# การควบคุมเครื่องจักรอัจฉริยะโดยใช้การสื่อสารระหว่างเครื่องจักรกับเครื่องจักร
# M2M - Intelligence Machine Control

ชื่อ-สกุล : นายอติชาติ ภู่นิเทศ

## 5/5: -- คำถามท้ายบทเพื่อทดสอบความเข้าใจ

### Quiz_301 – Start SCADA

< รูปอุปกรณ์ที่ใช้ทดสอบ ขณะทำการทดสอบ >

< รูปอุปกรณ์ที่ใช้ทดสอบ ขณะทำการทดสอบ >

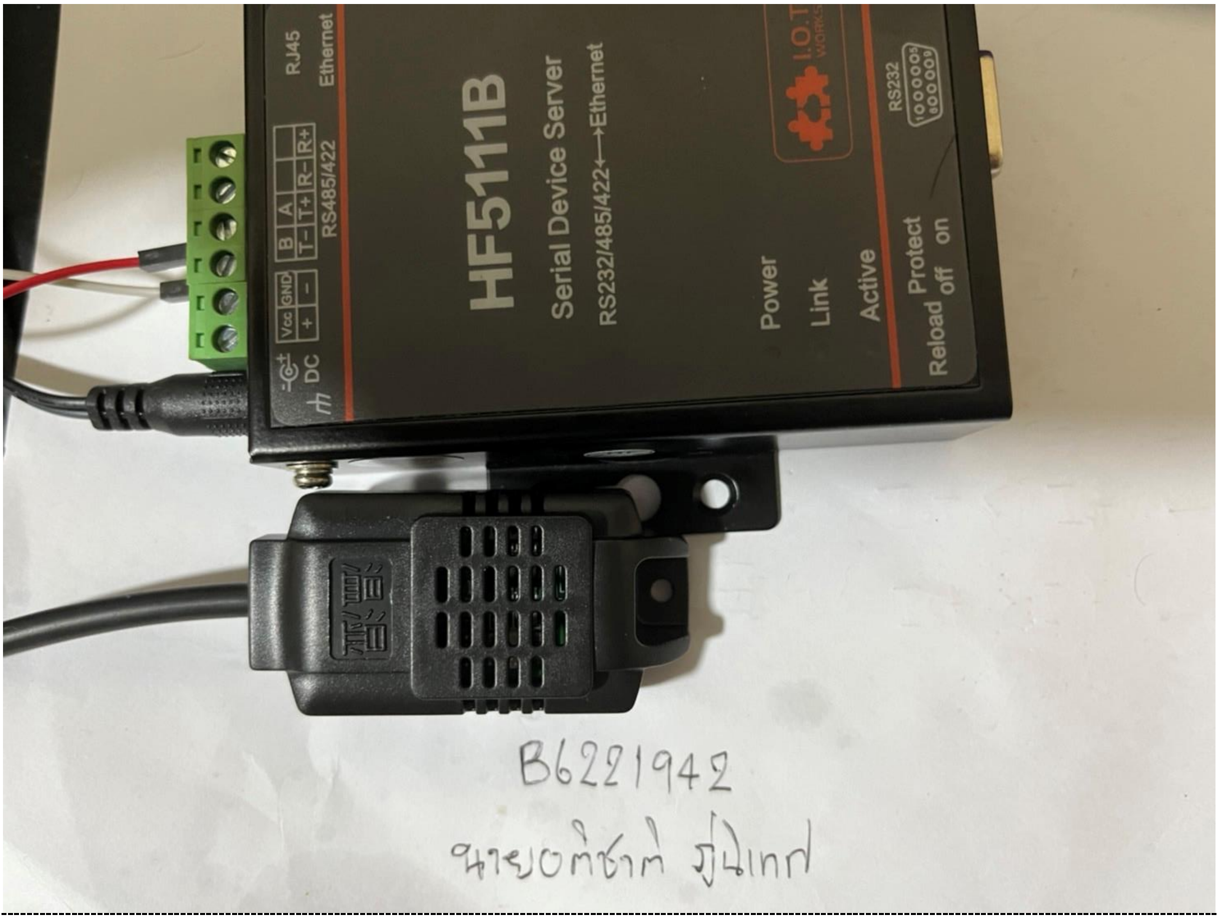
รายยละเอียดการทดสอบ

< โปรแกรมทดสอบ >

< ผลการทดสอบ >

### Quiz_302 – Modbus TCP Read/Write

< รูปอุปกรณ์ที่ใช้ทดสอบ ขณะทำการทดสอบ >



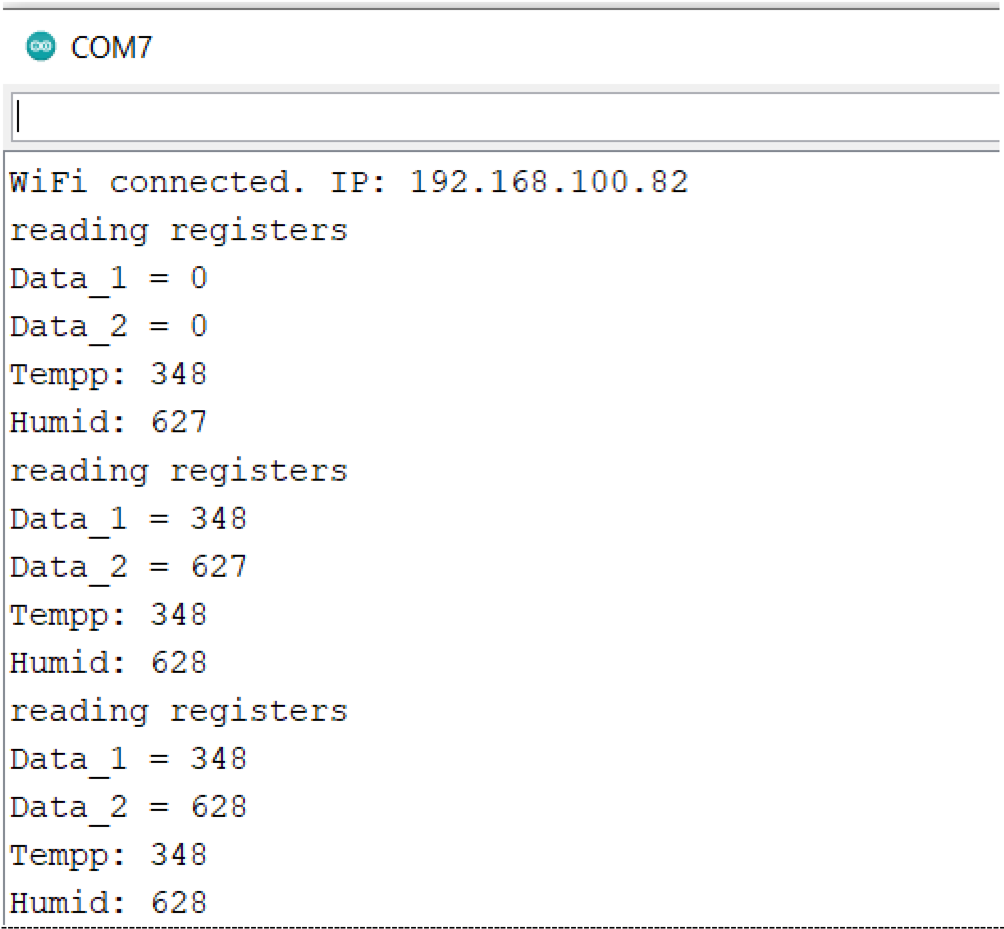
< รูปอุปกรณ์ที่ใช้ทดสอบ ขณะทำการทดสอบ >

รายละเอียดการทดสอบ

< โปรแกรมทดสอบ >

```
#include <Arduino.h>
#include <WiFi.h>
#include <esp32ModbusTCP.h>
char ssid[] = "LANTANIDEs-2.4G";
char pass[] = "0887040892";
bool WiFiConnected = false;
esp32ModbusTCP sunnyboy(1, {192, 168, 100, 183}, 502);

enum smaType {
  ENUM, // enumeration
  UFIX0, // unsigned 2 Byte, no decimals
  SFIX0, // signed 4 Byte, no decimals
};

struct smaData {
  const char* name;
  uint16_t address;
  uint16_t length;
  smaType type;
  uint16_t packetId;
};

smaData smaRegisters[] = {
 "Tempp", 0, 1, UFIX0, 0,
 "Humid", 1, 1, UFIX0, 0
};

Uint8_t numberSmaRegisters = sizeof(smaRegisters) / sizeof(smaRegisters[0]);
