Modbus RTU Relay

Electrical and Relay Safety Instructions

- This product must be operated by professional electricians or qualified personnel. During use, ensure electrical safety, leakage protection, and proper insulation.
- Before installing, maintaining, or replacing the relay device, always turn off the power and unplug the device.
- Do not attempt to disassemble the relay device to avoid damage or the risk of electric shock.
- Properly install and place the relay device. Do not use it in humid, overheated, flammable, or explosive environments to prevent accidents caused by improper installation or use.

1. Load Matching

- Ensure the relay's rated voltage and current match the load. Do not exceed the rated capacity.
- For inductive loads (motors, coils, lamps, etc.), the starting current may be much higher than the rated current. Choose a relay with sufficient current margin.

2. Short Circuit and Overcurrent Protection

- Install a **fuse** or **circuit breaker** in the relay circuit to prevent damage due to short circuits or accidental overcurrent.
- Ensure the load circuit has no short circuits during wiring, and select protection components with appropriate current ratings if necessary.

3. Arc and Switching Protection

- Relay switching generates arcs, which can cause contact wear or welding.
- For inductive loads, it is recommended to use **RC snubber circuits** or **varistors** for arc suppression.

4. Installation Environment

- Do not use the relay in humid, high-temperature, flammable, explosive, or dusty environments.
- Install the relay securely to avoid vibrations or shocks that may cause misoperation or damage.

5. Power-Off Operation

- Always cut off power before maintenance, wiring, or replacing the relay to ensure personnel and device safety.
- Latching relays are only powered when changing state. Avoid strong vibrations or strong magnetic fields while the relay is unpowered.

6. Status Confirmation

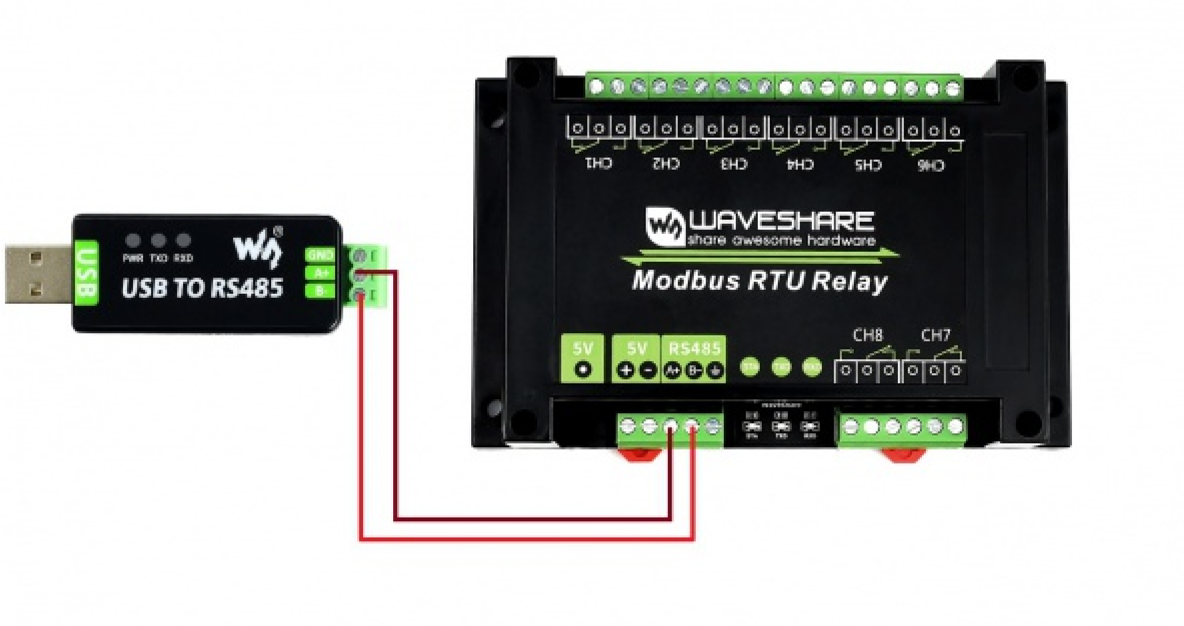
- After powering on, confirm or reset the relay status as needed to prevent abnormal operation caused by transportation, installation, or external disturbances.
- Avoid power interruption during relay operation to prevent uncertain status or contact damage.

7. Regular Inspection

- Periodically inspect relay contacts, terminals, and insulation to ensure proper operation.
- If abnormal heating, odor, or burn marks are detected, immediately cut off power and replace the relay.

Hardware Connection

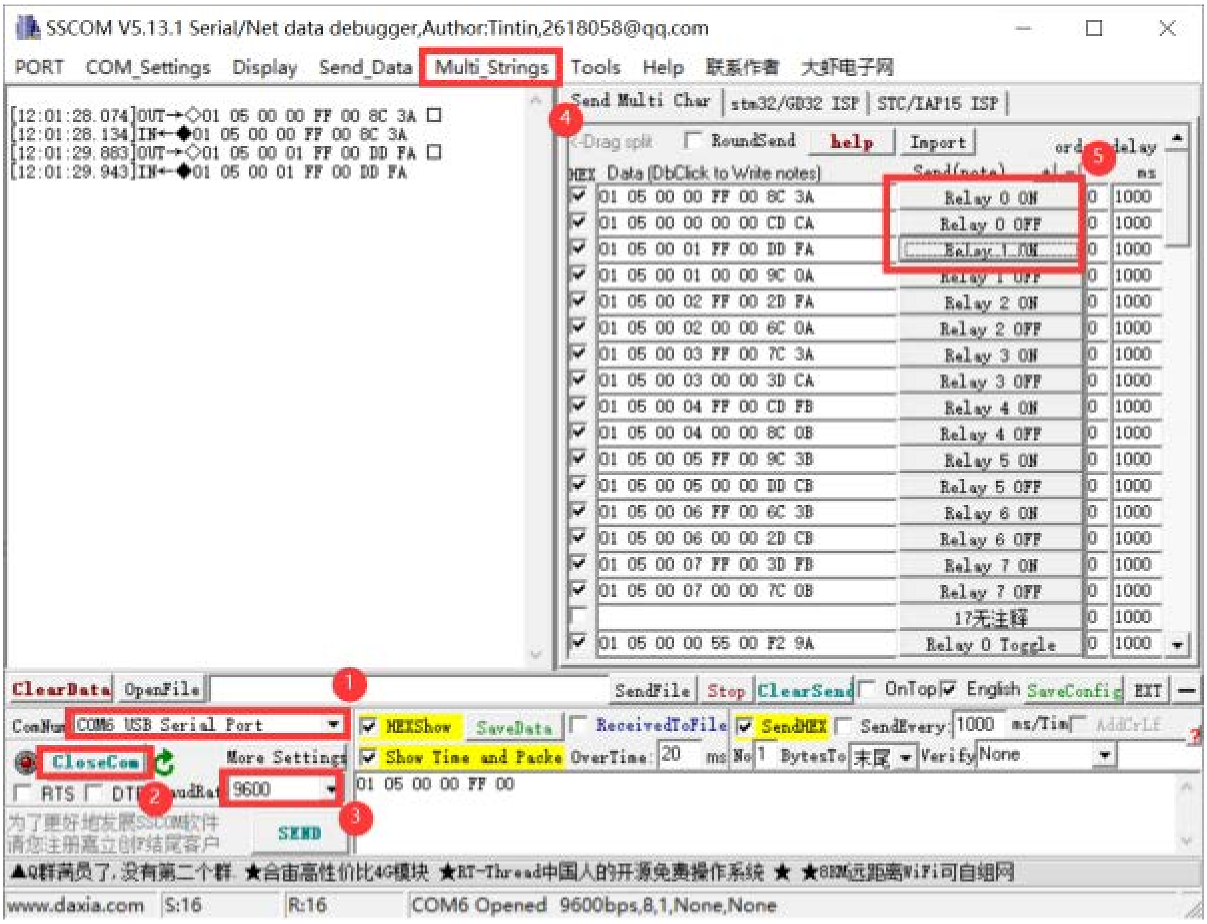
- Connect the USB TO 485 to the target boards via cables, A-A and B-B connected as shown below:



Example Demonstration

SSCOM Serial Port Debugging Assistant

- Download SSCOM Serial port debugging assistant and open it on the computer, open the corresponding port number, and set the baud rate as 9600. Click Multi-Strings to open the Send Multi-Char window, and click the function to send the corresponding command.



```
[12:01:28.074]OUT→◇01 05 00 00 FF 00 8C 3A □
[12:01:28.134]IN←◆01 05 00 00 FF 00 8C 3A
[12:01:29.883]OUT→◇01 05 00 01 FF 00 DD FA □
[12:01:29.943]IN←◆01 05 00 01 FF 00 DD FA
```

HEX	Data (DbClick to Write notes)	Send(note)	order	delay ms
☑	01 05 00 00 FF 00 8C 3A	Relay 0 ON	0	1000
☑	01 05 00 00 00 00 CD CA	Relay 0 OFF	0	1000
☑	01 05 00 01 FF 00 DD FA	Relay 1 ON	0	1000
☑	01 05 00 01 00 00 9C 0A	Relay 1 OFF	0	1000
☑	01 05 00 02 FF 00 2D FA	Relay 2 ON	0	1000
☑	01 05 00 02 00 00 6C 0A	Relay 2 OFF	0	1000
☑	01 05 00 03 FF 00 7C 3A	Relay 3 ON	0	1000
☑	01 05 00 03 00 00 3D CA	Relay 3 OFF	0	1000
☑	01 05 00 04 FF 00 CD FB	Relay 4 ON	0	1000
☑	01 05 00 04 00 00 8C 0B	Relay 4 OFF	0	1000
☑	01 05 00 05 FF 00 9C 3B	Relay 5 ON	0	1000
☑	01 05 00 05 00 00 DD CB	Relay 5 OFF	0	1000
☑	01 05 00 06 FF 00 6C 3B	Relay 6 ON	0	1000
☑	01 05 00 06 00 00 2D CB	Relay 6 OFF	0	1000
☑	01 05 00 07 FF 00 3D FB	Relay 7 ON	0	1000
☑	01 05 00 07 00 00 7C 0B	Relay 7 OFF	0	1000
□		17无注释	0	1000
☑	01 05 00 00 55 00 F2 9A	Relay 0 Toggle	0	1000

