

การควบคุมเครื่องจักรอัจฉริยะโดยใช้การสื่อสารระหว่างเครื่องจักรกับเครื่องจักร

M2M - Intelligence Machine Control

ชื่อ-สกุล : ญัฐพงศ์ โต๊ะแอ รหัสนักศึกษา : B6310158

4/4: -- คำถามท้ายบทเพื่อทดสอบความเข้าใจ

Quiz_201 – Read Modbus RTU

< รูปอุปกรณ์ที่ใช้ทดสอบ ขณะทำการทดสอบ >



< รูปอุปกรณ์ที่ใช้ทดสอบ ขณะทำการทดสอบ >



< โปรแกรมทดสอบ >

The IDE status bar at the bottom shows '30°C', '1942', and '11/6/2566'.

```
#include "ModbusMaster.h" //https://github.com/4-20ma/ModbusMaster
```

```
#define Slave_ID 11
```

```
#define MAX485_RE_NEG 5
```

```
#define RX_PIN 16
```

```
#define TX_PIN 17
```

```
ModbusMaster modbus;
```

```
void preTransmission() {
```

```
    digitalWrite(MAX485_RE_NEG, HIGH); //Switch to transmit data
```

```
}
```

```
void postTransmission() {
```

```
    digitalWrite(MAX485_RE_NEG, LOW); //Switch to receive data
```

```
}
```

```
void setup() {
```

```
    pinMode(MAX485_RE_NEG, OUTPUT);
```

```
    digitalWrite(MAX485_RE_NEG, LOW);
```

```
    Serial.begin(115200, SERIAL_8N1);
```

```
    Serial2.begin(9600, SERIAL_8N1, RX_PIN, TX_PIN);
```

```
    modbus.begin(Slave_ID, Serial2);
```

```
    modbus.preTransmission(preTransmission);
```

```
    modbus.postTransmission(postTransmission);
```

```
}
```

```
long lastMillis = 0;
```

```
void loop() {
```

```
    long currentMillis = millis();
```

```
    if (currentMillis - lastMillis > 1000) {
```

```
        uint8_t result = modbus.readInputRegisters(1, 2);
```

```
        if (getResultMsg(&modbus, result)) {
```

```
            Serial.println();
```

```
            double res_dbl = modbus.getResponseBuffer(0) / 10;
```

```
            String res = "Temperature: " + String(res_dbl) + " C\r\n";
```

```
            res_dbl = modbus.getResponseBuffer(1) / 10;
```

```
            res += "Humidity: " + String(res_dbl) + " %";
```

```
            Serial.println(res);
```

```
        }
```

```
        lastMillis = currentMillis;
```

```
    }
```

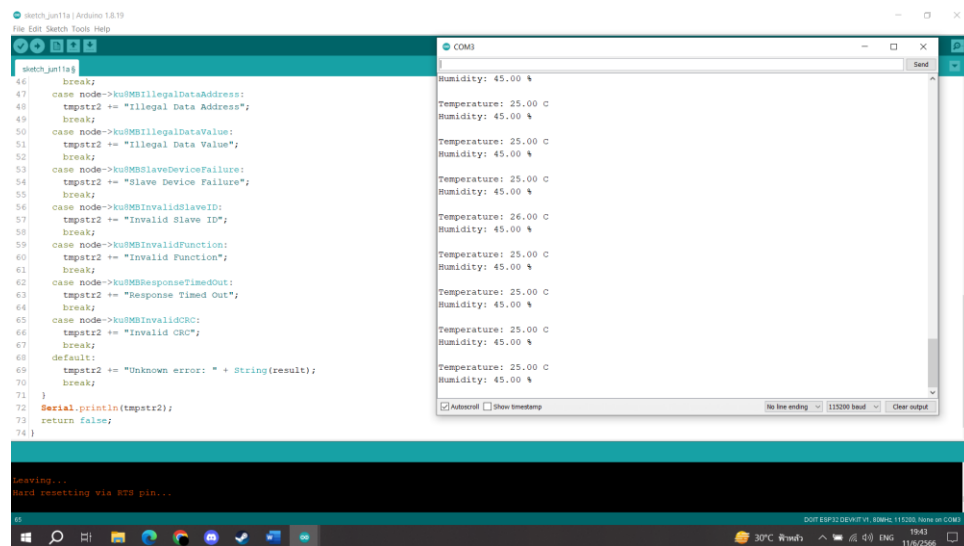
```
}
```

```

bool getResultMsg(ModbusMaster *node, uint8_t result) {
    String tmpstr2 = "\r\n";
    switch (result) {
        case node->ku8MBSuccess:
            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 >



Quiz_202 – Write Modbus RTU

< รูปอุปกรณ์ที่ใช้ทดสอบ ขณะทำการทดสอบ >



< รูปอุปกรณ์ที่ใช้ทดสอบ ขณะทำการทดสอบ >



< โปรแกรมทดสอบ >