Uint8_t currentSmaRegister = 0;
uint16_t ResultData[3];

void setup() {
  Serial.begin(115200);
  WiFi.disconnect(true); // delete old config
  sunnyboy.onData([](uint16_t packet, uint8_t slave, esp32Modbus::FunctionCode fc , uint8_t* data ,
uint16_t len) {
    for (uint8_t i = 0; i < numberSmaRegisters; ++i) {
      if (smaRegisters[i].packetId == packet) {
        smaRegisters[i].packetId = 0;
        switch (smaRegisters[i].type) {
          case ENUM:
          case UFIX0: {
              uint32_t value = 0; // 2-Byte Data
              value = (data[0] << 8) | (data[1]); // 2-Byte Data
              Serial.printf("%s: %u\n", smaRegisters[i].name, value);
              ResultData[i] = value;
              break;
            }
          case SFIX0: {
              int32_t value = 0;
              value = (data[0] << 24) | (data[1] << 16) | (data[2] << 8) | (data[3]);
              Serial.printf("%s: %i\n", smaRegisters[i].name, value);
              break;
            }
        }
        return;
      }
    }
  });
  sunnyboy.onError([](uint16_t packet, esp32Modbus::Error e) {
    Serial.printf("Error packet %u: %02x\n", packet, e);
  });