ComNum COM6 USB Serial Port

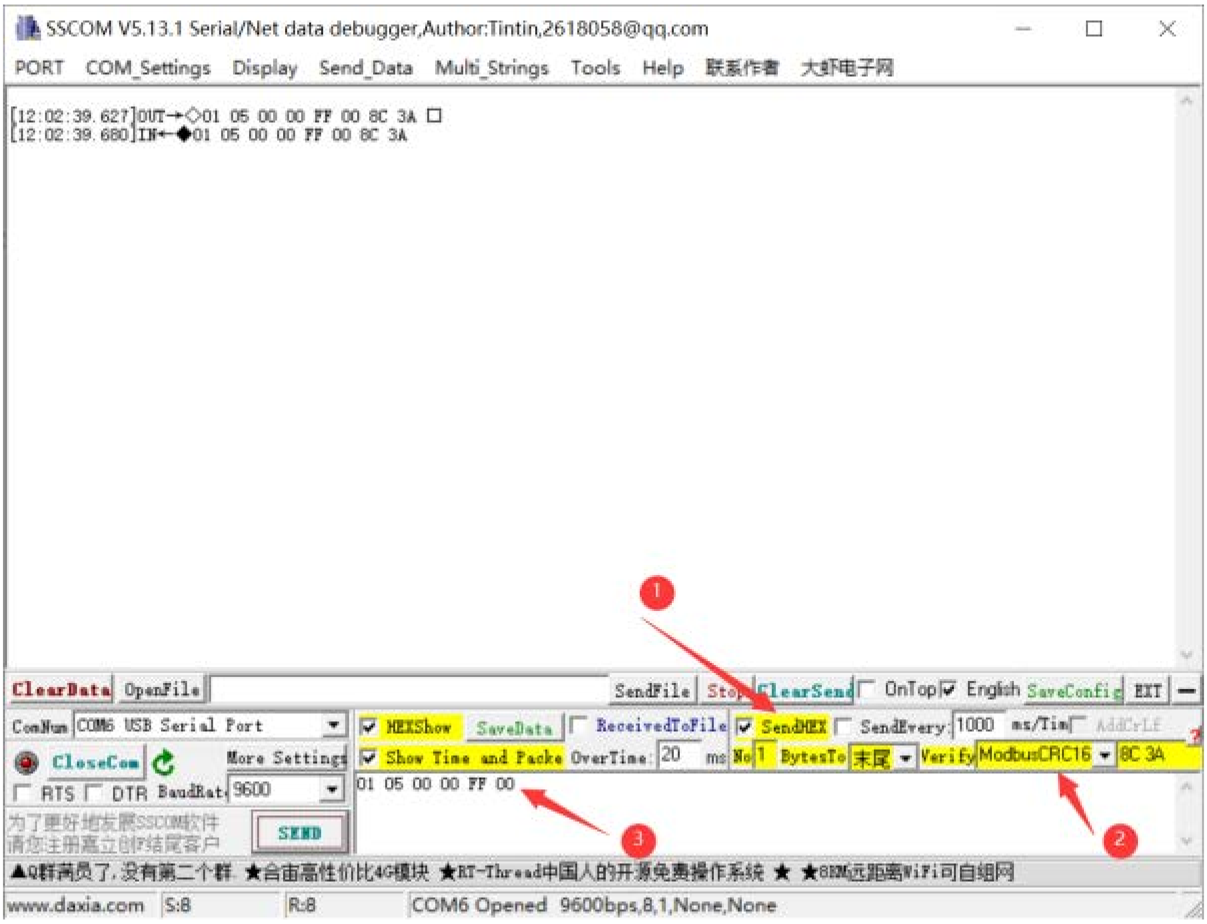
BaudRate 9600

☑ HEXShow ☑ Show Time and Packe OverTime: 20 ms No 1 BytesTo 末尾 ☑ SendHEX □ SendEvery: 1000 ms/Tim Verify None

01 05 00 00 FF 00

www.daxia.com S:16 R:16 COM6 Opened 9600bps,8,1,None,None

- If you need to send other commands, choose SendHEX. For checksum validation, select ModbusCRC16. After entering the first six bytes of the command, clicking SEND will automatically add the CRC check code.



SSCOM V5.13.1 Serial/Net data debugger,Author:Tintin,2618058@qq.com

```
[12:02:39.627]OUT→◇01 05 00 00 FF 00 8C 3A □
[12:02:39.680]IN←◆01 05 00 00 FF 00 8C 3A
```

ComNum: COM6 USB Serial Port

BaudRate: 9600

☑ HEXShow

☑ SendHEX

☑ Show Time and Packe OverTime: 20 ms

Verify: ModbusCRC16 8C 3A

```
01 05 00 00 FF 00
```

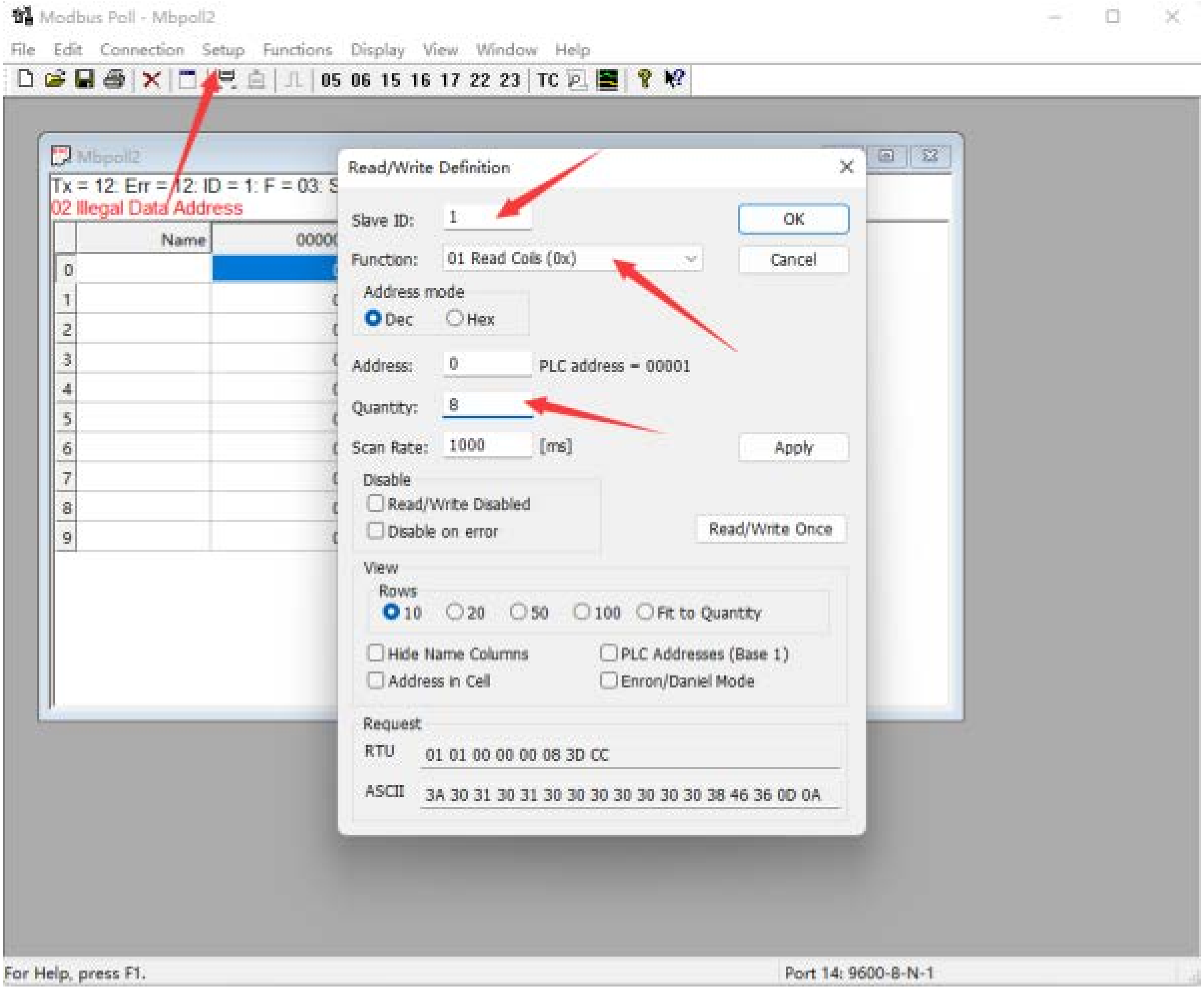
www.daxia.com S:8 R:8 COM6 Opened 9600bps,8,1,None,None

- For detailed control commands, please see the development protocol.

From September 2024, the V3 development protocol will be adopted, which will be compatible with the V2 version, and some new features will be added.

Modbus Poll Software

- It is not convenient to use the SSCOM software for observing the data, you can select Modbus Poll software to read the data. Download and install the Modbus Poll software.
- Open the software, select Setup->Read/Write Definition, select the actual device address for Slave ID, select 01 Read Coils function code for Function, and change Quantity to 8 channels. Click OK to confirm.



Tx = 12: Err = 12: ID = 1: F = 03: S

02 Illegal Data Address

Read/Write Definition

Slave ID: 1

Function: 01 Read Coils (0x)

Address mode ● Dec ○ Hex

Address: 0 PLC address = 00001

Quantity: 8

Scan Rate: 1000 [ms]

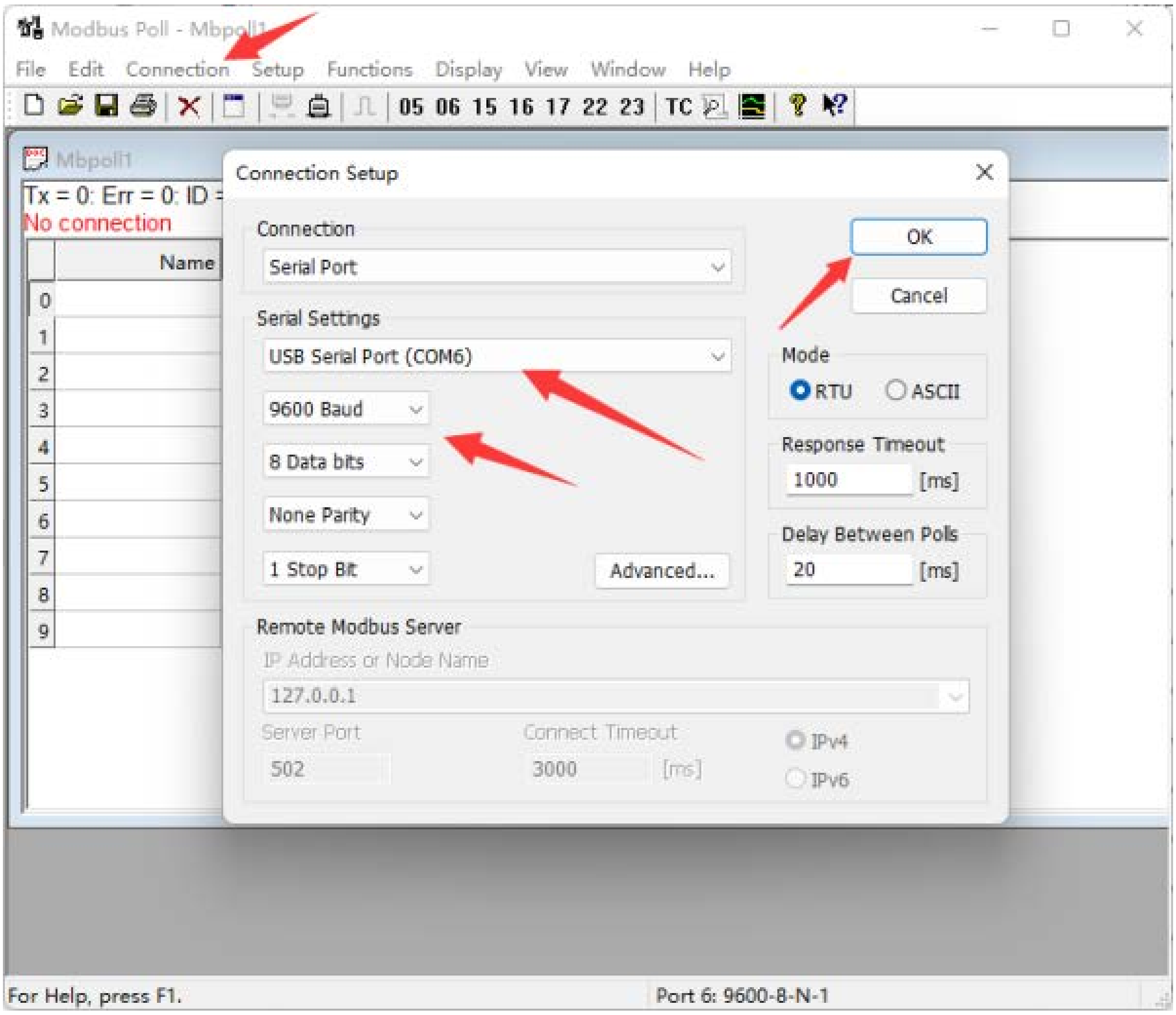
Disable
☐ Read/Write Disabled
☐ Disable on error

View
Rows ● 10 ○ 20 ○ 50 ○ 100 ○ Fit to Quantity
☐ Hide Name Columns ☐ PLC Addresses (Base 1)
☐ Address in Cell ☐ Enron/Daniel Mode

Request
RTU 01 01 00 00 00 08 3D CC
ASCII 3A 30 31 30 31 30 30 30 30 30 30 30 38 46 36 0D 0A

For Help, press F1. Port 14: 9600-8-N-1

- Select Connection->Connect..., choose the corresponding serial port, set the baud rate to 9600, and select 8 Data bits and None Parity. Click OK to connect.



