waveshare.com

Modbus RTU Relay (D) - Waveshare Wiki

31–39 минут

Overview

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.

This is an industrial 8-channel relay module controlled via RS485 bus, with 8-ch digital input, utilizing Modbus RTU protocol. It features embedded protection circuits such as power isolation, magnetical isolation, TVS, etc. It also comes with an ABS enclosure. This relay module is very easy to

use. Due to its fast communication, stability, reliability, and safety, it is an ideal choice for industrial control equipments and/or applications with high communication requirements.

Specifications

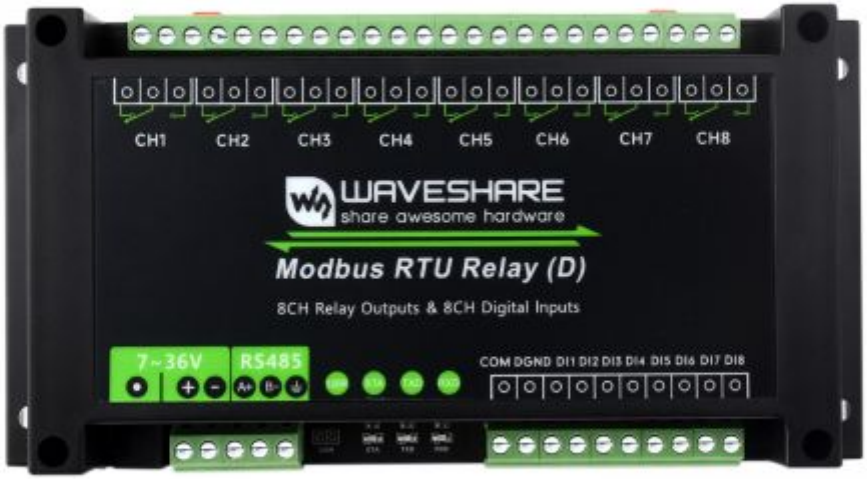
Power Supply	DC 7~36V
Communication Interface	RS485
Baudrate	4800, 9600, 19200, 38400, 57600, 115200, 128000, 256000
Default Communication Format	9600, N, 8, 1
Relay Channels	8 Channels
Contact Form	1NO 1NC
Contact Load	≤10A 250V AC or ≤10A 30V DC
Digital Input	8DI, 5~36V, passive/active input (NPN or PNP), built-in bi-directional optocoupler
Modbus Protocol	Standard Modbus RTU protocol
RS485 Address Setting	1~255
Indicator	STA: MCU indicator, keep flashing when MCU normally working TXD: Transmit Data indicator, lights up when transmitting data RXD: Receive Data indicator, lights up when receiving data

Primary Functions

Supports reading digital input by sending Modbus RTU protocol commands via RS485, and can control relay output based on input.

