การควบคุมเครื่องจักรอัจฉริยะโดยใช้การสื่อสารระหว่างเครื่องจักรกับเครื่องจักร

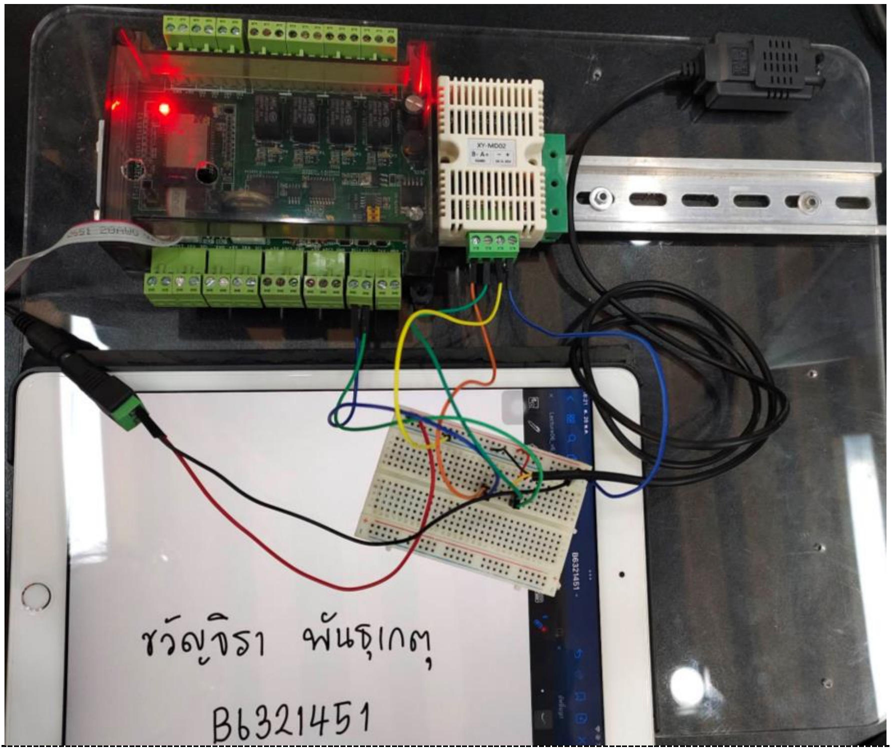
M2M - Intelligence Machine Control

ชื่อ-สกุล : ขวัญจิรา พันธุเกตุ **รหัสนักศึกษา :** B6321451

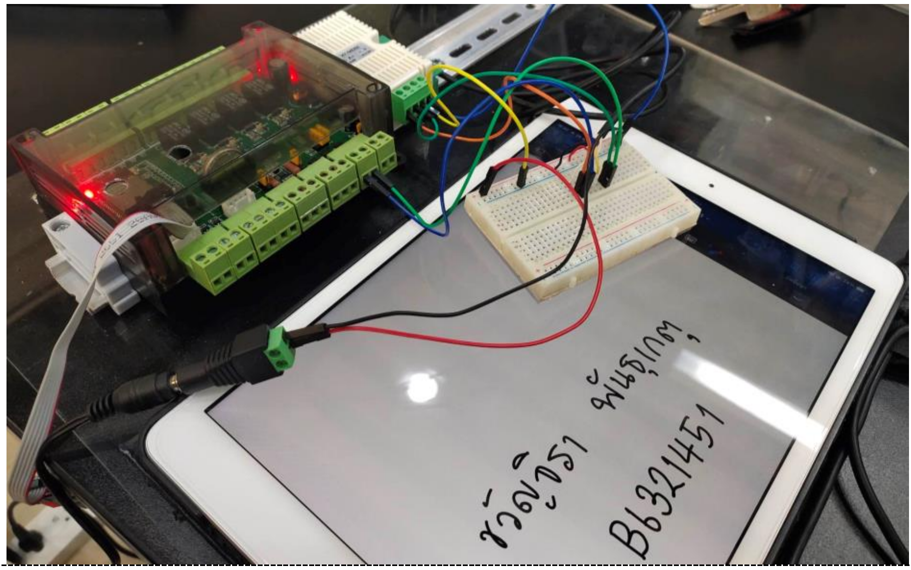
4/4: -- คำถามท้ายบทเพื่อทดสอบความเข้าใจ

Quiz_201 – Read Modbus RTU

< รูปอุปกรณ์ที่ใช้ทดสอบ ขณะทำการทดสอบ >



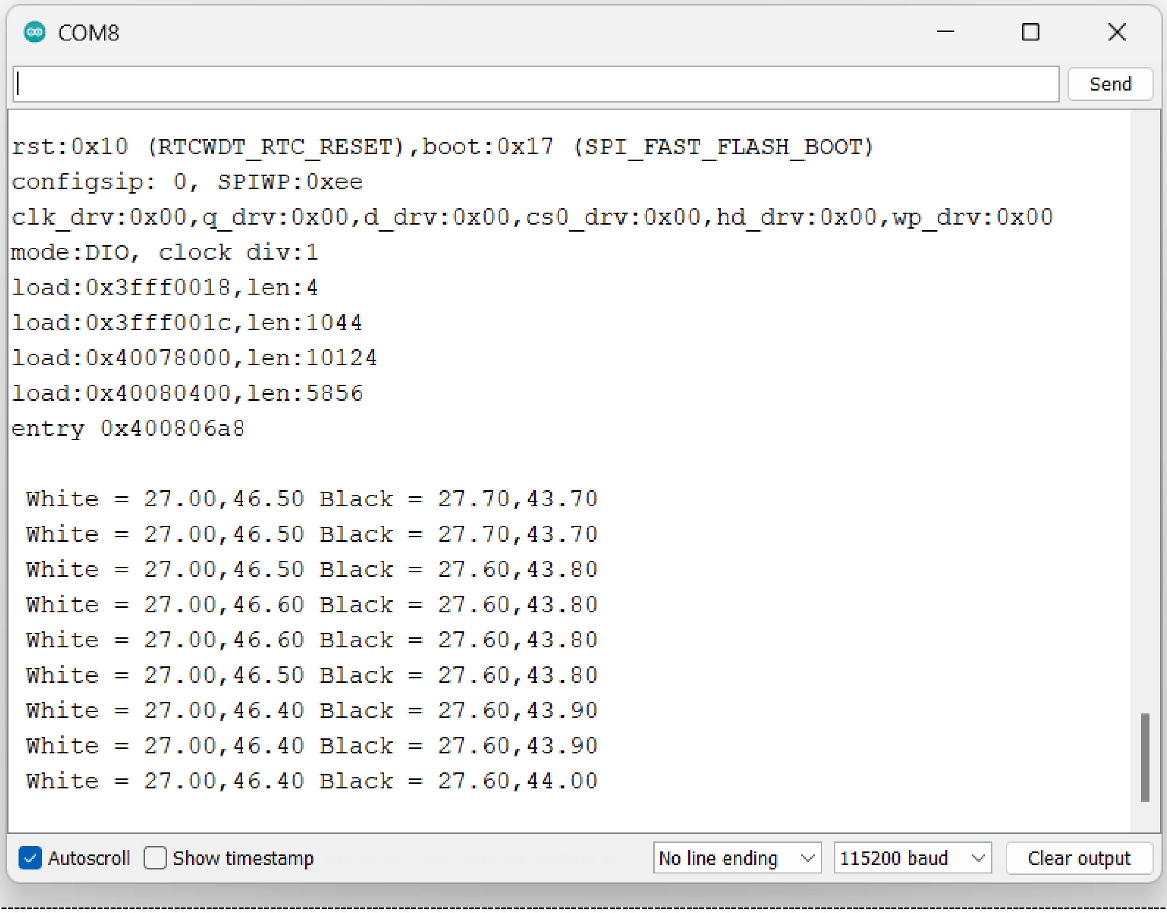
< รูปอุปกรณ์ที่ใช้ทดสอบ ขณะทำการทดสอบ >



โปรแกรมทดสอบ

```
#include "ModbusMaster.h" //hlps://github.com/4-20ma/ModbusMaster
#define SlaveID_White 11
#define SlaveID_Black 12
#define RX_PIN 26
#define TX_PIN 27
ModbusMaster modbus1, modbus2;
void setup() {
  Serial.begin(115200, SERIAL_8N1);
  Serial2.begin(9600, SERIAL_8N1, RX_PIN, TX_PIN);
  modbus1.begin(SlaveID_White, Serial2);
  modbus2.begin(SlaveID_Black, Serial2);
}
double GetData_DBL;
uint8_t result;
void loop() {
  Serial.println();
  delay(2000);
  Serial.print(" White = ");
  result = modbus1.readInputRegisters(1, 2);
  if (getResultMsg(&modbus1, result)) {
    GetData_DBL = modbus1.getResponseBuffer(0) / 10.0;
    Serial.print(GetData_DBL);
    Serial.print(",");
    GetData_DBL = modbus1.getResponseBuffer(1) / 10.0;
    Serial.print(GetData_DBL);
  }
  delay(2000);
  Serial.print(" Black = ");
  result = modbus2.readInputRegisters(1, 2);
  if (getResultMsg(&modbus2, result)) {
    GetData_DBL = modbus2.getResponseBuffer(0) / 10.0;
    Serial.print(GetData_DBL);
    Serial.print(",");
    GetData_DBL = modbus2.getResponseBuffer(1) / 10.0;
    Serial.print(GetData_DBL);
  }
}
bool getResultMsg(ModbusMaster *node, uint8_t result) {
  String tmpstr2 = "\r\n";
  switch (result) {
    case node->ku8MBSuccess:
      return true;
      break;
    case node->ku8MBIllegalFunction:
      tmpstr2 += "Illegal FuncSon";
      break;
    case node->ku8MBIllegalDataAddress:
      tmpstr2 += "Illegal Data Address";
      break;
    case node->ku8MBIllegalDataValue:
      tmpstr2 += "Illegal Data Value";
      break;
    case node->ku8MBSlaveDeviceFailure:
      tmpstr2 += "Slave Device Failure";
      break;
    case node->ku8MBInvalidSlaveID:
      tmpstr2 += "Invalid Slave ID";
      break;
    case node->ku8MBInvalidFunction:
      tmpstr2 += "Invalid FuncSon";
      break;
    case node->ku8MBResponseTimedOut:
      tmpstr2 += "Response Timed Out";
      break;
    case node->ku8MBInvalidCRC:
      tmpstr2 += "Invalid CRC";
      break;
    default:
      tmpstr2 += "Unknown error: " + String(result);
    break;
  }
  Serial.println(tmpstr2);
  return false;
}
```

< ผลการทดสอบ >

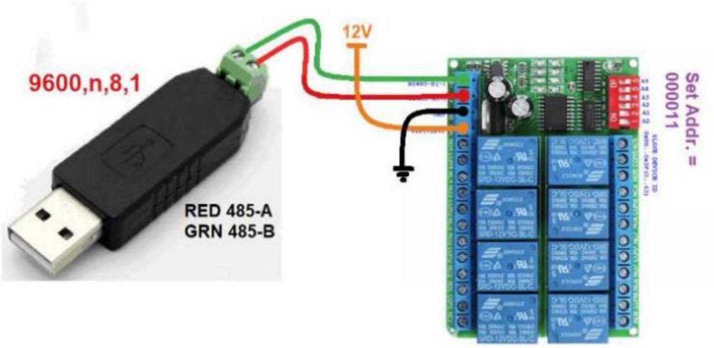


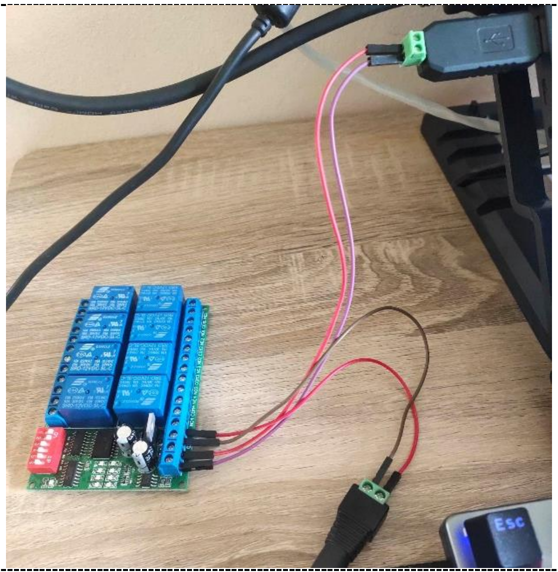
```
rst:0x10 (RTCWDT_RTC_RESET),boot:0x17 (SPI_FAST_FLASH_BOOT)
configsip: 0, SPIWP:0xee
clk_drv:0x00,q_drv:0x00,d_drv:0x00,cs0_drv:0x00,hd_drv:0x00,wp_drv:0x00
mode:DIO, clock div:1
load:0x3fff0018,len:4
load:0x3fff001c,len:1044
load:0x40078000,len:10124
load:0x40080400,len:5856
entry 0x400806a8

 White = 27.00,46.50 Black = 27.70,43.70
 White = 27.00,46.50 Black = 27.70,43.70
 White = 27.00,46.50 Black = 27.60,43.80
 White = 27.00,46.60 Black = 27.60,43.80
 White = 27.00,46.60 Black = 27.60,43.80
 White = 27.00,46.50 Black = 27.60,43.80
 White = 27.00,46.40 Black = 27.60,43.90
 White = 27.00,46.40 Black = 27.60,43.90
 White = 27.00,46.40 Black = 27.60,44.00
```

