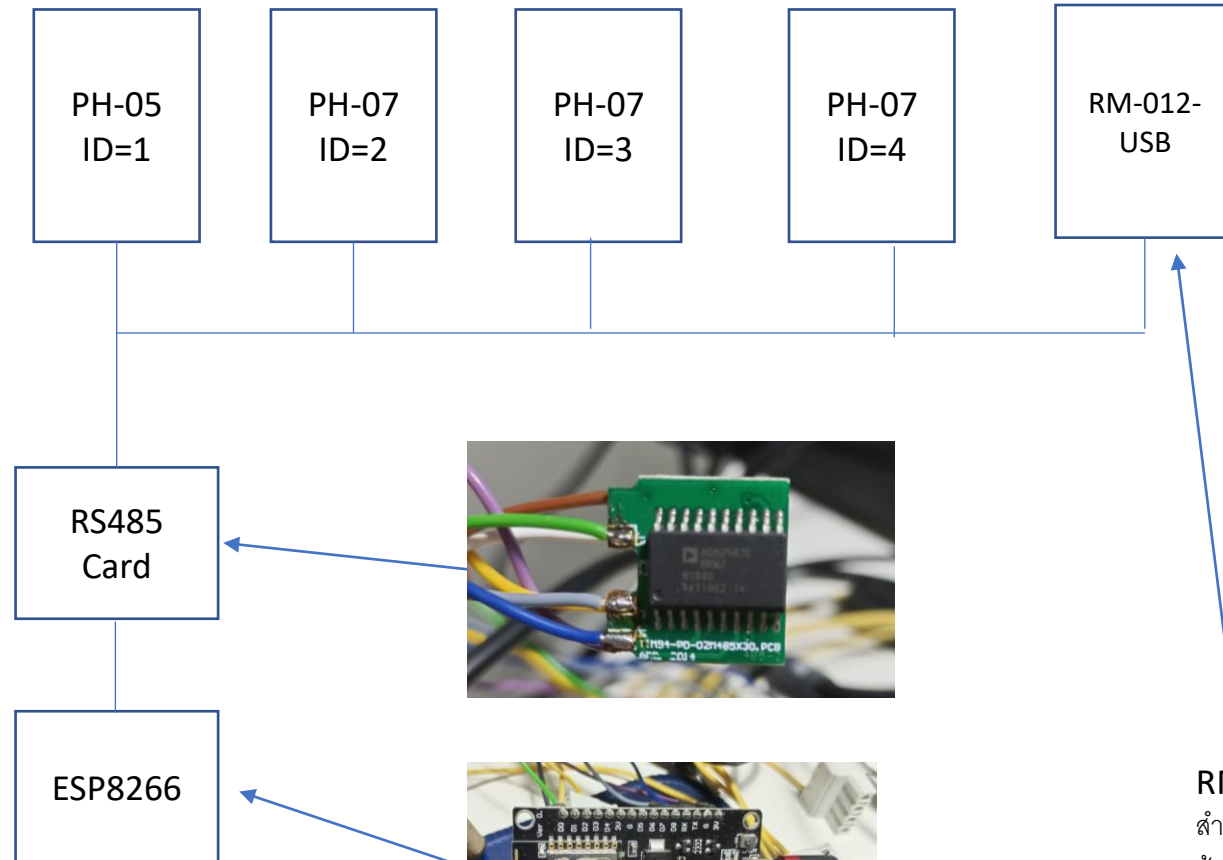


Example ESP8266 modbus
with PH-05 & PH-07

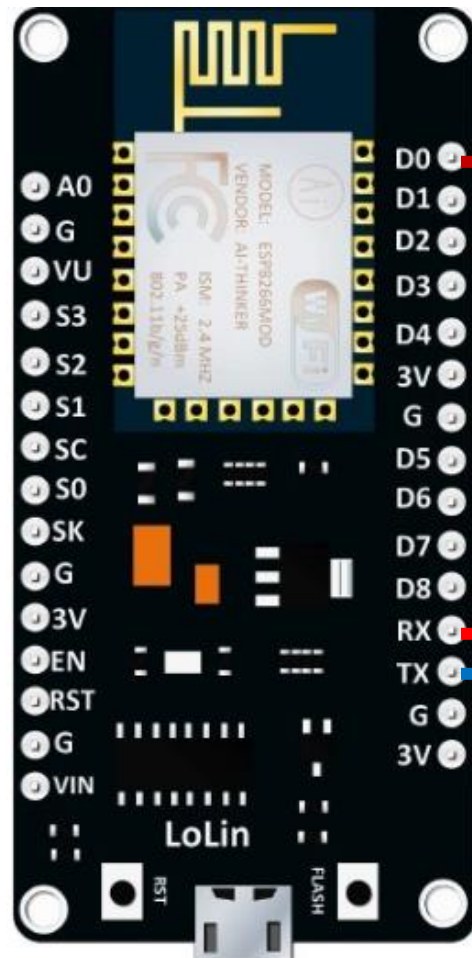
Lab test diagram



RM-012-USB

สำหรับ Capture Serial message ใน Bus และใช้ Serial port monitor เพื่อแสดงข้อมูลเป็น HEX

