

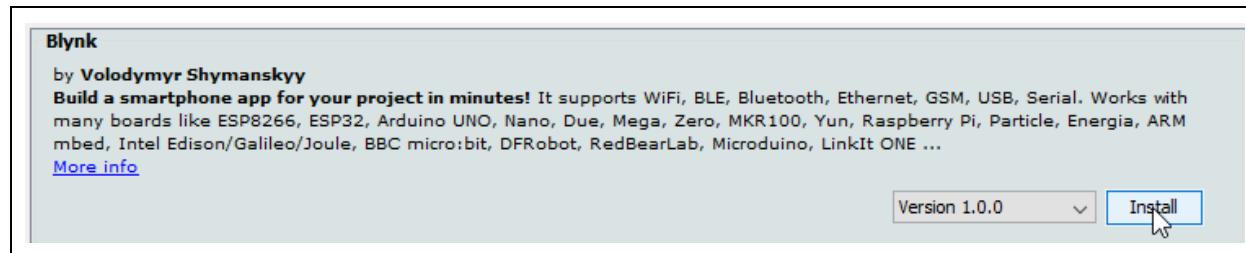
**การควบคุมเครื่องจักรอัจฉริยะโดยใช้การสื่อสารระหว่างเครื่องจักรกับเครื่องจักร
M2M - Intelligence Machine Control**

4/4 – Control and Monitor Modbus Device via The IoTs Platform

- การโปรแกรมใช้งาน Blynk เพื่อตรวจสอบ/สั่งงาน อุปกรณ์ Modbus
- การโปรแกรมใช้งาน Ubidots เพื่อตรวจสอบ/สั่งงาน อุปกรณ์ Modbus
- คำแนะนำท้ายบทเพื่อทดสอบความเข้าใจ

การโปรแกรมใช้งาน Blynk เพื่อตรวจสอบ/สั่งงาน อุปกรณ์ Modbus**Test 1/6. Remote Control ESP-32 via Blynk**

- ตรวจสอบให้แน่ใจว่ามี Blynk Library เรียบร้อยแล้ว



- ทดสอบการควบคุมผ่าน Blynk ไปยัง Build in LED

```
#define BLYNK_PRINT Serial
#include <WiFi.h>
#include <WiFiClient.h>
#include <BlynkSimpleEsp32.h>

char auth[] = "YD3FmnLek5vdhs-BeQIWwrACl8gXNgXK";
char ssid[] = "Mue.Home";
char pass[] = "pk1212312121";

#define testLED 2
#define testSW 0

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);
}

void loop() {
    Blynk.run();
    delay(100);
}
```

Draw!

Draw!

3. ทดสอบอ่านค่า SW0 และควบคุม D2 พร้อมกันผ่าน Blynk

```
#define BLYNK_PRINT Serial
#include <WiFi.h>
#include <WiFiClient.h>
#include <BlynkSimpleEsp32.h>

char auth[] = "YD3FmnLekSvdhs-BeQIWwrACI8gXNgXK";
char ssid[] = "Mue.Home";
char pass[] = "pk1212312121";

#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);
}

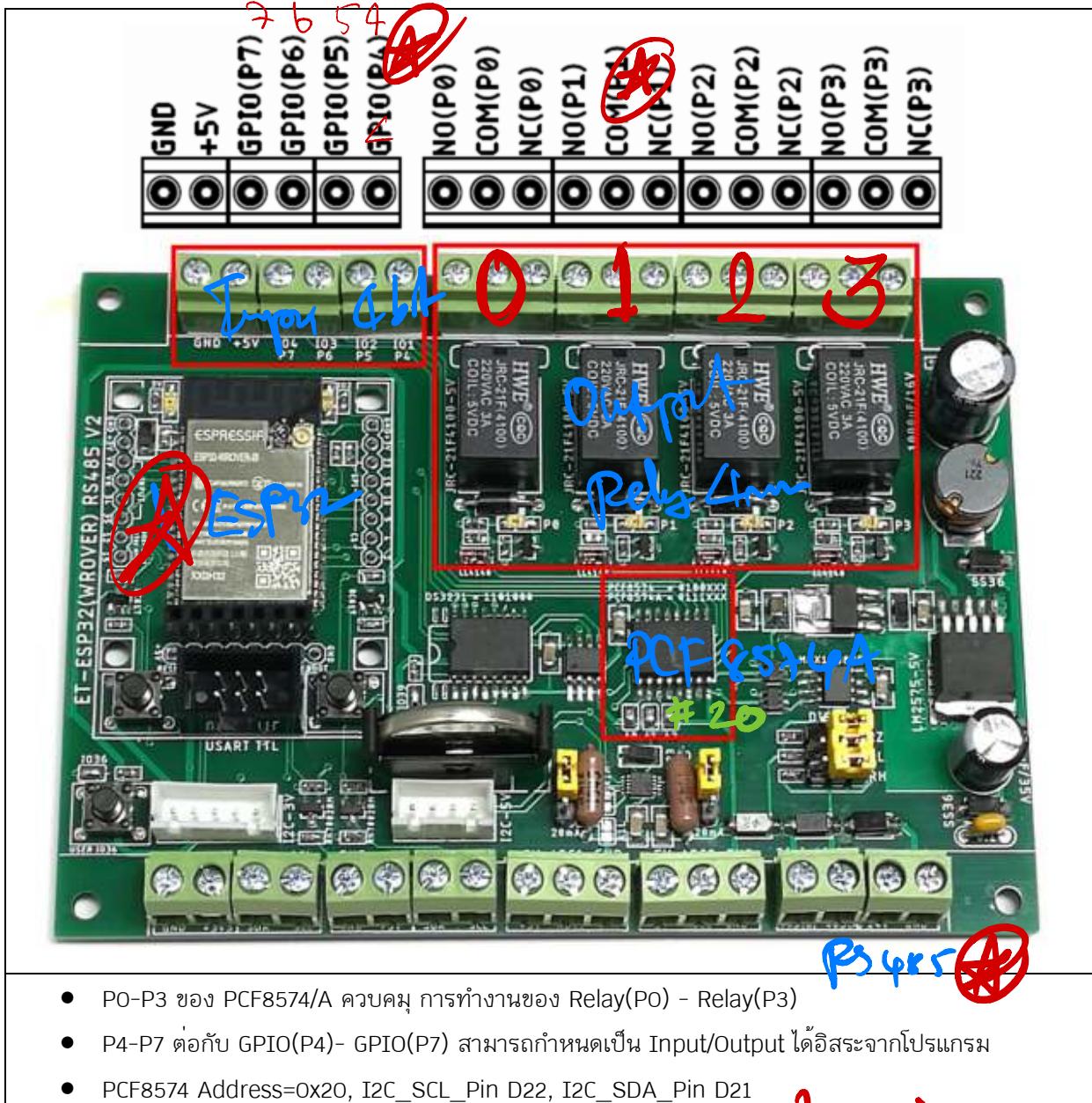
int loopCount = 10;
void loop() {
    Blynk.run();
    if (loopCount < 0) {
        loopCount = 20;
    }
    //int stsTestSW = digitalRead(testSW);
    int stsTestSW = random(2);
    Serial.println("stsTestSW = " + String(stsTestSW));
    if (stsTestSW == 0)
        LED_V4.off();
    else
        LED_V4.on();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

Red side

Jf

Test 2/6. Remote Control and Monitor ET-ESP32-RS485 V2 Board via Blynk

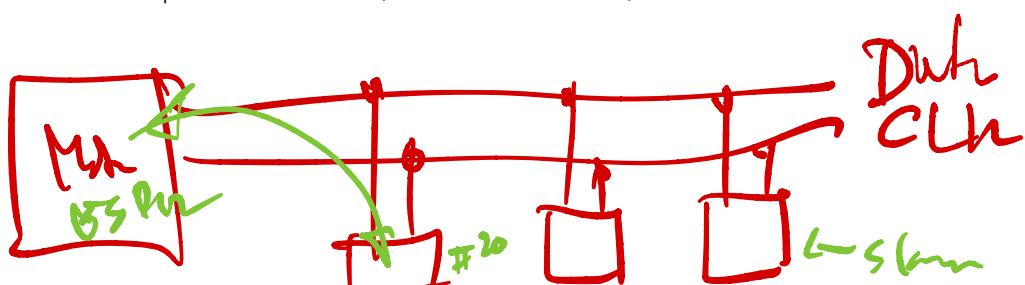
4. บอร์ด ET-ESP32-RS485 V2 เป็นบอร์ดที่เหมาะสมในการแปลงการสื่อสารระหว่าง Modbus กับ IoT เนื่องจาก มีตัวชี้วัดทางสื่อสารทำให้เรียกใช้งานได้ทันที



5. Add WROVER Chip ให้ Arduino IDE (หากกำหนดแล้วให้ข้าม)

$I^2C \Rightarrow SCA, SCL$

- File → Preference ... กำหนด https://dl.espressif.com/dl/package_esp32_index.json
- Tools → Board → Board Manager ... ทำการเพิ่ม esp32 บอร์ด
- Add WROVER Chip ให้ Arduino IDE (หากกำหนดแล้วให้ข้าม)



6. เลือกบอร์ดเป็น ESP32 Wrover Module ทดสอบการทำงานด้วย Blink Example Code

