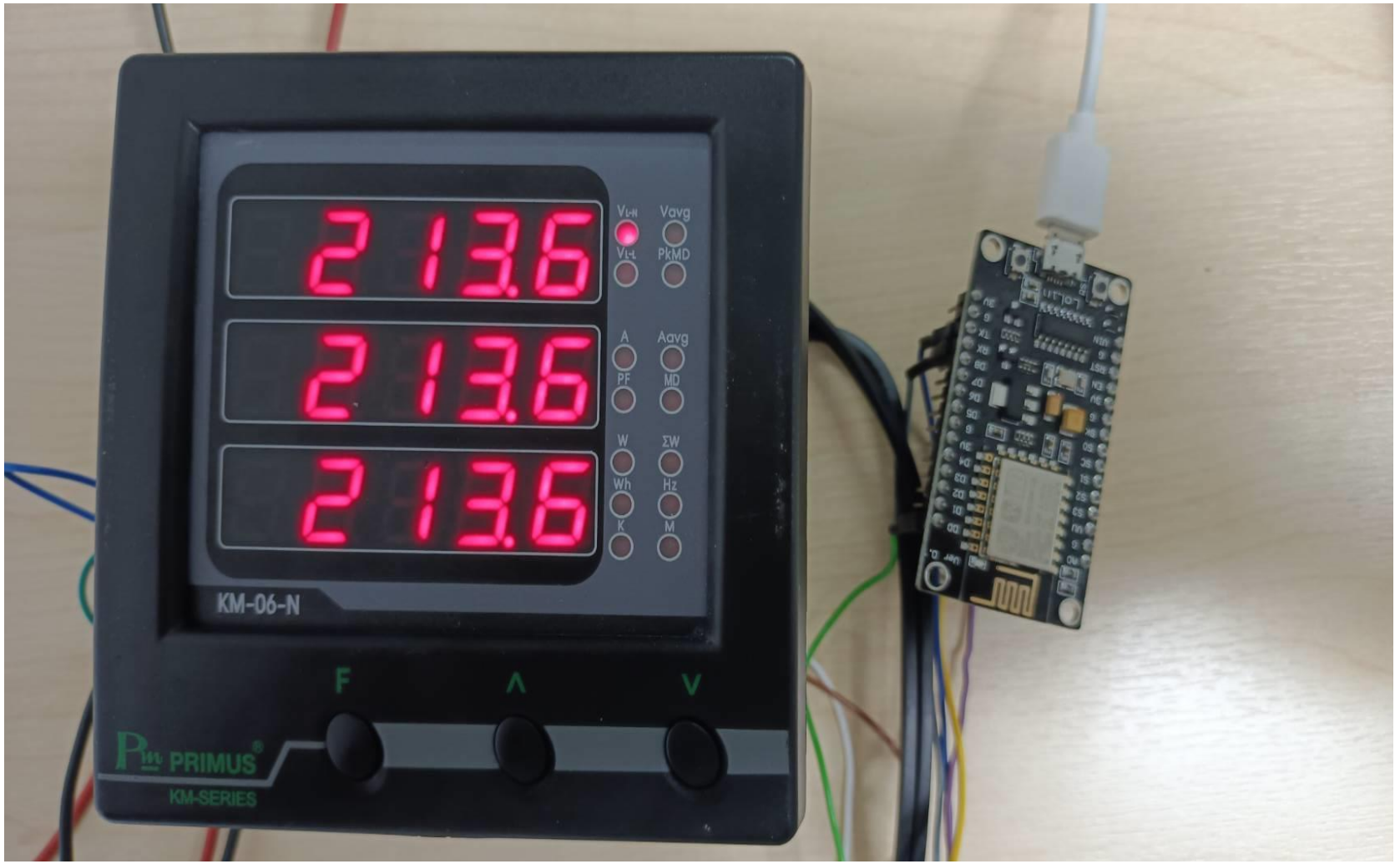


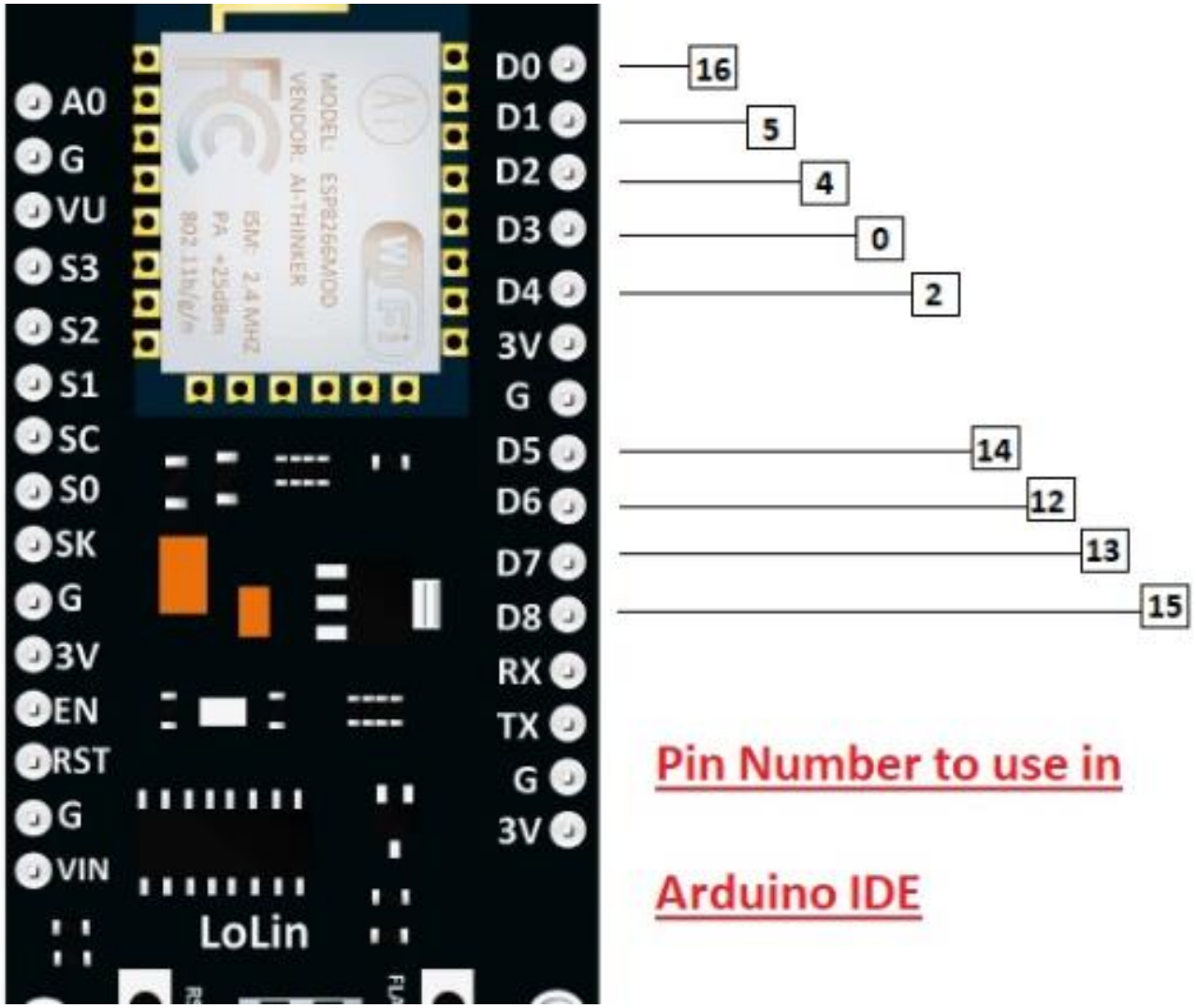
DEMO CODE ARDUNIO ESP8266 MODBUS RTU MASTER MODE

KM-06N PRIMUS

ESP8266 MODBUS MASTER WITH KM-06N

ส่งที่ใช้สำหรับการ run demo code

1. ESP8266
   - UART RX,TX
   - D0 (Digital GPIO16)
2. Card RS485
3. KM-06N



WIRING DIAGRAM