Linkage Control

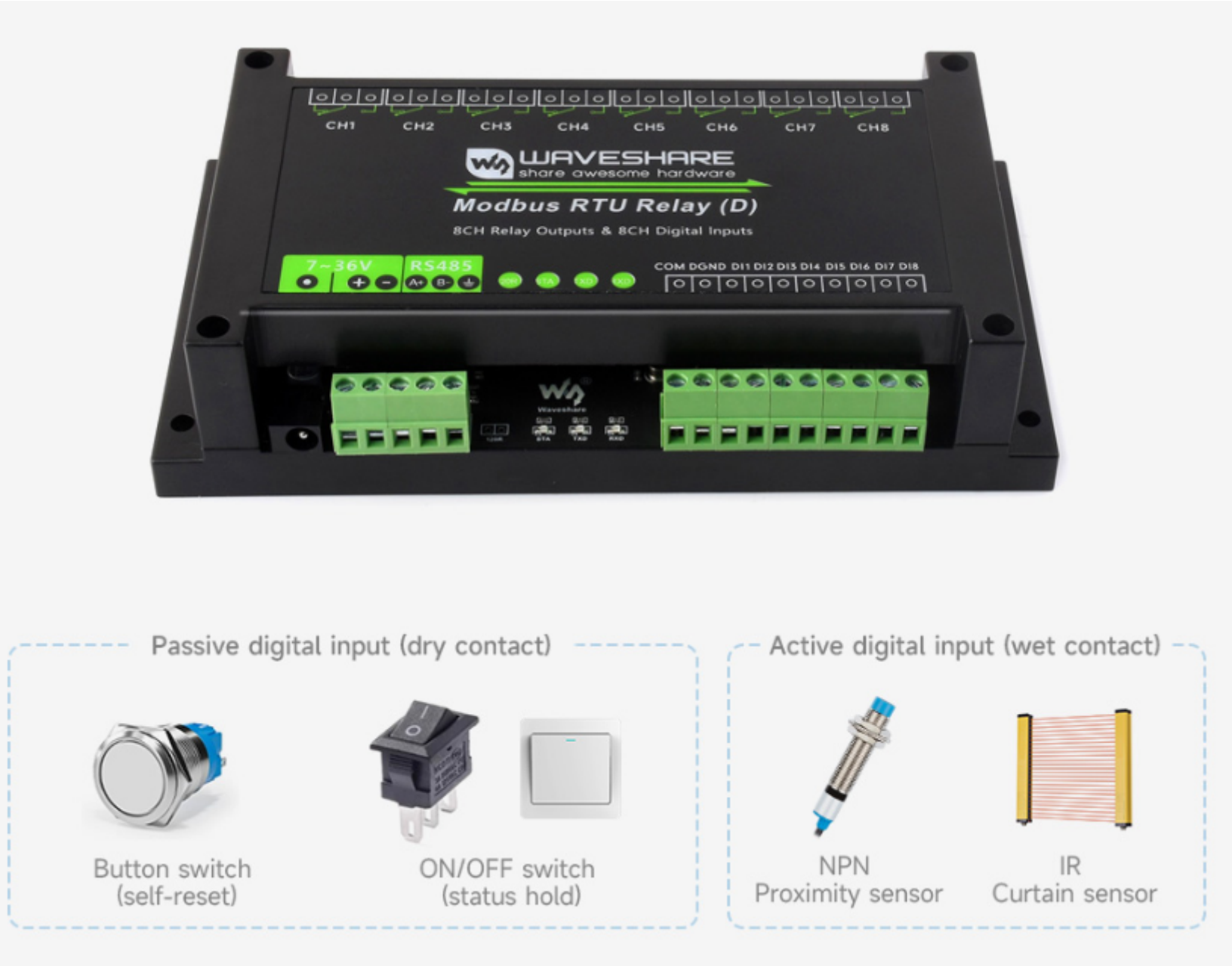
Wiring Description



DI1-DI8 are the 8-channel signal input terminals, and DGND is the signal terminal ground. COM is the common terminal for the input signal, it can be NC (Not Connected), connected to the positive or negative poles of the power supply, directly powered from the power supply or connected to an independent power supply.