```
#define testLED 2
void setup() {
    pinMode(testLED, OUTPUT);
}

void loop() {
    digitalWrite(testLED, HIGH); delay(1000);
    digitalWrite(testLED, LOW); delay(1000);
}
```

Blynk

7. ควบคุมการทำงานของ Relay

```
#include <Wire.h>

#define PCF8574_Addr 0x20
#define I2C_SCL_Pin 22
#define I2C_SDA_Pin 21
#define testLED 2

void setup() {
    pinMode(testLED, OUTPUT);
    Wire.begin(I2C_SDA_Pin, I2C_SCL_Pin);
}

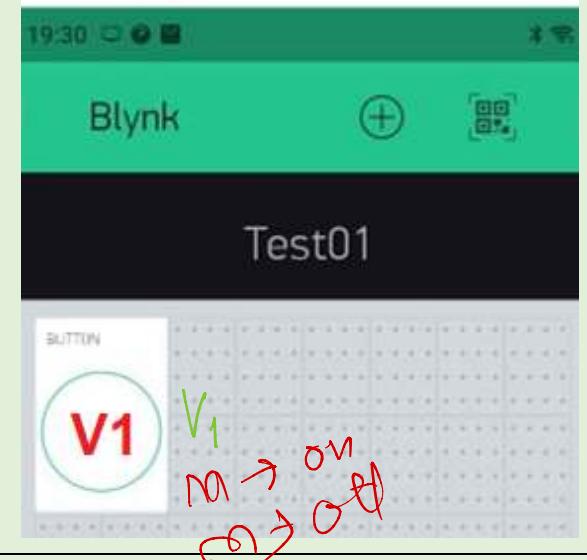
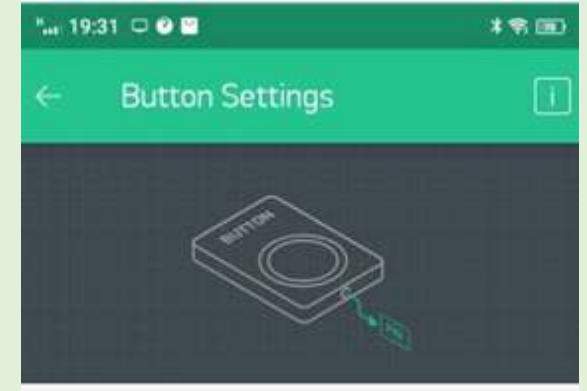
void loop() {
    digitalWrite(testLED, HIGH);
    Wire.beginTransmission(PCF8574_Addr);
    Wire.write(0xAA);
    Wire.endTransmission();
    delay(1000);

    digitalWrite(testLED, LOW);
    Wire.beginTransmission(PCF8574_Addr);
    Wire.write(0x55);
    Wire.endTransmission();
    delay(1000);
}
```

Annotations:

- Handwritten note: "Addr in I²C" with an arrow pointing to the line "#define PCF8574_Addr 0x20".
- Handwritten note: "pin in I²C" with an arrow pointing to the line "#define I2C_SCL_Pin 22".
- Handwritten note: "int (I²C)" with an arrow pointing to the "Wire" object.
- Handwritten note: "Addr" with an arrow pointing to the "addr" parameter in the "beginTransmission" call.
- Handwritten note: "0010 1010" with an arrow pointing to the value "0xAA".
- Handwritten note: "0010 0101" with an arrow pointing to the value "0x55".
- Handwritten note: "AA" with an arrow pointing to the value "0xAA".
- Handwritten note: "DD" with an arrow pointing to the value "0x55".

8. ทดสอบการควบคุมผ่าน Blynk ไปยัง PCF8517 Relay_O

Button

OUTPUT

V1 0 1

MODE

PUSH **SWITCH**

```
#define BLYNK_PRINT Serial
#include <WiFi.h>
#include <WiFiClient.h>
#include <BlynkSimpleEsp32.h>
#include <Wire.h>

#define PCF8574_Adr 0x20
#define I2C_SCL_Pin 22
#define I2C_SDA_Pin 21
#define testLED 2

char auth[] = "GzWPpqdufO2I4WFA4kT4FjTh_AT";      // Token Key
char ssid[] = "Test1234";                          // AP Name
char pass[] = "0816601234";                        // Wifi-Password

byte statusRelay = 0x0ff; // All Off

void RelayUpdate(int idRelay, int stsRelay) {
    byte stsRelayTemp;
    Serial.print("Relay_");
    Serial.print(idRelay);
    if (stsRelay == 1) {
        stsRelayTemp = ~ (1 << idRelay);
        statusRelay |= stsRelayTemp;
        Serial.println("Force: On");
    }
    else {
        stsRelayTemp = 1 << idRelay;
        statusRelay &= stsRelayTemp;
        Serial.println("Force: Off");
    }
    Wire.beginTransmission(PCF8574_Adr);
    Wire.write(statusRelay);
    Wire.endTransmission();
}

BLYNK_WRITE(V1) {
    int pinValue = param.asInt();
    RelayUpdate(0, pinValue);
    digitalWrite(testLED, pinValue);
}

void setup() {
    Serial.begin(115200);
    pinMode(testLED, OUTPUT);
    Blynk.begin(auth, ssid, pass);
    Wire.begin(I2C_SDA_Pin, I2C_SCL_Pin);
}