Tx = 0: Err = 0: ID =

No connection

Connection Setup

Connection: Serial Port

Serial Settings: USB Serial Port (COM6)

9600 Baud

8 Data bits

None Parity

1 Stop Bit

Mode: ● RTU ○ ASCII

Response Timeout: 1000 [ms]

Delay Between Polls: 20 [ms]

Remote Modbus Server

IP Address or Node Name: 127.0.0.1

Server Port: 502

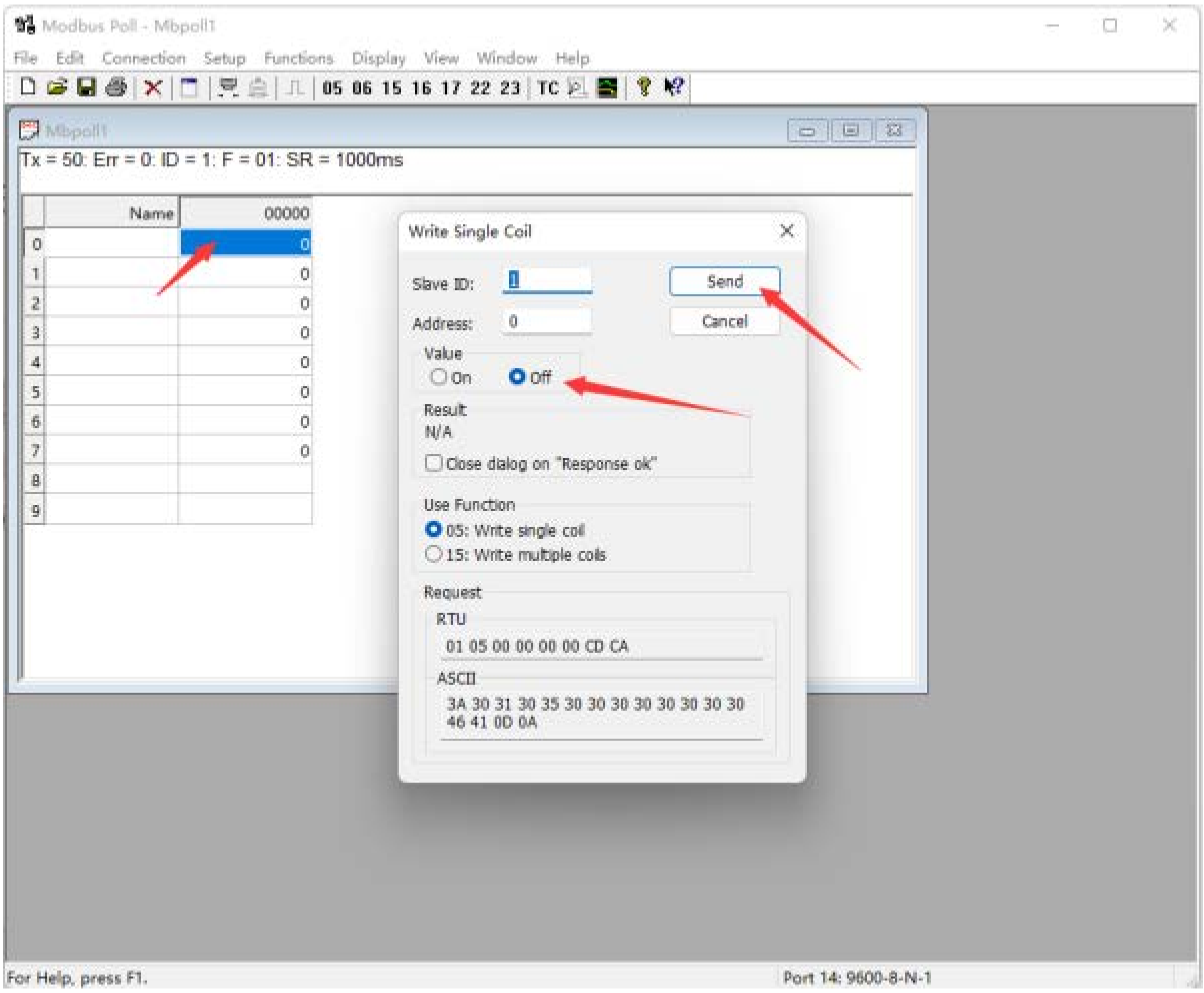
Connect Timeout: 3000 [ms]

● IPv4 ○ IPv6

For Help, press F1.

Port 6: 9600-8-N-1

- After the connection is normal, you can check the current relay status. Select the corresponding channel, then double-click the status value to pop up the send page. Choose On or Off, then click Send to control the relay opening and closing.


Modbus Poll - Mbpoll1
File Edit Connection Setup Functions Display View Window Help
05 06 15 16 17 22 23 TC
Mbpoll1
Send
Cancel

Tx = 50: Err = 0: ID = 1: F = 01: SR = 1000ms

	Name	00000
0		0
1		0
2		0
3		0
4		0
5		0
6		0
7		0
8		
9		

Write Single Coil

Slave ID: 1

Address: 0

Value

○ On ● Off

Result

N/A

□ Close dialog on "Response ok"

Use Function

● 05: Write single coil

○ 15: Write multiple coils

Request

RTU

```
01 05 00 00 00 00 CD CA
```

ASCII

```
3A 30 31 30 35 30 30 30 30 30 30 30 30
46 41 0D 0A
```

For Help, press F1.

Port 14: 9600-8-N-1

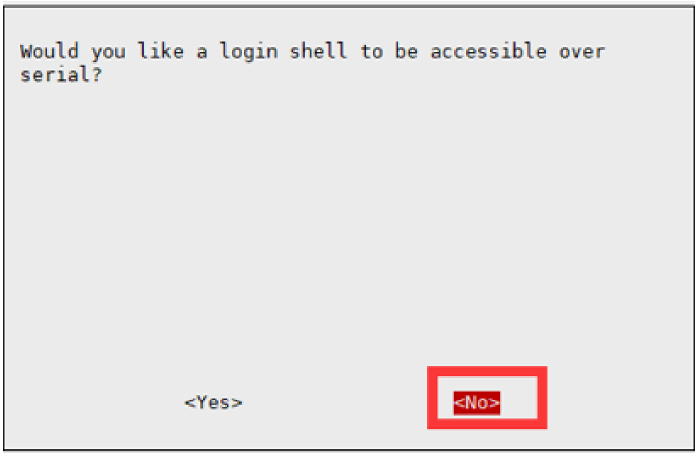
Demo Test

Note: RS485 can not be directly connected to the serial port of the Raspberry Pi, otherwise it may burn the device, you need to add 485 level conversion. For Raspberry Pi, it is recommended to work with the RS485 CAN HAT module. For NUCLEO-F103RB and Arduino, it is recommended to work with the RS485 CAN Shield module.

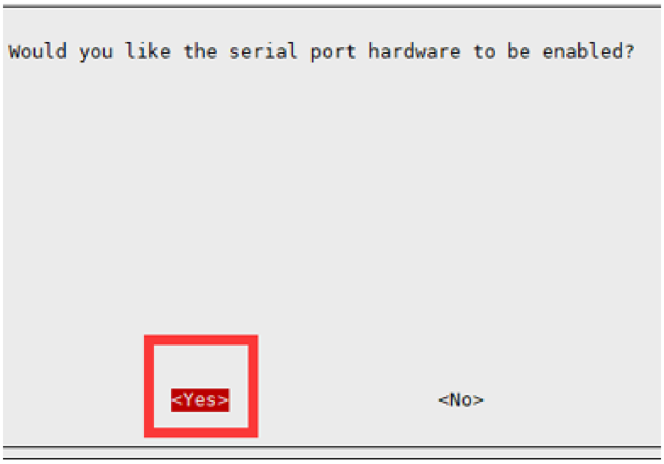
Raspberry Pi

Open the Raspberry Pi terminal and enter the following command to enter the configuration interface

```
sudo raspi-config

Select Interfacing Options -> Serial, disable shell access, and enable the hardw
are serial port
```



Would you like a login shell to be accessible over serial?



Would you like the serial port hardware to be enabled?

Then restart Raspberry Pi:

```
sudo reboot
```

Open the /boot/config.txt file, find the following configuration statement to enable the serial port, if not, you can add it to the end of the file.

```
enable_uart=1
```

For Raspberry Pi 3B users, the serial port is used for Bluetooth and needs to comment out:

```
#dtoverlay=pi3-miniuart-bt
```

Then restart Raspberry Pi:

```
sudo reboot
```

Insert the RS485 CAN HAT into the Raspberry Pi, and connect the Modbus RTU Relay module to the RS485 CAN HAT through A and B.
If you are using other 485 devices, make sure to connect A-A, B-B.
Run the following commands to run the demo:

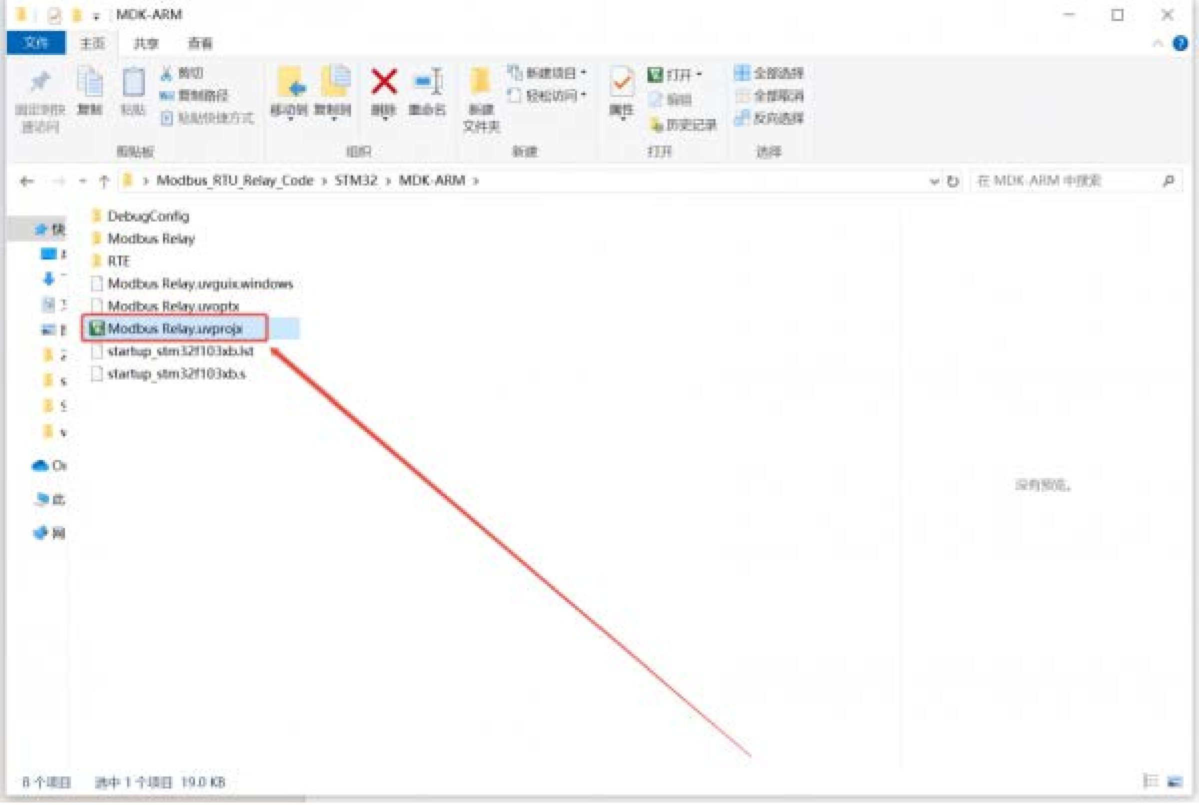
```
sudo apt-get install unzip
wget https://files.waveshare.com/wiki/Modbus-RTU-Relay/Modbus_RTU_Relay_Code.zip
unzip Modbus_RTU_Relay_Code.zip
cd Modbus_RTU_Relay_Code/Python3
```

```
sudo python3 main.py
```

STM32

Note: The STM32 demo is based on the NUCLEO-F103RB and RS485 CAN Shield module.

1. Download Demo, find the STM32 project file Modbus Relay.uvprojx in the path Modbus_RTU_Relay_Code\STM32\MDK-ARM, and double-click to open the STM32 project file. Note that you should ensure Keil5 software is installed on your computer before using it.

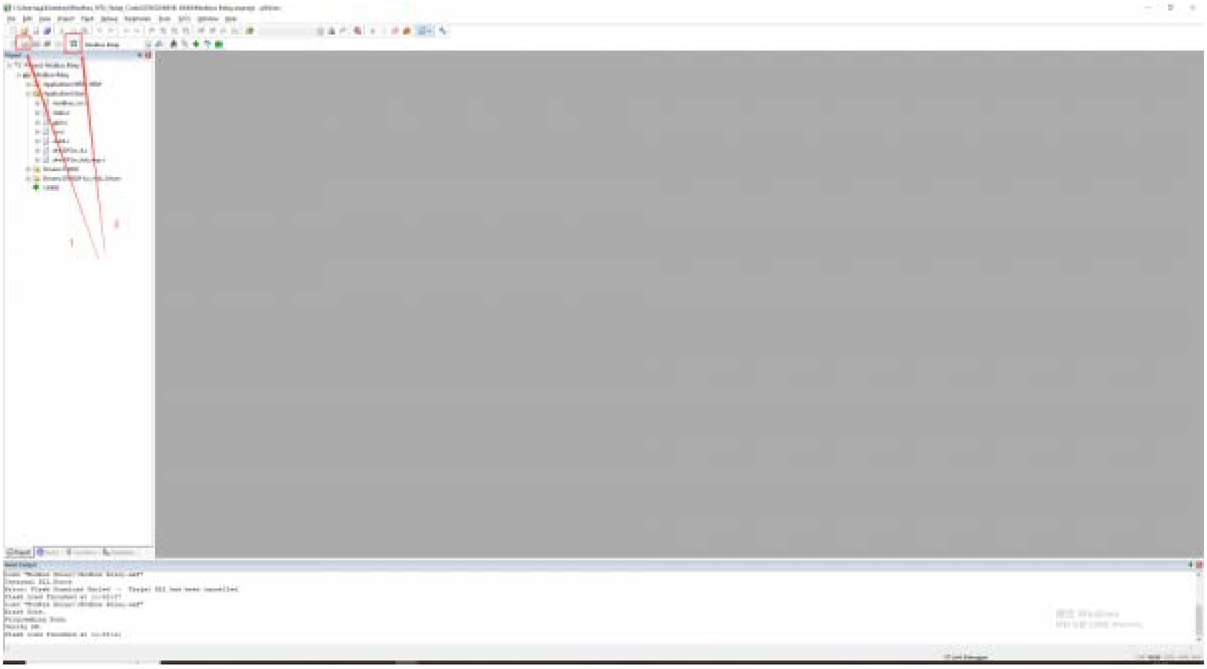


Modbus_RTU_Relay_Code › STM32 › MDK-ARM ›

DebugConfig
Modbus Relay
RTE
Modbus Relay.uvguix.windows
Modbus Relay.uvoptx
Modbus Relay.uvprojx
startup_stm32f103xb.lst
startup_stm32f103xb.s

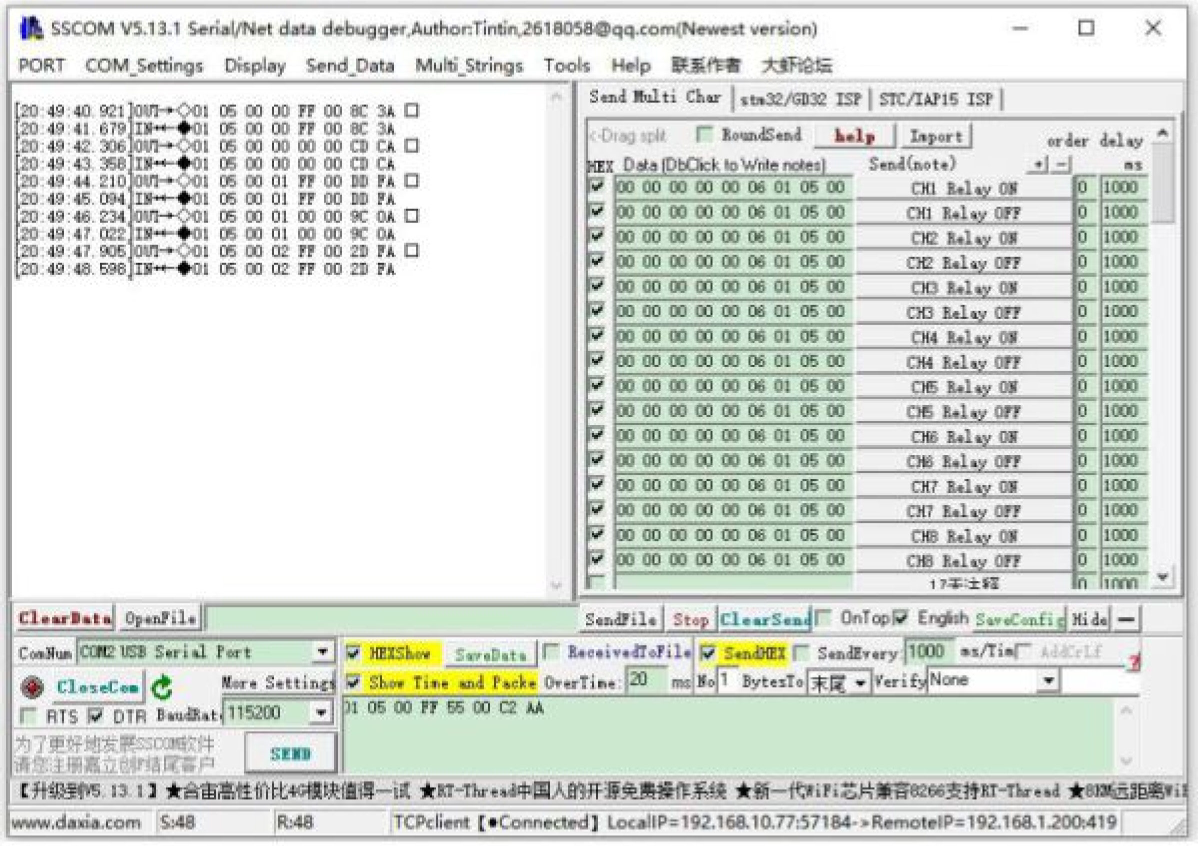
8 个项目 选中 1 个项目 19.0 KB

2. Connect the STM32 to a computer via the STM32 download and debug probe. Compile and download the demo to the STM32.



3. Install the RS485 CAN Shield module on the STM32. Connect the RS485_A on the RS485 CAN Shield module to the RS485_A on the Modbus RTU Relay via a wire, and connect the RS485_B on the RS485 CAN Shield module to the RS485_B on the Modbus RTU Relay via a wire. Then power on the Modbus RTU Relay and the STM32 sequentially.

4. After powering on, the serial port will output the sent commands, which can be observed using a serial port assistant. At the same time, the relays will turn on in sequence (e.g., 1→2→3→4), and after all of them are turned on, they will then turn off in sequence (e.g., 1→2→3→4). You can observe whether the relay is normally engaged through the LED indicator light.



SSCOM V5.13.1 Serial/Net data debugger,Author:Tintin,2618058@qq.com(Newest version)

```
[20:49:40.921]OUT→◇01 05 00 00 FF 00 8C 3A □
[20:49:41.679]IN←◆01 05 00 00 FF 00 8C 3A
[20:49:42.306]OUT→◇01 05 00 00 00 00 CD CA □
[20:49:43.358]IN←◆01 05 00 00 00 00 CD CA
[20:49:44.210]OUT→◇01 05 00 01 FF 00 DD FA □
[20:49:45.094]IN←◆01 05 00 01 FF 00 DD FA
[20:49:46.234]OUT→◇01 05 00 01 00 00 9C 0A □
[20:49:47.022]IN←◆01 05 00 01 00 00 9C 0A
[20:49:47.905]OUT→◇01 05 00 02 FF 00 2D FA □
[20:49:48.598]IN←◆01 05 00 02 FF 00 2D FA
```

HEX	Data (DbClick to Write notes)	Send(note)	order	delay ms
☑	00 00 00 00 00 06 01 05 00	CH1 Relay ON	0	1000
☑	00 00 00 00 00 06 01 05 00	CH1 Relay OFF	0	1000
☑	00 00 00 00 00 06 01 05 00	CH2 Relay ON	0	1000
☑	00 00 00 00 00 06 01 05 00	CH2 Relay OFF	0	1000
☑	00 00 00 00 00 06 01 05 00	CH3 Relay ON	0	1000
☑	00 00 00 00 00 06 01 05 00	CH3 Relay OFF	0	1000
☑	00 00 00 00 00 06 01 05 00	CH4 Relay ON	0	1000
☑	00 00 00 00 00 06 01 05 00	CH4 Relay OFF	0	1000
☑	00 00 00 00 00 06 01 05 00	CH5 Relay ON	0	1000
☑	00 00 00 00 00 06 01 05 00	CH5 Relay OFF	0	1000
☑	00 00 00 00 00 06 01 05 00	CH6 Relay ON	0	1000
☑	00 00 00 00 00 06 01 05 00	CH6 Relay OFF	0	1000
☑	00 00 00 00 00 06 01 05 00	CH7 Relay ON	0	1000
☑	00 00 00 00 00 06 01 05 00	CH7 Relay OFF	0	1000
☑	00 00 00 00 00 06 01 05 00	CH8 Relay ON	0	1000
☑	00 00 00 00 00 06 01 05 00	CH8 Relay OFF	0	1000
□		17无注释	0	1000