```

sketch_jun10g
1 #define RS485TX HIGH
2 #define RS485RX LOW
3 #define RS485CTRL 5
4 #define LED_MONITOR 2
5 int stepcount = 0;
6 int eindex = 0;
7 byte echo[20];
8 byte slaveID = 0x03;
9 byte modbusCMD = 0x06;
10 byte h_relayID = 0x00;
11 byte l_relayID = 0x03;
12 byte relay_on = 0x01;
13 byte relay_off = 0x02;
14 byte on_off_delay = 0x00;
15 byte h_byteCRC = 0;
16 byte l_byteCRC = 0;
17 void setup() {
18   pinMode(RS485CTRL, OUTPUT);
19   pinMode(LED_MONITOR, OUTPUT);
20   Serial.begin(9600);
21   Serial2.begin(9600);
22   digitalWrite(RS485CTRL, RS485RX);
23   Serial.println("Start Test MODBUS RTU");
24 }
25 uint16_t CRC16_Update(uint16_t tempCRC, uint8_t inData) {
26   tempCRC ^= inData;
27   for (int i = 0; i < 8; i++) {
28     if (tempCRC & 1) {
29       tempCRC = (tempCRC >> 1) ^ 0xA001;
30     }
31     else {
32       tempCRC = tempCRC >> 1;
33     }
34   }
35   return tempCRC;
36 }
37 uint16_t sendByte_CRCUpdate(uint16_t tempCRC, uint8_t inData) {

```

```

#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) {

```

```

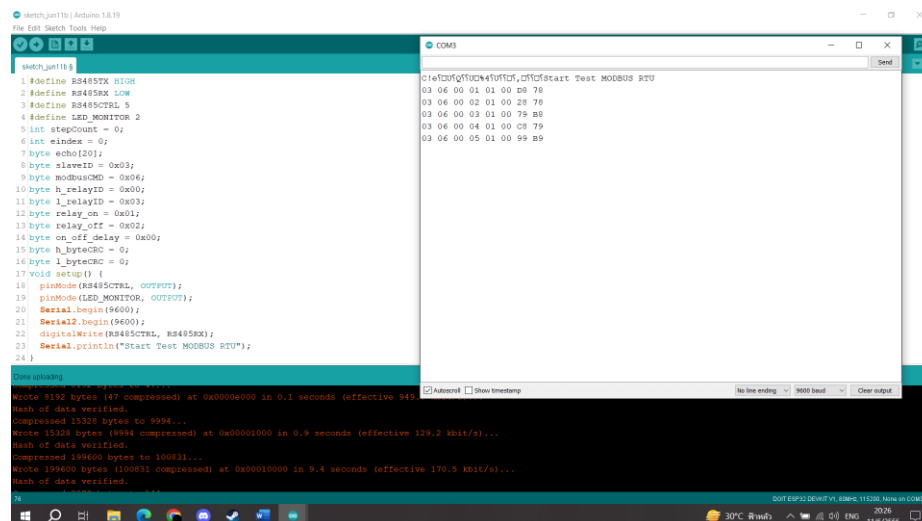
Serial2.write(inData);
if (inData < 0x10) Serial.print("0");
Serial.print(inData, HEX);
Serial.print(" ");
tempCRC = CRC16_Update(tempCRC, inData);
return tempCRC;
}