void loop() {
    Blynk.run();
}
```

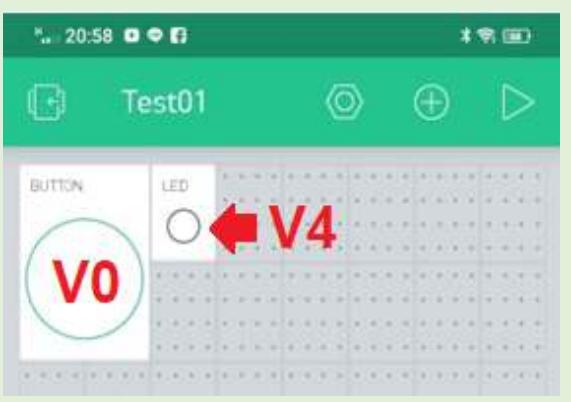
Handwritten annotations:

- A blue arrow points from the word "V1" in the Blynk app's button label to the "V1" in the BLYNK_WRITE block.
- A green arrow points from the "V1" in the BLYNK_WRITE block to the "V1" in the Blynk app's button label.
- A red handwritten note "Relay #1 not in Error" is placed near the BLYNK_WRITE block.

9. ปรับแก้เป็นควบคุม 4-Switch สำหรับ 4-Relay

| | |
|--|---|
|  | <pre> #define BLYNK_PRINT Serial #include <WiFi.h> #include <WiFiClient.h> #include <BlynkSimpleEsp32.h> #include <Wire.h> #define PCF8574_Addr 0x20 #define I2C_SCL_Pin 22 #define I2C_SDA_Pin 21 #define testLED 2 char auth[] = "GzWPpqdufO2I4WFA4kT4fJTh_AT"; // Token Key char ssid[] = "Test1234"; // AP Name char pass[] = "0816601234"; // Wifi-Password byte statusRelay = 0x0ff; // All Off void RelayUpdate(int idRelay, int stsRelay) { byte stsRelayTemp; Serial.print("Relay_"); Serial.print(idRelay); if (stsRelay == 1) { stsRelayTemp = ~(1 << idRelay); statusRelay &= stsRelayTemp; Serial.println("Force: On"); } else { stsRelayTemp = 1 << idRelay; statusRelay = stsRelayTemp; Serial.println("Force: Off"); } Wire.beginTransmission(PCF8574_Addr); Wire.write(statusRelay); Wire.endTransmission(); } BLYNK_WRITE (V0) { int pinValue = param.asInt(); RelayUpdate(0, pinValue); digitalWrite(testLED, pinValue); } BLYNK_WRITE (V1) { int pinValue = param.asInt(); RelayUpdate(1, pinValue); } BLYNK_WRITE (V2) { int pinValue = param.asInt(); RelayUpdate(2, pinValue); } BLYNK_WRITE (V3) { int pinValue = param.asInt(); RelayUpdate(3, pinValue); } void setup() { Serial.begin(115200); pinMode(testLED, OUTPUT); Blynk.begin(auth, ssid, pass); Wire.begin(I2C_SDA_Pin, I2C_SCL_Pin); } void loop() { Blynk.run(); } </pre> |
|--|---|

10. ทดสอบอ่านค่า GPIO(P4) และควบคุม Relay-0 พร้อมกันผ่าน Blynk



The screenshot shows the Blynk mobile application interface. At the top, it displays "Test01". Below that, there are two widgets: a "BUTTON" labeled "V0" and an "LED" labeled "V4". A red arrow points from the "V4" label towards the right-hand code area.

```

#define BLYNK_PRINT Serial
#include <WiFi.h>
#include <WiFiClient.h>
#include <BlynkSimpleEsp32.h>
#include <Wire.h>

#define PCF8574_Addr 0x20
#define I2C_SCL_Pin 22
#define I2C_SDA_Pin 21
#define testLED 2

char auth[] = "GzWPpqdufO2I4WFA4kT4FjTh_AT"; // Token Key
char ssid[] = "Test1234"; // AP Name
char pass[] = "0816601234"; // Wifi-Password

byte statusRelay = 0x0ff; // All Off

WidgetLED Input4(V4);
BlynkTimer timer;

//-----
void expanderWrite(byte _data) {
    Wire.beginTransmission(PCF8574_Addr);
    Wire.write(_data);
    Wire.endTransmission();
}

//-----
byte expanderRead() {
    byte _data;
    Wire.requestFrom(PCF8574_Addr, 1);
    if (Wire.available()) {
        _data = Wire.read();
    }
    return _data;
}

//-----
void RelayUpdate(int idRelay, int stsRelay) {
    byte stsRelayTemp;
    Serial.print("Relay_");
    Serial.print(idRelay);
    if (stsRelay == 1) {
        stsRelayTemp = ~(1 << idRelay);
        statusRelay &= stsRelayTemp;
        Serial.println("Force: On");
    } else {
        stsRelayTemp = 1 << idRelay;
        statusRelay |= stsRelayTemp;
        Serial.println("Force: Off");
    }
    expanderWrite(statusRelay);
}

//-----
BLYNK_WRITE(V0) {
    int pinValue = param.asInt();
    RelayUpdate(0, pinValue);
    digitalWrite(testLED, pinValue);
}

//-----
void loopReadInput()
{
    byte stsRead = expanderRead();
    if ((stsRead >> 4) & 1){
        Input4.off(); Serial.println("Input V4: off");
    } else {
        Input4.on(); Serial.println("Input V4: on");
    }
}

//=====
void setup() {
    Serial.begin(115200);
    pinMode(testLED, OUTPUT);
    Blynk.begin(auth, ssid, pass);
    Wire.begin(I2C_SDA_Pin, I2C_SCL_Pin);
    timer.setInterval(1000L, loopReadInput);
}

void loop() {
    Blynk.run();
    timer.run();
}

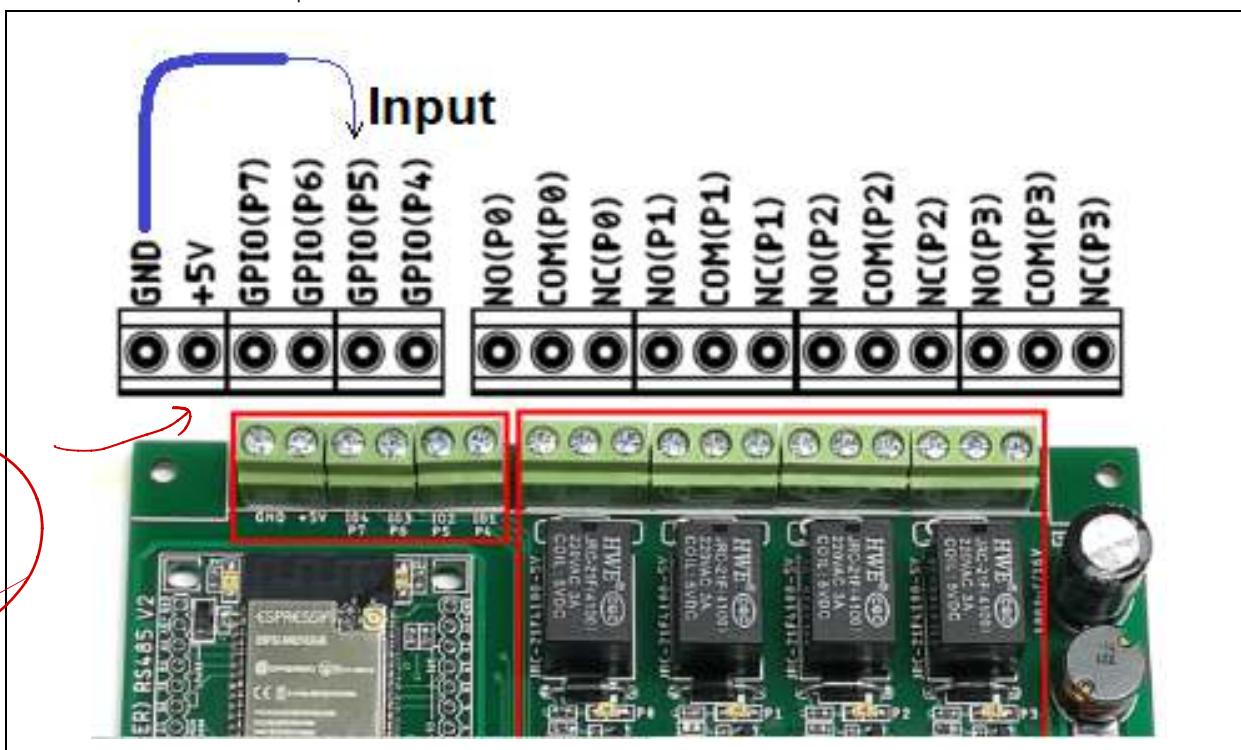
```

The code is written in C++ and includes definitions for WiFi connection, Blynk library, and I2C communication. It handles button presses on the Blynk app and updates an external relay via an expander chip connected to pins 22 and 21. The Blynk app also features a "LED Settings" screen with three LED icons and a "LED" input field containing "V4". A red arrow points from the "V4" label in the app to the corresponding line of code in the sketch.

A red box highlights the line "if ((stsRead >> 4) & 1){". Above this box, a red annotation "#20" is written above the line "Input4.off();". To the right of the box, another red annotation "min 164" is written above the line "Serial.println("Input V4: off");".

A large red arrow originates from the "V4" label in the Blynk app and points down to the "BLYNK_WRITE(V0)" function in the code. Another red arrow points from the "Input4.on();" line in the "loopReadInput()" function to the "1000ms" handwritten note at the bottom right of the code area.

11. การอ่านค่าจาก Input Connector



Test 3/7. Remote monitor and control Modbus RTU device via Blynk

12. ให้แน่ใจว่าติดตั้ง ModbusMaster V 2.0.1 ของ Doc Walker



13. ทดสอบอ่าน MODBUS RTU SENSOR H/T