Quiz_202 – Write Modbus RTU

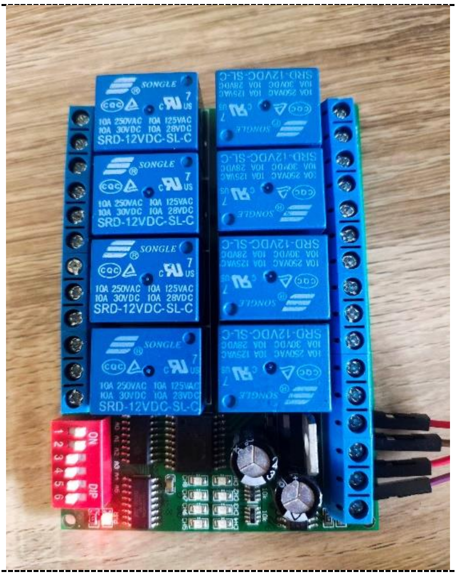
การตั้งค่าอุปกรณ์ด้วย Modbus Poll

รูปการต่ออุปกรณ์ Relay 8 way

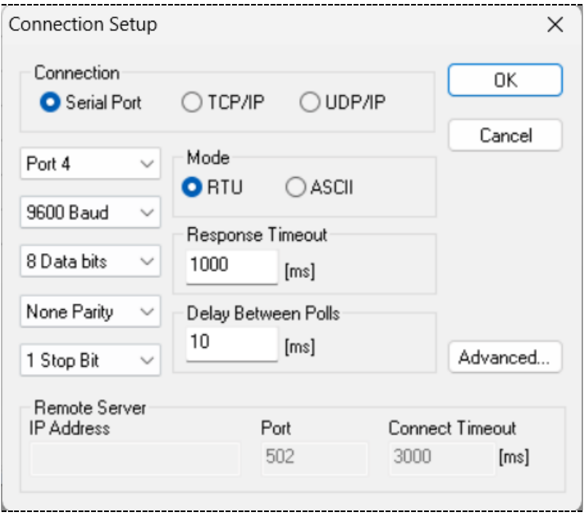




กำหนดหมายเลข slave ให้กับอุปกรณ์ Relay 8 way



สร้างการเชื่อมต่อกับอุปกรณ์ Relay 8 way



Connection Setup

Connection: ● Serial Port ○ TCP/IP ○ UDP/IP

Port 4

9600 Baud

8 Data bits

None Parity

1 Stop Bit

Mode: ● RTU ○ ASCII

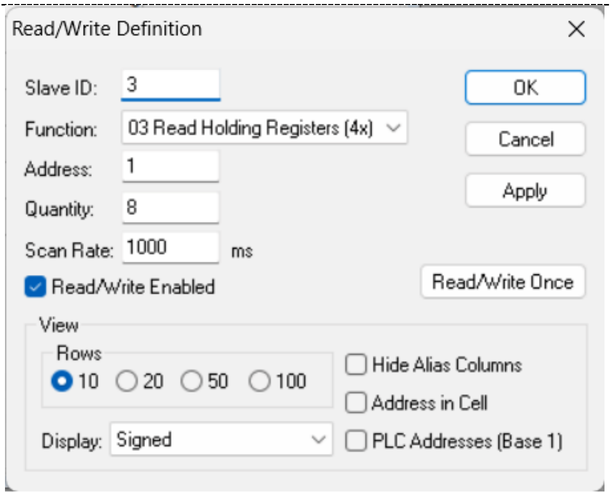
Response Timeout: 1000 [ms]

Delay Between Polls: 10 [ms]

Remote Server

IP Address	Port	Connect Timeout
	502	3000 [ms]

กำหนดการอ่านค่าของอุปกรณ์ Relay 8 way



Read/Write Definition

Slave ID: 3

Function: 03 Read Holding Registers (4x)

Address: 1

Quantity: 8

Scan Rate: 1000 ms

☑ Read/Write Enabled

View

Rows: ● 10 ○ 20 ○ 50 ○ 100

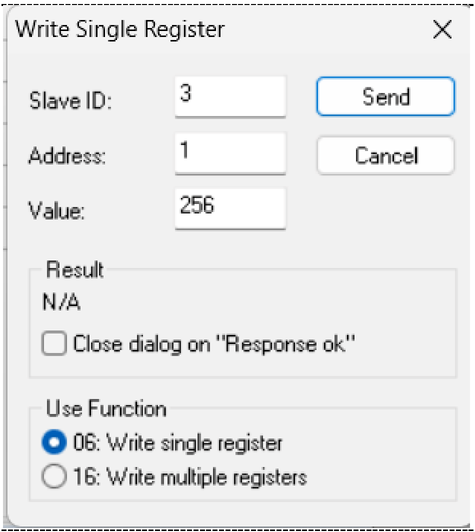
☐ Hide Alias Columns

☐ Address in Cell

☐ PLC Addresses (Base 1)

Display: Signed

ทดสอบการเขียนค่าให้relay ตัวที่ 1 ติด



Write Single Register

Slave ID: 3

Address: 1

Value: 256

Result

N/A

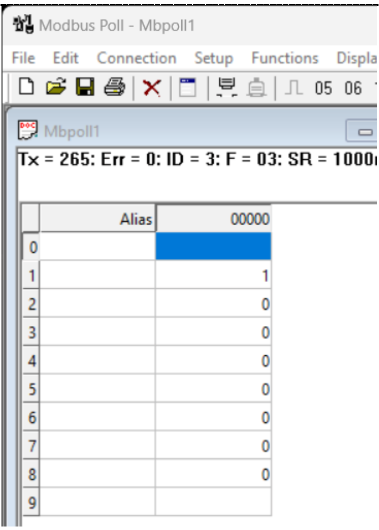
☐ Close dialog on "Response ok"

Use Function

● 06: Write single register

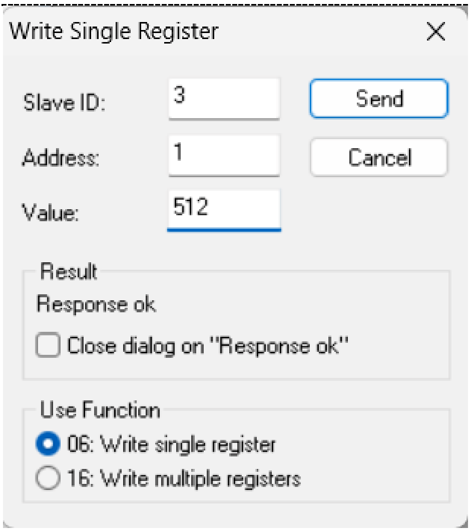
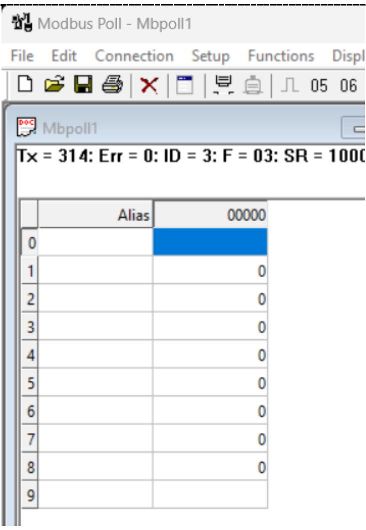
○ 16: Write multiple registers

อ่านค่าจากที่เขียนให้relay ตัวที่ 1



```
Tx = 265: Err = 0: ID = 3: F = 03: SR = 1000
```

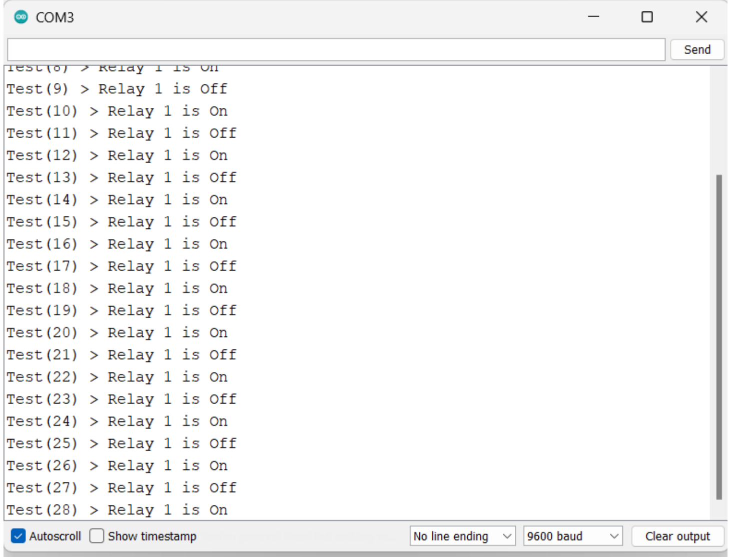
	Alias	00000
0		
1		1
2		0
3		0
4		0
5		0
6		0
7		0
8		0
9		

ทดสอบการเขียนค่าให้relay ตัวที่ 1 ดับ	อ่านค่าจากที่เขียนให้relay ตัวที่ 1
 Write Single Register Slave ID: 3 Address: 1 Value: 512 Result Response ok ☐ Close dialog on "Response ok" Use Function ● 06: Write single register ○ 16: Write multiple registers	 Tx = 314: Err = 0: ID = 3: F = 03: SR = 100(Alias | 00000 0 | 1 | 0 2 | 0 3 | 0 4 | 0 5 | 0 6 | 0 7 | 0 8 | 0 9 |