  delay(1000);
  WiFi.onEvent([](WiFiEvent_t event, WiFiEventInfo_t info) {
    Serial.print("WiFi connected. IP: ");
    Serial.println(IPAddress(info.got_ip.ip_info.ip.addr));
    WiFiConnected = true;
  }, WiFiEvent_t::SYSTEM_EVENT_STA_GOT_IP);
  WiFi.onEvent([](WiFiEvent_t event, WiFiEventInfo_t info) {
    Serial.print("WiFi lost connection. Reason: ");
    Serial.println(info.disconnected.reason);
    WiFi.disconnect();
    WiFiConnected = false;
  }, WiFiEvent_t::SYSTEM_EVENT_STA_DISCONNECTED);
  WiFi.begin(ssid, pass);
  Serial.println();
  Serial.println("Connecting to WiFi... ");
}
void loop() {
  static uint32_t lastMillis = 0;
  if ((millis() - lastMillis > 1000 && WiFiConnected)) {
    lastMillis = millis();
    Serial.print("reading registers\n");
    for (uint8_t i = 0; i < numberSmaRegisters; ++i) {
      uint16_t packetId = sunnyboy.readHoldingRegisters(smaRegisters[i].address,
smaRegisters[i].length);
      if (packetId > 0) {
        smaRegisters[i].packetId = packetId;
      } else {
        Serial.print("reading error\n");
      }
    }
    Serial.println("Data_1 = "+ String(ResultData[0]));
    Serial.println("Data_2 = " + String(ResultData[1]));
  }
}
```