```
// ET-ESP32(WROVER).RS485.V2
// Read MODBUS RTU SENSOR H/T

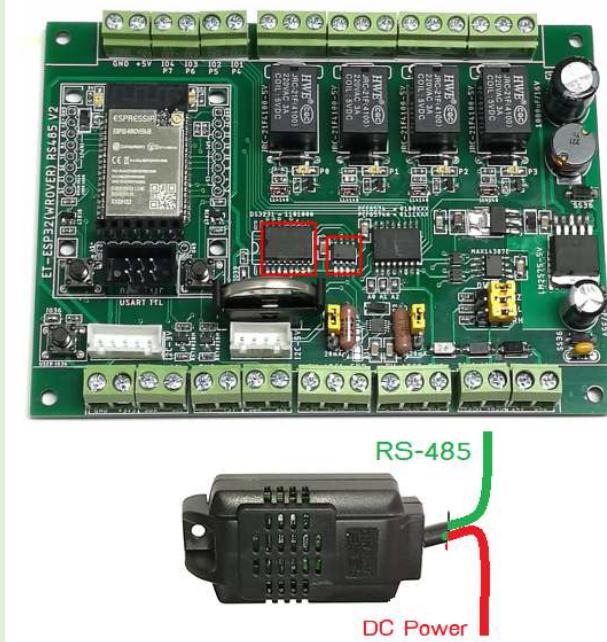
#include <ModbusMaster.h>
#define Rx2_Pin 26
#define Tx2_Pin 27

#define ModbusID 1
#define StartAdd 0
#define nWordRead 2

ModbusMaster node;

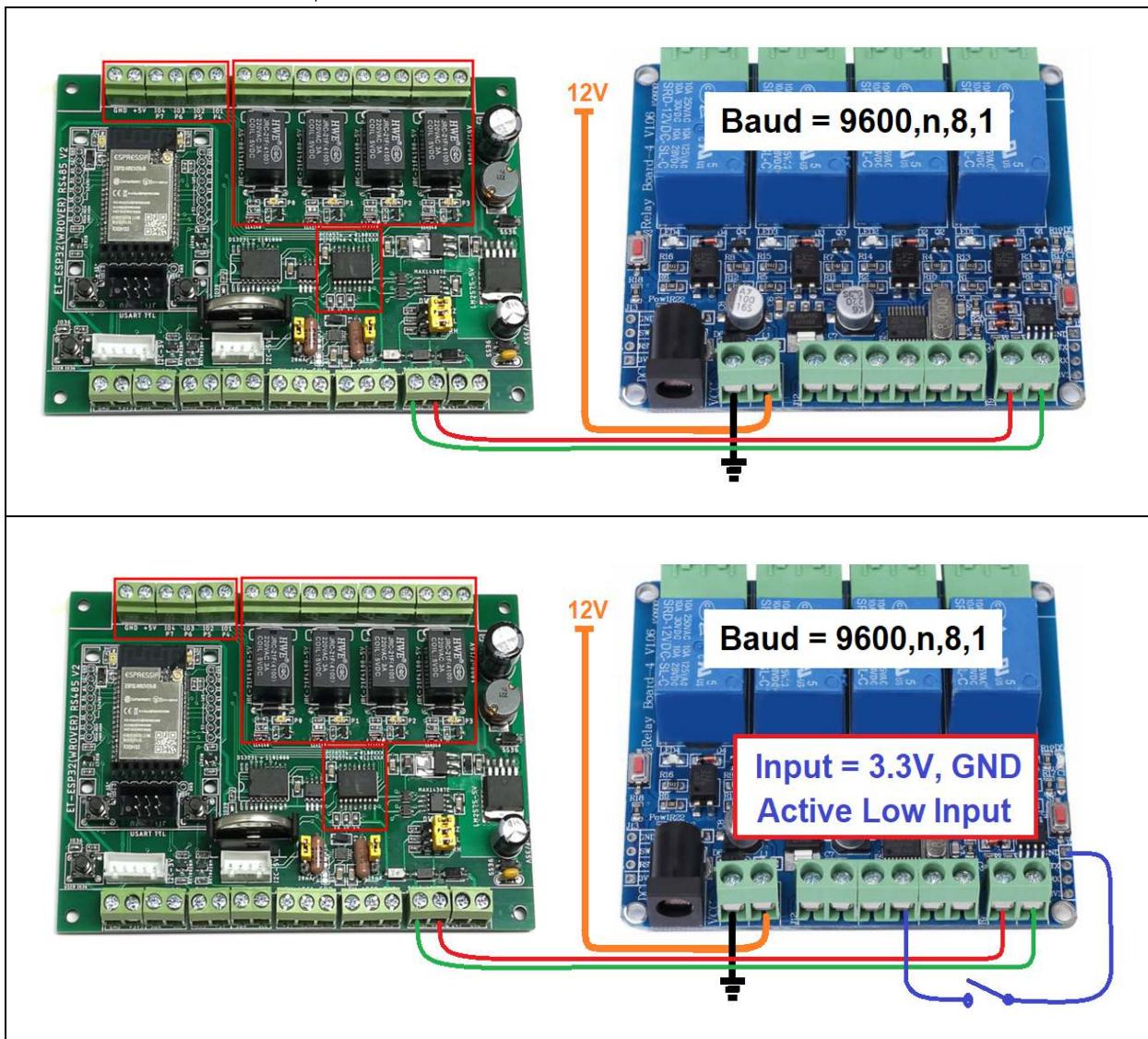
void setup() {
  Serial.begin(115200);
  Serial2.begin(9600, SERIAL_8N1, Rx2_Pin, Tx2_Pin);
  node.begin(ModbusID, Serial2);
}

void loop() {
  uint16_t data[6];
  int result = node.readHoldingRegisters(StartAdd, nWordRead);
  if (result == node.ku8MBSuccess) {
    for (int j = 0; j < nWordRead; j++)
      data[j] = node.getResponseBuffer(j);
    float Tempp = data[0] / 10.0;
    float Humid = data[1] / 10.0;
    Serial.println();
    Serial.println("Tempp = " + String(Tempp, 1) + "C");
    Serial.println("Humid = " + String(Humid, 1) + "%");
  }
  delay(10000);
}
```