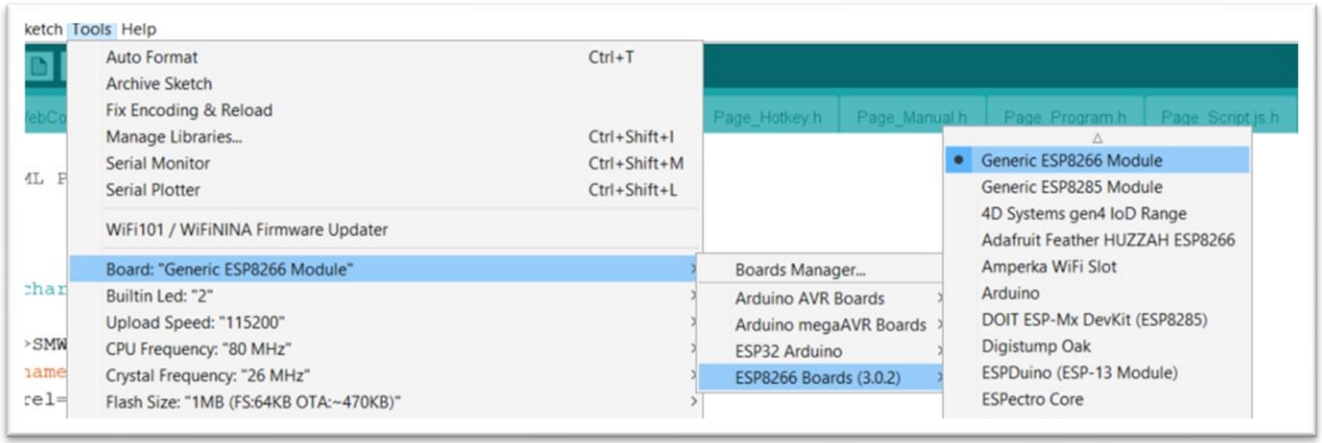
KM-06N ตั้ง Addr = 1 Baud =19200

ARDUINO IDE PREPARATION

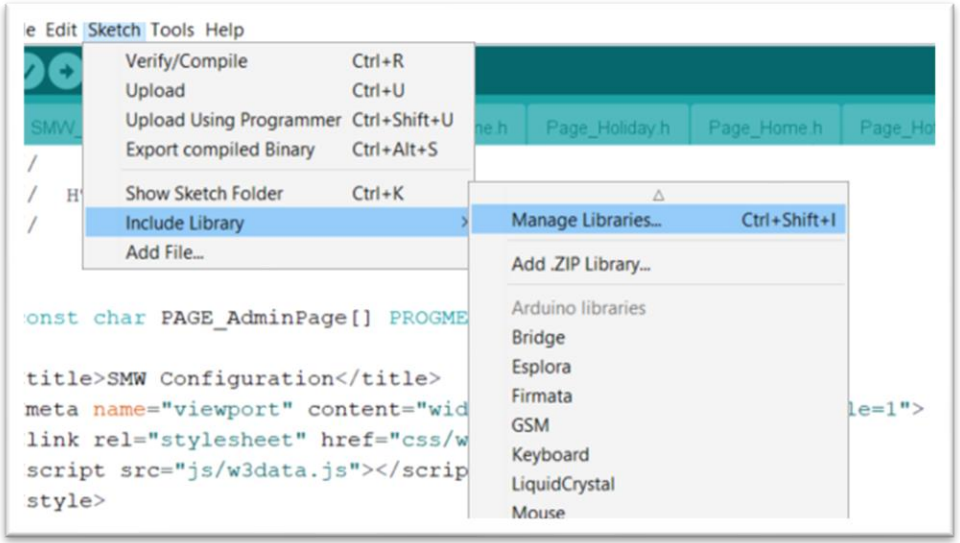
- Install Arduino IDE
- Install ESP8266
  - รายละเอียดสามารถดูได้จาก link
  - https://randomnerdtutorials.com/how-to-install-esp8266-board-arduino-ide/
- Install MODBUS MASTER Library