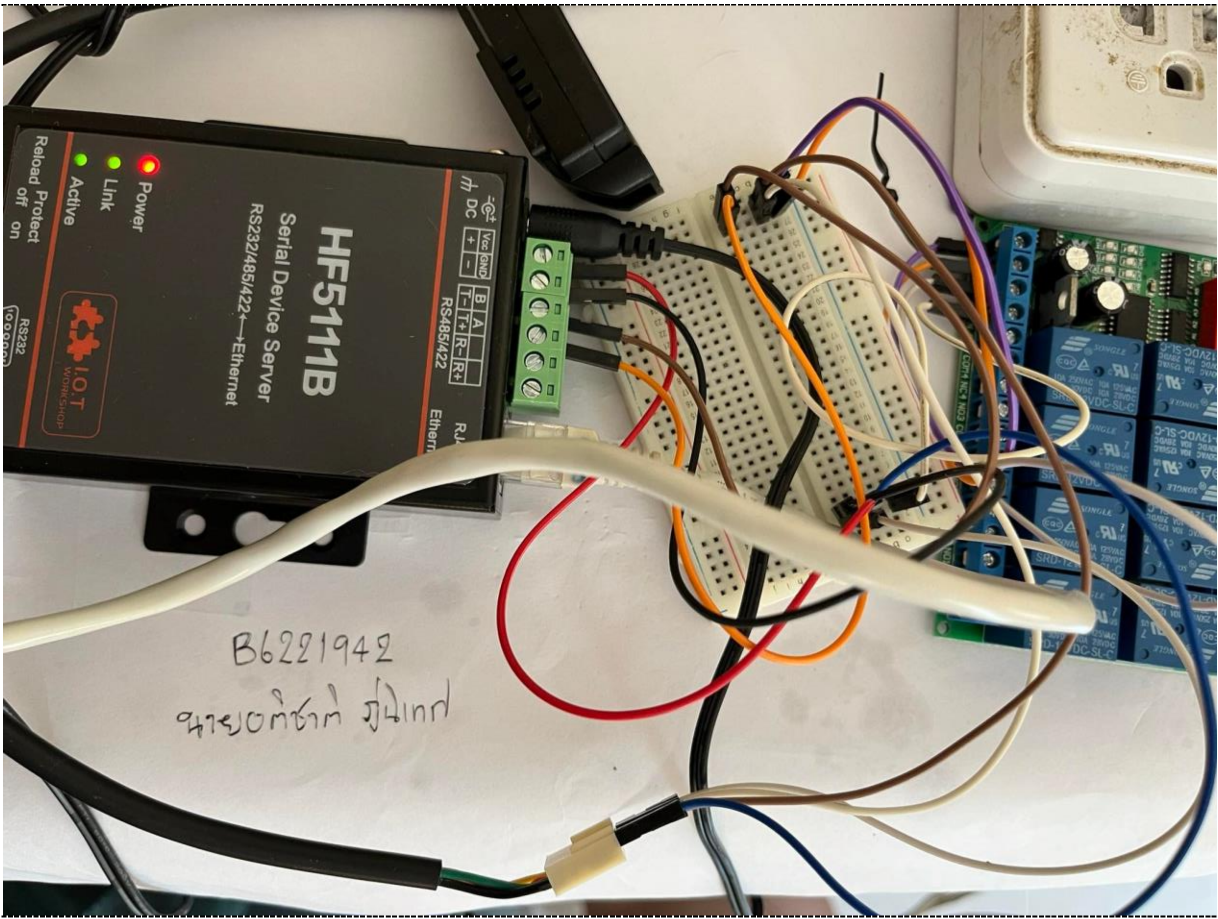
< ผลการทดสอบ >



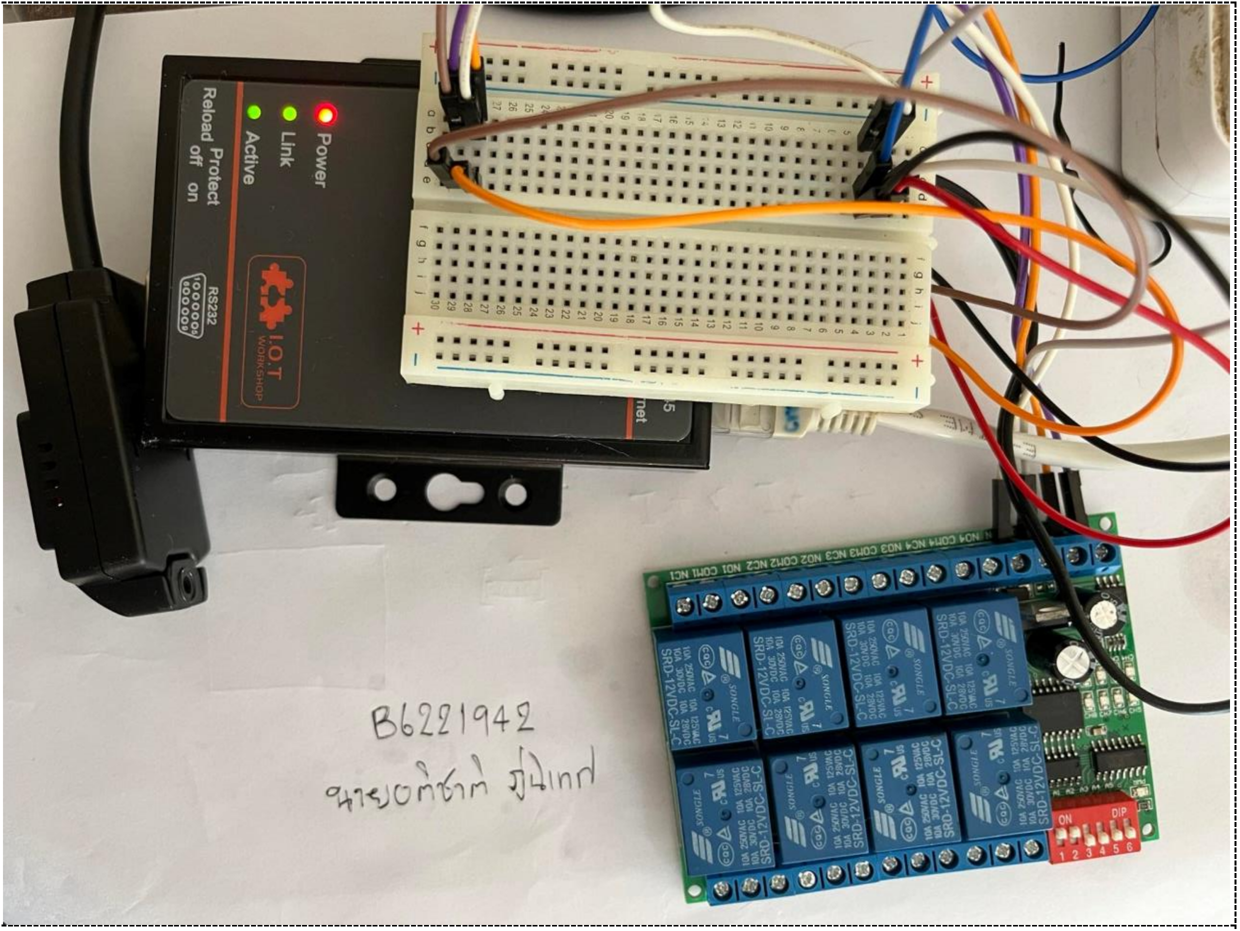
```
WiFi connected. IP: 192.168.100.82
reading registers
Data_1 = 0
Data_2 = 0
Tempp: 348
Humid: 627
reading registers
Data_1 = 348
Data_2 = 627
Tempp: 348
Humid: 628
reading registers
Data_1 = 348
Data_2 = 628
Tempp: 348
Humid: 628
```