□ OnTop ☑ English SaveConfig Hide

ComNum COM2 USB Serial Port

☑ HEXShow SaveData □ ReceivedToFile ☑ SendHEX □ SendEvery 1000 ms/Tim □ AddCrLf

☑ Show Time and Packe OverTime: 20 ms No 1 BytesTo 末尾 Verify None

□ RTS ☑ DTR BaudRat 115200

```
01 05 00 FF 55 00 C2 AA
```

为了更好地发展SSCOM软件
请您注册嘉立创P结尾客户

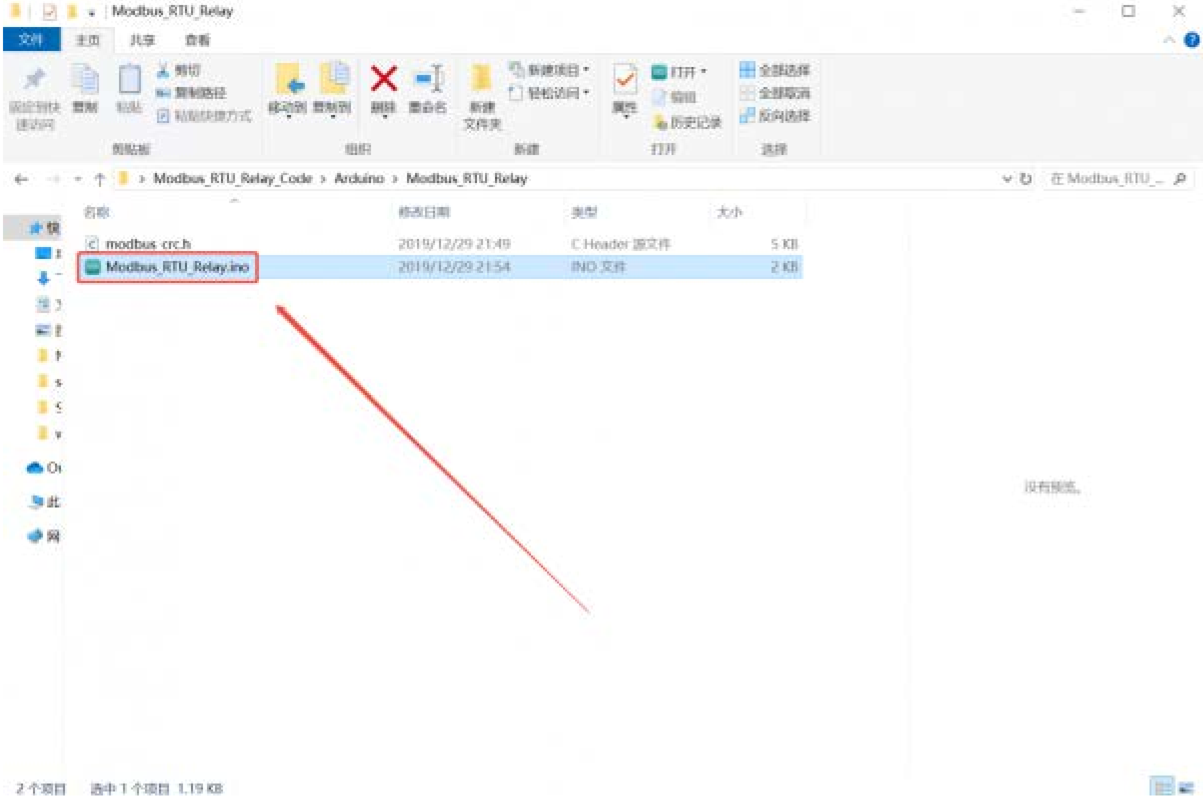
【升级到V5.13.1】★合宙高性价比4G模块值得一试 ★RT-Thread中国人的开源免费操作系统 ★新一代WiFi芯片兼容8266支持RT-Thread ★8KM远距离WiFi

www.daxia.com S:48 R:48 TCPclient [•Connected] LocalIP=192.168.10.77:57184->RemoteIP=192.168.1.200:419

Arduino

Note: The Arduino demo is based on the UNO PLUS and RS485 CAN Shield module.

1. Download Demo, find the Arduino project file Modbus_RTU_Relay.ino in the path Modbus_RTU_Relay_Code\Arduino\Modbus_RTU_Relay, and double-click to open the Arduino project file. Note that you should ensure Arduino IDE software is installed on your computer before using it.

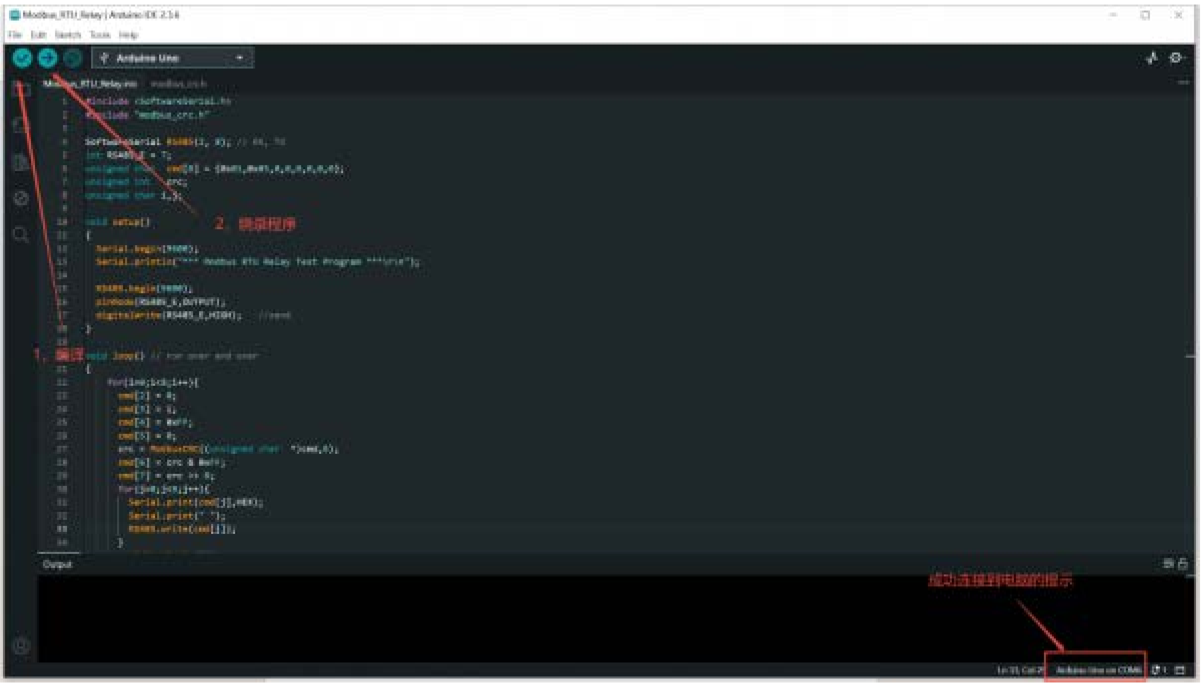


名称	修改日期	类型	大小
modbus_crc.h	2019/12/29 21:49	C Header 源文件	5 KB
Modbus_RTU_Relay.ino	2019/12/29 21:54	INO 文件	2 KB

2 个项目 选中 1 个项目 1.19 KB

2. Connect the Arduino to the computer via a USB cable. In the Arduino IDE software, select the Arduino board model under Tools->Board. Choose the COM port that the Arduino is connected to under Tools->Port.

3. After seeing the prompt to connect to the computer in the lower right corner, click to compile and flash the program, and wait for the flashing to complete.

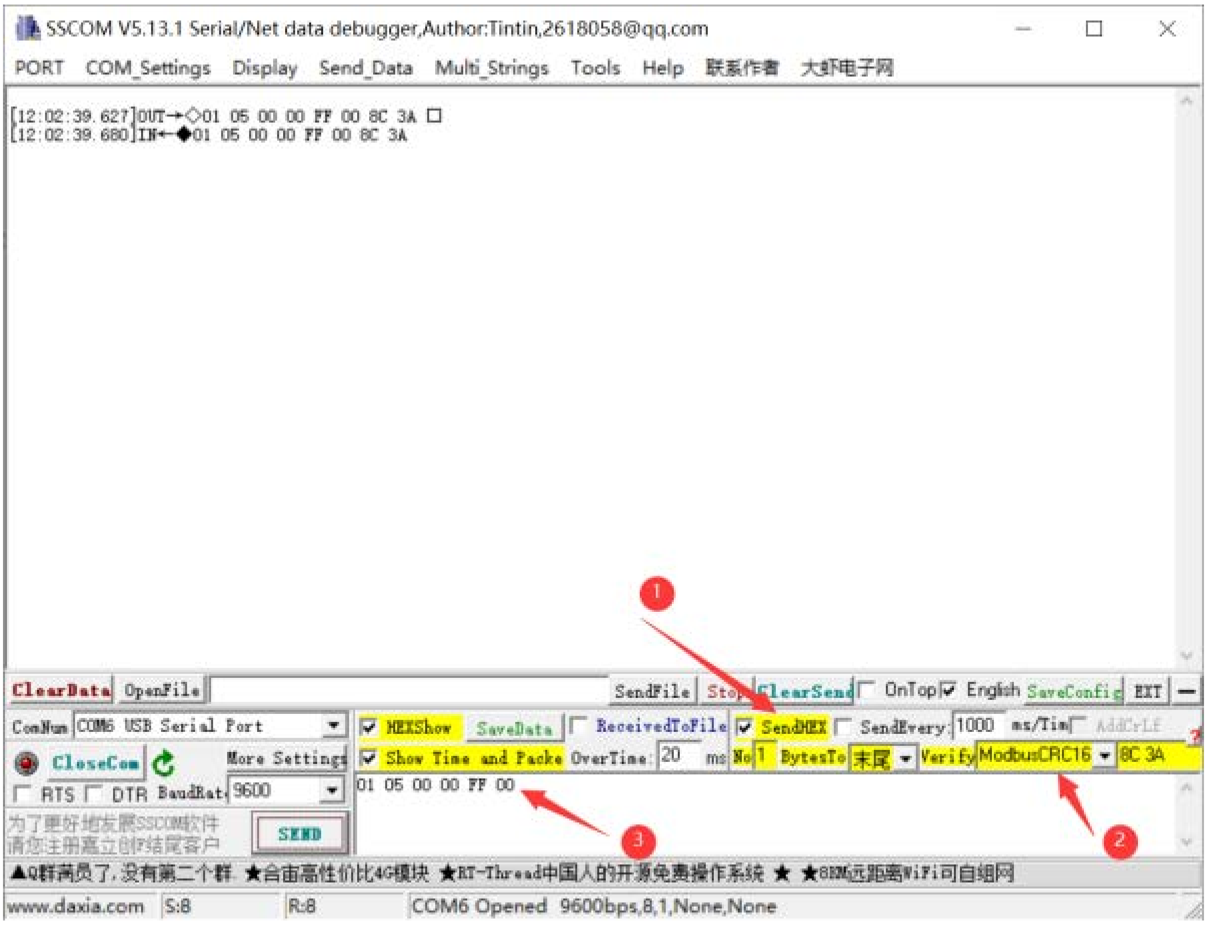


```
#include <SoftwareSerial.h>
#include "modbus_crc.h"

SoftwareSerial RS485(2, 8); // RX, TX
int RS485_E = 7;
unsigned char cmd[8] = {0x01,0x05,0,0,0,0,0,0};
unsigned int crc;
unsigned char i,j;

void setup()
{
  Serial.begin(9600);
  Serial.println("**** Modbus RTU Relay Test Program ***\r\n");

  RS485.begin(9600);
  pinMode(RS485_E,OUTPUT);
  digitalWrite(RS485_E,HIGH);   //send
}

void loop() // run over and over
{
    for(i=0;i<4;i++){
      cmd[2] = 0;
      cmd[3] = i;
      cmd[4] = 0xFF;
      cmd[5] = 0;
      crc = ModbusCRC((unsigned char *)cmd,6);
      cmd[6] = crc & 0xFF;
      cmd[7] = crc >> 8;
      for(j=0;j<8;j++){
        Serial.print(cmd[j],HEX);
        Serial.print(" ");
        RS485.write(cmd[j]);
      }
```

4. Install the RS485 CAN Shield module on the Arduino. Connect the RS485_A on the RS485 CAN Shield module to the RS485_A on the Modbus RTU Relay via a wire, and connect the RS485_B on the RS485 CAN Shield module to the RS485_B on the Modbus RTU Relay via a wire. Then power on the Modbus RTU Relay and the Arduino sequentially.