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 >



Quiz_203 – Read/Write Modbus RTU

< รูปอุปกรณ์ที่ใช้ทดสอบ ขณะทำการทดสอบ >



< รูปอุปกรณ์ที่ใช้ทดสอบ ขณะทำการทดสอบ >



< โปรแกรมทดสอบ >

The screenshot displays the Arduino IDE interface. The top menu bar includes 'File', 'Edit', 'Sketch', 'Tools', and 'Help'. The toolbar contains icons for opening files, saving, compiling, and uploading. The main text area shows the following C++ code:

```
sketch_jun15a
1 #define RS485TX HIGH
2 #define RS485RX LOW
3 #define RS485CTRL 5
4 #define LED_MONITOR 2
5 int stepCount = 0;
6 int eindex = 0;
7 byte echo[20];
8 byte slaveID = 0x05;
9 byte modbusCMD = 0x05;
10 byte h_relayID = 0x00;
11 byte l_relayID = 0x00;
12 byte relay_on = 0xFF;
13 byte relay_off = 0x00;
14 byte on_off_delay = 0x00;
15 byte h_byteCRC = 0;
16 byte l_byteCRC = 0;
17 void setup() {
18   pinMode(RS485CTRL, OUTPUT);
19   pinMode(LED_MONITOR, OUTPUT);
20   Serial.begin(9600);
21   Serial2.begin(9600);
22   digitalWrite(RS485CTRL, RS485RX);
23   Serial.println("Start Test MODBUS RTU");
24 }
25 uint16_t CRC16_Update(uint16_t tempCRC, uint8_t inData) {
26   tempCRC ^= inData;
27   for (int i = 0; i < 8; i++) {
28     if (tempCRC & 1) {
29       tempCRC = (tempCRC >> 1) ^ 0xA001;
30     }
31   }
32 }
33 void loop() {
34   // ... (code is partially obscured)
35 }
```

At the bottom of the IDE, the status bar shows 'DOT: ESP32 DevKit v1, Board: 115200, None on COM1'. The Windows taskbar at the very bottom indicates the system time as 13:58 on 15/6/2566.

```
#define RS485TX HIGH
#define RS485RX LOW
#define RS485CTRL 5
#define LED_MONITOR 2
int stepCount = 0;
int eindex = 0;
byte echo[20];
byte slaveID = 0x05;
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 < 6; 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);
}