INSTALL MODBUS MASTER LIBRARY

1. เปิดโปรแกรม Arduino ไปยัง Menu Tools เลือก Board Generic ESP8266 Module

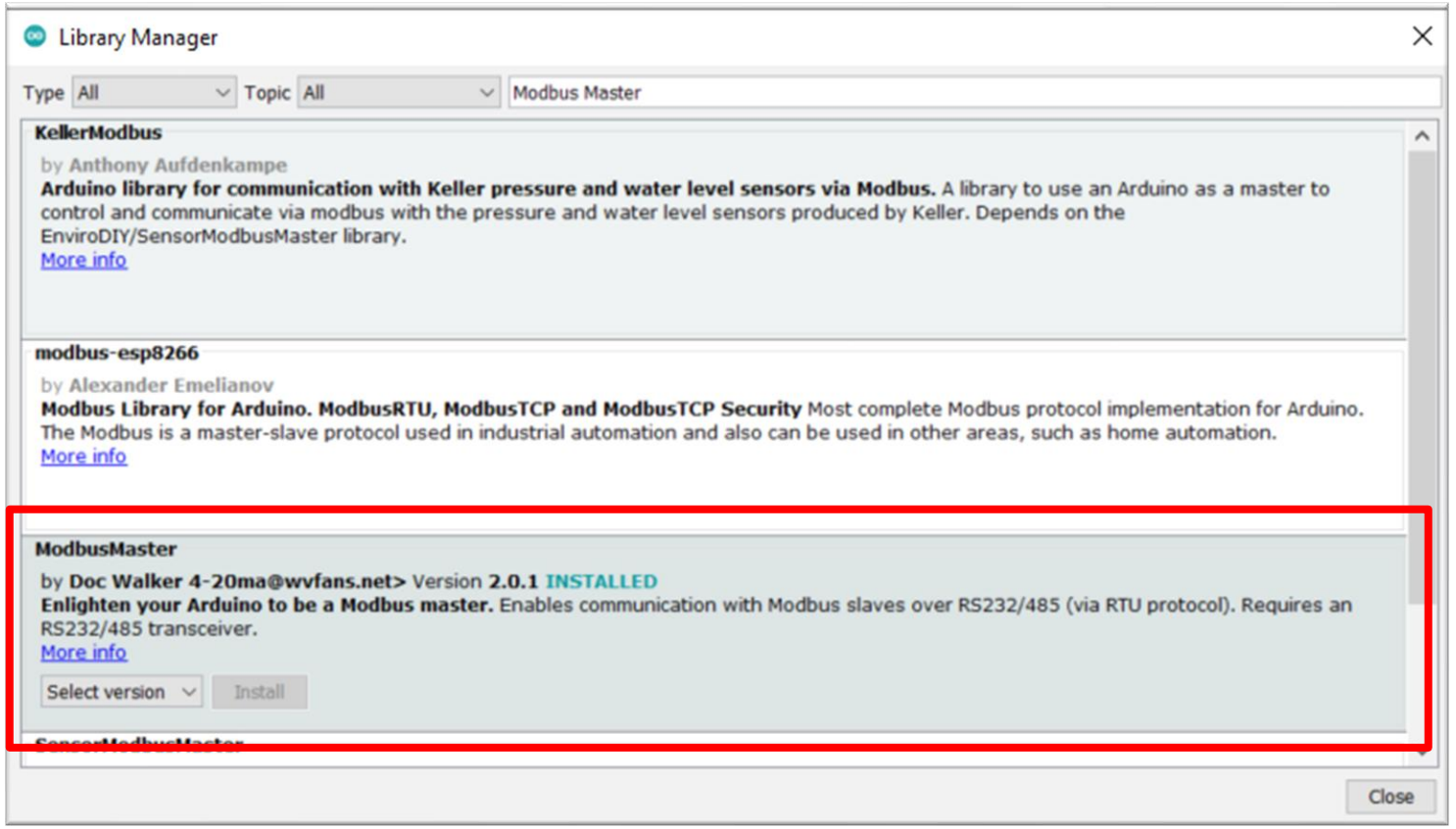


2. เปิดโปรแกรม Arduino ไปยัง Menu Sketch เลือก Include Library -> Manage Libraries



INSTALL MODBUS MASTER LIBRARY

- ค้นหา Modbus Master และ Click Install ModbusMaster by Doc Walker 4-20ma@wvfans.net



