Think Hub Programming

Test Date:29.06.2022

Tester:

Durjoy Talukder, Jr. Design Engineer

Md. Kamal Hossain, Technician

Program:

#include <Arduino.h>

#include <ModbusMaster.h> //need to include Modbus Master library

#include <HardwareSerial.h>

HardwareSerial Serial3(PC6); //FTDA Serial

HardwareSerial Serial1(PA10, PA9); // RS485 Serial

HardwareSerial Serial2(PA3, PA2); //Modem Serial

//HardwareSerial Serial2(PA3, PA2);

#define number 3 //Device Number

#define MAX485\_Enable PC7

// instantiate ModbusMaster object

ModbusMaster node[number];

word CRC16 (const byte \*nData, word wLength);

uint8\_t slave [] = {0x001,0x002,0x003};

struct Structure

{

uint8\_t PhaseA\_Voltage;

uint8\_t PhaseB\_Voltage;

uint8\_t PhaseC\_Voltage;

uint8\_t PhaseA\_Current;

uint8\_t PhaseB\_Current;

uint8\_t PhaseC\_Current;

uint8\_t PhaseA\_ActivePower;

uint8\_t PhaseB\_ActivePower;

uint8\_t PhaseC\_ActivePower;

uint8\_t Total\_ActivePower;

};

void preTransmission()

{

digitalWrite(MAX485\_Enable, 1); //Transmit data

}

void postTransmission()

{

digitalWrite(MAX485\_Enable, 0); //Receive Data

}

void setup ()

{

pinMode(MAX485\_Enable, OUTPUT);

// pinMode(MAX485\_DE, OUTPUT);

// Init in receive mode

// digitalWrite(MAX485\_RE\_NEG, 0);

// digitalWrite(MAX485\_DE, 0);

// Modbus communication runs at 115200 baud

Serial1.begin(9600, SERIAL\_8N2);

Serial2.begin(115200, SERIAL\_8N2);

Serial3.begin(115200, SERIAL\_8N2);

pinMode (PC11, OUTPUT);

pinMode (PA1, OUTPUT);

pinMode (PC0, INPUT);

modem\_start ();

// Modbus slave ID 1

for (int i=0; i<number; i++)

{

node[i].begin(slave[i], Serial1);

// Callbacks allow us to configure the RS485 transceiver correctly

node[i].preTransmission(preTransmission);

node[i].postTransmission(postTransmission);

}

}

bool state = true;

void loop ()

{

uint8\_t result [256];

uint8\_t data [6];

Structure sys;

// uint8\_t Voltage1, Voltage2, Voltage2, Current1, Current2, Current3;

// // Toggle the coil at address 0x0002 (Manual Load Control)

// result = node.writeSingleCoil(0x0002, state);

// state = !state;

for(int i=0;i<number;i++){

result[i] = node[i].readHoldingRegisters(0x0061, 20);

if (result[i] == node[i].ku8MBSuccess)

{

// Read 16 registers starting at 0x3100)

sys.PhaseA\_Voltage=node[i].getResponseBuffer(0x00)/10.0f;

sys.PhaseB\_Voltage=node[i].getResponseBuffer(0x01)/10.0f;

sys. PhaseC\_Voltage=node[i].getResponseBuffer(0x02)/10.0f;

sys. PhaseA\_Current=node[i].getResponseBuffer(0x03)/100.0f;

sys.PhaseB\_Current=node[i].getResponseBuffer(0x04)/100.0f;

sys.PhaseC\_Current=node[i].getResponseBuffer(0x05)/100.0f;

sys.PhaseA\_ActivePower=node[i].getResponseBuffer(0x06)/10.0f;

sys.PhaseB\_ActivePower=node[i].getResponseBuffer(0x07)/10.0f;

sys.PhaseC\_ActivePower=node[i].getResponseBuffer(0x08)/10.0f;

sys.Total\_ActivePower=node[i].getResponseBuffer(0x09)/10.0f;

// delay(100);

// Serial3.print("Phase A Voltage: ");

// Serial3.println(sys.PhaseA\_Voltage);

// Serial3.print("Phase B Voltage: ");

// Serial3.println(sys.PhaseB\_Voltage);

// Serial3.print("Phase C Voltage: ");

// Serial3.println(sys.PhaseC\_Voltage);

// Serial3.print("Phase A Current: ");

// Serial3.println(sys.PhaseA\_Current);

// Serial3.print("Phase B Current: ");

// Serial3.println(sys.PhaseB\_Current);

// Serial3.print("Phase C Current: ");

// Serial3.println(sys.PhaseC\_Current);

// Serial3.print("Phase A Power: ");

// Serial3.println(sys.PhaseA\_ActivePower);

// Serial3.print("Phase B Power: ");