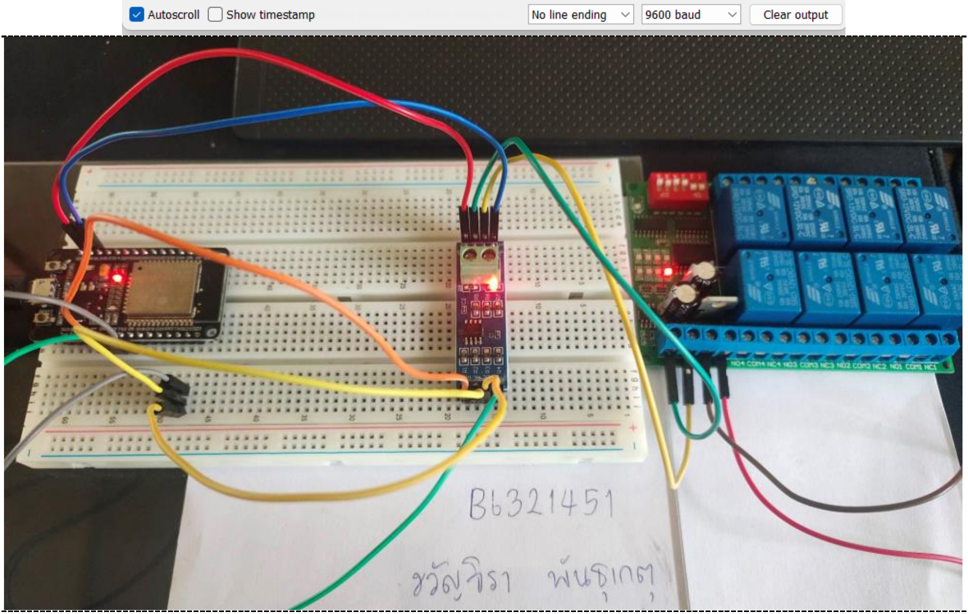
โปรแกรมที่ใช้ทดสอบ 1 - ใช้ ESP32 เขียนค่าให้Relay 8 way โดยไม่มีการคำนวณ CRC

```
#define RS485TX HIGH
#define RS485RX LOW
#define RS485CTRL 5
#define LED_MONITOR 2
int stepCount = 0;
int eindex = 0;
byte byteSend;
byte echo[20];
byte cmd_on[] = {0x03, 0x06, 0x00, 0x01, 0x01, 0x00, 0xD8, 0x78};
byte cmd_off[] = {0x03, 0x06, 0x00, 0x01, 0x02, 0x00, 0xD8, 0x88};
byte cmd_read[] = {0x03, 0x03, 0x00, 0x01, 0x00, 0x01, 0xD4, 0x28};
String mapData(byte number) {
  switch (number) {
    case 1 : return "On";
    case 2 : return "Off";
    default : return "Invalid State";
  }
}
void setup() {
  pinMode(RS485CTRL, OUTPUT);
  pinMode(LED_MONITOR, OUTPUT);
  Serial.begin(9600);
  Serial2.begin(9600);
  digitalWrite(RS485CTRL, RS485RX);
  Serial.println("Start Test MODBUS RTU");
}
void loop() {
  Serial.print("\nTest(");
  Serial.print(++stepCount);
  Serial.print(") > ");
  digitalWrite(LED_MONITOR, HIGH);
  digitalWrite(RS485CTRL, RS485TX);
  delay(10);
  if ((stepCount % 2) == 0) {
    for (int i = 0; i < sizeof(cmd_on); i++) {
      Serial2.write(cmd_on[i]);
    }
  }
  else {
    for (int i = 0; i < sizeof(cmd_off); i++) {
      Serial2.write(cmd_off[i]);
    }
  }
  delay(10);
  digitalWrite(RS485CTRL, RS485RX);
  digitalWrite(LED_MONITOR, LOW);
  delay(10);
  digitalWrite(LED_MONITOR, HIGH);
  digitalWrite(RS485CTRL, RS485TX);
  delay(10);
  for (int i = 0; i < sizeof(cmd_read); i++) {
    Serial2.write(cmd_read[i]);
  }
  delay(10);
  digitalWrite(RS485CTRL, RS485RX);
  digitalWrite(LED_MONITOR, LOW);
  eindex = 0;
  for (long int i = 0; i < 600000; i++) {
    if (Serial2.available())
      echo[eindex++] = Serial2.read();
    if (eindex > 12) i = 999999;
  }
  Serial.print("Relay " + String(echo[1]) + " is " + mapData(echo[2]));


  delay(5000);
}
```

ผลการทดสอบ



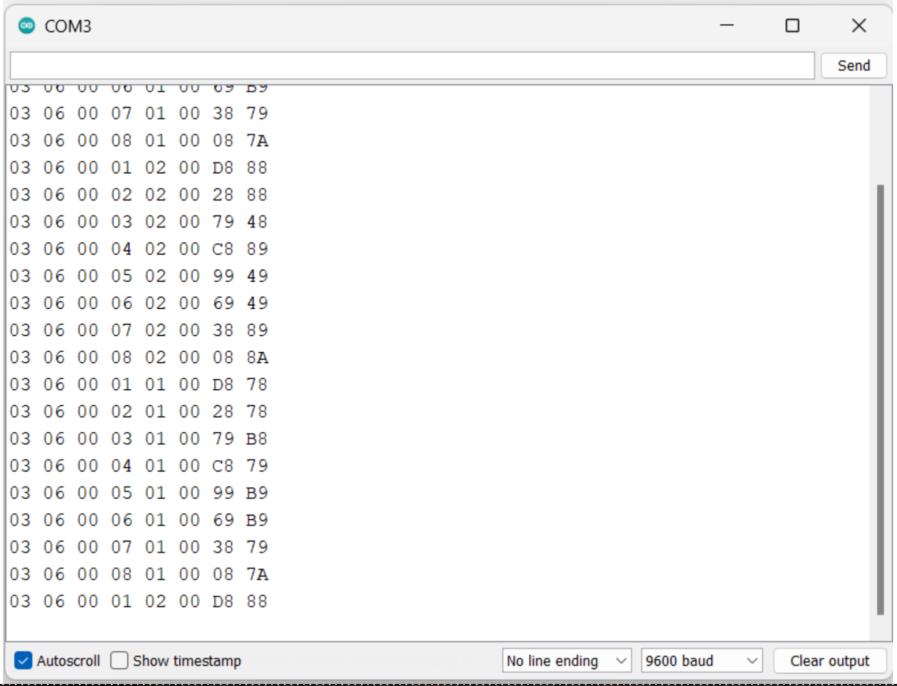
```
Test(8) > Relay 1 is On
Test(9) > Relay 1 is Off
Test(10) > Relay 1 is On
Test(11) > Relay 1 is Off
Test(12) > Relay 1 is On
Test(13) > Relay 1 is Off
Test(14) > Relay 1 is On
Test(15) > Relay 1 is Off
Test(16) > Relay 1 is On
Test(17) > Relay 1 is Off
Test(18) > Relay 1 is On
Test(19) > Relay 1 is Off
Test(20) > Relay 1 is On
Test(21) > Relay 1 is Off
Test(22) > Relay 1 is On
Test(23) > Relay 1 is Off
Test(24) > Relay 1 is On
Test(25) > Relay 1 is Off
Test(26) > Relay 1 is On
Test(27) > Relay 1 is Off
Test(28) > Relay 1 is On
```



โปรแกรมที่ใช้ทดสอบ **2 -** ใช้ **ESP32** เขียน **Relay 8 way** โดยมีการคำนวณ **CRC** อัตโนมัติ

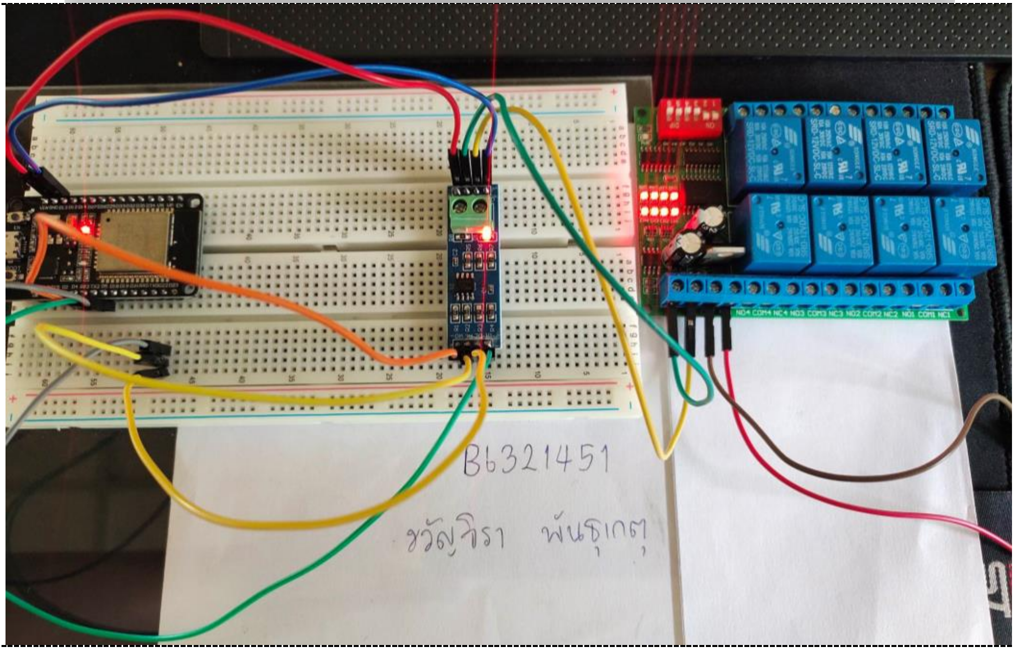
```
#define RS485TX HIGH
#define RS485RX LOW
#define RS485CTRL 5
#define LED_MONITOR 2
int stepCount = 0;
int eindex = 0;
byte echo[20];
byte slaveID = 0x03;
byte modbusCMD = 0x06;
byte h_relayID = 0x00;
byte l_relayID = 0x03;
byte relay_on = 0x01;
byte relay_off = 0x02;
byte on_off_delay = 0x00;
byte h_byteCRC = 0;
byte l_byteCRC = 0;
void setup() {
 pinMode(RS485CTRL, OUTPUT);
 pinMode(LED_MONITOR, OUTPUT);
 Serial.begin(9600);
 Serial2.begin(9600);
 digitalWrite(RS485CTRL, RS485RX);
 Serial.println("Start Test MODBUS RTU");
}
uint16_t CRC16_Update(uint16_t tempCRC, uint8_t inData) {
 tempCRC ^= inData;
 for(int i = 0; i < 8; i++) {
 if(tempCRC & 1) {
 tempCRC = (tempCRC >> 1) ^ 0xA001;
 }
 else {
 tempCRC = tempCRC >> 1;
 }
 }
 return tempCRC;
}
uint16_t sendByte_CRCUpdate(uint16_t tempCRC, uint8_t inData) {
 Serial2.write(inData);
 if(inData < 0x10) Serial.print("0");
 Serial.print(inData, HEX);
 Serial.print(" ");
 tempCRC = CRC16_Update(tempCRC, inData);
 return tempCRC;
}
void relayCTRL(int relay_id, byte relay_cmd) {
 uint16_t calculateCRC = 0xFFFF;
 h_relayID = highByte(relay_id);
 l_relayID = lowByte(relay_id);
 digitalWrite(LED_MONITOR, HIGH);
 digitalWrite(RS485CTRL, RS485TX);
 delay(10);
 calculateCRC = sendByte_CRCUpdate(calculateCRC, slaveID);
 calculateCRC = sendByte_CRCUpdate(calculateCRC, modbusCMD);
 calculateCRC = sendByte_CRCUpdate(calculateCRC, h_relayID);
 calculateCRC = sendByte_CRCUpdate(calculateCRC, l_relayID);
 calculateCRC = sendByte_CRCUpdate(calculateCRC, relay_cmd);
 calculateCRC = sendByte_CRCUpdate(calculateCRC, on_off_delay);
 h_byteCRC = highByte(calculateCRC);
 l_byteCRC = lowByte(calculateCRC);
 calculateCRC = sendByte_CRCUpdate(calculateCRC, l_byteCRC);
 calculateCRC = sendByte_CRCUpdate(calculateCRC, h_byteCRC);
 delay(10);
 digitalWrite(RS485CTRL, RS485RX);
 digitalWrite(LED_MONITOR, LOW);
 Serial.println();
}
void loop() {
 for(int relay = 1; relay <= 8; relay++) {
 relayCTRL(relay, relay_on);
 delay(3000);
 }
 for(int relay = 1; relay <= 8; relay++) {
 relayCTRL(relay, relay_off);
 delay(3000);
 }
}
```

ผลการทดสอบ



```
03 06 00 06 01 00 69 B9
03 06 00 07 01 00 38 79
03 06 00 08 01 00 08 7A
03 06 00 01 02 00 D8 88
03 06 00 02 02 00 28 88
03 06 00 03 02 00 79 48
03 06 00 04 02 00 C8 89
03 06 00 05 02 00 99 49
03 06 00 06 02 00 69 49
03 06 00 07 02 00 38 89
03 06 00 08 02 00 08 8A
03 06 00 01 01 00 D8 78
03 06 00 02 01 00 28 78
03 06 00 03 01 00 79 B8
03 06 00 04 01 00 C8 79
03 06 00 05 01 00 99 B9
03 06 00 06 01 00 69 B9
03 06 00 07 01 00 38 79
03 06 00 08 01 00 08 7A
03 06 00 01 02 00 D8 88
```