5. After powering on, the serial port will output the sent commands, which can be observed using a serial port assistant. At the same time, the relays will turn on in sequence (e.g., 1→2→3→4), and after all of them are turned on, they will then turn off in sequence (e.g., 1→2→3→4). You can observe whether the relay is normally engaged through the LED indicator light.

Note: The data in the figure below is in ASCII code containing spaces and line breaks, and is not the actual data sent.



SSCOM V5.13.1 Serial/Net data debugger,Author:Tintin,2618058@qq.com

```
[12:02:39.627]OUT→◇01 05 00 00 FF 00 8C 3A □
[12:02:39.680]IN←◆01 05 00 00 FF 00 8C 3A
```

ComNum: COM6 USB Serial Port

BaudRate: 9600

☑ HEXShow

☑ Show Time and Packe OverTime: 20 ms No 1 BytesTo 末尾

☑ SendHEX

□ SendEvery: 1000 ms/Tim

Verify ModbusCRC16 8C 3A

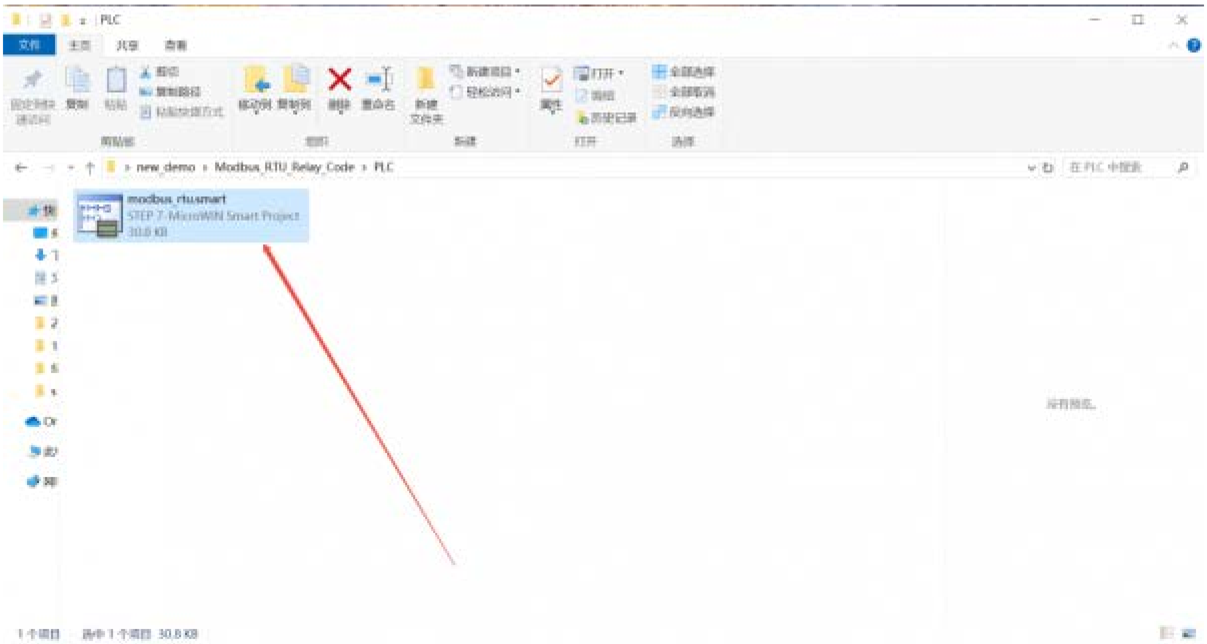
```
01 05 00 00 FF 00
```

www.daxia.com S:8 R:8 COM6 Opened 9600bps,8,1,None,None

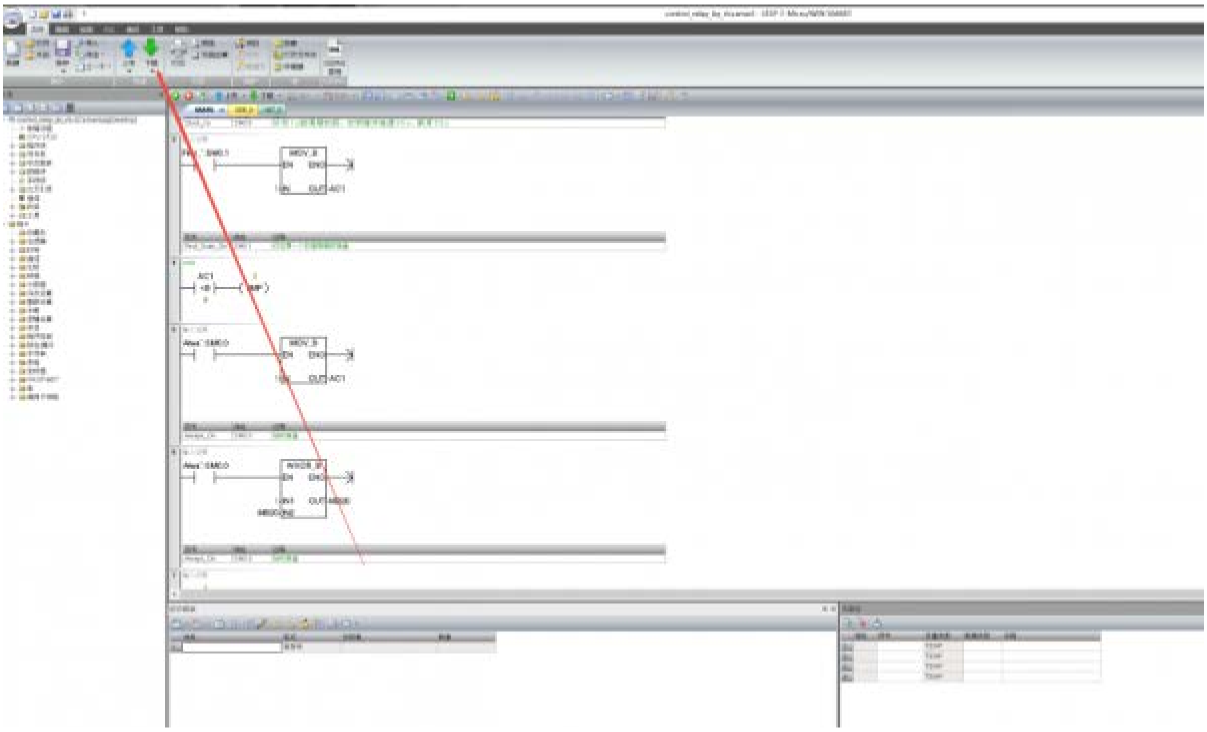
PLC

Note: The PLC demo is based on SIMATIC S7-200 SMART.

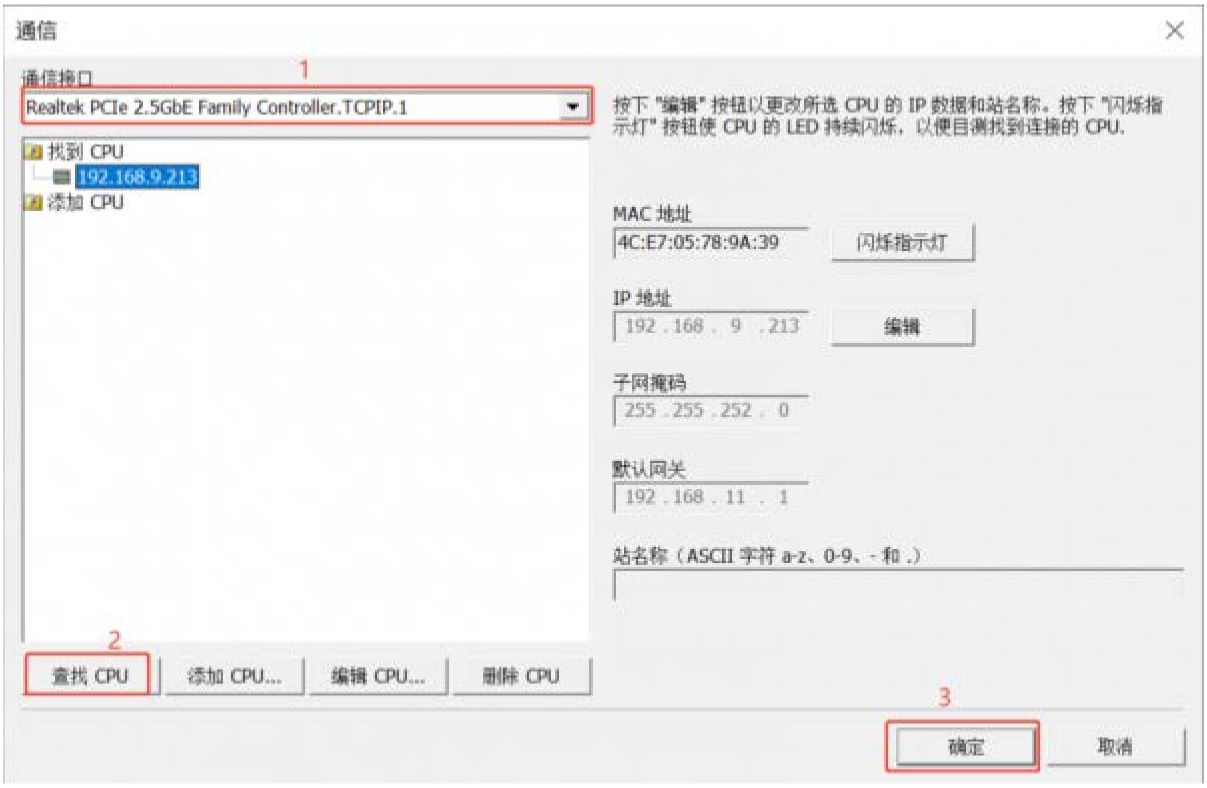
1. Download Demo, find the PLC project file modbus_rtu.smart in the path Modbus_RTU_Relay_Code\PLC, and double-click to open the PLC project file. Note that you should ensure STEP 7-MicroWIN SMART software is installed on your computer before using it.