```

< ผลการทดสอบ >

The screenshot shows the Arduino IDE interface. The main window displays the sketch code, which includes CRC calculations and relay control logic. The serial monitor window is open, showing the output of the sketch. The output consists of hexadecimal data pairs separated by a double arrow, indicating the state of the relay and the board readings.

```

// sketch_jun15a | Arduino 1.8.19 (Windows Store 1.8.37.0)
File Edit Sketch Tools Help

sketch_jun15a
75: calculateCRC = sendByte_CRCUpdate(calculateCRC, 0x00);
76: calculateCRC = sendByte_CRCUpdate(calculateCRC, 0x00);
77: calculateCRC = sendByte_CRCUpdate(calculateCRC, 0x08);
78: h_byteCRC = highByte(calculateCRC);
79: l_byteCRC = lowByte(calculateCRC);
80: calculateCRC = sendByte_CRCUpdate(calculateCRC, l_byteCRC);
81: calculateCRC = sendByte_CRCUpdate(calculateCRC, h_byteCRC);
82: delay(10);
83: digitalWrite(RS485CTRL, RS485RX);
84: digitalWrite(LED_MONITOR, LOW);
85: eindex = 0;
86: while (Serial2.available()) {
87:   echo[eindex] = Serial2.read();
88:   eindex++;
89: }
90: Serial.print(">> ");
91: for (int i = 0; i < 6; i++) {
92:   if (echo[i] < 0x10) Serial.print("0");
93:   Serial.print(echo[i], HEX);
94:   Serial.print(" ");
95: }
96: Serial.println();
97:
98: void loop() {
99:   relayCTRL(0, relay_on); delay(1500); readBoard(); delay(1500);
100:  relayCTRL(0, relay_off); delay(1500); readBoard(); delay(1500);
101:  relayCTRL(1, relay_on); delay(1500); readBoard(); delay(1500);
102:  relayCTRL(1, relay_off); delay(1500); readBoard(); delay(1500);
103: }

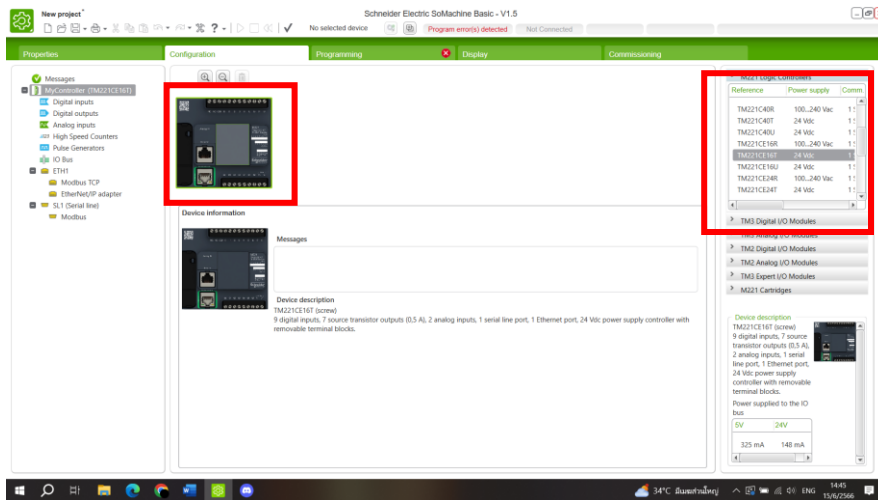
Uploading...
Board resetting via STSP pin...

COM1
05 02 00 00 00 08 78 48 >> 90 00 00 48 78 8E
05 05 00 01 FF 00 DC 7E
05 02 00 00 00 08 78 48 >> 90 00 00 48 78 8E
05 05 00 01 00 00 9D 8E
05 02 00 00 00 08 78 48 >> 90 00 00 48 78 8E
05 05 00 00 FF 00 8D 8E
05 02 00 00 00 08 78 48 >> 90 00 00 48 78 8E
05 05 00 01 FF 00 DC 7E
05 02 00 00 00 08 78 48 >> 90 00 00 48 78 8E
05 05 00 01 00 00 9D 8E
05 02 00 00 00 08 78 48 >> 90 00 00 48 78 8E
05 05 00 00 FF 00 8D 8E
05 02 00 00 00 08 78 48 >> 90 00 00 48 78 8E
05 05 00 00 00 00 CC 4E
05 02 00 00 00 08 78 48 >> 90 00 00 48 78 8E
05 05 00 01 FF 00 DC 7E
05 02 00 00 00 08 78 48 >> 90 00 00 48 78 8E
05 05 00 01 FF 00 DC 7E
Autoscroll Show timestamp Newline 9600 baud Clear output

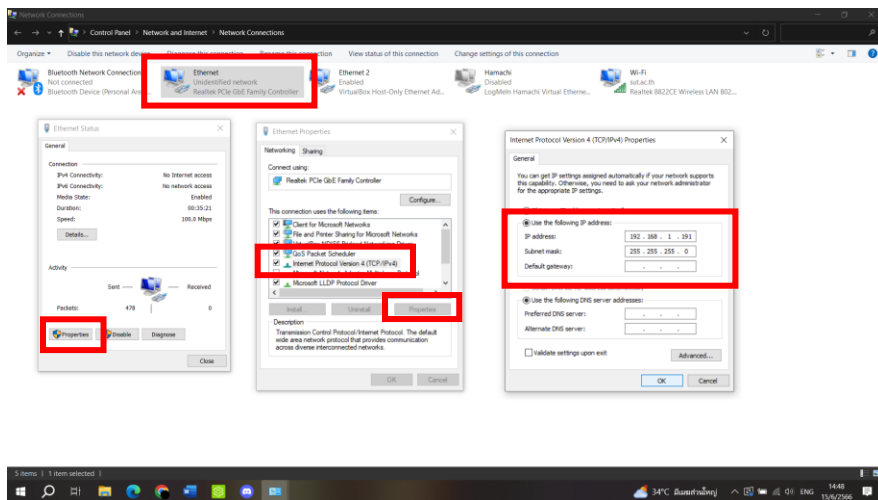
```

Quiz_204 – PLC Test

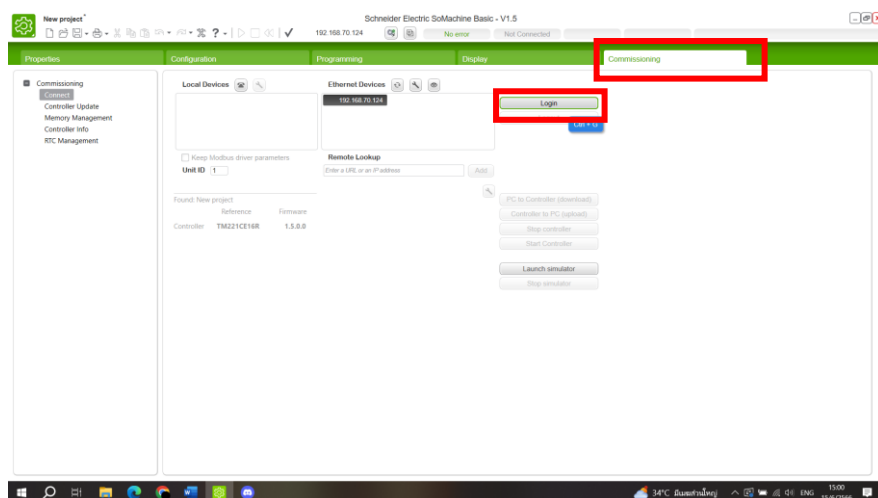
เปิดโปรแกรม SoMachine → เลือก TM221CE16R