```
Tempp = 28.9'C
Humid = 58.9%
Tempp = 28.9'C
Humid = 58.6%
```

14. วงจรทดสอบการควบคุม และวงจรการอ่านค่า กับ Modbus Device “MODBUS RTU RELAY4/IN4”



15. ทดสอบการควบคุม Modbus Device แบบ Direct Code ไม่ใช้ Library และแบบใช้ Modbus Master Lib.

| | |
|---|---|
| <pre> // ET-ESP32(WROVER).RS485.V2 // Control MODBUS RTU RELAY4/IN4 #include <ModbusMaster.h> #define Rx2_Pin 26 #define Tx2_Pin 27 #define HTSensor_ID 1 #define R4I4Brd_ID 5 #define HTStartAdd 0 #define HTnWordRead 2 ModbusMaster nodeHT; ModbusMaster nodeR4I4; float Tempp, Humid; int state = 1; void Read_HT_Sensor() { uint16_t data[6]; int result = nodeHT.readHoldingRegisters(HTStartAdd, HTnWordRead); if (result == nodeHT.ku8MBSuccess) { for (int j = 0; j < HTnWordRead; j++) data[j] = nodeHT.getResponseBuffer(j); Tempp = data[0] / 10.0; Humid = data[1] / 10.0; } } void setup() { Serial.begin(115200); Serial2.begin(9600, SERIAL_8N1, Rx2_Pin, Tx2_Pin); nodeHT.begin(HTSensor_ID, Serial2); nodeR4I4.begin(R4I4Brd_ID, Serial2); } void loop() { Read_HT_Sensor(); Serial.println(); Serial.println("Tempp = " + String(Tempp, 1) + "C"); Serial.println("Humid = " + String(Humid, 1) + "%"); for (int i = 0; i <= 3; i++) { nodeR4I4.writeSingleCoil(i, state); Serial.println("Addr-" + (String)(i) + " >> " + (String)(state)); delay(2000); } state = 1 - state; } </pre> | <p style="background-color: #e0f2e0; padding: 10px;"> Tempp = 28.7'C Humid = 56.8% Addr-0 >> 1 Addr-1 >> 1 Addr-2 >> 1 Addr-3 >> 1 Tempp = 28.7'C Humid = 56.9% Addr-0 >> 0 Addr-1 >> 0 </p> |
|---|---|

16. วงจรทดสอบ การควบคุมและวางแผนการอ่านค่า กับ Modbus Device

```

// ET-ESP32(WROVER).RS485.V2
// Control MODBUS RTU RELAY4/IN4

#include <ModbusMaster.h>
#define Rx2_Pin 26
#define Tx2_Pin 27

#define HTSensor_ID 1
#define R4I4Brd_ID 5
#define HTStartAdd 0
#define HTnWordRead 2

ModbusMaster nodeHT;
ModbusMaster nodeR4I4;

float Tempp, Humid;
int DataIn4, coilState = 1, IO_State = 1;

void Read_HT_Sensor() {
    uint16_t data[6];
    int result = nodeHT.readHoldingRegisters(HTStartAdd, HTnWordRead);
    if (result == nodeHT.ku8MBSuccess) {
        for (int j = 0; j < HTnWordRead; j++)
            data[j] = nodeHT.getResponseBuffer(j);
        Tempp = data[0] / 10.0;
        Humid = data[1] / 10.0;
    }
}

void Read_R4I4_Board() {
    int result;
    uint16_t data[6];
    // Toggle the coil at address (Manual Load Control)
    result = nodeR4I4.writeSingleCoil(R4I4Brd_ID, coilState);
    coilState = 1 - coilState;
    result = nodeR4I4.readDiscreteInputs(0, 4); // Start=0, nByte=4
    if (result == nodeR4I4.ku8MBSuccess)
        DataIn4 = nodeR4I4.getResponseBuffer(0);
}

void setup() {
    Serial.begin(115200);
    Serial2.begin(9600, SERIAL_8N1, Rx2_Pin, Tx2_Pin);
    nodeHT.begin(HTSensor_ID, Serial2);
    nodeR4I4.begin(R4I4Brd_ID, Serial2);
}

void loop() {
    Serial.println();

    Read_HT_Sensor();
    Serial.println("Tempp = " + String(Tempp, 1) + "C");
    Serial.println("Humid = " + String(Humid, 1) + "%");

    Read_R4I4_Board();
    Serial.print("DataIn4 = ");
    Serial.print(DataIn4 >> 3 & 1); Serial.print(DataIn4 >> 2 & 1);
    Serial.print(DataIn4 >> 1 & 1); Serial.print(DataIn4 >> 0 & 1);

    for (int i = 0; i <= 3; i++) {
        nodeR4I4.writeSingleCoil(i, IO_State);
        Serial.print("\n Addr-" + (String)(i) + " >> " + (String)(IO_State));
        delay(2000);
    }
    IO_State = 1 - IO_State;
}

```

```

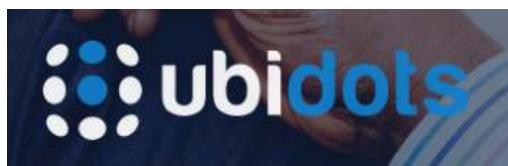
Tempp = 28.7'C
Humid = 57.0%
DataInput = 1100
Addr-0 >> 0
Addr-1 >> 0
Addr-2 >> 0
Addr-3 >> 0
Tempp = 28.7'C
Humid = 57.1%
DataInput = 1100
Addr-0 >> 1

```

การโปรแกรมใช้งาน Ubidots เพื่อตรวจสอบ/สั่งงาน อุปกรณ์ Modbus

Test 5/6. Remote Control and Monitor via Ubidots IoTs Platform

- Ubidots -- <https://ubidots.com/>



Dot platform →

- Signed Up (or Signed In) -- <https://industrial.ubidots.com/accounts/signin/>

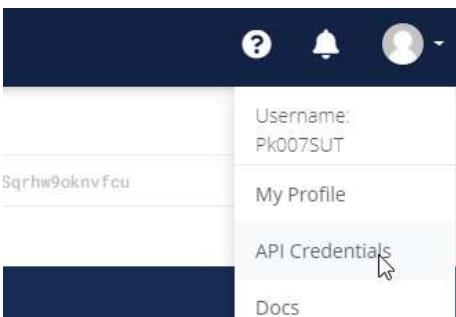
Device → Variable
Topic

- Create Device and Variable

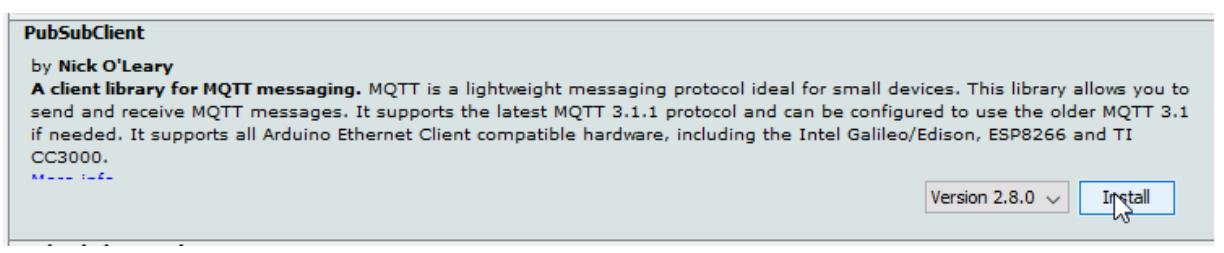
| Devices → Device | Device → Variable |
|--|---|
| <p>Devices - Data -</p> <p>Devices Groups</p> <p>Device name: pk007test (X)</p> <p>Device label: pk007test</p> <p><i>pk007test</i></p> | <p>Create Device Blank Device</p> <p>Device Name = pk007test Device Label = pk007test <all lowercase alphabet></p> |
| <p>humid</p> <p>tempp</p> <p>Last activity: No last activity</p> | <p>Click pk007test Device Add Variable → raw humid tempp <all lowercase alphabet></p> <p><i>Variable</i></p> |

| | | |
|---|---|---|
| humid | tempp | Click ... Set API Label = humid Set API Label = tempp <all lowercase alphabet> |
| Description Change description | Description Change description | |

4. Get Tokens Key

| | |
|---|-----------------|
|  | API Credentials |
| Tokens Default token BBFF-JD5UkJRKay8zKaP3TSqrhw9oknvfcu | Get Tokens Key |

5. ตรวจสอบว่าติดตั้ง PubSubClient by Nick O'Leary – V2.8.0

| |
|--|
|  |
|--|

6. Coding Send Random Data to Ubidots

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

const char *My_SSID = "Test1234";
const char *My_Pass = "0816601929";
const char *MQTT_Server = "things.ubidots.com";
const char *MQTT_User = "BBFF-YgcQcuoAiO9g46ehWh12TxVwyTjCx";
const char *MQTT_Pass = "BBFF-YgcQcuoAiO9g46ehWh12TxVwyTjCx";
const char *PTopic1 = "/v2.0/devices/pk007test"; Topic
const char *STopic1 = "/v2.0/devices/pk007test/humid"; Sub
const char *STopic2 = "/v2.0/devices/pk007test/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); sub
    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 setup()
{ Serial.begin(115200);
  Setup_Wifil();
  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;
    float xTemp = random(2000, 4000) / 100.0;
    float xHumid = random(6000, 8000) / 100.0;
    sprintf(msg, 75, "{ \"humid\" : %5.2f, \"temp\" : %5.2f}", xHumid, xTemp);
    Serial.print("Publish message: ");
    Serial.println(msg);
    client.publish(PTopic1, msg);
  }
}
Random new msg json