☑ Autoscroll □ Show timestamp | No line ending | 9600 baud | Clear output



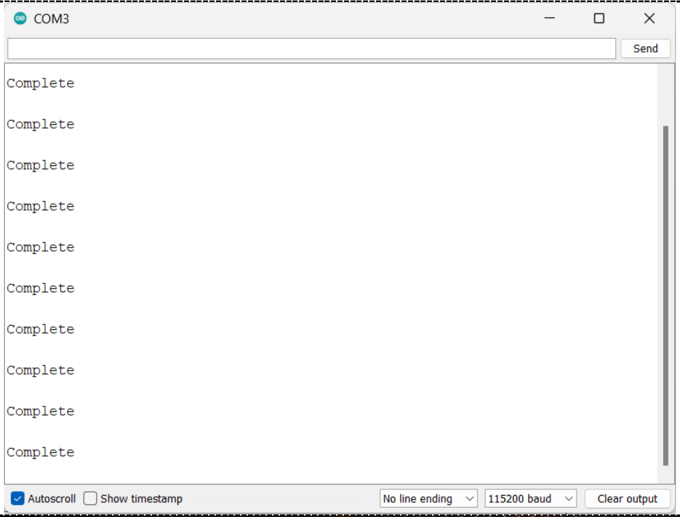
โปรแกรมที่ใช้ทดสอบ 3 - ใช้ ESP32 เขียนค่าให้Relay 8 way โดยใช้ library

```
#include "ModbusMaster.h"
#define SLAVE_ID 3
#define CTRL 5
#define RX 16
#define TX 17
ModbusMaster modbus;
```

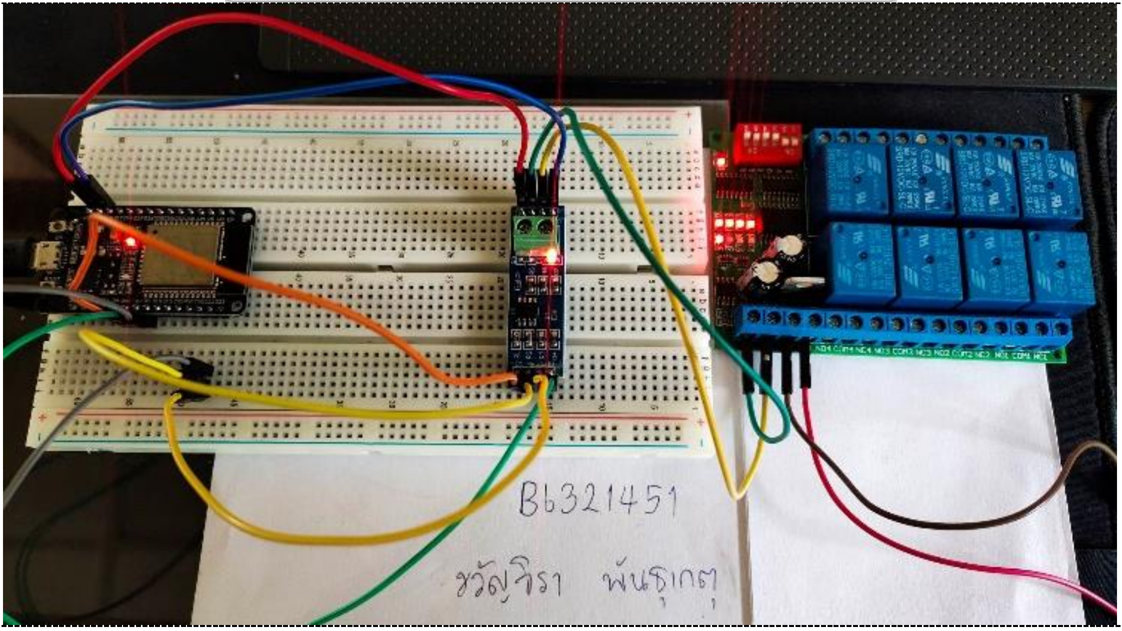
```
void preTransmission() {
  digitalWrite(CTRL, HIGH);
}
void postTransmission() {
  digitalWrite(CTRL, LOW);
}
void setup() {
  pinMode(CTRL, OUTPUT);
  digitalWrite(CTRL, LOW);
  Serial.begin(115200);
  Serial2.begin(9600, SERIAL_8N1, RX, TX);
  modbus.begin(SLAVE_ID, Serial2);
  modbus.preTransmission(preTransmission);
  modbus.postTransmission(postTransmission);
}
void loop() {
  uint8_t result;
  result = modbus.writeSingleRegister(1, 0x0100);
  getResultMsg(&modbus, result);
  delay(5000);
  result = modbus.writeSingleRegister(1, 0x0200);
  getResultMsg(&modbus, result);
  delay(5000);
}
bool getResultMsg(ModbusMaster *node, uint8_t result) {
  String tmpstr2 = "\r\n";
  switch (result) {
    case node->ku8MBSuccess:
      tmpstr2 += "Complete";
      Serial.println(tmpstr2);
      return true;
      break;
    case node->ku8MBIllegalFunction:
      tmpstr2 += "Illegal Function";
      break;
    case node->ku8MBIllegalDataAddress:
      tmpstr2 += "Illegal Data Address";
      break;
    case node->ku8MBIllegalDataValue:
      tmpstr2 += "Illegal Data Value";
      break;
    case node->ku8MBSlaveDeviceFailure:
      tmpstr2 += "Slave Device Failure";
      break;
    case node->ku8MBInvalidSlaveID:
      tmpstr2 += "Invalid Slave ID";
      break;
    case node->ku8MBInvalidFunction:
```

```
    tmpstr2 += "Invalid Function";
    break;
  case node->ku8MBResponseTimedOut:
    tmpstr2 += "Response Timed Out";
    break;
  case node->ku8MBInvalidCRC:
    tmpstr2 += "Invalid CRC";
    break;
  default:
    tmpstr2 += "Unknown error: " + String(result);
    break;
 }
 Serial.println(tmpstr2);
 return false;
}
```

ผลการทดสอบ



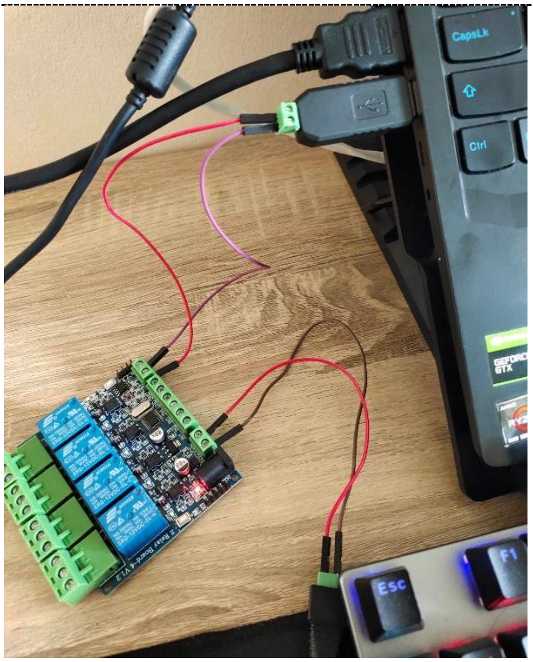
```
Complete

Complete

Complete

Complete

Complete

Complete

Complete

Complete

Complete

Complete
```

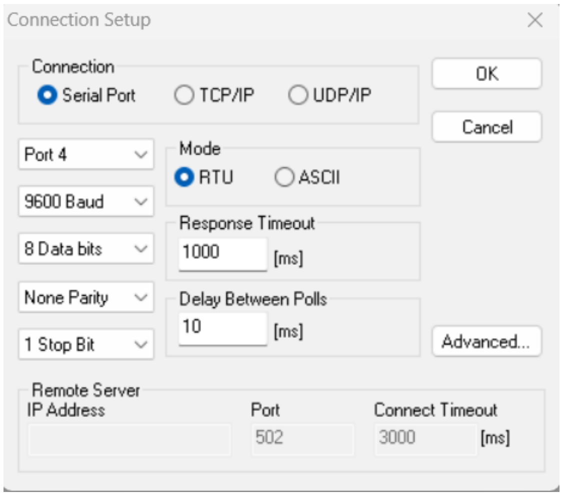


Quiz_203 – Read/Write Modbus RTU

การตั้งค่าอุปกรณ์ด้วย Modbus Poll

สร้างการเชื่อมต่อกับอุปกรณ์ Relay 4 way





Connection: ● Serial Port ○ TCP/IP ○ UDP/IP

Port 4

9600 Baud

8 Data bits

None Parity

1 Stop Bit

Mode: ● RTU ○ ASCII

Response Timeout: 1000 [ms]

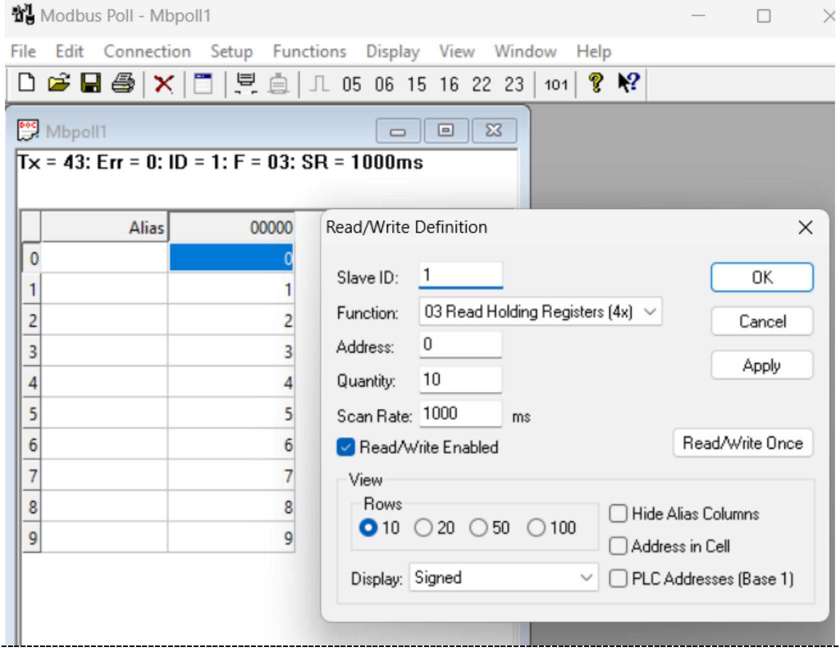
Delay Between Polls: 10 [ms]

Remote Server

IP Address:

Port: 502

Connect Timeout: 3000 [ms]



Tx = 43: Err = 0: ID = 1: F = 03: SR = 1000ms

	Alias	00000
0		0
1		1
2		2
3		3
4		4
5		5
6		6
7		7
8		8
9		9

Slave ID: 1

Function: 03 Read Holding Registers (4x)

Address: 0

Quantity: 10

Scan Rate: 1000 ms

☑ Read/Write Enabled

View

Rows: ● 10 ○ 20 ○ 50 ○ 100

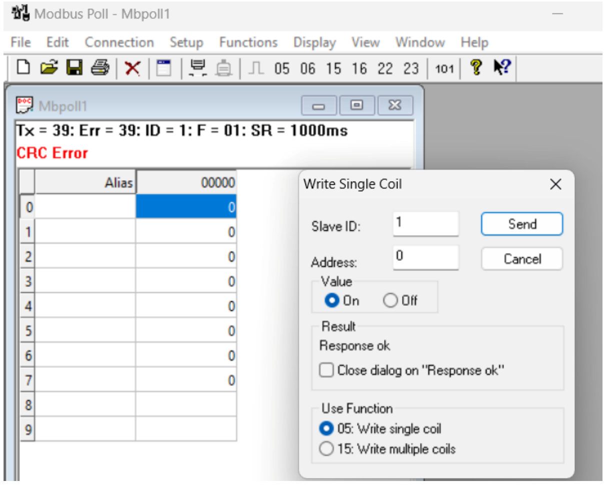
□ Hide Alias Columns

□ Address in Cell

Display: Signed

□ PLC Addresses (Base 1)