RUN DEMO CODE

- UnZip SimpleMBServer.rar
- เปิด file sketch : SimpleMBServer.ino

- ตั้ง WIFI SSID และ WIFI PASSWORD

- Define MbSlaveID เป็น 1
- ตั้ง Addr ของ KM-06N เป็น 1

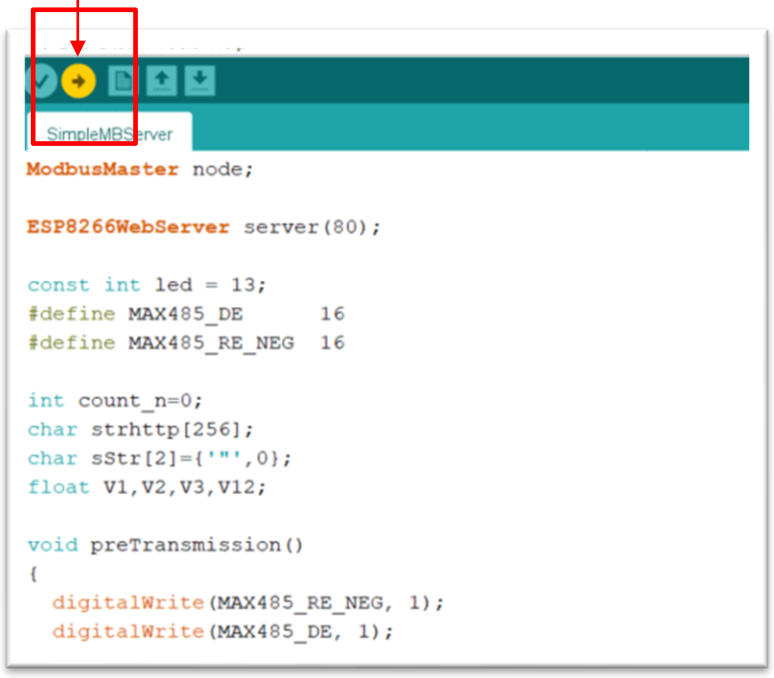
- Define MAX485_DE และ MAX485_RE_NEG เป็น 16 คือ ให้ GPIO 16 เป็นสัญญาณควบคุม MAX485

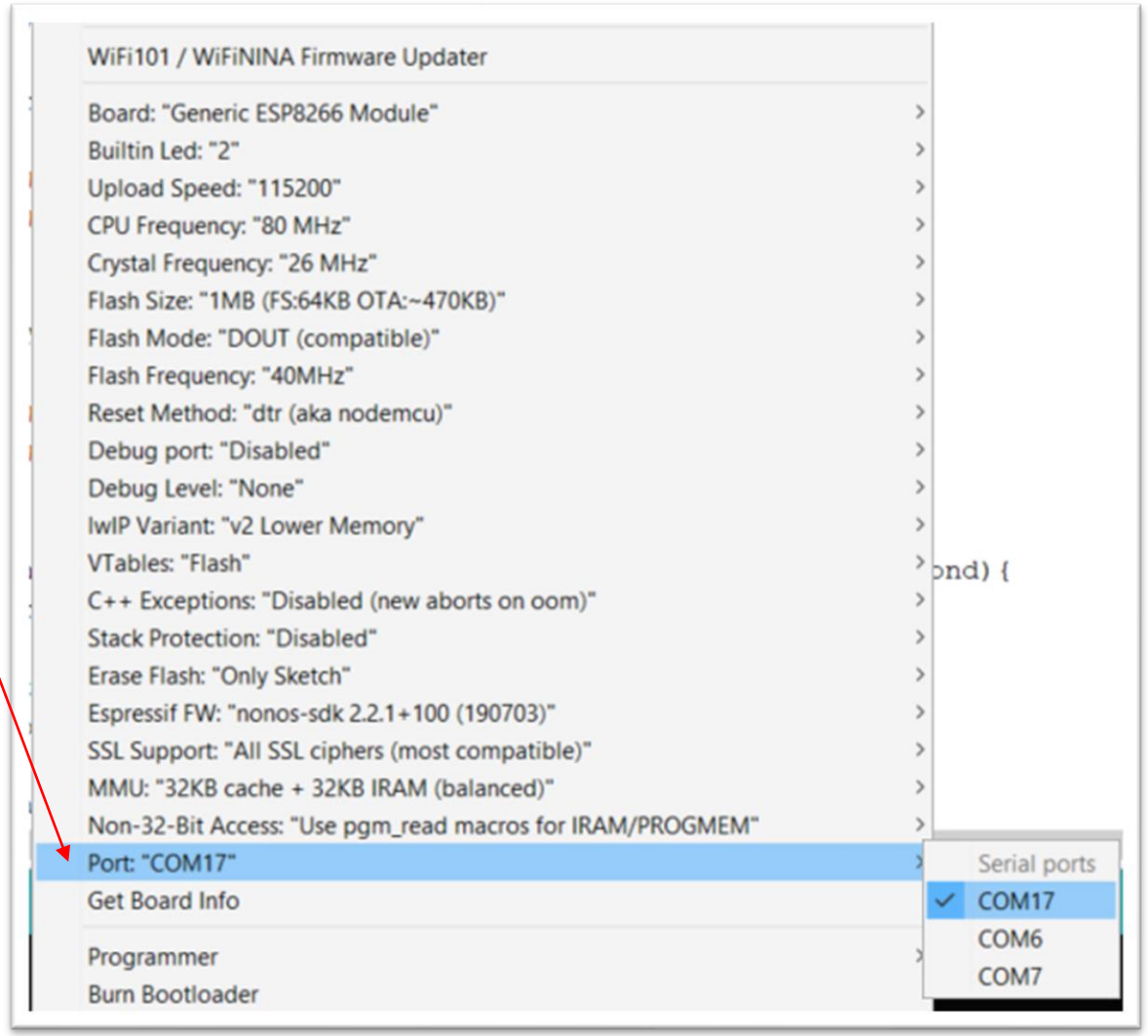
SimpleMBServer

```
#include <ESP8266WiFi.h>
#include <WiFiClient.h>
#include <ESP8266WebServer.h>
#include <ESP8266mDNS.h>
#include <ModbusMaster.h>

#ifndef STASSID
#define STASSID "Your WIFI SSID"
#define STAPSK  "Your WIFI PASSWORD"
#endif
#define MbSlaveID 1

const char* ssid = STASSID;
const char* password = STAPSK;
// instantiate ModbusMaster object
ModbusMaster node;

ESP8266WebServer server(80);

const int led = 13;
#define MAX485_DE      16
#define MAX485_RE_NEG  16
```