```

Connecting to Test1234

.....

WiFi connected

IP address:

192.168.1.22

Attempting MQTT connection...connected

Message arrived [/v2.0/devices/pk007test/humid] {"value": 69.53, "timestamp": 16259896}

Message arrived [/v2.0/devices/pk007test/temp] {"value": 32.73, "timestamp": 16259896}

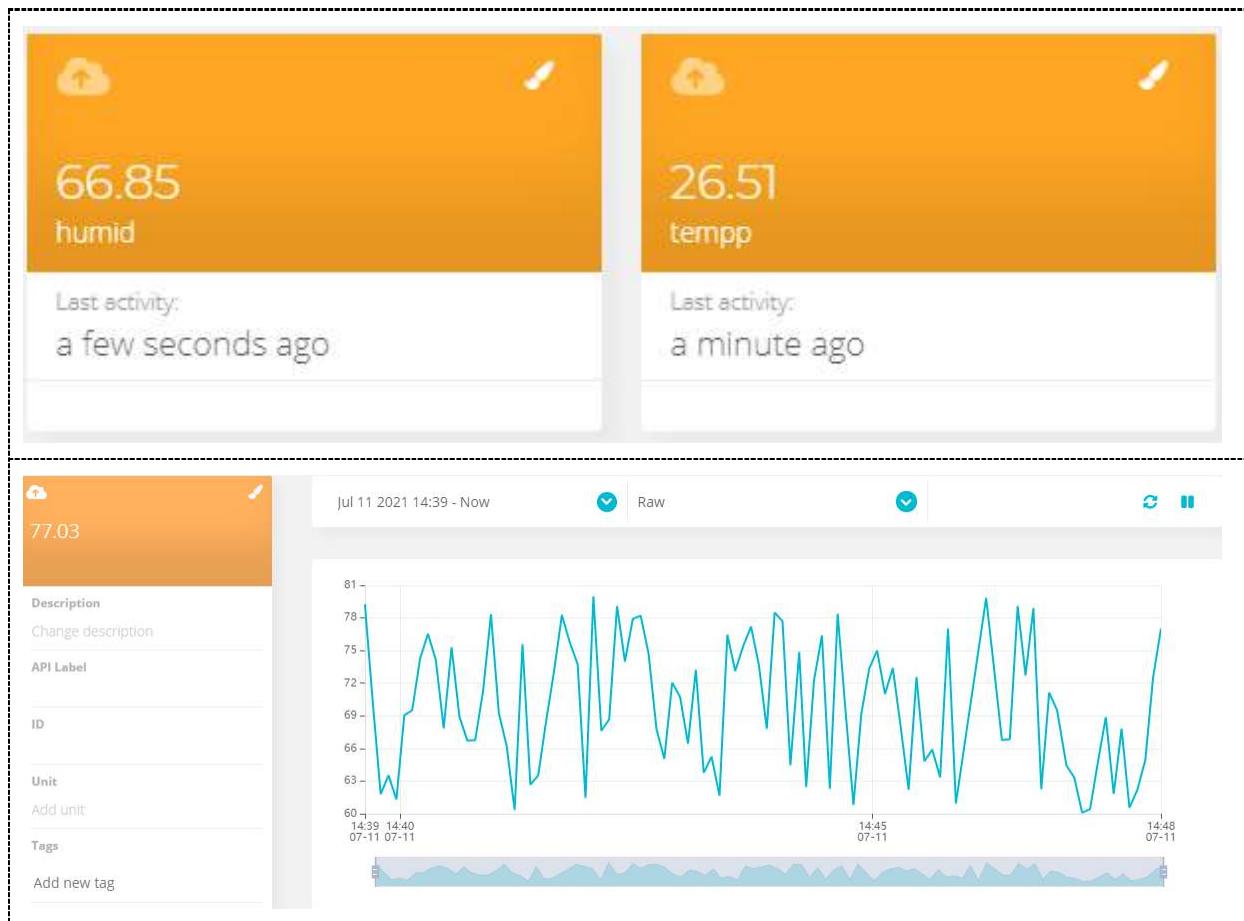
Publish message: { "humid" : 64.44, "temp" : 30.80}

Message arrived [/v2.0/devices/pk007test/humid] {"value": 64.44, "timestamp": 16259896}

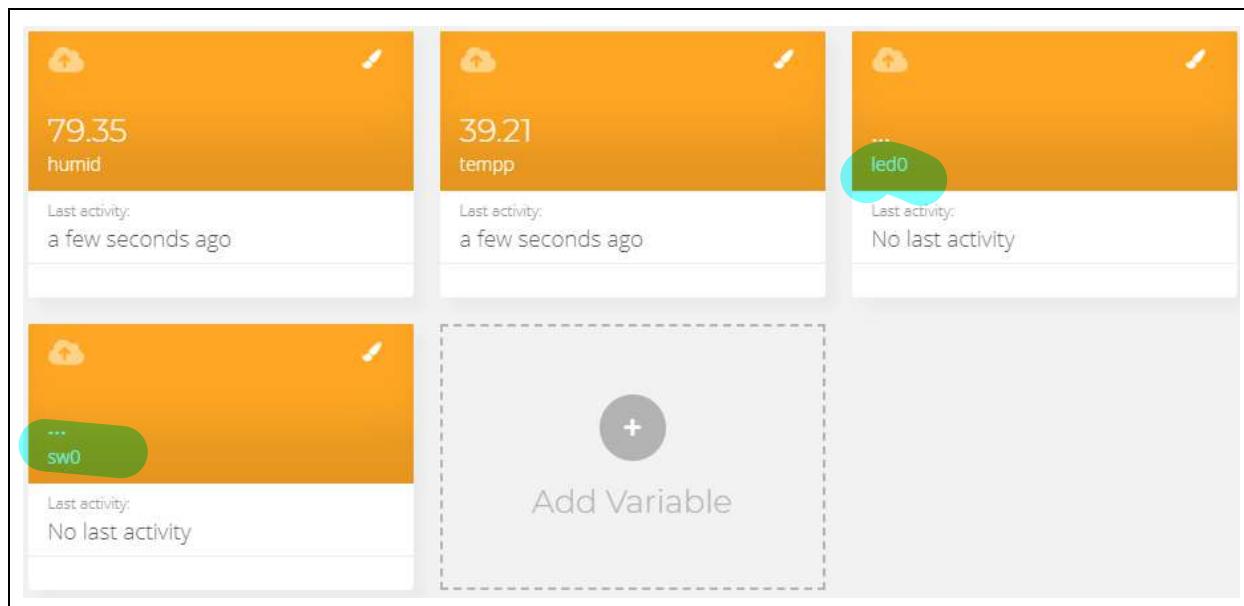
Message arrived [/v2.0/devices/pk007test/temp] {"value": 30.8, "timestamp": 16259896}

Publish message: { "humid" : 63.31, "temp" : 34.42}

below.