ทดสอบการเขียนค่าให้ relay ตัวที่ 1 ติด



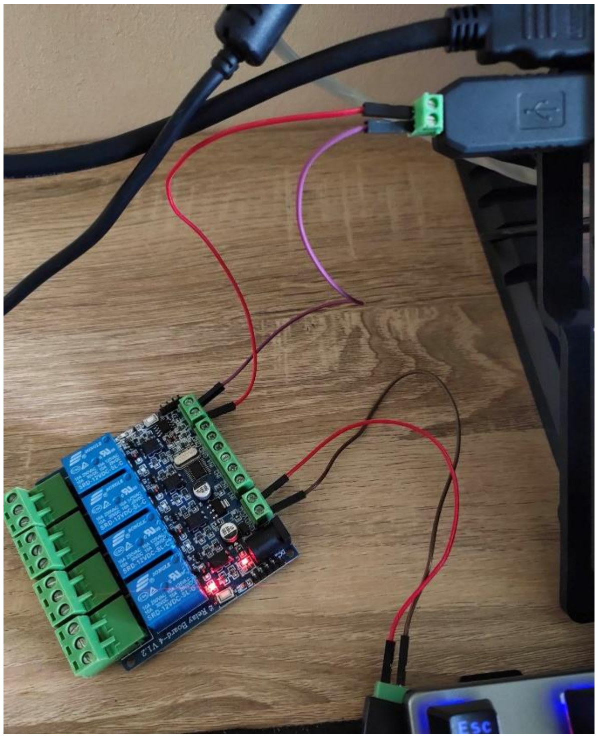
Tx = 39: Err = 39: ID = 1: F = 01: SR = 1000ms
CRC Error

	Alias	00000
0		0
1		0
2		0
3		0
4		0
5		0
6		0
7		0
8		
9		

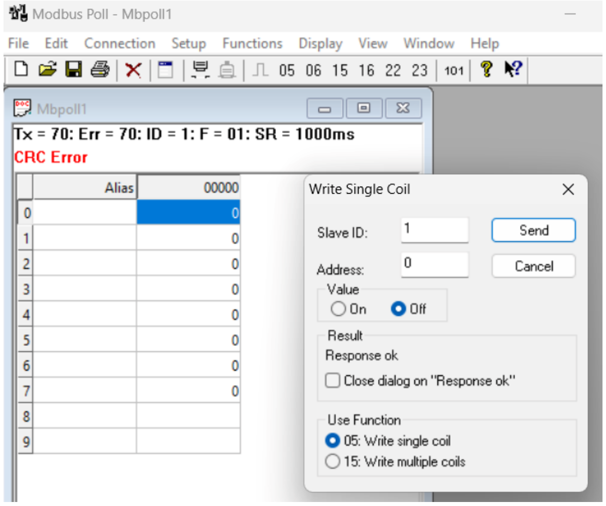
Slave ID: 1
Address: 0
Value: ● On ○ Off
Result: Response ok
□ Close dialog on "Response ok"
Use Function:
● 05: Write single coil
○ 15: Write multiple coils

CRC Error แต่อุปกรณ์จริง มีการ ON OFF ปกติ

relay ตัวที่ 1 ติด



ทดสอบการเขียนค่าให้ relay ตัวที่ 1 ดับ

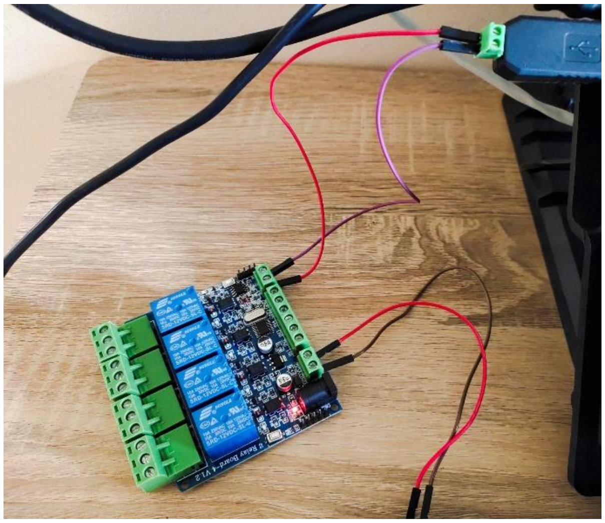


Tx = 70: Err = 70: ID = 1: F = 01: SR = 1000ms
CRC Error

	Alias	00000
0		0
1		0
2		0
3		0
4		0
5		0
6		0
7		0
8		
9		

Slave ID: 1
Address: 0
Value: ○ On ● Off
Result: Response ok
□ Close dialog on "Response ok"
Use Function:
● 05: Write single coil
○ 15: Write multiple coils

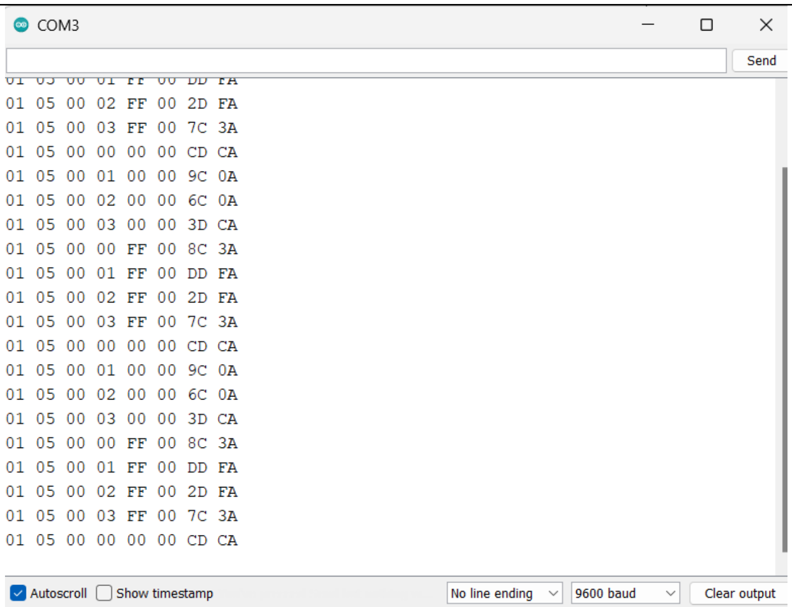
relay ตัวที่ 1 ดับ



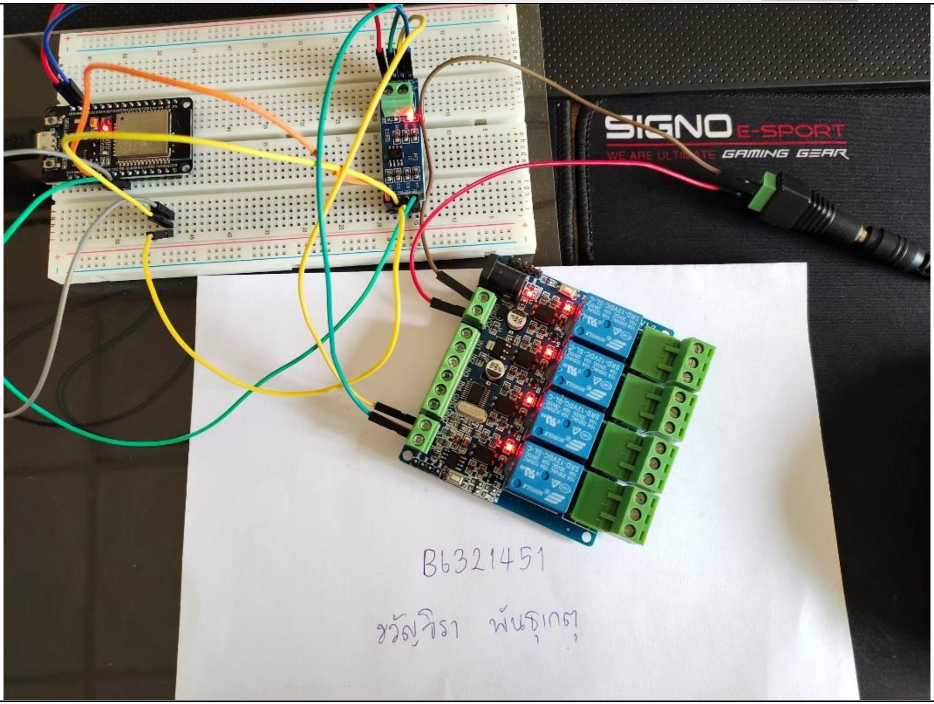
โปรแกรมที่ใช้ทดสอบ 1 – ใช้ ESP32 เขียน relay 2 way เพื่อเปิด/ปิด relay

```
#define RS485TX HIGH
#define RS485RX LOW
#define RS485CTRL 5
#define LED_MONITOR 2
int stepCount = 0;
int eindex = 0;
byte echo[20];
byte slaveID = 0x01;
byte modbusCMD = 0x05;
byte h_relayID = 0x00;
byte l_relayID = 0x00;
byte relay_on = 0xFF;
byte relay_off = 0x00;
byte on_off_delay = 0x00;
byte h_byteCRC = 0;
byte l_byteCRC = 0;
void setup() {
  pinMode(RS485CTRL, OUTPUT);
  pinMode(LED_MONITOR, OUTPUT);
  Serial.begin(9600);
  Serial2.begin(9600);
  digitalWrite(RS485CTRL, RS485RX);
  Serial.println("Start Test MODBUS RTU");
}
uint16_t CRC16_Update(uint16_t tempCRC, uint8_t inData) {
  tempCRC ^= inData;
  for (int i = 0; i < 8; i++) {
    if (tempCRC & 1) {
      tempCRC = (tempCRC >> 1) ^ 0xA001;
    }
    else {
      tempCRC = tempCRC >> 1;
    }
  }
  return tempCRC;
}
uint16_t sendByte_CRCUpdate(uint16_t tempCRC, uint8_t inData) {
  Serial2.write(inData);
  if (inData < 0x10) Serial.print("0");
  Serial.print(inData, HEX);
  Serial.print(" ");
  tempCRC = CRC16_Update(tempCRC, inData);
  return tempCRC;
}
void relayCTRL(int relay_id, byte relay_cmd) {
  uint16_t calculateCRC = 0xFFFF;
  h_relayID = highByte(relay_id);
  l_relayID = lowByte(relay_id);
  digitalWrite(LED_MONITOR, HIGH);
  digitalWrite(RS485CTRL, RS485TX);
  delay(10);
  calculateCRC = sendByte_CRCUpdate(calculateCRC, slaveID);
  calculateCRC = sendByte_CRCUpdate(calculateCRC, modbusCMD);
  calculateCRC = sendByte_CRCUpdate(calculateCRC, h_relayID);
  calculateCRC = sendByte_CRCUpdate(calculateCRC, l_relayID);
  calculateCRC = sendByte_CRCUpdate(calculateCRC, relay_cmd);
  calculateCRC = sendByte_CRCUpdate(calculateCRC, on_off_delay);
  h_byteCRC = highByte(calculateCRC);
  l_byteCRC = lowByte(calculateCRC);
  calculateCRC = sendByte_CRCUpdate(calculateCRC, l_byteCRC);
  calculateCRC = sendByte_CRCUpdate(calculateCRC, h_byteCRC);
  delay(10);
  digitalWrite(RS485CTRL, RS485RX);
  digitalWrite(LED_MONITOR, LOW);
  Serial.println();
}
void loop() {
  for (int relay = 0; relay < 4; relay++) {
    relayCTRL(relay, relay_on);
    delay(3000);
  }
  for (int relay = 0; relay < 4; relay++) {
    relayCTRL(relay, relay_off);
    delay(3000);
  }
}
```

