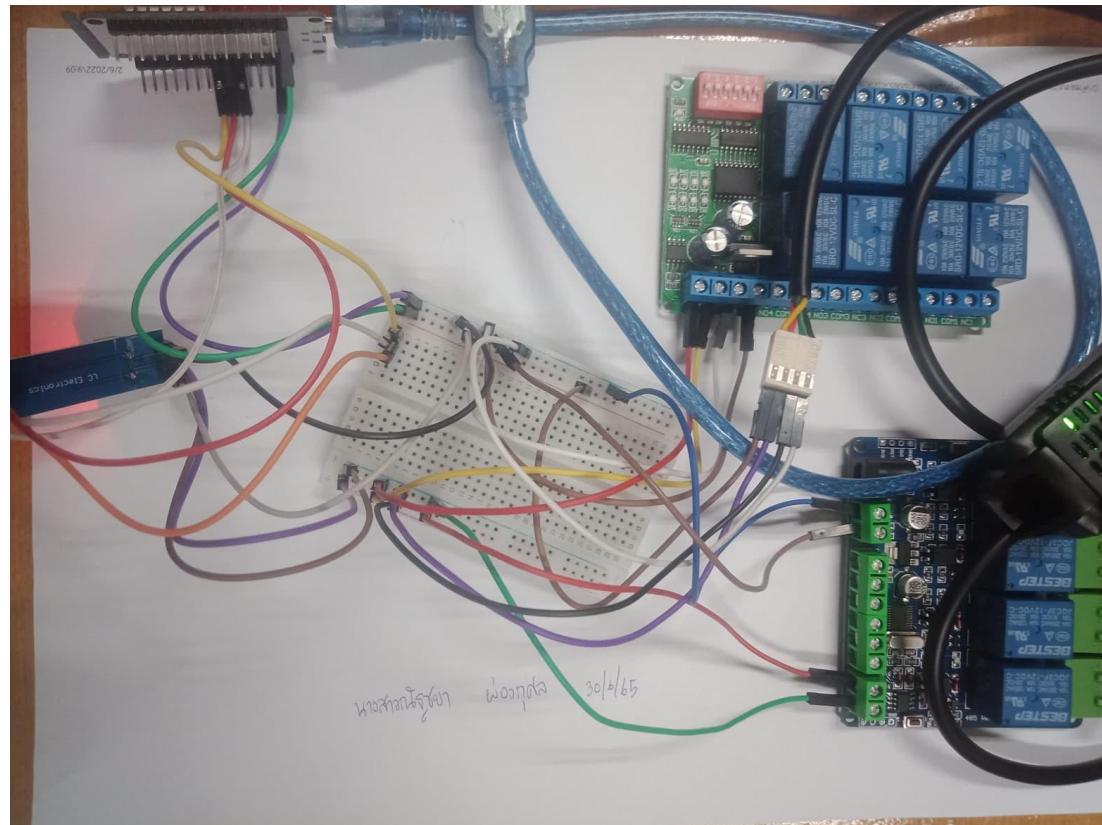


### Quiz\_203 – Read/Write Modbus RTU

< รูปอุปกรณ์ที่ใช้ทดสอบ ขณะทำการทดสอบ >



< รูปอุปกรณ์ที่ใช้ทดสอบ ขณะทำการทดสอบ >



< โปรแกรมทดสอบ >

D1 | Arduino 1.8.13

File Edit Sketch Tools Help

```

D1

1 #include <ModbusMaster.h>
2 #define RS485Transmit HIGH
3 #define RS485Receive LOW
4 #define RS485Control 4 //RS485 Direction control
5 #define Pin_LEDMonitor 2
6 #define Slave_Sensor_ID 1
7 #define Slave_Relay8_ID 3
8 #define Slave_Ry4In4_ID 5
9 int state = 0;
10 float CTempp, Hudmid;
11 bool DgInput0, DgInput1, DgInput2, DgInput3;
12 ModbusMaster node_Sensor;
13 ModbusMaster node_Relay8;
14 ModbusMaster node_Ry4In4;
15 void preTransmission() {
16   digitalWrite(RS485Control, RS485Transmit);
17 }
18 void postTransmission() {
19   digitalWrite(RS485Control, RS485Receive);
20 }
21 void setup() {
22   pinMode(RS485Control, OUTPUT);
23   pinMode(Pin_LEDMonitor, OUTPUT);
24   Serial.begin(115200);
25   Serial2.begin(9600);
26   postTransmission();
#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_Relay8_ID 3
#define Slave_Ry4In4_ID 5
int state = 0;
float CTempp, Hudmid;
bool DgInput0, DgInput1, DgInput2, DgInput3;
ModbusMaster node_Sensor;
ModbusMaster node_Relay8;
ModbusMaster node_Ry4In4;
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_Relay8.begin(Slave_Relay8_ID, Serial2); // Modbus slave ID=3
    node_Relay8.preTransmission(preTransmission);
    node_Relay8.postTransmission(postTransmission);
    node_Ry4In4.begin(Slave_Ry4In4_ID, Serial2); // Modbus slave ID=5
    node_Ry4In4.preTransmission(preTransmission);
    node_Ry4In4.postTransmission(postTransmission);
}

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(0x0000, 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 ReadDigitalInput(void) {
    uint8_t result;