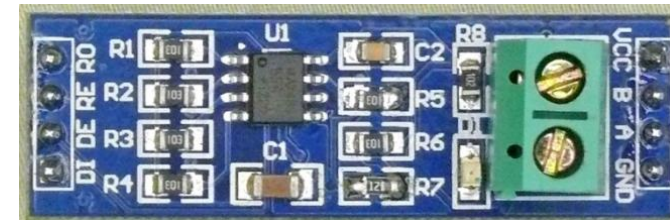
ESP8266 wiring diagram



PIN Control Module RS485
D0->(DE/RE)

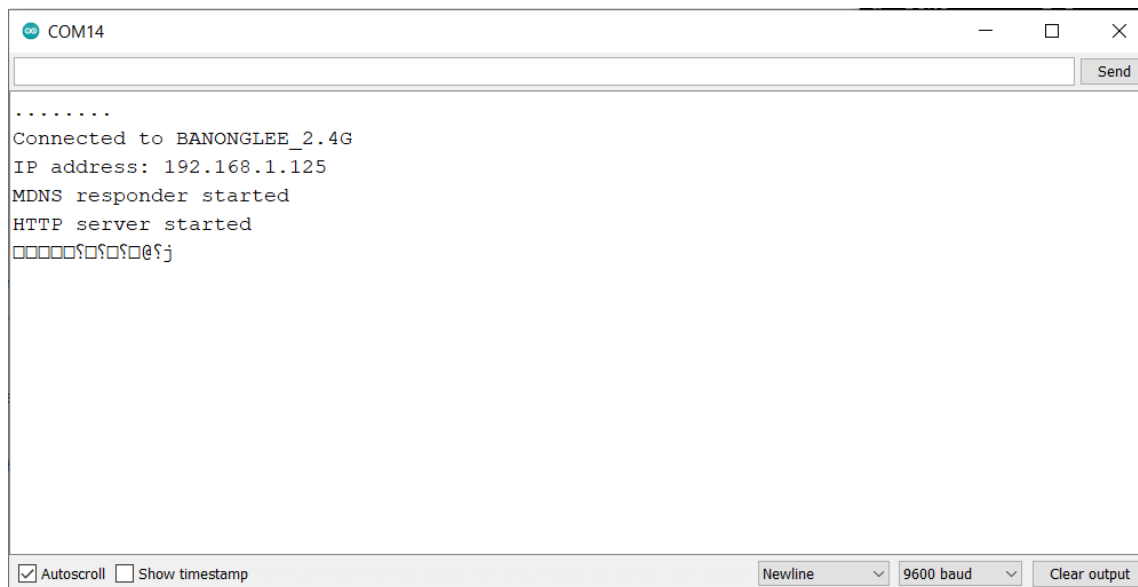
PIN RX -> (RO)

PIN TX -> (DI)



ESP8266

- Modbus library
 - ModbusMaster
- แสดงผลจำนวน request ทั้ง READ/WRITE และจำนวน Receive บน Web
- เริ่มทำงานให้เปิด Serial Monitor Port ที่โปรแกรมเพื่อรับข้อความแสดง IP ของ Web



```
.....
Connected to BANONGLEE_2.4G
IP address: 192.168.1.125
MDNS responder started
HTTP server started
[garbled characters]
```

ESP8266

เปิด IP ที่ได้รับมาจาก Serial monitor ใน web จะพบการแสดงผลจำนวน request และ respond ที่ได้รับตอบกลับ ทั้งแบบแยก module และ รวมทุก module

ESP8266 Web Server									
esp8266 test Modbus Master			total request=118			total receive=118			no
Register read back	PH-05 request	PH-05 receive	PH-07_2 request	PH-07_2 receive	PH-07_3 request	PH-07_3 receive	none	none	
raw data (Word)	30	30	30	30	28	28	30	30	
Modbus libraries is ModbusMaster									

กรณีที่ request แล้วมีการตอบกลับถูกต้อง
จำนวน request = receive

ESP8266 Web Server									
esp8266 test Modbus Master			total request=708			total receive=700			none
Register read back	PH-05 request	PH-05 receive	PH-07_2 request	PH-07_2 receive	PH-07_3 request	PH-07_3 receive	none	none	
raw data (Word)	178	170	176	176	176	176	178	178	
Modbus libraries is ModbusMaster									

กรณีที่ request แล้วมีการตอบกลับไม่ถูกต้อง หรือไม่
ตอบกลับ จำนวน request != receive