ผลการทดสอบ



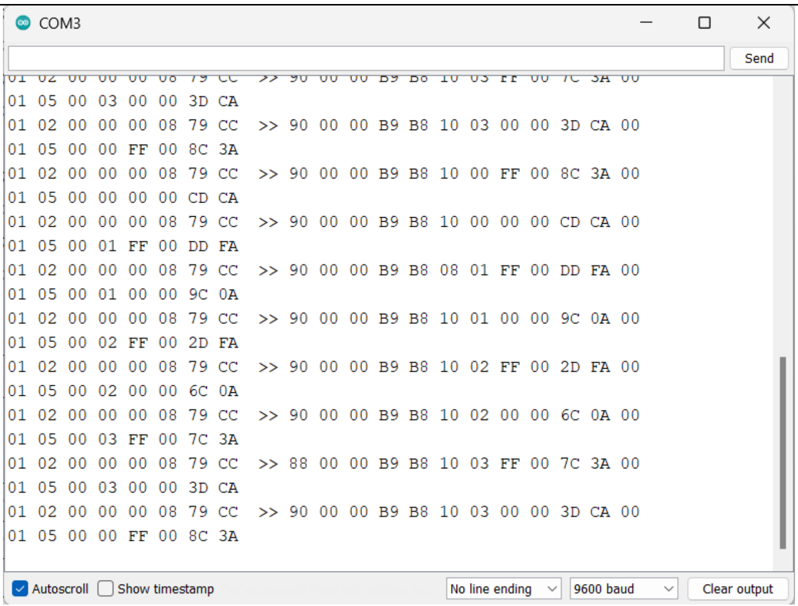
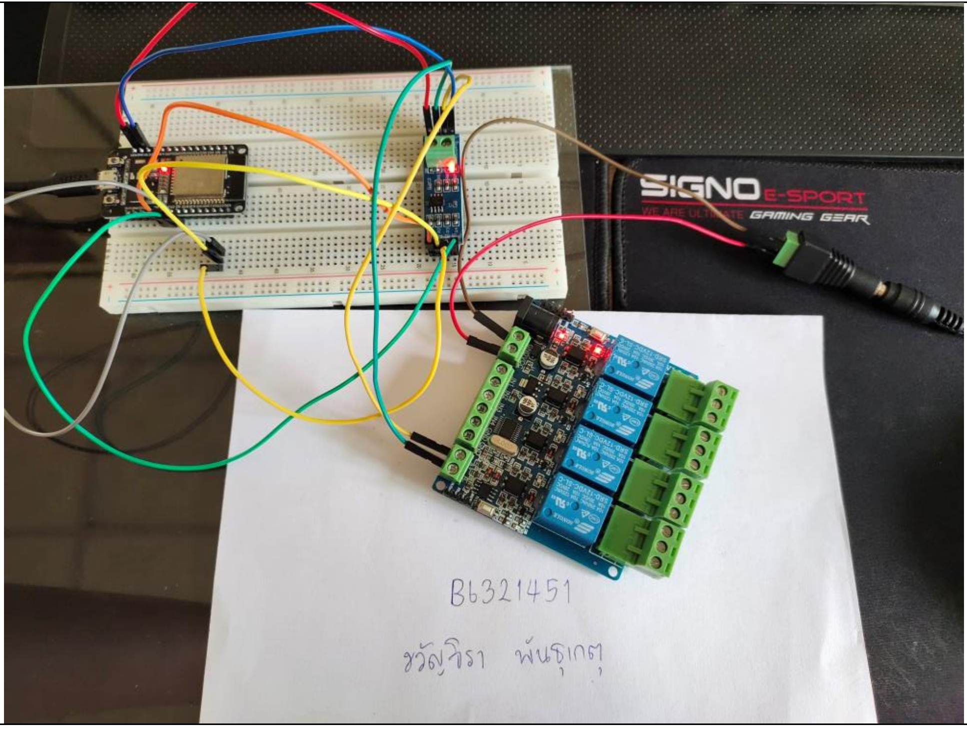
```
01 05 00 01 FF 00 DD FA
01 05 00 02 FF 00 2D FA
01 05 00 03 FF 00 7C 3A
01 05 00 00 00 00 CD CA
01 05 00 01 00 00 9C 0A
01 05 00 02 00 00 6C 0A
01 05 00 03 00 00 3D CA
01 05 00 00 FF 00 8C 3A
01 05 00 01 FF 00 DD FA
01 05 00 02 FF 00 2D FA
01 05 00 03 FF 00 7C 3A
01 05 00 00 00 00 CD CA
01 05 00 01 00 00 9C 0A
01 05 00 02 00 00 6C 0A
01 05 00 03 00 00 3D CA
01 05 00 00 FF 00 8C 3A
01 05 00 01 FF 00 DD FA
01 05 00 02 FF 00 2D FA
01 05 00 03 FF 00 7C 3A
01 05 00 00 00 00 CD CA
```



https://www.youtube.com/shorts/0Q09EvQ-7_s

โปรแกรมที่ใช้ทดสอบ 2 - ใช้ ESP32 อ่านค่า input จาก relay 2 way

```
#define RS485TX HIGH
#define RS485RX LOW
#define RS485CTRL 5
#define LED_MONITOR 2
int stepCount = 0;
int eindex = 0;
byte echo[20];
byte slaveID = 0x01;
byte modbusCMD = 0x05;
byte h_relayID = 0x00;
byte l_relayID = 0x00;
byte relay_on = 0xFF;
byte relay_off = 0x00;
byte on_off_delay = 0x00;
byte h_byteCRC = 0;
byte l_byteCRC = 0;
void setup() {
 pinMode(RS485CTRL, OUTPUT);
 pinMode(LED_MONITOR, OUTPUT);
 Serial.begin(9600);
 Serial2.begin(9600);
 digitalWrite(RS485CTRL, RS485RX);
 Serial.println("Start Test MODBUS RTU");
}
uint16_t CRC16_Update(uint16_t tempCRC, uint8_t inData) {
 tempCRC ^= inData;
 for(int i = 0; i < 8; i++) {
 if(tempCRC & 1) {
 tempCRC = (tempCRC >> 1) ^ 0xA001;
 }
 else {
 tempCRC = tempCRC >> 1;
 }
 }
 return tempCRC;
}
uint16_t sendByte_CRCUpdate(uint16_t tempCRC, uint8_t inData) {
 Serial2.write(inData);
 if(inData < 0x10) Serial.print("0");
 Serial.print(inData, HEX);
 Serial.print(" ");
 tempCRC = CRC16_Update(tempCRC, inData);
 return tempCRC;
}
void relayCTRL(int relay_id, byte relay_cmd) {
 uint16_t calculateCRC = 0xFFFF;
 h_relayID = highByte(relay_id);
 l_relayID = lowByte(relay_id);
 digitalWrite(LED_MONITOR, HIGH);
 digitalWrite(RS485CTRL, RS485TX);
 delay(10);
 calculateCRC = sendByte_CRCUpdate(calculateCRC, slaveID);
 calculateCRC = sendByte_CRCUpdate(calculateCRC, modbusCMD);
 calculateCRC = sendByte_CRCUpdate(calculateCRC, h_relayID);
 calculateCRC = sendByte_CRCUpdate(calculateCRC, l_relayID);
 calculateCRC = sendByte_CRCUpdate(calculateCRC, relay_cmd);
 calculateCRC = sendByte_CRCUpdate(calculateCRC, on_off_delay);
 h_byteCRC = highByte(calculateCRC);
 l_byteCRC = lowByte(calculateCRC);
 calculateCRC = sendByte_CRCUpdate(calculateCRC, l_byteCRC);
 calculateCRC = sendByte_CRCUpdate(calculateCRC, h_byteCRC);
 delay(10);
 digitalWrite(RS485CTRL, RS485RX);
 digitalWrite(LED_MONITOR, LOW);
 Serial.println();
}
void readBoard() {
 uint16_t calculateCRC = 0xFFFF;
 digitalWrite(LED_MONITOR, HIGH);
 digitalWrite(RS485CTRL, RS485TX);
 delay(10);
 calculateCRC = sendByte_CRCUpdate(calculateCRC, slaveID);
 calculateCRC = sendByte_CRCUpdate(calculateCRC, 0x02);
 calculateCRC = sendByte_CRCUpdate(calculateCRC, 0x00);
 calculateCRC = sendByte_CRCUpdate(calculateCRC, 0x00);
 calculateCRC = sendByte_CRCUpdate(calculateCRC, 0x00);
 calculateCRC = sendByte_CRCUpdate(calculateCRC, 0x08);
 h_byteCRC = highByte(calculateCRC);
 l_byteCRC = lowByte(calculateCRC);
 calculateCRC = sendByte_CRCUpdate(calculateCRC, l_byteCRC);
 calculateCRC = sendByte_CRCUpdate(calculateCRC, h_byteCRC);
 delay(10);
 digitalWrite(RS485CTRL, RS485RX);
 digitalWrite(LED_MONITOR, LOW);
 eindex = 0;
 while(Serial2.available()) {
 echo[eindex] = Serial2.read();
 eindex++;
 }
 Serial.print(" >> ");
 for(int i = 0; i < 12; i++) {
 if(echo[i] < 0x10) Serial.print("0");
 Serial.print(echo[i], HEX);
 Serial.print(" ");
 }
 Serial.println();
}
void loop() {
 relayCTRL(0, relay_on); delay(1500); readBoard(); delay(1500);
 relayCTRL(0, relay_off); delay(1500); readBoard(); delay(1500);
 relayCTRL(1, relay_on); delay(1500); readBoard(); delay(1500);
 relayCTRL(1, relay_off); delay(1500); readBoard(); delay(1500);

 relayCTRL(2, relay_on); delay(1500); readBoard(); delay(1500);
 relayCTRL(2, relay_off); delay(1500); readBoard(); delay(1500);
 relayCTRL(3, relay_on); delay(1500); readBoard(); delay(1500);
 relayCTRL(3, relay_off); delay(1500); readBoard(); delay(1500);


}
```

ผลการทดสอบ


https://www.youtube.com/shorts/ADORrd32xww

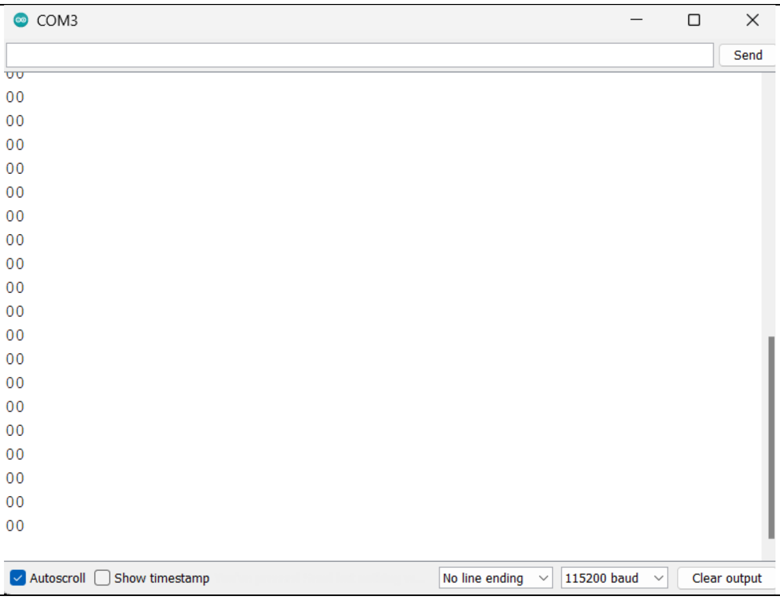
```
01 02 00 00 00 08 79 CC  >> 90 00 00 B9 B8 10 03 FF 00 7C 3A 00
01 05 00 03 00 00 3D CA
01 02 00 00 00 08 79 CC  >> 90 00 00 B9 B8 10 03 00 00 3D CA 00
01 05 00 00 FF 00 8C 3A
01 02 00 00 00 08 79 CC  >> 90 00 00 B9 B8 10 00 FF 00 8C 3A 00
01 05 00 00 00 00 CD CA
01 02 00 00 00 08 79 CC  >> 90 00 00 B9 B8 10 00 00 00 CD CA 00
01 05 00 01 FF 00 DD FA
01 02 00 00 00 08 79 CC  >> 90 00 00 B9 B8 08 01 FF 00 DD FA 00
01 05 00 01 00 00 9C 0A
01 02 00 00 00 08 79 CC  >> 90 00 00 B9 B8 10 01 00 00 9C 0A 00
01 05 00 02 FF 00 2D FA
01 02 00 00 00 08 79 CC  >> 90 00 00 B9 B8 10 02 FF 00 2D FA 00
01 05 00 02 00 00 6C 0A
01 02 00 00 00 08 79 CC  >> 90 00 00 B9 B8 10 02 00 00 6C 0A 00
01 05 00 03 FF 00 7C 3A
01 02 00 00 00 08 79 CC  >> 88 00 00 B9 B8 10 03 FF 00 7C 3A 00
01 05 00 03 00 00 3D CA
01 02 00 00 00 08 79 CC  >> 90 00 00 B9 B8 10 03 00 00 3D CA 00
01 05 00 00 FF 00 8C 3A
```