// Serial3.println(sys.PhaseB\_ActivePower);

// Serial3.print("Phase C Power: ");

// Serial3.println(sys.PhaseC\_ActivePower);

// Serial3.print("TOTAL Power: ");

// Serial3.println(sys.Total\_ActivePower);

delay(100);

String string ;

string +="{";

string +="\"Id\":";

string += String(i);

string +=",";

string +="\"VA\":";

string += String(node[i].getResponseBuffer(0x00)/10.0f);

string +=",";

string +="\"VB\":";

string += String(node[i].getResponseBuffer(0x01)/10.0f);

string +=",";

string +="\"VC\":";

string += String(node[i].getResponseBuffer(0x02)/10.0f);

string +=",";

string +="\"CA\":";

string += String(node[i].getResponseBuffer(0x03)/100.0f);

string +=",";

string +="\"CB\":";

string += String(node[i].getResponseBuffer(0x04)/100.0f);

string +=",";

string +="\"CC\":";

string += String(node[i].getResponseBuffer(0x05)/100.0f);

string +=",";

string +="\"PA\":";

string += String(node[i].getResponseBuffer(0x06)/10.0f);

string +=",";

string +="\"PB\":";

string += String(node[i].getResponseBuffer(0x07)/10.0f);

string +=",";

string +="\"PC\":";

string += String(node[i].getResponseBuffer(0x08)/10.0f);

string +=",";

string +="\"Total\_p\":";

string += String(node[i].getResponseBuffer(0x09)/10.0f);

string += "}";

int data\_length = string.length()+2 ;

Serial3.println(data\_length);

//Serial2.println("AT+CMQTTPAYLOAD=0,51\r\n");

String payload ;

payload+="AT+CMQTTPAYLOAD=0,";

payload+=String(data\_length);

payload+="\r\n";

Serial3.println(payload);

char\* ptopic= "v1/devices/me/attributes”; //Publish Topic

char\* stopic = "v1/devices/me/attributes/response/+"; //Subscribe Topic

Serial3.println("Modem Started...");

delay(10);

Serial2.println("AT\r\n");

delay(10);

//AT Commands for Status Control

Serial2.println("AT+CSQ\r\n"); //Query signal quality

delay(10);

Serial2.println("AT+CPIN?\r\n"); //Enter PIN

delay(10);

// AT Commands for Network

Serial2.println("AT+COPS?\r\n"); //Operator selection

delay(10);

Serial2.println("AT+CREG=1\r\n"); //Network registration, 1 -registered, home network

delay(10);

Serial2.println("AT+CREG?\r\n");

delay(10);

Serial2.println("AT+CGMI\r\n"); //Request manufacturer identification

delay(10);

Serial2.println("AT+CNMP=2\r\n"); //Preferred mode selection, 2 – Automatic

delay(1000);

Serial2.println("AT+CMQTTSTART\r\n"); //Start MQTT server

delay(1000);

Serial2.println("AT+CMQTTACCQ=0,\"test102\"\r\n"); //Acquire a client

delay(1000);

Serial2.println("AT+CMQTTCONNECT=0,\"tcp://thinkiot.com.bd\",20,1,\"test11\",\"test11\"\r\n "); //Connect to the server

delay(5000);

Serial3.println(mdm\_rpl());

Serial2.println("AT+CMQTTTOPIC=0,24\r\n"); //publish topic

delay(1000);

Serial2.println(ptopic); //SET topic

delay(1000);

// Serial2.println("AT+CMQTTPAYLOAD=0,51\r\n"); //Set publishing message

Serial2.println(payload);

delay(1000);

Serial2.println(string); //Set publishing message

delay(1000);

Serial2.println("AT+CMQTTPUB=0,1,60\r\n"); //Send a PUBLISH message to server

delay(1000);

Serial3.println(mdm\_rpl());

// Serial2.println("AT+CMQTTSUBTOPIC=0,35,1\r\n");

// Serial2.println(stopic); //SET topic

// delay(1000);

// Serial2.println("AT+CMQTTSUB=0\r\n ");

// delay(1000);

Serial2.println("AT+CMQTTDISC=0,120\r\n");

delay(1000);

Serial2.println("AT+CMQTTREL=0\r\n");

delay(1000);

Serial2.println ("AT+CMQTTSTOP\r\n");

delay(1000);

}

}

}

//FTDA Serial Print

String mdm\_rpl()

{

if(Serial2.available())

{

String reply = Serial2.readString();

reply.trim();

return reply;

}

}

//Modem Start When Enable Pin PA1 High, PWRKEY PC11 High few Second and then Low

void modem\_start(void)

{

digitalWrite(PA1,HIGH);

delay(500);

digitalWrite(PC11,HIGH);

delay(500);

digitalWrite(PC11,LOW);

}