PLC
文件
主页
共享
查看
剪贴板
组织
新建
打开
选择
new_demo › Modbus_RTU_Relay_Code › PLC
在 PLC 中搜索
modbus_rtu.smart
STEP 7-MicroWIN Smart Project
30.8 KB
没有预览。
1 个项目 选中 1 个项目 30.8 KB

2. Connect the PLC to the computer via a network cable. Click to download



3. Select the communication interface in the communication popup, find your device, and click Confirm



通信

通信接口

Realtek PCIe 2.5GbE Family Controller.TCPIP.1

找到 CPU

192.168.9.213

添加 CPU

按下 "编辑" 按钮以更改所选 CPU 的 IP 数据和站名称。按下 "闪烁指示灯" 按钮使 CPU 的 LED 持续闪烁，以便目测找到连接的 CPU。

MAC 地址: 4C:E7:05:78:9A:39

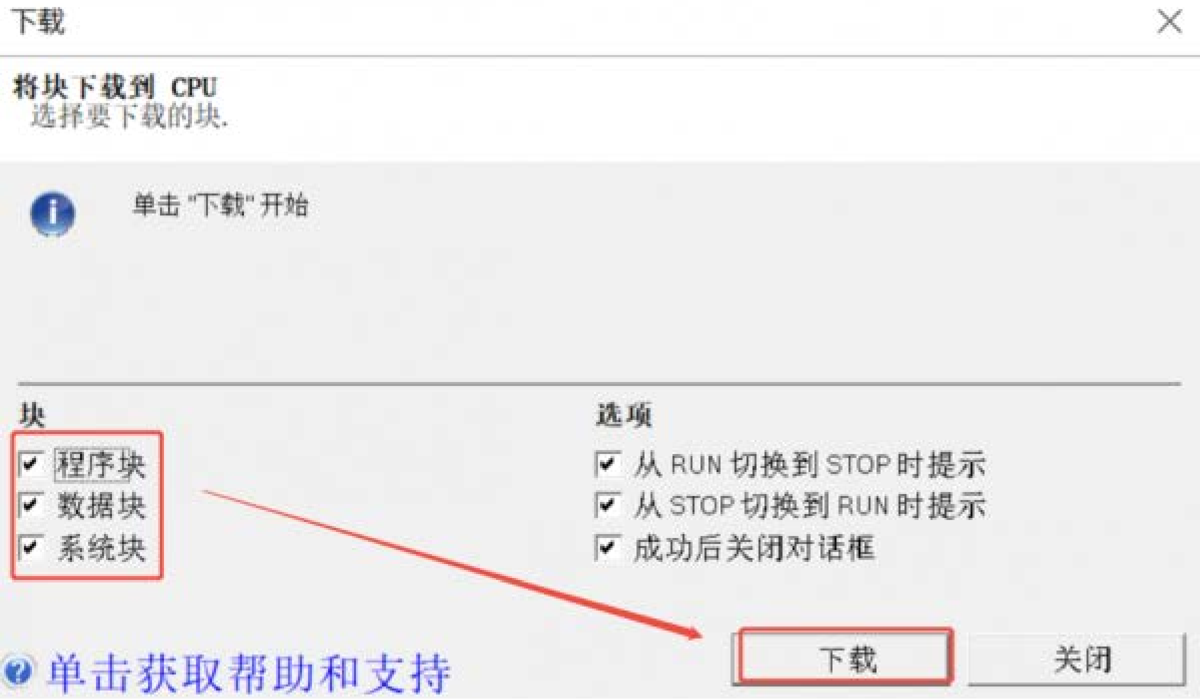
IP 地址: 192 . 168 . 9 . 213

子网掩码: 255 . 255 . 252 . 0

默认网关: 192 . 168 . 11 . 1

站名称（ASCII 字符 a-z、0-9、- 和 .）

4. In the download pop-up window, check the boxes for program blocks, data blocks, and system blocks, and then click Download.



下载

将块下载到 CPU

选择要下载的块.

单击 "下载" 开始

块

☑ 程序块
☑ 数据块
☑ 系统块

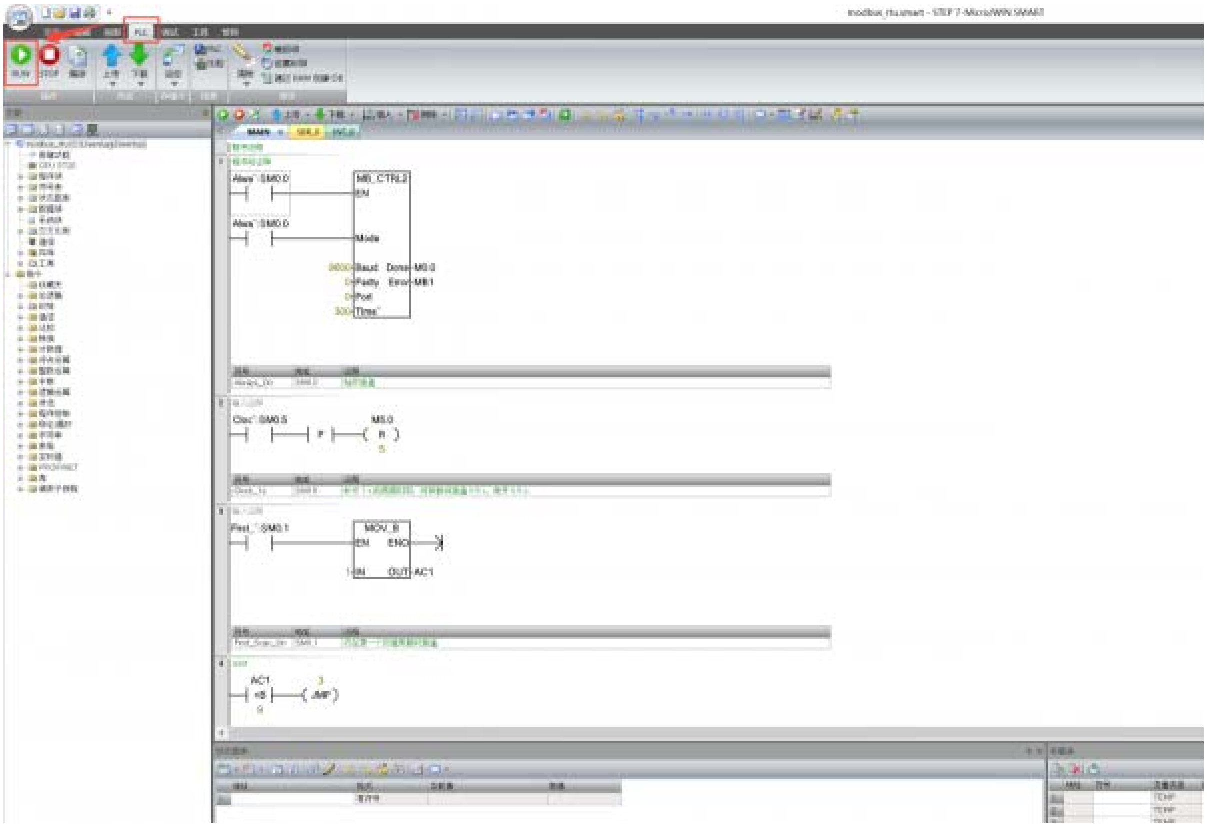
选项

☑ 从 RUN 切换到 STOP 时提示
☑ 从 STOP 切换到 RUN 时提示
☑ 成功后关闭对话框

单击获取帮助和支持

5. Connect the 485_A on the PLC module to RS485_A on the Modbus RTU Relay with a wire, and connect the 485_B on the PLC module to RS485_B on the Modbus RTU Relay with a wire. Then power on the Modbus RTU Relay.

6. After powering on the Modbus RTU Relay, in the STEP 7-MicroWIN SMART software, find the "PLC" tab in the upper menu bar, switch to this tab, and click the green "RUN" button in the lower "Operation" area to perform the operation to make the PLC enter the running state



7. After the PLC runs, the relays will turn on in sequence (e.g., 1→2→3→4), and after all of them are turned on, they will then turn off in sequence (e.g., 1→2→3→4). You can observe whether the relay is normally engaged through the LED indicator light.

Development Protocol V3

Function Code Introduction

Function Code	Description	Note
01	Read coil status	Read relay status
03	Read holding register	Read the address and version
05	Write single coil	Write single relay
06	Write single register	Set the baud rate and address
0F	Write multiple coils	Write all relays

Register Address Introduction

Address (HEX)	Address storage content	Register value	Permission	Modbus Function Code
0x0000 …… 0x0007	Channels 1~8 relay addresses	0xFF00: relay on 0x0000: relay off 0x5500: relay toggle	Read/Write	0x01, 0x05, 0x0F
0x00FF	Control all relays	0xFF00: all relays on 0x0000: all relays off 0x5500: all relays toggle	Write	0x05
0x0100 …… 0x0107	Channels 1~8 relays toggle	0xFF00: relay toggle 0x0000: relay unchanged	Write	0x05, 0x0F
0x01FF	All relays toggle	0xFF00: all relays toggle 0x0000: all relays unchanged	Write	0x05
0x0200 …… 0x0207	Channels 1~8 relays flash on	Delay time: data*100ms Value: 0x0007, delay time: 7*100MS = 700MS	Write	0x05
0x0400 …… 0x0407	Channels 1~8 relays flash off	Delay time: data*100ms Value: 0x0007, delay time: 7*100MS = 700MS	Write	0x05
4x2000	UART Parameter	The high eight bits indicate the parity mode: 0x00~0x02 The low eight bits indicate the baud rate mode: 0x00~0x07	Read/Write	0x03, 0x06
4x4000	Device Address	Directly store Modbus address Device address: 0x0001-0x00FF	Read/Write	0x03, 0x06
4x8000	Software Version	Converting to decimal and then shifting the decimal point two places to the left will represent the software version 0x0064 = 100 = V1.00	Read	0x03

Modbus RTU Command Introduction

Control Single Relay

Send code: 01 05 00 00 FF 00 8C 3A

Field	Description	Note
01	Device Address	0x00 indicates the broadcast address, 0x01-0xFF indicates the device address
05	05 Command	Relay control
00 00	Address	The register address of the relay to be controlled, 0x0000-0x0007

FF 00	Command	0xFF00: relay on;
		0x0000: relay off;
		0x5500: relay toggle

8C 3A	CRC16	The CRC16 checksum of the first 6 bytes of data

Return code: 01 05 00 00 FF 00 8C 3A

Field	Description	Note
01	Device Address	0x00 indicates the broadcast address, 0x01-0xFF indicates the device address
05	05 Command	Relay control
00 00	Address	The register address of the relay to be controlled, 0x0000-0x0007

FF 00	Command	0xFF00: relay on;
		0x0000: relay off;
		0x5500: relay toggle

8C 3A	CRC16	The CRC16 checksum of the first 6 bytes of data

For example: [Address 1 device]:

```
Relay 0 on: 01 05 00 00 FF 00 8C 3A
Relay 0 off: 01 05 00 00 00 00 CD CA
Relay 1 on: 01 05 00 01 FF 00 DD FA
Relay 1 off: 01 05 00 01 00 00 9C 0A
Relay 2 on: 01 05 00 02 FF 00 2D FA
Relay 2 off: 01 05 00 02 00 00 6C 0A
Relay 3 on: 01 05 00 03 FF 00 7C 3A
Relay 3 off: 01 05 00 03 00 00 3D CA
Relay 0 toggle: 01 05 00 00 55 00 F2 9A
Relay 1 toggle: 01 05 00 01 55 00 A3 5A
Relay 2 toggle: 01 05 00 02 55 00 53 5A
Relay 3 toggle: 01 05 00 03 55 00 02 9A
```