โปรแกรมที่ใช้ทดสอบ 3 - ใช้ ESP32 อ่านและเขียน relay 2 way (รวมโปรแกรม 1 และ 2)

```
#include "ModbusMaster.h"
#define SLAVE_ID 1
#define CTRL 5
#define RX 16
#define TX 17
#define LED_MONITOR 2
ModbusMaster node;
void preTransmission() {
 digitalWrite(CTRL, HIGH);
}
void postTransmission() {
 digitalWrite(CTRL, LOW);
}
void setup() {
 pinMode(CTRL, OUTPUT);
 digitalWrite(CTRL, LOW);
 Serial.begin(115200);
 Serial2.begin(9600, SERIAL_8N1, RX, TX);
 node.begin(SLAVE_ID, Serial2);
 node.preTransmission(preTransmission);
 node.postTransmission(postTransmission);
}
int read_relay() {
 uint8_t result;
 uint8_t value = 0xff;
 result = node.readDiscreteInputs(0, 8); // Start=0, nByte=4
 if(result == node.ku8MBSuccess) {
 value = node.getResponseBuffer(0); // Read return from 0_Byte
 }
 return value;
}
void binDisplay(int dataIn) {
 if(dataIn == 0xff) {
 Serial.println("Read Error");
 }
 else {
 Serial.print(dataIn >> 1 & 1);
 Serial.print(dataIn >> 0 & 1);
```

```
  Serial.println();
 }
}
void loop() {
 node.writeSingleCoil(0, 0x00FF); delay(2000); // On Relay0
 binDisplay(read_relay());
 node.writeSingleCoil(0, 0x0000); delay(2000); // Off Relay0

 node.writeSingleCoil(1, 0x00FF); delay(2000); // On Relay1
 binDisplay(read_relay());
 node.writeSingleCoil(1, 0x0000); delay(2000); // Off Relay1

 node.writeSingleCoil(2, 0x00FF); delay(2000); // On Relay0
 binDisplay(read_relay());
 node.writeSingleCoil(2, 0x0000); delay(2000); // Off Relay0

 node.writeSingleCoil(3, 0x00FF); delay(2000); // On Relay1
 binDisplay(read_relay());
 node.writeSingleCoil(3, 0x0000); delay(2000); // Off Relay1
}
```

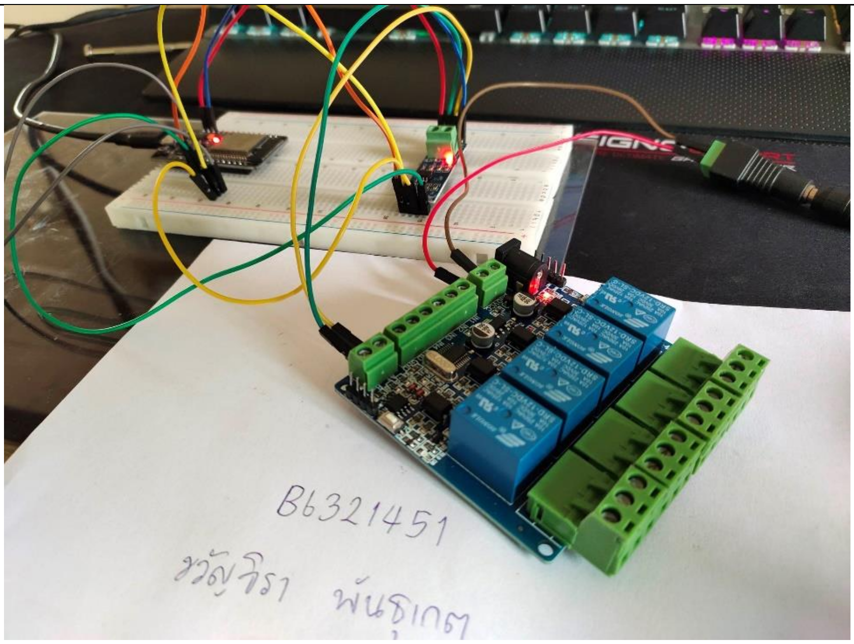
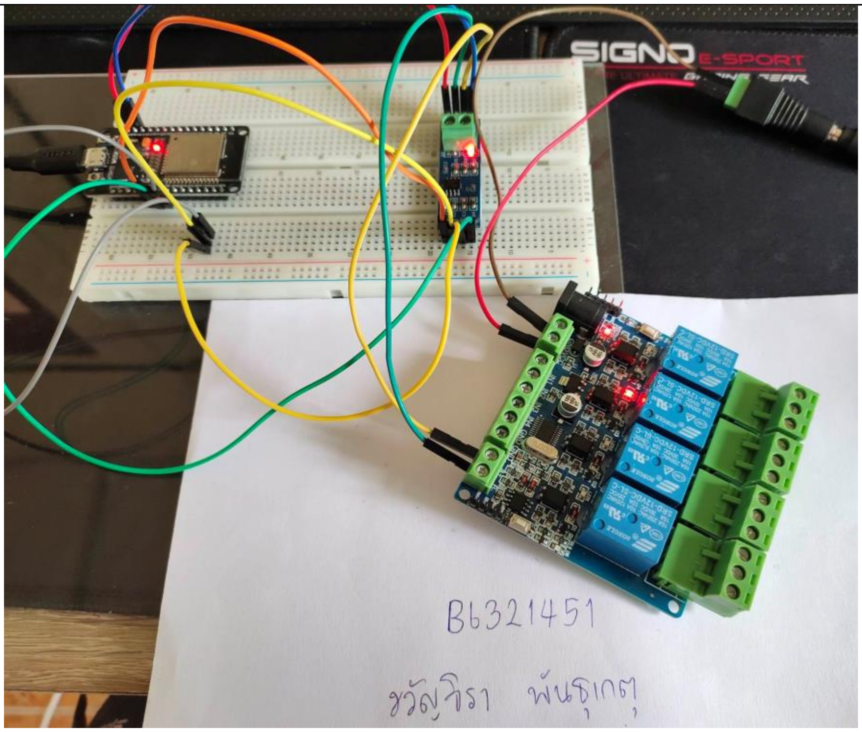
ผลการทดสอบ



No line ending

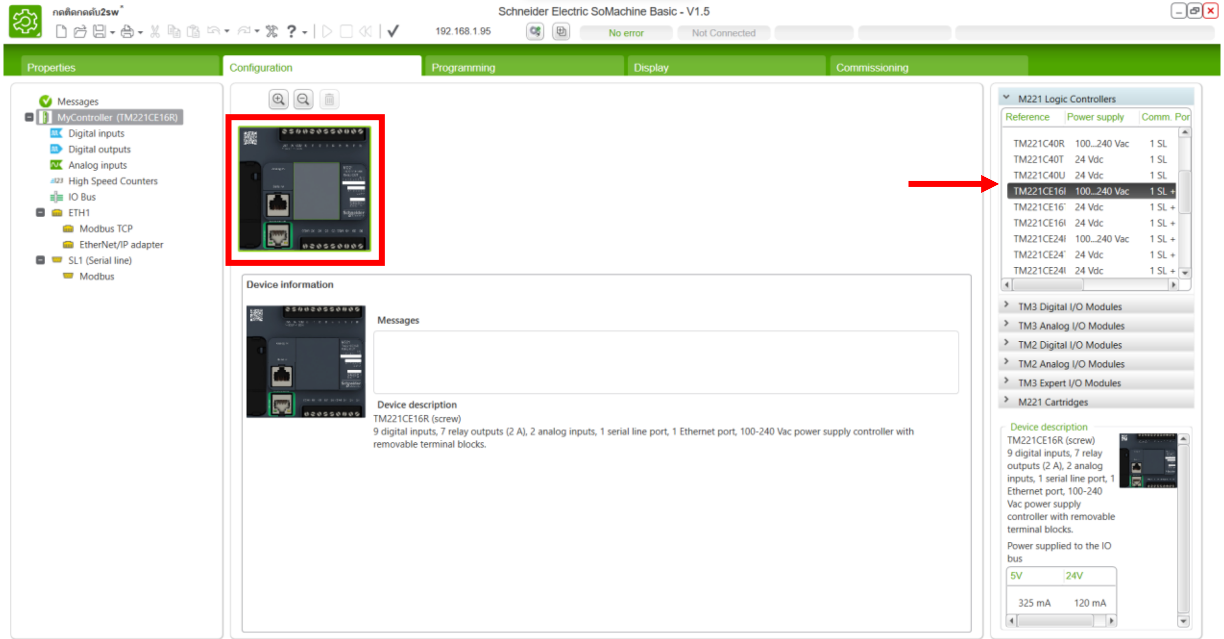
115200 baud

```
00
00
00
00
00
00
00
00
00
00
00
00
00
00
00
00
00
00
00
00
```

รูปการต่อวงจร



Quiz_204 – PLC Test

เปิดโปรแกรม SoMachine → เลือก TM221CE16R



Schneider Electric SoMachine Basic - V1.5

192.168.1.95 · No error · Not Connected

Reference	Power supply	Comm. Por
TM221C40R	100...240 Vac	1 SL
TM221C40T	24 Vdc	1 SL
TM221C40U	24 Vdc	1 SL
TM221CE16I	100...240 Vac	1 SL +
TM221CE16	24 Vdc	1 SL +
TM221CE16I	24 Vdc	1 SL +
TM221CE24I	100...240 Vac	1 SL +
TM221CE24	24 Vdc	1 SL +
TM221CE24I	24 Vdc	1 SL +

Device information

Messages

Device description

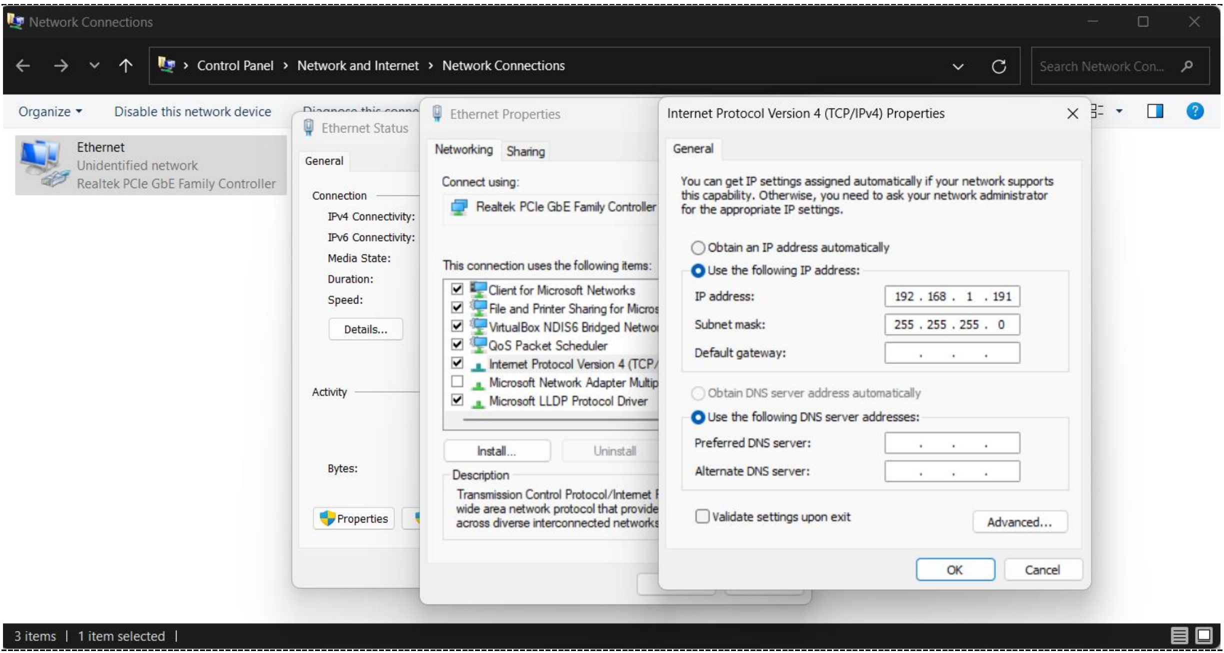
TM221CE16R (screw)

9 digital inputs, 7 relay outputs (2 A), 2 analog inputs, 1 serial line port, 1 Ethernet port, 100-240 Vac power supply controller with removable terminal blocks.

Power supplied to the IO bus

5V	24V
325 mA	120 mA

กำหนด IP address ของ PC ให้อยู่ในวง LAN เดียวกันกับ PLC



Network Connections

Control Panel › Network and Internet › Network Connections

Connect using:

Realtek PCIe GbE Family Controller

This connection uses the following items:

- ☑ Client for Microsoft Networks
- ☑ File and Printer Sharing for Micros
- ☑ VirtualBox NDIS6 Bridged Netwo
- ☑ QoS Packet Scheduler
- ☑ Internet Protocol Version 4 (TCP/
- ☐ Microsoft Network Adapter Multip
- ☑ Microsoft LLDP Protocol Driver

Description

Transmission Control Protocol/Internet I wide area network protocol that provide across diverse interconnected networks

Internet Protocol Version 4 (TCP/IPv4) Properties

General

You can get IP settings assigned automatically if your network supports this capability. Otherwise, you need to ask your network administrator for the appropriate IP settings.