7. Update Code for monitor and control via Ubidots
8. Add 2 Variable → led0 and swo {ตัวเลือกเท่านั้น}



```

Message arrived [/v2.0/devices/pk007test/tempp] {"value": 32.01, "timestamp": 162599041340
Message arrived [/v2.0/devices/pk007test/sw0] {"value": 1.0, "timestamp": 1625990413404, ' Publish message: { "humid" : 72.10, "tempp": 24.22, "sw0": 0}
Message arrived [/v2.0/devices/pk007test/sw0] {"value": 0.0, "timestamp": 1625990418408, ' Publish message: { "humid" : 74.82, "tempp": 31.04, "sw0": 1}
Message arrived [/v2.0/devices/pk007test/sw0] {"value": 1.0, "timestamp": 1625990423400, ' Message arrived [/v2.0/devices/pk007test/humid] {"value": 72.1, "timestamp": 1625990418408
Message arrived [/v2.0/devices/pk007test/tempp] {"value": 31.04, "timestamp": 1625990423400
Message arrived [/v2.0/devices/pk007test/sw0] {"value": 1.0, "timestamp": 1625990423400, ' Message arrived [/v2.0/devices/pk007test/humid] {"value": 74.82, "timestamp": 1625990423400
Message arrived [/v2.0/devices/pk007test/tempp] {"value": 31.04, "timestamp": 1625990423400

```

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

```

const char *My_SSID = "Test1234";
const char *My_Pass = "0816601929";
const char *MQTT_Server = "things.ubidots.com";
const char *MQTT_User = "BBFF-YgQcuqoAiO9g46ehWh12TxVwyTjCx";
const char *MQTT_Pass = "BBFF-YgQcuqoAiO9g46ehWh12TxVwyTjCx";
const char *PTopic1 = "/v2.0/devices/pk007test";
const char *STopic1 = "/v2.0/devices/pk007test/humid";
const char *STopic2 = "/v2.0/devices/pk007test/tempp";
const char *STopic3 = "/v2.0/devices/pk007test/led0";
const char *STopic4 = "/v2.0/devices/pk007test/sw0"; } + Vandu
```

```

#define MQTT_Port 1883

#define Test_LED0 2
#define Test_SW00 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);
}
}
}

```

Call back

```

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]) { A 23 = LED
Serial.print(" -LED1-> ");
Serial.print((char)payload[10]);
if (payload[10] == '1')
digitalWrite(Test_LED0, HIGH);
else
}

```

on / off

```

digitalWrite(Test_LED0, LOW);
}

Serial.println();
}

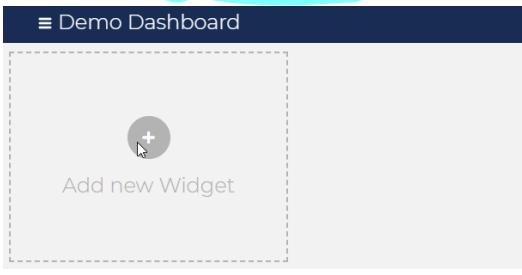
void setup()
{ pinMode(Test_LED0,OUTPUT);
pinMode(Test_SW00,INPUT_PULLUP);
Serial.begin(115200);
Setup_Wifi();
client.setServer(MQTT_Server, MQTT_Port);
client.setCallback(callback);
}

void loop()
{ if (!client.connected()) reconnect();
client.loop();
long now = millis();
if (now - lastMsg > 5000)
{ lastMsg = now;
float xTemp = random(2000, 4000) / 100.0;
float xHumid = random(6000, 8000) / 100.0;
int stsSW = digitalRead(Test_SW00);
snprintf (msg, 75, "{\"humid\":%5.2f,\"temp\":%5.2f,\"sw0\":%d}", xHumid, xTemp, stsSW);
Serial.print("Publish message: ");
Serial.println(msg);
client.publish(PTopic1, msg);
}
}