RUN DEMO CODE

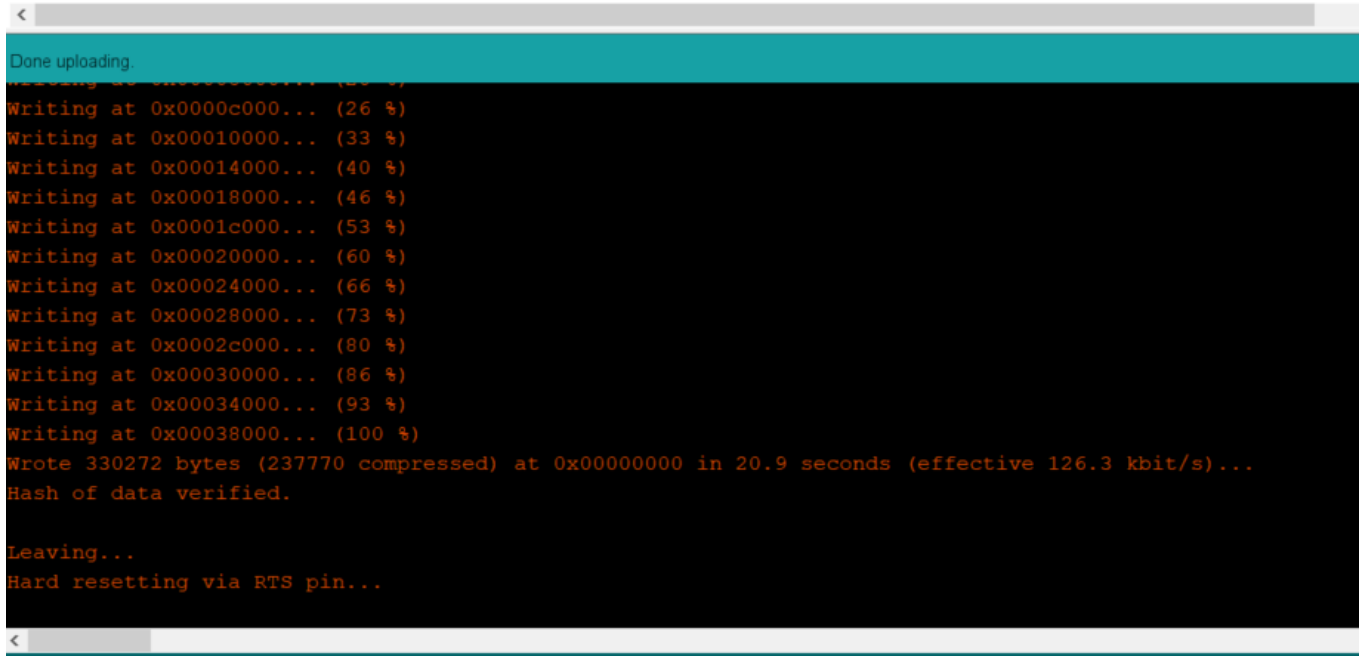
- เลือก Serial Comport ที่ใช้สำหรับ Program ESP8266
- Click compile and Upload program





RUN DEMO CODE

- เมื่อ Compile และ Upload สำเร็จ console จะแสดงดังรูป

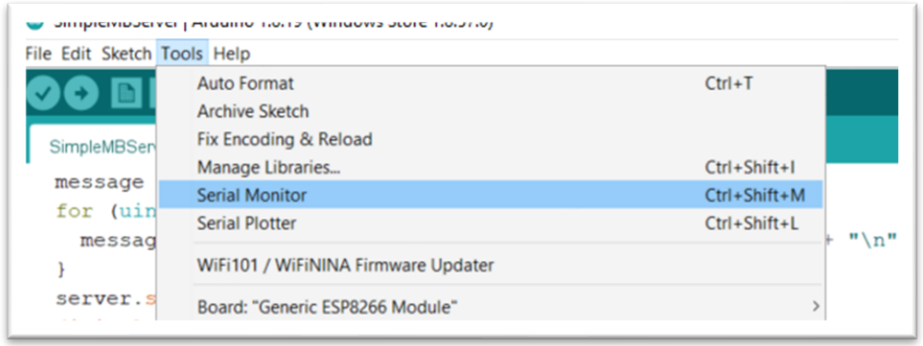


Trick

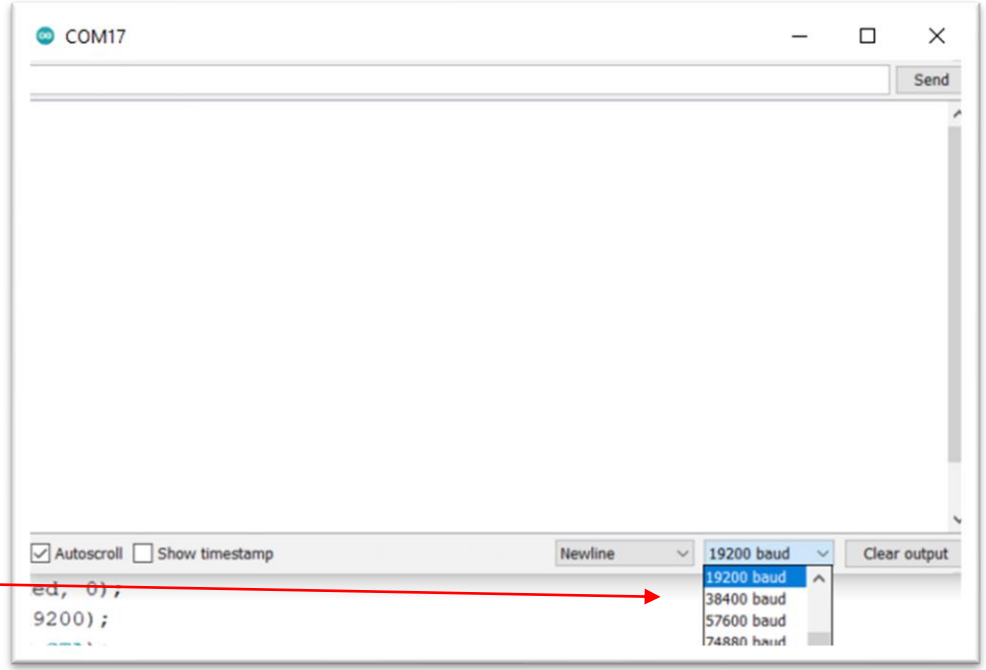
ในกรณีที่ compile สำเร็จแต่ไม่สามารถ Upload ได้ให้ตรวจสอบ port ที่เลือกใช้งานอาจจำเป็นต้องถอดและเสียบ USB ของ ESP8266 ใหม่ หรือเป็นเพราะ Serial UART บน Board ESP8266 ที่ใช้งานช่องเดียวกับการ Upload โปรแกรม ให้ถอดสาย TX บน Board เวลา Upload โปรแกรม