## Quiz_303 – Modbus RTU/ASCII/TCP with IoTs

< รูปอุปกรณ์ที่ใช้ทดสอบ ขณะทำการทดสอบ >

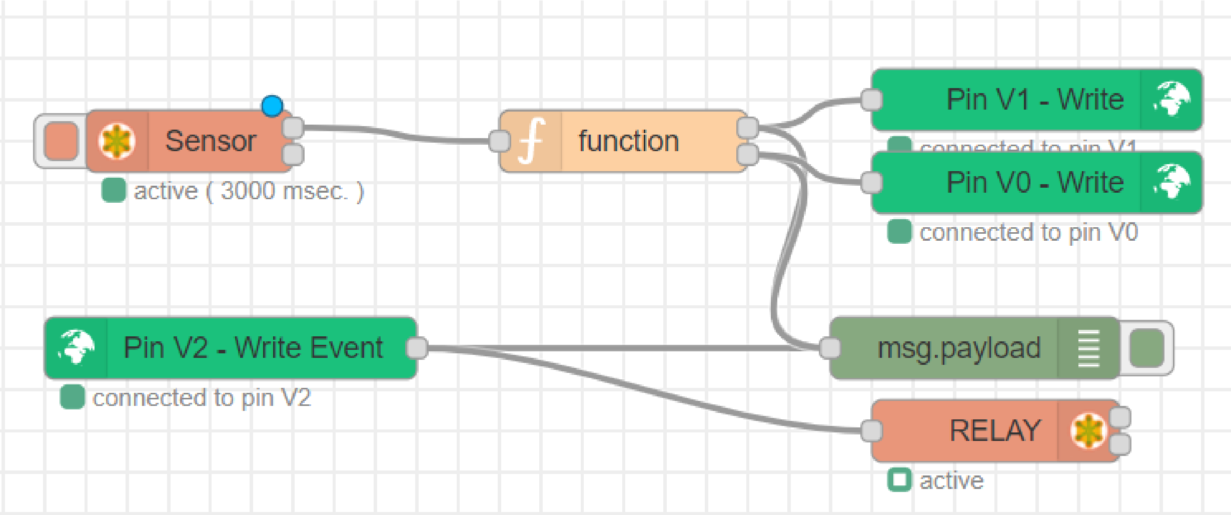


< รูปอุปกรณ์ที่ใช้ทดสอบ ขณะทำการทดสอบ >



รายละเอียดการทดสอบ

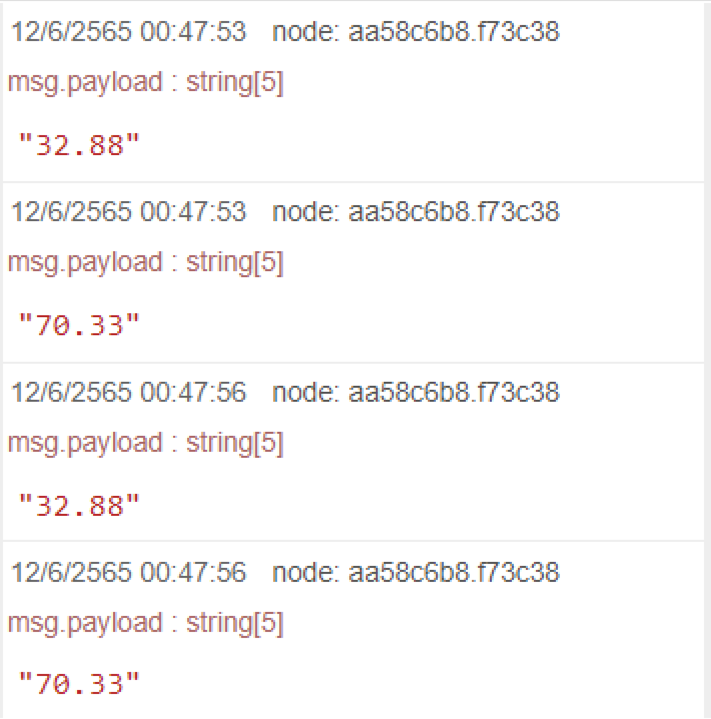
< โปรแกรมทดสอบ >



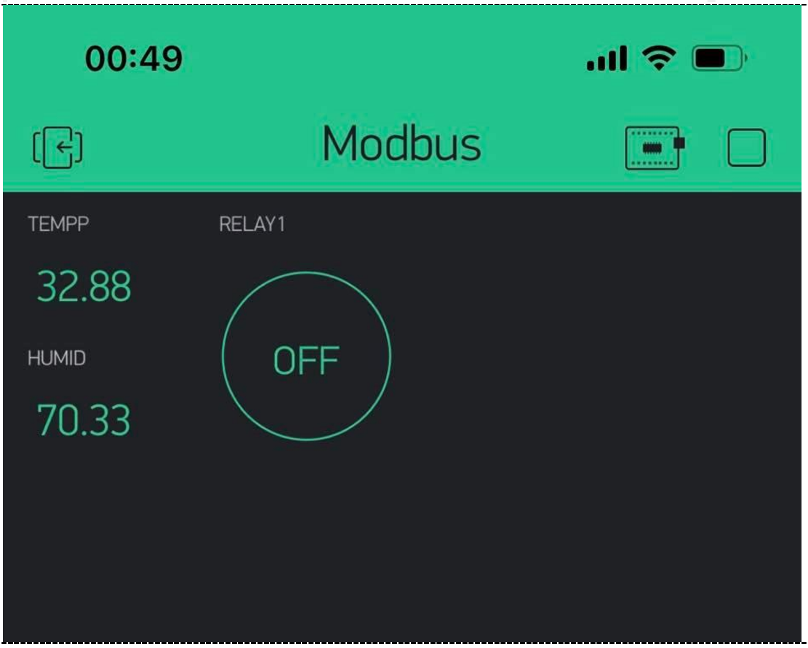
[{"id":"4605fa2b.2d5cc4","type":"tab","label":"Flow 1","disabled":false,"info":""},{"id":"684d7b9c.7e98e4","type":"modbus-read","z":"4605fa2b.2d5cc4","name":"Sensor","topic":"","showStatusActivities":false,"logIOActivities":false,"showErrors":false,"unitid":"1","dataType":"HoldingRegister","adr":"0","quantity":"2","rate":"3000","rateUnit":"ms","delayOnStart":false,"startDelayTime":"","server":"ac4a4b6f.cc29d8","useIO

```
File":false,"ioFile":"","useIOForPayload":false,"emptyMsgOnFail":false,"x":210,"y":200,"wires":[["35
98daf.b84ce26"],[]]},{"id":"3598daf.b84ce26","type":"function","z":"4605fa2b.2d5cc4","name":"","f
unc":"var msg1 = {payload: String(msg.payload[0]/10)+String(msg.payload[0]%10)};\nvar msg2 =
{payload: String(msg.payload[1]/10)+String(msg.payload[1]%10)}\n\nreturn
[msg1,msg2];","outputs":2,"noerr":0,"initialize":"","finalize":"","libs":[],"x":420,"y":200,"wires":[["
fccec6f9.9d9548","aa58c6b8.f73c38"],["aa58c6b8.f73c38","28992304.ca038c"]]},{"id":"fccec6f9.9d9
548","type":"blynk-ws-out-
write","z":"4605fa2b.2d5cc4","name":"","pin":"1","pinmode":0,"client":"3ee168b6.099918","x":620,
"y":180,"wires":[]},{"id":"aa58c6b8.f73c38","type":"debug","z":"4605fa2b.2d5cc4","name":"","activ
e":true,"tosidebar":true,"console":false,"tostatus":false,"complete":"false","statusVal":"","statusType
":"auto","x":590,"y":300,"wires":[]},{"id":"af1b298c.441ba8","type":"blynk-ws-in-
write","z":"4605fa2b.2d5cc4","name":"","pin":"2","pin_all":0,"client":"3ee168b6.099918","x":230,"y