```

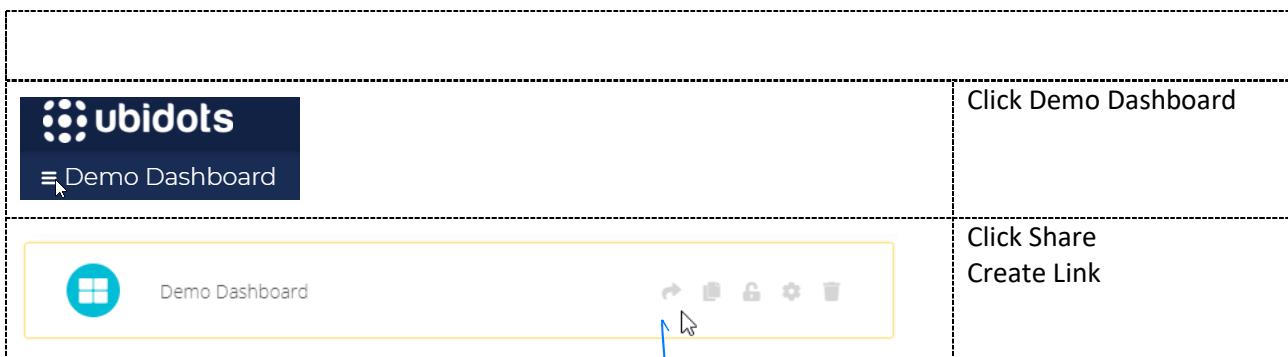
Srinath

8. Create Dashboard

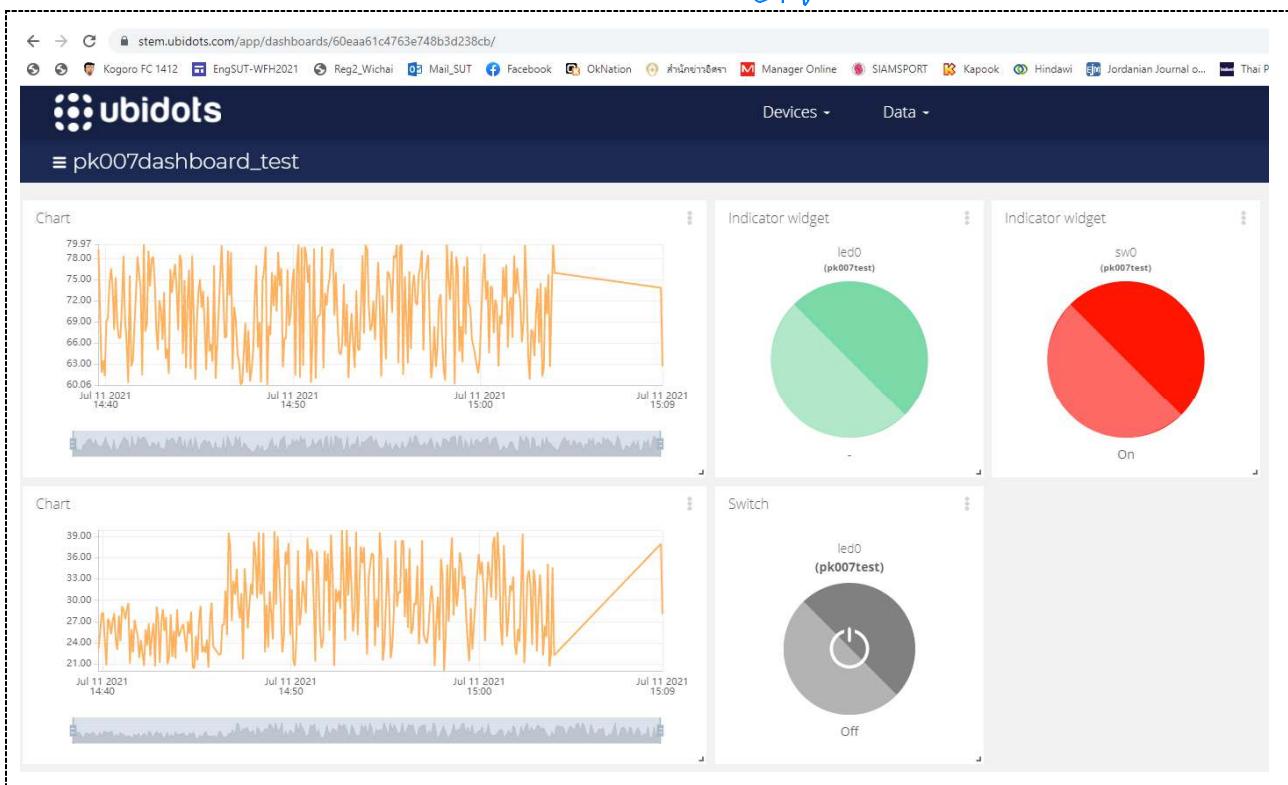
| | |
|--|---|
|  <p>Add new Widget</p> | <p>Data Dashboard Add New Dashboard</p> <p>Add new Widget</p> |
|  <p>Line chart</p> | <p>Line chart</p> |
|  <p>Widget Creation Select Variables</p> <p>Search <input type="text"/> Q</p> <p>pk007test 2 Variables</p> <p>humid</p> | <p>Add Value → humid</p> |

| | |
|--|--|
| <p>Widget Creation</p> <h3>Select Variables</h3> | <p>Line chart Add Value → temp</p> |
| | <p>Switch Add Value → led0</p> <p>Indicator Add Value → led0 Add Value → sw0</p> |
| | |

9. Test Dashboard and Create Share Dashboard Link



10. Test with Browser



Test 6/6. Remote monitor and control Modbus RTU device via Ubidots

9. Create Device → Variable {relay0, relay1, relay2, relay3} → dashboard on Ubidots

The screenshot shows the Ubidots Device creation interface. It displays four variables: relay0, relay1, relay2, and relay3. Each variable has a yellow header with a cloud icon and a edit pen. Below each header, the variable name is listed with three dots above it. Underneath the names, the status 'Last activity: No last activity' is shown. To the right of the variables is a dashed box containing a plus sign and the text 'Add Variable'.

The screenshot shows the Ubidots dashboard titled 'pk007dashboard_test'. The dashboard is organized into a grid of eight widgets:

- Row 1: Four 'Indicator widget' for relay0, relay1, relay2, and relay3. Each shows a pie chart where the red slice represents the 'On' state.
- Row 2: Four 'Switch' widgets for relay0, relay1, relay2, and relay3. Each switch has a cyan circle on the left labeled 'On' and a grey circle on the right labeled 'Off'.

10. Test This Code for Relay Control

```

#define WiFi.h
#define PubSubClient.h
#define ModbusMaster.h

#define MAX485_Monitor 2
#define MAX485_Ctrl 5 // Pin Ctrl 1=Tx and 0=RxD2
#define MAX485_Rx 16 // Pin RxD2
#define MAX485_Tx 17 // Pin TXD2
#define Slave_ID 5 // Slave ID
ModbusMaster node; // instantiate ModbusMaster object

const char *My_SSID = "Test1234";
const char *My_Pass = "0816601929";
const char *MQTT_Server = "things.ubidots.com";
const char *MQTT_User = "BBFF-YgcQcuqoAiO9g46ehWh12TxVwyTjCx";
const char *MQTT_Pass = "BBFF-YgcQcuqoAiO9g46ehWh12TxVwyTjCx";
const char *PTopic1 = "/v2.0/devices/pk007test";
const char *STopic1 = "/v2.0/devices/pk007test/relay0";
const char *STopic2 = "/v2.0/devices/pk007test/relay1";
const char *STopic3 = "/v2.0/devices/pk007test/relay2";
const char *STopic4 = "/v2.0/devices/pk007test/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.println("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);
}

```

↑ 4 Wands

Φ Wands

ID Pending Status

↓ On/Off RTU

ID

STS

```

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;
  }
}

```

Connecting to Test1234

....
 WiFi connected
 IP address:
 192.168.1.22
 Attempting MQTT connection...connected
 Message arrived [/v2.0/devices/pk007test/relay0] {"value": 1.0, "timestamp": 162599406757
 RlyID-0 >> RlyStatus-1
 Message arrived [/v2.0/devices/pk007test/relay2] {"value": 1.0, "timestamp": 162599407163
 RlyID-2 >> RlyStatus-1
 Message arrived [/v2.0/devices/pk007test/relay1] {"value": 1.0, "timestamp": 162599406918
 RlyID-1 >> RlyStatus-1
 Message arrived [/v2.0/devices/pk007test/relay3] {"value": 1.0, "timestamp": 162599407315
 RlyID-3 >> RlyStatus-1

Topne

29

0123456789..

payla

10

Echo - Sh

11. Test Control and Monitor Modbus Device

Test 7/7. Remote monitor and control Modbus RTU/ASCII/TCP device via Ubidots

12. Test Control and Monitor Modbus Device

< ลงบริบัปโปรแกรมด้วยตัวท่านเอง >

การควบคุมเครื่องจักรอัจฉริยะโดยใช้การสื่อสารระหว่างเครื่องจักรกับเครื่องจักร
M2M - Intelligence Machine Control

ชื่อ-สกุล :

4/4: -- คำถ้ามท้ายบทเพื่อทดสอบความเข้าใจ

Quiz_401 – test Blynk

< รูปอุปกรณ์ที่ใช้ทดสอบ ขณะทำการทดสอบ >

< รูปอุปกรณ์ที่ใช้ทดสอบ ขณะทำการทดสอบ >

< รูปหน้าจอ Blynk >

รายละเอียดการทดสอบ

< โปรแกรมทดสอบ >

< ผลการทดสอบ >

Quiz_402 – test Ubidot with ESP32

< รูปอุปกรณ์ที่ใช้ทดสอบ ขณะทำการทดสอบ >

< รูปอุปกรณ์ที่ใช้ทดสอบ ขณะทำการทดสอบ >

< รูปหน้าจอ Ubidots >

รายละเอียดการทดสอบ

< โปรแกรมทดสอบ >

< ผลการทดสอบ >

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

< รูปอุปกรณ์ที่ใช้ทดสอบ ขณะทำการทดสอบ >

< รูปอุปกรณ์ที่ใช้ทดสอบ ขณะทำการทดสอบ >

< รูปหน้าจอ Ubidots >

รายละเอียดการทดสอบ

< โปรแกรมทดสอบ >

< ผลการทดสอบ >

Quiz_404 – Application

< อธิบายแนวคิด การนำไปใช้เกี่ยวกับงานที่รับผิดชอบ >