Control All Relays

Send code: 01 05 00 FF FF 00 BC 0A

Field	Description	Note
01	Device Address	0x00 indicates the broadcast address, 0x01-0xFF indicates the device address
05	05 Command	Relay control
00 FF	Address	Fixed 0x00FF

FF 00	Command	0xFF00: relay on;
		0x0000: relay off;
		0x5500: relay toggle

BC 0A	CRC16	The CRC16 checksum of the first 6 bytes of data

Return code: 01 05 00 FF FF 00 BC 0A

Field	Description	Note
01	Device Address	0x00 indicates the broadcast address, 0x01-0xFF indicates the device address
05	05 Command	Relay control

00 FF	Address	Fixed 0x00FF

FF 00	Command	0xFF00: relay on;
		0x0000: relay off;
		0x5500: relay toggle

BC 0A	CRC16	The CRC16 checksum of the first 6 bytes of data

For example: [Address 1 device]:

```
All relays on: 01 05 00 FF FF 00 BC 0A

All relays off: 01 05 00 FF 00 00 FD FA

All relays toggle: 01 05 00 FF 55 00 C2 AA
```

Read Relay Status

Send code: 01 01 00 00 00 08 3D CC

Field	Description	Note
01	Device Address	0x00 indicates the broadcast address, 0x01-0xFF indicates the device address
01	01 Command	Query relay status
00 00	Relay Start Address	The register address of the relay, 0x0000 - 0x0007
00 08	Relay Number	The number of relays to be read, which must not exceed the maximum number of relays
3D CC	CRC16	The CRC16 checksum of the first 6 bytes of data

Receive code: 01 01 01 00 51 88

Field	Description	Note
01	Device Address	0x00 indicates the broadcast address, 0x01-0xFF indicates the device address

01	01 Command	Query relay status
01	Byte Number	The number of all bytes of the returned status information
00	Query status	Received relay status Bit0: the first relay status; Bit1: the second relay status; And so on, with the idle high bit being zero
51 88	CRC16	The CRC16 checksum of the first 6 bytes of data

For example: [Address 1 device]

```
Send: 01 01 00 00 00 08 3D CC
Receive: 01 01 01 00 51 88       //all relays off
Send: 01 01 00 00 00 08 3D CC
Receive: 01 01 01 01 90 48       //Relay 0 is on, others are off
Send: 01 01 00 00 00 08 3D CC
Receive: 01 01 01 41 91 B8       //Relay 0 and 6 are on, others are off
```

Write Relay Status

Send code: 01 0F 00 00 00 08 01 FF BE D5

Field	Description	Note
01	Device Address	0x00 indicates the broadcast address, 0x01-0xFF indicates the device address
0F	0F Command	Write relay status
00 00	Relay Start Address	The register address of the relay to be controlled, 0x0000 - 0x0007
00 08	Relay Number	The number of relays to be operated, which must not exceed the maximum number of relays
01	Byte Number	The byte number of the status

FF	Relay Status	Bit0: the first relay status; Bit1: the second relay status; And so on, with the idle high bit being zero
BE D5	CRC16	The CRC16 checksum of the first 6 bytes of data

Receive code: 01 0F 00 00 00 08 54 0D

Field	Description	Note
01	Device Address	0x00 indicates the broadcast address, 0x01-0xFF indicates the device address
0F	0F Command	Control all registers
00 00	Relay Start Address	The register address of the relay to be controlled, 0x0000 - 0x0007
00 08	Relay Number	The number of relays to be operated
54 0D	CRC16	The CRC16 checksum of the first 6 bytes of data

For example: [Address 1 device]

```
All relays on: 01 0F 00 00 00 08 01 FF BE D5
All relays off: 01 0F 00 00 00 08 01 00 FE 95
0-1 on; 2-7 off: 01 0F 00 00 00 08 01 03 BE 94
```

Relay Flash ON/OFF Command

Send code: 01 05 02 00 00 07 8D B0

Field	Description	Note
01	Device Address	0x00 indicates the broadcast address, 0x01-0xFF indicates the device address
05	05 Command	Single control command

02	Flash on flash off	02: flash on, 04: flash off
00	Relay Address	The address of the relay to be controlled, 0x00~0x07
00 07	Delay Time	The delay time: data*100ms Value: 0x0007, delay time: 7*100MS = 700MS The maximum setting for the flash-on flash-off time is 0x7FFF
8D B0	CRC16	The CRC16 checksum of the first 6 bytes of data

Receive code: 01 05 02 00 00 07 8D B0

Field	Description	Note
01	Device Address	0x00 indicates the broadcast address, 0x01-0xFF indicates the device address
05	05 Command	Single control command
02	Flash on flash off	02: flash on, 04: flash off
00	Relay Address	The address of the relay to be controlled, 0x00~0x07
00 07	Delay Time	The delay time: data*100ms Value: 0x0007, delay time: 7*100MS = 700MS The maximum setting for the flash-on flash-off time is 0x7FFF
8D B0	CRC16	The CRC16 checksum of the first 6 bytes of data

For example: [Address 1 device]

```
Relay 0 flash on: 01 05 02 00 00 07 8D B0   //700MS = 7*100MS = 700MS
Relay 1 flash on: 01 05 02 01 00 08 9C 74   //800MS
Relay 0 flash off: 01 05 04 00 00 05 0C F9  //500MS
Relay 1 flash off: 01 05 04 01 00 06 1D 38  //600MS
```

Set Baudrate Command

Send code: 01 06 20 00 00 05 42 09

Field	Description	Note
01	Device Address	0x00 indicates the broadcast address, 0x01-0xFF indicates the device address
06	06 command	Set the baud rate and device address
20 00	Baud rate register	0x2000: set the baud rate
00	Parity Method	0x00: no parity, 0x01: even parity; 0x02: odd parity
05	Baud Rate Value	Correspondence of baud rate values 0x00: 4800 0x01: 9600 0x02: 19200 0x03: 38400 0x04: 57600 0x05: 115200 0x06: 128000 0x07: 256000
42 09	CRC16	The CRC16 checksum of the first 6 bytes of data

Receive code: 01 06 20 00 00 05 42 09

Field	Description	Note
01	Device Address	0x00 indicates the broadcast address, 0x01-0xFF indicates the device address
06	06 command	Set the baud rate and device address
20 00	Baud rate register	0x2000: set the baud rate
00	Parity Method	0x00: no parity, 0x01: odd parity; 0x02: even parity
05	Baud Rate	Correspondence of baud rate values 0x00: 4800 0x01: 9600 0x02: 19200

		0x03: 38400 0x04: 57600 0x05: 115200 0x06: 128000 0x07: 256000
42 09	CRC16	The CRC16 checksum of the first 6 bytes of data

For example: [Address 1 device]

```
Set the baud rate as 4800: 01 06 20 00 00 00 82 0A
Set the baud rate as 9600: 01 06 20 00 00 01 43 CA
Set the baud rate as 115200: 01 06 20 00 00 05 42 09
```

Set Device Address Command

Send code: 01 06 40 00 00 01 5D CA

Field	Description	Note
01	Device Address	0x00 indicates the broadcast address, 0x01-0xFF indicates the device address
06	06 command	Set the baud rate and device address
40 00	Device address register	0x4000: set the device address
00 01	Device Address	Set the device address, 0x0001-0x00FF
5D CA	CRC16	The CRC16 checksum of the first 6 bytes of data

Receive code: 01 06 40 00 00 01 5D CA

Field	Description	Note
01	Device Address	0x00 indicates the broadcast address, 0x01-0xFF indicates the device address

06	06 command	Set the baud rate and device address
40 00	Device address register	0x4000: set the device address
00 01	Device Address	Set the device address, 0x0001-0x00FF
5D CA	CRC16	The CRC16 checksum of the first 6 bytes of data

For example: [Address 1 device]

```
Set the device address as 0x01: 00 06 40 00 00 01 5C 1B
Set the device address as 0x02: 00 06 40 00 00 02 1C 1A
Set the device address as 0x03: 00 06 40 00 00 03 DD DA
```

Read Device Address Command

Send code: 00 03 40 00 00 01 90 1B

Field	Description	Note
00	Device Address	0x00 indicates the broadcast address, 0x01-0xFF indicates the device address
03	03 Command	Read the device address
40 00	Device address register	0x4000: read the device address
00 01	Byte Number	Fixed 0x0001
90 1B	CRC16	The CRC16 checksum of the first 6 bytes of data

Receive code: 00 03 02 00 01 44 44

Field	Description	Note

00	Device Address	0x00 indicates the broadcast address, 0x01-0xFF indicates the device address
03	03 Command	Read the software version and device address
02	Byte Number	The number of bytes returned
00 01	Device Address	Set the device address, 0x0001-0x00FF
44 44	CRC16	The CRC16 checksum of the first 6 bytes of data

For example: [Address 2 device]

```
Send: 00 03 40 00 00 01 90 1B
Receive: 00 03 02 00 02 04 45     //Address: 0x02
```

Read Software Version Command

Send code: 01 03 80 00 00 01 AD CA

Field	Description	Note
01	Device Address	0x00 indicates the broadcast address, 0x01-0xFF indicates the device address
03	03 Command	Read the software version and device address
80 00	Software version register	0x4000: read the device address, 0x8000: read software version
00 01	Byte Number	Fixed 0x0001
AD CA	CRC16	The CRC16 checksum of the first 6 bytes of data

Receive code: 01 03 02 01 2C B8 09

Field	Description	Note

01	Device Address	0x00 indicates the broadcast address, 0x01-0xFF indicates the device address
03	03 Command	Read the software version and device address
02	Byte Number	The number of bytes returned
01 2C	Software Version	Converting to decimal and then shifting the decimal point two places to the left will represent the software version 0x012C = 300 = V3.00
B8 09	CRC16	The CRC16 checksum of the first 6 bytes of data

For example:

```
Send: 01 03 80 00 00 01 AD CA
Receive: 01 03 02 01 2C B8 09       //0x012C = 300 =V3.00
```

Exception Function Code

When the received command is incorrect or the device is abnormal, an exception response will be returned in the following format:

Return code: 01 85 03 02 91

Field	Description	Note
01	Device Address	0x00 indicates the broadcast address, 0x01-0xFF indicates the device address
85	Exception Function Code	Exception function code = Request function code + 0x80
03	Byte Number	Exception Code
02 91	CRC16	The CRC16 checksum of the first 6 bytes of data

An exception code is a single-byte value that indicates the type of error. Several commonly used exception codes defined by the Modbus protocol:

Exception Code	Name	Description
0x01	Illegal Function	The requested function code is not supported
0x02	Illegal Data Address	The requested data address is incorrect
0x03	Illegal Data Value	The requested data value or operation cannot be executed
0x04	Server Failure	Server equipment failure
0x05	Response	The request has been received and is being processed
0x06	Device Busy	The device is currently busy and cannot perform the requested operation

Resources

Demo

- Demo

Software

- Sscom software for Modbus RTU Relay V2
- Sscom software for Modbus RTU Relay
- Modbus Poll Software
- SecureCRT Software

Related resources

- Modbus Protocol Introduction
- Modbus Series BootLoader Instructions
- Development Protocol V2

- Development Protocol V1
- Using Modbus RTU Relay with Home Assistant