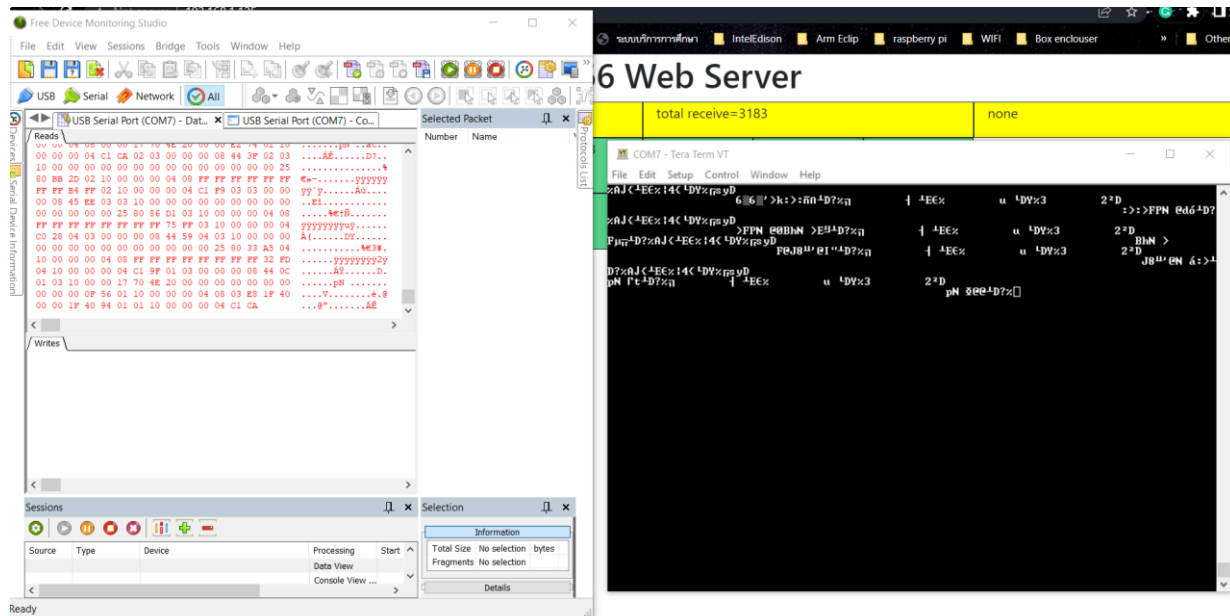
Note

เนื่องจากใช้ Html แบบง่ายโดย refresh ทุก 5 วิ ทำให้ web มีอาการค้างได้
ต้องกดปุ่ม refresh ที่ browser เองเพื่อแก้ไข

ESP8266

การ Monitor Message ใช้ Serial Port อีกชุดมาพวงสัญญาณเพื่อป้องกันการ กวนกัน ระหว่าง ESP8266 UART และ USB Serial บนตัว ESP8266

1. เปิด Port ที่ต่อพวงสัญญาณด้วย Program เช่น HyperTerminal/Putty/Tera Term etc.
2. เปิดโปรแกรม Free Device Monitoring Studio เลือก Port ที่ต้องการ Monitor เลือกแสดง Visualizers เป็น Data View เพื่อแสดง message ในรูปแบบ HEX ได้



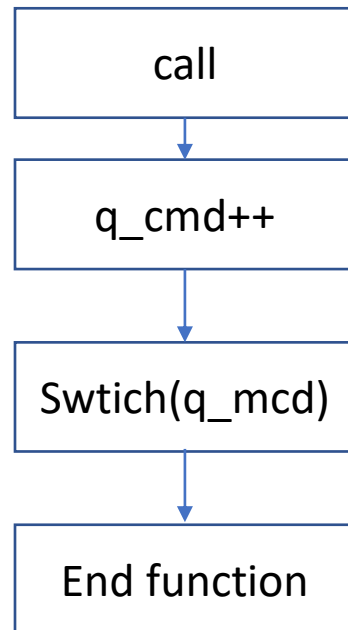
Note

1. การแสดงเป็น Read อย่างเดียวไม่มีการ Write เพราะ Port นี้จะทำหน้าที่อ่าน Message บน Bus เท่านั้นไม่มีการส่งเขียนข้อความได้
2. การส่งข้อมูลและตอบกลับที่ละข้อความจะทำให้วิเคราะห์ได้สะดวกขึ้น

ESP8266

```
void loop(void) {  
  //ArduinoOTA.handle();  
  server.handleClient();  
  MDNS.update();  
  if(millis() - timeStamp > timeIntervall/4){  
    mb_run();  
    timeStamp = millis(); // reset the timer  
  }  
}
```

การเปลี่ยน Speed ในการ request แต่ละครั้งทำได้ที loop()



q_cmd =0 -> read PH-05 ID=1 จำนวน 8 register
q_cmd =1 -> multiwrite PH-05 ID=1 จำนวน 8 register
q_cmd =2 -> read PH-07 ID=2 จำนวน 8 register
q_cmd =3 -> multiwrite PH-07 ID=2 จำนวน 8 register
q_cmd =4 -> read PH-07 ID=3 จำนวน 8 register
q_cmd =5 -> multiwrite PH-07 ID=3 จำนวน 8 register
q_cmd =6 -> read PH-07 ID=4 จำนวน 8 register
q_cmd =7 -> multiwrite PH-07 ID=4 จำนวน 8 register