|  |
| --- |
| **การควบคุมเครื่องจักรอัจฉริยะโดยใช้การสื่อสารระหว่างเครื่องจักรกับเครื่องจักร**  **M2M - Intelligence Machine Control** |
| **ขื่อ-สกุล : นายธนพล กาศักดิ์** |

**4/4: -- คำถามท้ายบทเพื่อทดสอบความเข้าใจ**

**Quiz\_201 – Read Modbus RTU**

|  |
| --- |
| < รูปอุปกรณ์ที่ใช้ทดสอบ ขณะทำการทดสอบ > |
| < รูปอุปกรณ์ที่ใช้ทดสอบ ขณะทำการทดสอบ >A picture containing cable, electrical wiring, electronics, electronic engineering  Description automatically generated |
| < โปรแกรมทดสอบ >  #define RS485Transmit HIGH #define RS485Receive LOW #define RS485Control 4 //RS485 Direction control #define Pin\_LEDMonitor 2int Wr\_Index, StepConut = 0; byte Request[] = {0x01, 0x03, 0x00, 0x00, 0x00, 0x02, 0xC4, 0x0B}; byte Echo[20];void setup() {  pinMode(Pin\_LEDMonitor, OUTPUT);  pinMode(RS485Control, OUTPUT);  Serial.begin(9600);  Serial2.begin(9600);  digitalWrite(RS485Control, RS485Receive);  Serial.println("\nStart Test MODBUS RTU"); } void loop() {  Serial.print("\nTest(");  Serial.print(++StepConut);  Serial.print(") >>");  digitalWrite(Pin\_LEDMonitor, HIGH);  digitalWrite(RS485Control, RS485Transmit);  delay(10);  for (int i = 0; i < sizeof(Request); i++) {  Serial2.write(Request[i]);}  delay(10);  digitalWrite(RS485Control, RS485Receive);  digitalWrite(Pin\_LEDMonitor, LOW);  Wr\_Index = 0;  for (long int i = 0; i < 300000; i++) {  if (Serial2.available())  Echo[Wr\_Index++] = Serial2.read();  if (Wr\_Index > 12) i = 999999;  delayMicroseconds(10);  }    for (int i = 0; i < Wr\_Index - 1; i++) {  Serial.print(" ");  if (Echo[i] < 0x10) Serial.print("0");  Serial.print(Echo[i], HEX);  }    float Tempp = (Echo[3] \* 256 + Echo[4]) / 10.0;  Serial.print(" > Tempp('C)="); Serial.print(Tempp, 2);  float Humid = (Echo[5] \* 256 + Echo[6]) / 10.0;  Serial.print(" > Humid(%)="); Serial.print(Humid, 2);    delay(5000); } |
| < ผลการทดสอบ > |

**Quiz\_202 – Write Modbus RTU**

|  |
| --- |
| < รูปอุปกรณ์ที่ใช้ทดสอบ ขณะทำการทดสอบ > |
| < รูปอุปกรณ์ที่ใช้ทดสอบ ขณะทำการทดสอบ > |
| < โปรแกรมทดสอบ >  #define RS485Transmit HIGH #define RS485Receive LOW #define RS485Control 4 //RS485 Direction control #define Pin\_LEDMonitor 2byte Board\_ID = 0x03; byte Mdbs\_Cmd = 0x06; byte H\_RelayID = 0x00; byte L\_RelayID = 0x03; byte Relay\_On = 0x01; byte Relay\_Off = 0x02; byte OnOff\_Dly = 0x00; byte HByte\_CRC = 00; byte LByte\_CRC = 00; int StepConut = 0; byte Echo[20];void setup() {  pinMode(Pin\_LEDMonitor, OUTPUT);  pinMode(RS485Control, OUTPUT);  Serial.begin(9600);  Serial2.begin(9600);  digitalWrite(RS485Control, RS485Receive);  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 RTU\_RelayCtrl(int rly\_ID, byte rly\_Cmd) {  uint16\_t Calc\_CRC = 0xffff; // the initial value  H\_RelayID = highByte(rly\_ID);  L\_RelayID = lowByte(rly\_ID);  digitalWrite(Pin\_LEDMonitor, HIGH);  digitalWrite(RS485Control, RS485Transmit); delay(10);  Calc\_CRC = SendByte\_CRCUpdate(Calc\_CRC, Board\_ID);  Calc\_CRC = SendByte\_CRCUpdate(Calc\_CRC, Mdbs\_Cmd);  Calc\_CRC = SendByte\_CRCUpdate(Calc\_CRC, H\_RelayID);  Calc\_CRC = SendByte\_CRCUpdate(Calc\_CRC, L\_RelayID);  Calc\_CRC = SendByte\_CRCUpdate(Calc\_CRC, rly\_Cmd);  Calc\_CRC = SendByte\_CRCUpdate(Calc\_CRC, OnOff\_Dly);  HByte\_CRC = highByte(Calc\_CRC);  LByte\_CRC = lowByte(Calc\_CRC);  Calc\_CRC = SendByte\_CRCUpdate(Calc\_CRC, LByte\_CRC);  Calc\_CRC = SendByte\_CRCUpdate(Calc\_CRC, HByte\_CRC);  delay(10);  digitalWrite(RS485Control, RS485Receive);  digitalWrite(Pin\_LEDMonitor, LOW);  Serial.println(); }void loop() {  for (int relay = 1; relay <= 8; relay++) {  RTU\_RelayCtrl(relay, Relay\_On);  delay(3000);  }  for (int relay = 1; relay <= 8; relay++) {  RTU\_RelayCtrl(relay, Relay\_Off);  delay(3000);  } } |
| < ผลการทดสอบ > |