RUN DEMO CODE

- การตรวจสอบว่า ESP8266 เชื่อมต่อกับ WIFI แล้วได้รับ IP อะไรจำเป็นต้องใช้ Tools -> Serial monitor บนโปรแกรม arduion IDE ช่วย

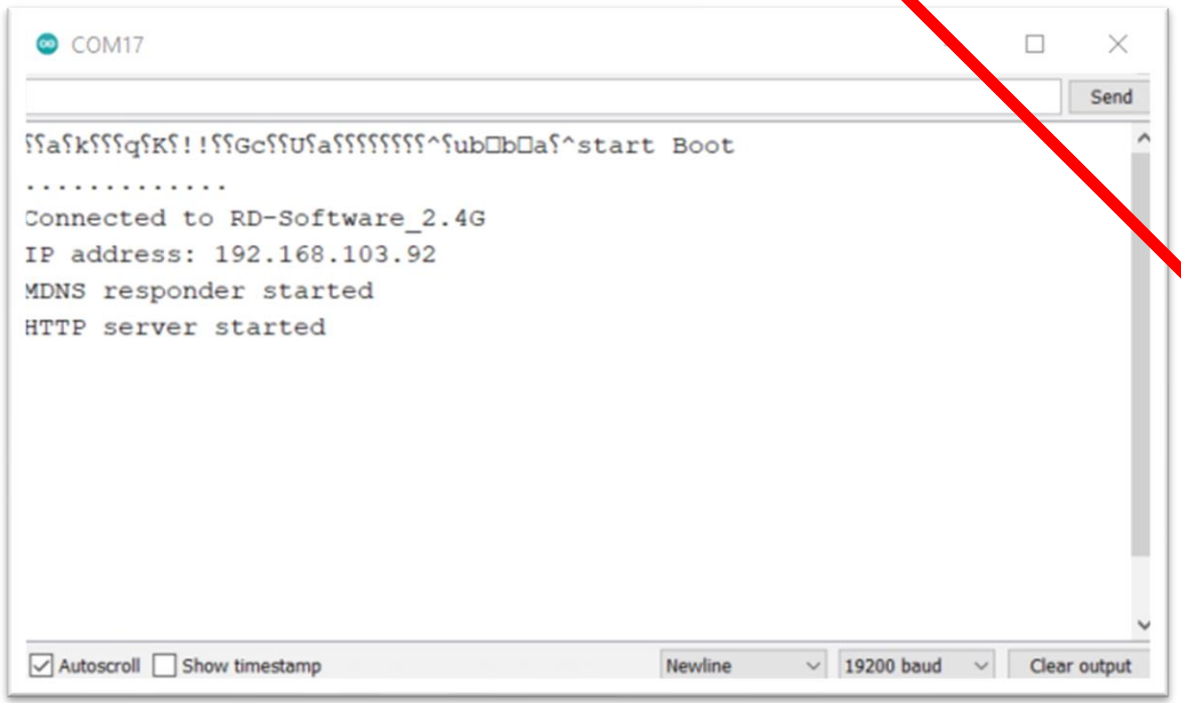


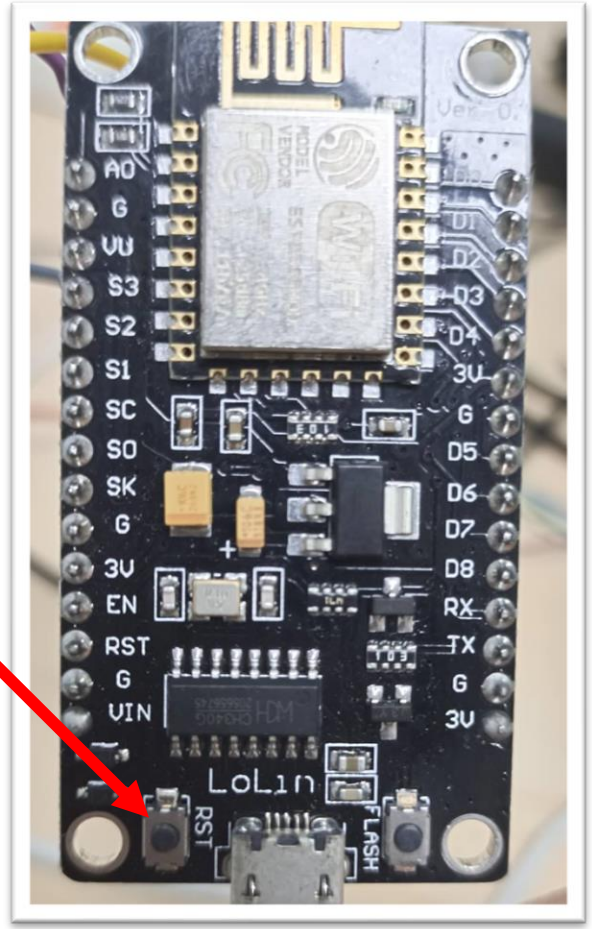
- เลือก Baud rate 19200



RUN DEMO CODE

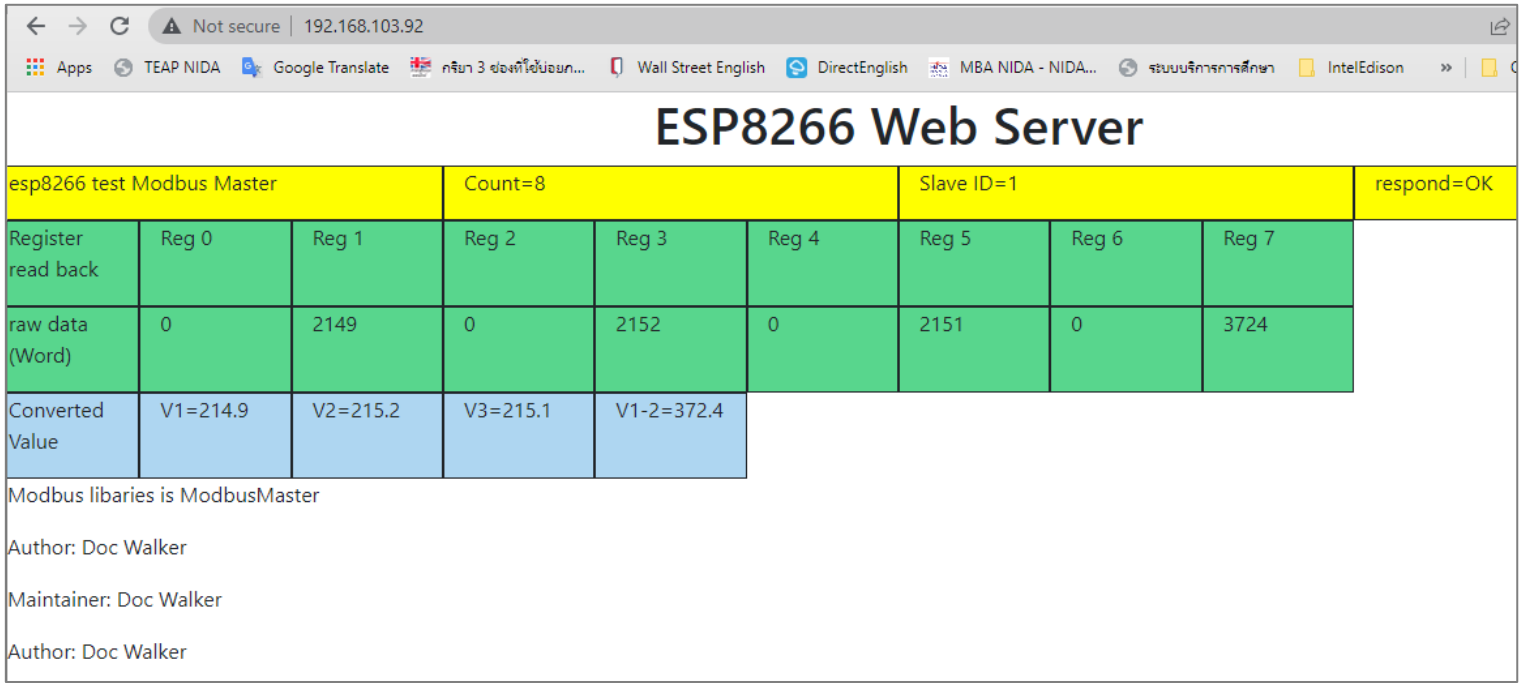
- การตรวจสอบว่า ESP8266 เชื่อมต่อกับ WIFI แล้วได้รับ IP (ต่อ)
  - กดปุ่ม Reset เพื่อดูผลการเชื่อมต่อ





RUN DEMO CODE

- ใน IP ที่ได้จาก Serial Monitor ไปเปิดบน web browser
- ทุกครั้งที่มีการ refresh browser โปรแกรมที่เขียนไว้จะสั่งอ่าน KM-06N หนึ่งครั้งและนำมาแสดงผล
- Count คือ จำนวนครั้งที่ทำการ refresh
- Slave ID คือ Addr ที่ตั้งบน KM-06N
- Respond คือ แสดงสถานะการอ่าน OK คือมีอุปกรณ์ KM-06N ตอบกลับมา หากเกิดปัญหา 226 คือไม่มีการตอบกับจาก KM-06N
- Raw data คือ ค่าที่อ่านกลับมาได้จาก Meter



CODE DESCRIPTION

- อธิบายการใช้ Code Modbus Master แบบย่อ

1. ทำการ define GPIO สำหรับสัญญาณควบคุม MAX485

```
ESP8266WebServer server(80);

const int led = 13;
#define MAX485_DE      16
#define MAX485_RE_NEG  16
```

2. เขียน Function ขับสัญญาณควบคุม คือ preTransmission ต้องสั่ง GPIO เป็น 1 และ postTransmission ต้องสั่ง GPIO เป็น 0

```
void preTransmission()
{
  digitalWrite(MAX485_RE_NEG, 1);
  digitalWrite(MAX485_DE, 1);
}

void postTransmission()
{
  digitalWrite(MAX485_RE_NEG, 0);
  digitalWrite(MAX485_DE, 0);
}
```