    // Toggle the coil at address (Manual Load Control)
    result = node_Ry4In4.writeSingleCoil(Slave_Sensor_ID, state);
    state = !state;
```

```
// Read 4 registers starting at 0x0000)
result = node_Ry4In4.readDiscreteInputs(0, 4); // Start=0, nByte=4
if (result == node_Ry4In4.ku8MBSuccess) {
    int DgTemp = node_Ry4In4.getResponseBuffer(0x00);
    DgInput3 = (DgTemp >> 3) & 1;
    DgInput2 = (DgTemp >> 2) & 1;
    DgInput1 = (DgTemp >> 1) & 1;
    DgInput0 = (DgTemp >> 0) & 1;
}
}

void RelayControl(int inputCase) {
    int rnMode = inputCase / 10;
    int nRelay = inputCase % 10;
    if (rnMode == 81) node_Relay8.writeSingleRegister(nRelay, 0x0100); // On RelayX
    if (rnMode == 80) node_Relay8.writeSingleRegister(nRelay, 0x0200); // Off RelayX
    if (rnMode == 41) node_Ry4In4.writeSingleRegister(nRelay, 0x0100); // On RelayX
    if (rnMode == 40) node_Ry4In4.writeSingleRegister(nRelay, 0x0000); // Off RelayX
}

void loop() {
    ReadTemperature();
    ReadDigitalInput();
    Serial.print("\n Temp(C): 26.99"); Serial.print(CTempp, 2);
    Serial.print(", Humid(%): 33.5"); Serial.print(Humid, 2);
    Serial.print(", Sensor[0:3]: 0-1-1-1"); Serial.print(DgInput3);
    Serial.print("-"); Serial.print(DgInput2);
    Serial.print("-"); Serial.print(DgInput1);
    Serial.print("-"); Serial.print(DgInput0);
    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 >

```

COM3
ets Jun  8 2016 00:22:57

rst:0x1 (POWERON_RESET),boot:0x13 (SPI_FAST_FLASH_BOOT)
config:ip: 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

TempC: 26.99, Humid(%): 33.5, Sensor[0:3]: 0-1-1-1

Autoscroll Show timestamp
Carriage return 115200 baud Clear output

```

### Quiz\_204 – PLC Test [ยังไม่ต้องทำ]

< รูปอุปกรณ์ที่ใช้ทดสอบ ขณะทำการทดสอบ >

< รูปอุปกรณ์ที่ใช้ทดสอบ ขณะทำการทดสอบ >

< โปรแกรมทดสอบ >

< ผลการทดสอบ >