**Quiz\_203 – Read/Write Modbus RTU**

|  |
| --- |
| < รูปอุปกรณ์ที่ใช้ทดสอบ ขณะทำการทดสอบ > |
| < รูปอุปกรณ์ที่ใช้ทดสอบ ขณะทำการทดสอบ > |
| < โปรแกรมทดสอบ >  #include <ModbusMaster.h> #define RS485Transmit HIGH #define RS485Receive LOW #define RS485Control 4 //RS485 Direction control #define Pin\_LEDMonitor 2 #define Slave\_Sensor\_ID 1 #define Slave\_Relay8\_ID 3 #define Slave\_Ry4In4\_ID 5 int state = 0; float CTempp, Hudmid; bool DgInput0, DgInput1, DgInput2, DgInput3; ModbusMaster node\_Sensor; ModbusMaster node\_Relay8; ModbusMaster node\_Ry4In4; void preTransmission() {  digitalWrite(RS485Control, RS485Transmit); } void postTransmission() {  digitalWrite(RS485Control, RS485Receive); }void setup() {  pinMode(RS485Control, OUTPUT);  pinMode(Pin\_LEDMonitor, OUTPUT);  Serial.begin(115200);  Serial2.begin(9600);  postTransmission();  node\_Sensor.begin(Slave\_Sensor\_ID, Serial2); // Modbus slave ID=1  node\_Sensor.preTransmission(preTransmission);  node\_Sensor.postTransmission(postTransmission);  node\_Relay8.begin(Slave\_Relay8\_ID, Serial2); // Modbus slave ID=3  node\_Relay8.preTransmission(preTransmission);  node\_Relay8.postTransmission(postTransmission);  node\_Ry4In4.begin(Slave\_Ry4In4\_ID, Serial2); // Modbus slave ID=5  node\_Ry4In4.preTransmission(preTransmission);  node\_Ry4In4.postTransmission(postTransmission); }void ReadTemperature(void) {  uint8\_t result;  // Toggle the coil at address (Manual Load Control)  result = node\_Sensor.writeSingleCoil(Slave\_Sensor\_ID, state);  state = !state;  // Read 2 registers starting at 0x0000)  result = node\_Sensor.readInputRegisters(0x0000, 2); // From=0, nByte=2  if (result == node\_Sensor.ku8MBSuccess) {  CTempp = node\_Sensor.getResponseBuffer(0x00) / 10.0f;  Hudmid = node\_Sensor.getResponseBuffer(0x01) / 10.0f;  } }void ReadDigitalInput(void) {  uint8\_t result;  // Toggle the coil at address (Manual Load Control)  result = node\_Ry4In4.writeSingleCoil(Slave\_Sensor\_ID, state);  state = !state;  // Read 4 registers starting at 0x0000)  result = node\_Ry4In4.readDiscreteInputs(0, 4); // Start=0, nByte=4  if (result == node\_Ry4In4.ku8MBSuccess) {  int DgTemp = node\_Ry4In4.getResponseBuffer(0x00);  DgInput3 = (DgTemp >> 3) & 1;  DgInput2 = (DgTemp >> 2) & 1;  DgInput1 = (DgTemp >> 1) & 1;  DgInput0 = (DgTemp >> 0) & 1;  } }void RelayControl(int inputCase) {  int rnMode = inputCase / 10;  int nRelay = inputCase % 10;  if (rnMode == 81) node\_Relay8.writeSingleRegister(nRelay, 0x0100); // On RelayX  if (rnMode == 80) node\_Relay8.writeSingleRegister(nRelay, 0x0200); // Off RelayX   if (rnMode == 41) node\_Ry4In4.writeSingleCoil(nRelay, 0x01); // On RelayX  if (rnMode == 40) node\_Ry4In4.writeSingleCoil(nRelay, 0x00); // Off RelayX }void loop() {  ReadTemperature();  ReadDigitalInput();  Serial.print("\n Tempp('C): "); Serial.print(CTempp, 2);  Serial.print(", Humid(%): "); Serial.print(Hudmid, 2);  Serial.print(", Sensor[0:3]: "); Serial.print(DgInput3);  Serial.print("-"); Serial.print(DgInput2);  Serial.print("-"); Serial.print(DgInput1);  Serial.print("-"); Serial.print(DgInput0);  if (Serial.available() > 0) {  int DataInput = Serial.parseInt();  Serial.print("\n >> XYZ > X={8,4}Board Name, Y={1,0}On,Off, Z={0-8}RlyID >> ");  Serial.println(DataInput);  RelayControl(DataInput);  }  delay(2000); } |
| < ผลการทดสอบ > |

**Quiz\_204 – PLC Test**

|  |
| --- |
| < รูปอุปกรณ์ที่ใช้ทดสอบ ขณะทำการทดสอบ > |
| < รูปอุปกรณ์ที่ใช้ทดสอบ ขณะทำการทดสอบ > |
| < โปรแกรมทดสอบ > |
| < ผลการทดสอบ > |