○ Obtain an IP address automatically

● Use the following IP address:

IP address: 192 . 168 . 1 . 191

Subnet mask: 255 . 255 . 255 . 0

Default gateway: . . .

○ Obtain DNS server address automatically

● Use the following DNS server addresses:

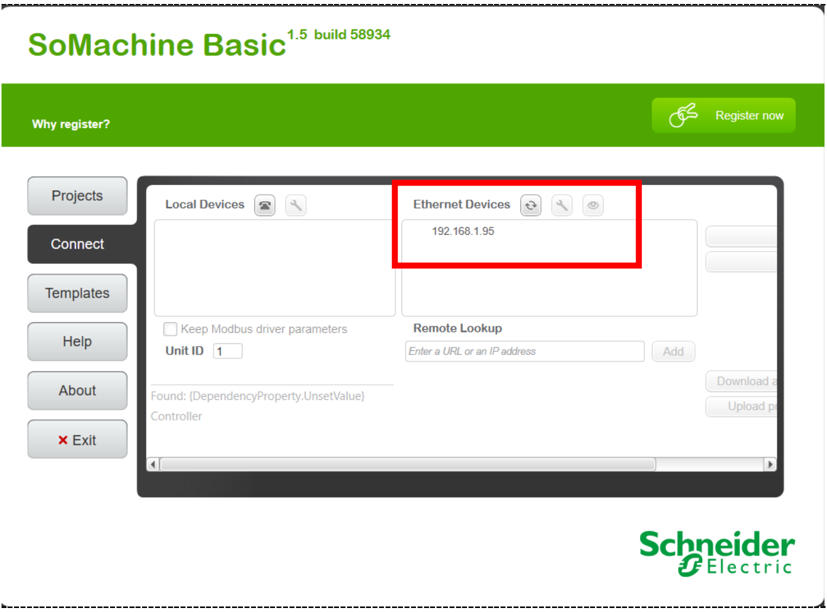
Preferred DNS server: . . .

Alternate DNS server: . . .

☐ Validate settings upon exit

Advanced...

OK Cancel



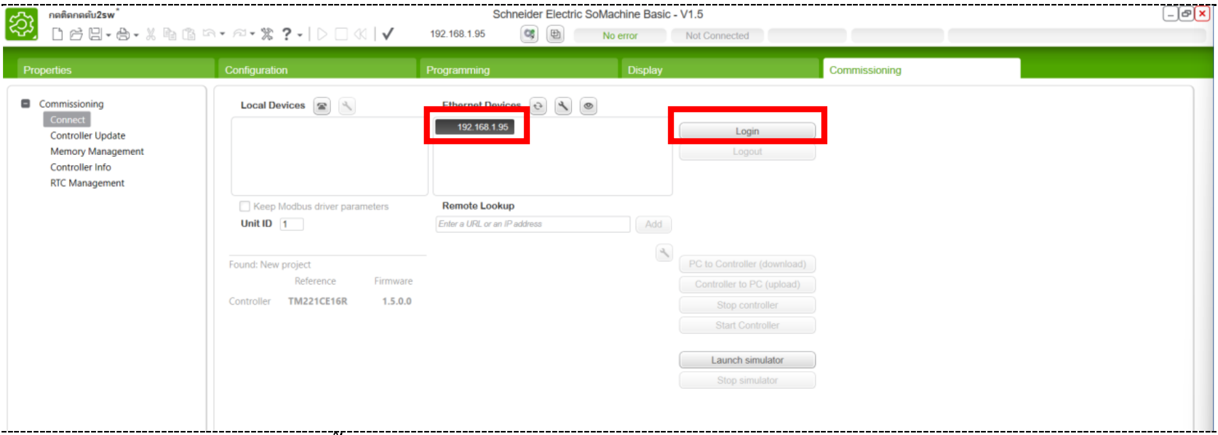
Ethernet Devices: 192.168.1.95

Unit ID: 1

Found: {DependencyProperty.UnsetValue}

Controller

ไปที่ Commissioning → login



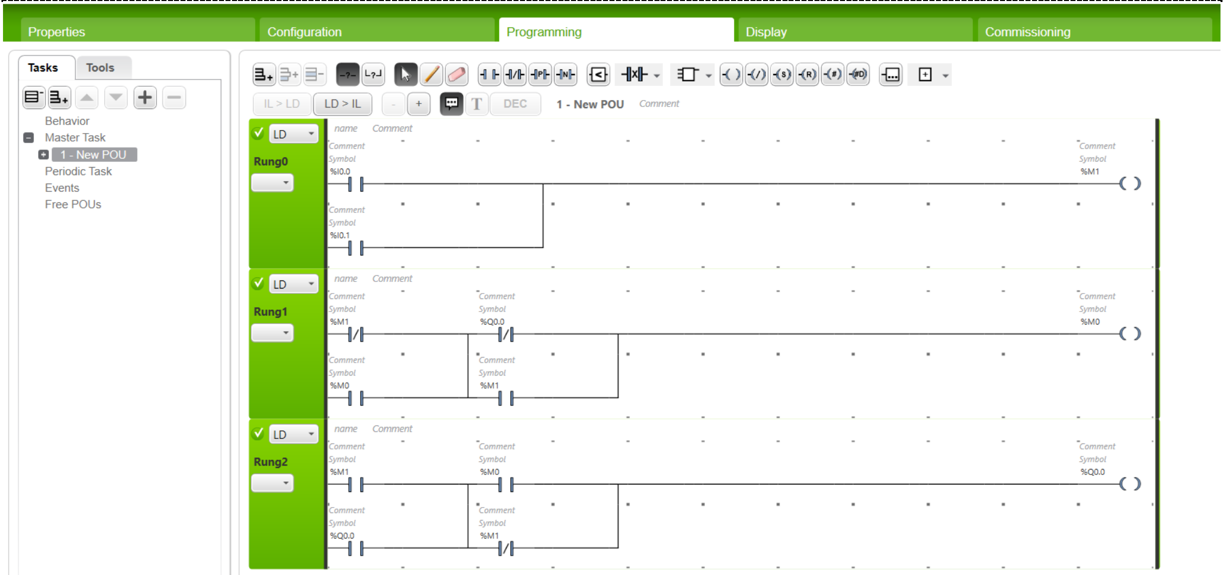
Ethernet Devices: 192.168.1.95

Unit ID: 1

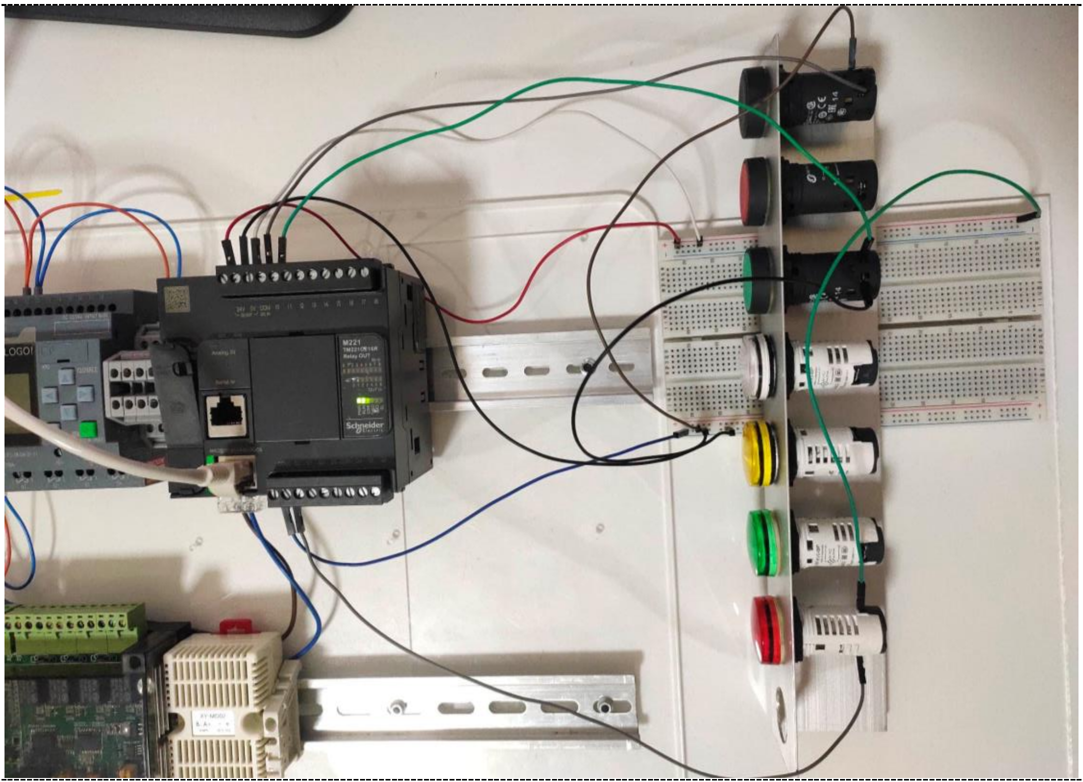
Found: New project

	Reference	Firmware
Controller	TM221CE16R	1.5.0.0

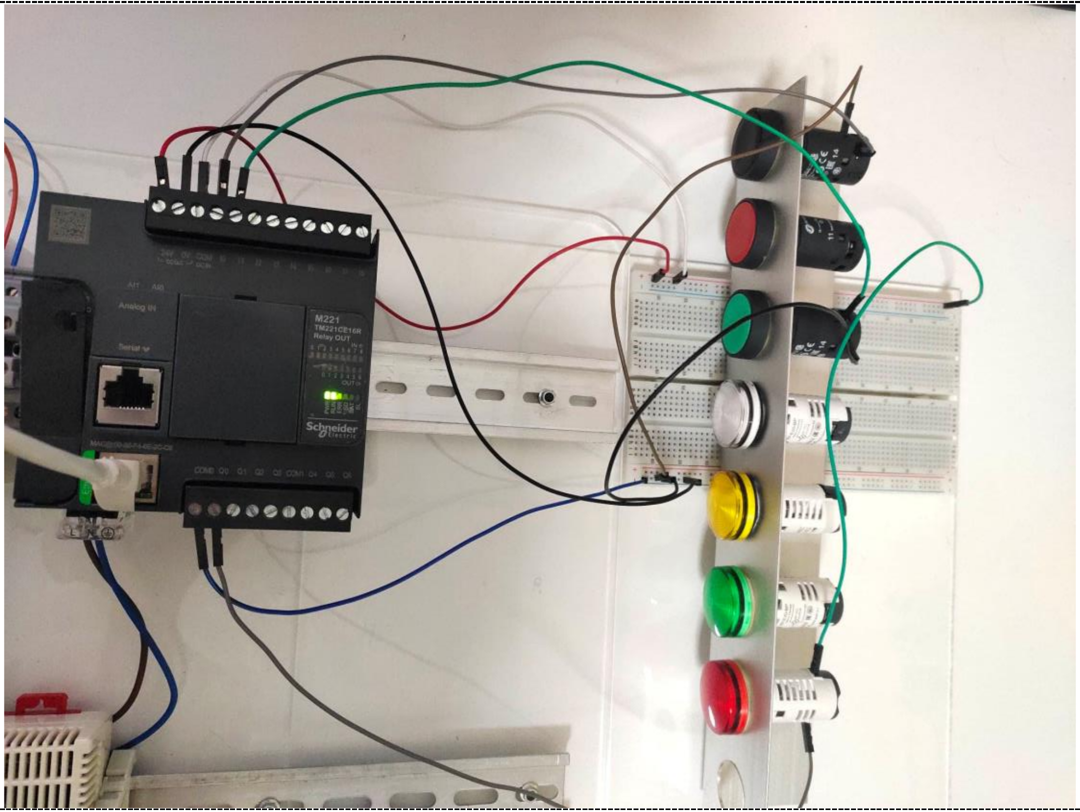
เขียนโปรแกรม ladder –> จากนั้นทำการ อัพโหลดไปยัง PLC



รูปวงจร 1



รูปวงจร 2



ผลการทดสอบ

กดติด กดอีกครั้งดับ 2 SW , 1LED