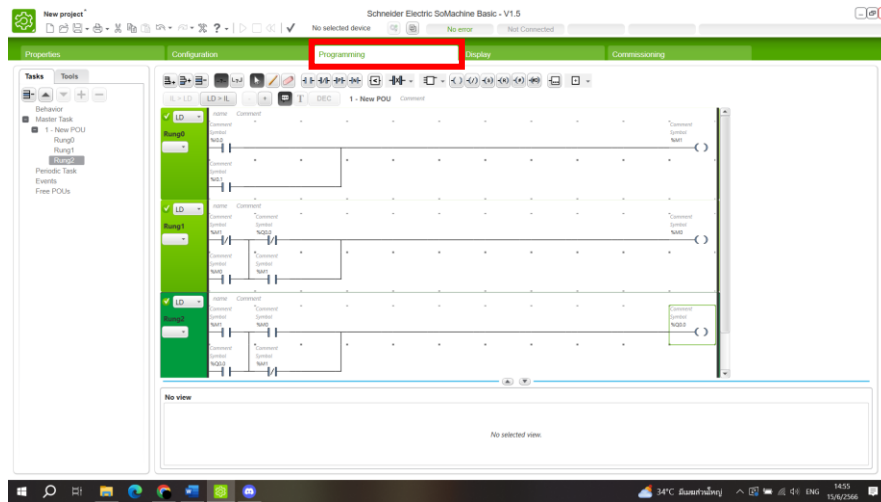
กำหนด IP address ของ PC ให้อยู่ในวง LAN เดียวกันกับ PLC



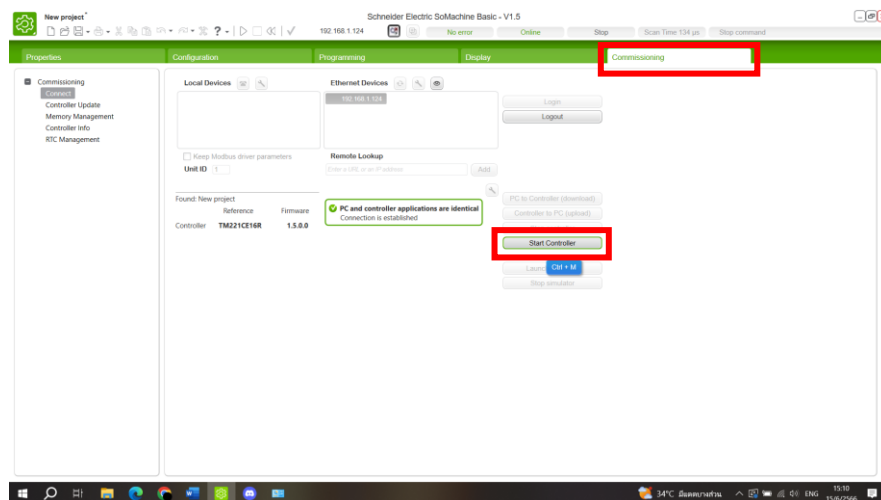
ไปที่ Commissioning --> login



กลับมาที่ Programming และ ทำการต่อวงจร



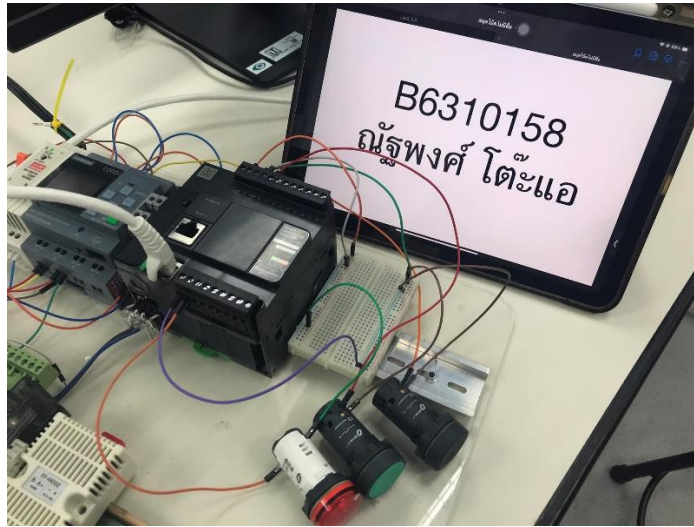
ทำการ Run เข้าไปที่ Commissioning --> Start Controller



รูปวงจร 1



รูปวงจร 2



รูปวงจร 3