":300,"wires":[["aa58c6b8.f73c38","5af5dd71.7d0164"]]},{"id":"5af5dd71.7d0164","type":"modbus-
write","z":"4605fa2b.2d5cc4","name":"RELAY","showStatusActivities":false,"showErrors":false,"uniti
d":"3","dataType":"HoldingRegister","adr":"1","quantity":"1","server":"ac4a4b6f.cc29d8","emptyMs
gOnFail":false,"keepMsgProperties":false,"x":600,"y":340,"wires":[[],[]]},{"id":"28992304.ca038c","t
ype":"blynk-ws-out-
write","z":"4605fa2b.2d5cc4","name":"","pin":"0","pinmode":0,"client":"3ee168b6.099918","x":620,
"y":220,"wires":[]},{"id":"ac4a4b6f.cc29d8","type":"modbus-
client","name":"abcd","clienttype":"tcp","bufferCommands":true,"stateLogEnabled":false,"queueLogE
nabled":false,"tcpHost":"192.168.100.183","tcpPort":"502","tcpType":"DEFAULT","serialPort":"/dev/t
tyUSB","serialType":"RTU-
BUFFERD","serialBaudrate":"9600","serialDatabits":"8","serialStopbits":"1","serialParity":"none","se
rialConnectionDelay":"100","serialAsciiResponseStartDelimiter":"0x3A","unit_id":"1","commandDelay
":"1","clientTimeout":"1000","reconnectOnTimeout":true,"reconnectTimeout":"2000","parallelUnitIds
Allowed":true},{"id":"3ee168b6.099918","type":"blynk-ws-client","name":"","path":"ws://blynk-
cloud.com/websockets","key":"10Dtg7C1TMQgqOjQTNaX0DrSe8uH9YLf","dbg_all":false,"dbg_read":fa
lse,"dbg_write":false,"dbg_notify":false,"dbg_mail":false,"dbg_prop":false,"dbg_sync":false,"dbg_brid
ge":false,"dbg_low":false,"dbg_pins":"","multi_cmd":false,"proxy_type":"no","proxy_url":"","enabled
":true}]
```

< ผลการทดสอบ >.



```
12/6/2565 00:47:53   node: aa58c6b8.f73c38
msg.payload : string[5]
 "32.88"
12/6/2565 00:47:53   node: aa58c6b8.f73c38
msg.payload : string[5]
 "70.33"
12/6/2565 00:47:56   node: aa58c6b8.f73c38
msg.payload : string[5]
 "32.88"
12/6/2565 00:47:56   node: aa58c6b8.f73c38
msg.payload : string[5]
 "70.33"
```



TEMPP
32.88
HUMID
70.33
RELAY1
OFF