CODE DESCRIPTION

3. ใน function void setup(void) ทำการ initial GPIO และ serial ที่ baud rate 19200

```
void setup(void) {
  pinMode(led, OUTPUT);
  pinMode(MAX485_RE_NEG, OUTPUT);
  pinMode(MAX485_DE, OUTPUT);
  // Init in receive mode
  digitalWrite(MAX485_RE_NEG, 0);
  digitalWrite(MAX485_DE, 0);

  digitalWrite(led, 0);
  Serial.begin(19200);
  WiFi.mode(WIFI_STA);
  WiFi.begin(ssid, password);
  Serial.println("start Boot");

  // Wait for connection
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println("");
```

initial GPIO

initial Serial

4. Call server.on("/", handleRoot); เพื่อให้ ESP8266 ไป run void handleRoot() มีการ request จาก web browser

```
SimpleMBServer §
Serial.begin(19200);
WiFi.mode(WIFI_STA);
WiFi.begin(ssid, password);
Serial.println("start Boot");

// Wait for connection
while (WiFi.status() != WL_CONNECTED) {
  delay(500);
  Serial.print(".");
}
Serial.println("");
Serial.print("Connected to ");
Serial.println(ssid);
Serial.print("IP address: ");
Serial.println(WiFi.localIP());

if (MDNS.begin("esp8266")) {
  Serial.println("MDNS responder started");
}

server.on("/", handleRoot);

server.on("/inline", []() {
```

initial WIFI

Set function สำหรับ request จาก web browser

CODE DESCRIPTION

4. การทำงานของ void handleRoot()

```
count_n++;
//====================Modbus master==============================//
node.begin(MbSlaveID, Serial);
// Callbacks allow us to configure the RS485 transceiver correctly
node.preTransmission(preTransmission);
node.postTransmission(postTransmission);


result = node.readHoldingRegisters(6, 10);
if (result == node.ku8MBSuccess)
{
  for (j = 0; j < 10; j++)
  {
    MBdata[j] = node.getResponseBuffer(j);
  }
//=========================================================//
server.send(200, "text/html", SendHTML(count_n,MBdata,result));
digitalWrite(led, 0);
```

count_n++ สำหรับนับจำนวนรอบที่มีการ request

node.begin(MbSlaveID, Serial) คือการกำหนดคุณสมบัติของ node ให้มี salve id และใช้ Serial อะไรในการส่งข้อมูล

```
const char* ssid = STASSID;
const char* password = STAPSK;
// instantiate ModbusMaster object
ModbusMaster node;
```

Trip

node เป็นตัวแปรที่ถูกประกาศขึ้นจาก class ของ library

กำหนดให้ node ทำรัน preTransmission และ postTransmission ด้วย function ที่ถูกเขียนเตรียมไว้แล้ว

สั่งให้ node อ่าน ข้อมูลจากตำแหนงเริ่มต้นคือ register ที่ 6 และอ่านไปอีกทั้งหมด 10 register

CODE DESCRIPTION

4. การทำงานของ void handleRoot() (ต่อ)

```
count_n++;
//=====================Modbus master==============================//
node.begin(MbSlaveID, Serial);
// Callbacks allow us to configure the RS485 transceiver correctly
node.preTransmission(preTransmission);
node.postTransmission(postTransmission);


result = node.readHoldingRegisters(6, 10);
if (result == node.ku8MBSuccess)
{
  for (j = 0; j < 10; j++)
  {
    MBdata[j] = node.getResponseBuffer(j);
  }
//=========================================================//
server.send(200, "text/html", SendHTML(count_n,MBdata,result));
digitalWrite(led, 0);
```

ถ้า result success ให้ทำการ load ข้อมูลไปยังตัวแปล Mbdata[] เพื่อเก็บไว้คำนวนใน function SendHTML()

ส่งข้อมูลไปยัง function SendHTML() เพื่อแปลงเป็น HTML สำหรับ web browser

CODE DESCRIPTION

5. การทำงานของ String SendHTML(uint16_t count_n,uint16_t *ptdata,uint8_t respond)

count_n : จำนวนรอบที่มีการ request

*ptdata : ตัวแปล pointer รับอ่านจาก MBdata ที่เก็บค่าตอบกลับจาก Slave ID

respond : ตัวแปลเป็นผลการอ่าน modbus ว่าสำเร็จหรือไม่

ถ้า respond == 0 (OK) ให้ทำการคำนวนค่า Volt จาก register ที่อ่านมาได้

```
uint32_t buffData;
if(respond==0)     // Slave respond OK
{
//=======  Calculation from register     ====///
  buffData=*(ptdata);
  buffData=buffData<<16;
  buffData+=*(ptdata+1);
  V1=buffData;
  V1/=10;

  buffData=*(ptdata+2);
  buffData=buffData<<16;
  buffData+=*(ptdata+3);
  V2=buffData;
  V2/=10;
```

ค่า Volt ที่คำนวนได้ถูกแปลงเป็น HTML Tag ด้วย sprint

```
// Table 1 row 4  Modbus Converted value //
ptr +="<div class=\"row\">";
ptr +="<div class=\"col-lg-1 col-md-1 col-sm-1\" style=\"background-color:#AED6F1; border: 1px solid\">";
ptr +="<p>Converted Value</p>\n";
ptr +="</div>\n";
// R0
ptr +="<div class=\"col-lg-1 col-md-1 col-sm-1\" style=\"background-color:#AED6F1; border: 1px solid\">";
sprintf(str,"<p>V1=%0.1f</p>\n",V1);
ptr +=str;
ptr +="</div>\n";
ptr +="<div class=\"col-lg-1 col-md-1 col-sm-1\" style=\"background-color:#AED6F1; border: 1px solid\">";
sprintf(str,"<p>V2=%0.1f</p>\n",V2);
ptr +=str;
ptr +="</div>\n";
ptr +="<div class=\"col-lg-1 col-md-1 col-sm-1\" style=\"background-color:#AED6F1; border: 1px solid\">";
sprintf(str,"<p>V3=%0.1f</p>\n",V3);
ptr +=str;
ptr +="</div>\n";
ptr +="<div class=\"col-lg-1 col-md-1 col-sm-1\" style=\"background-color:#AED6F1; border: 1px solid\">";
sprintf(str,"<p>V1-2=%0.1f</p>\n",V12);
```