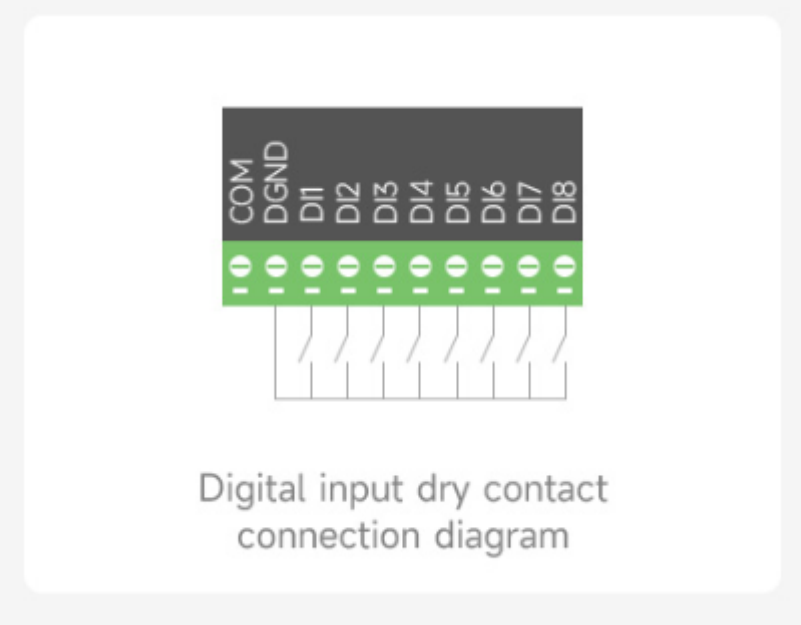
- NC: dry contact passive input.
- Connect to the positive pole of power supply: low active, NPN wet contact active input, voltage: 5V-30V DC.
- Connect to the negative pole of power supply: high active, PNP wet contact active input.

Digital Input Wiring



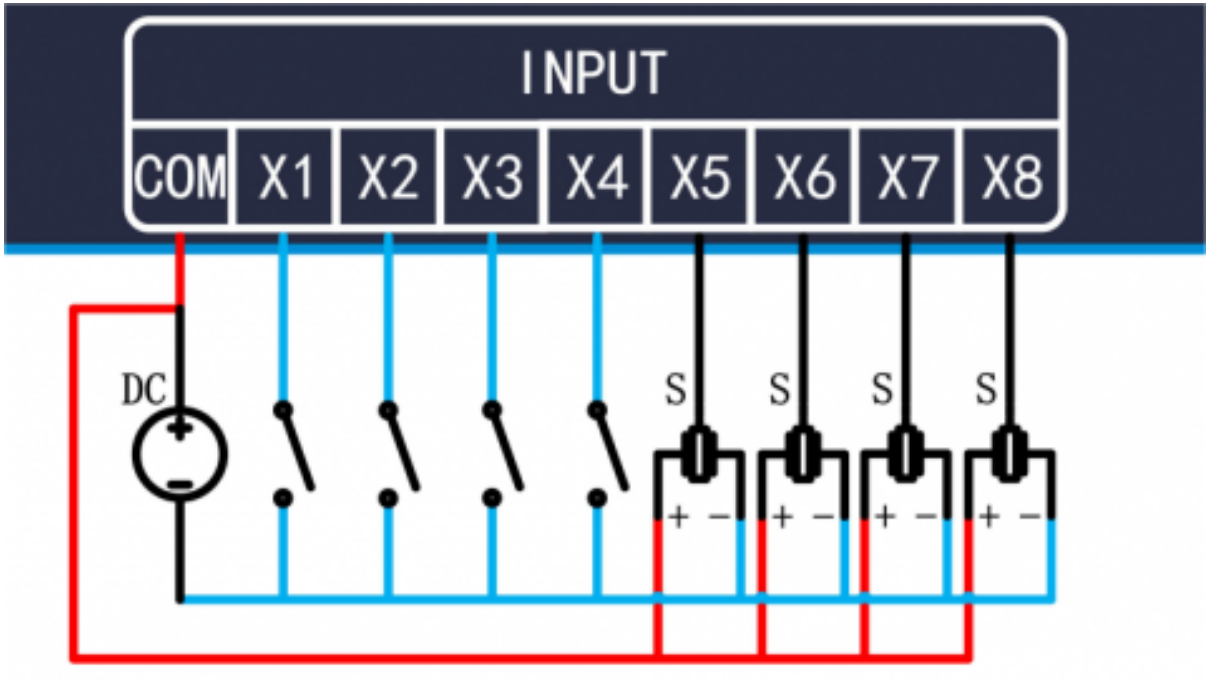
Passive Dry Contact Wiring

- Passive dry contact input

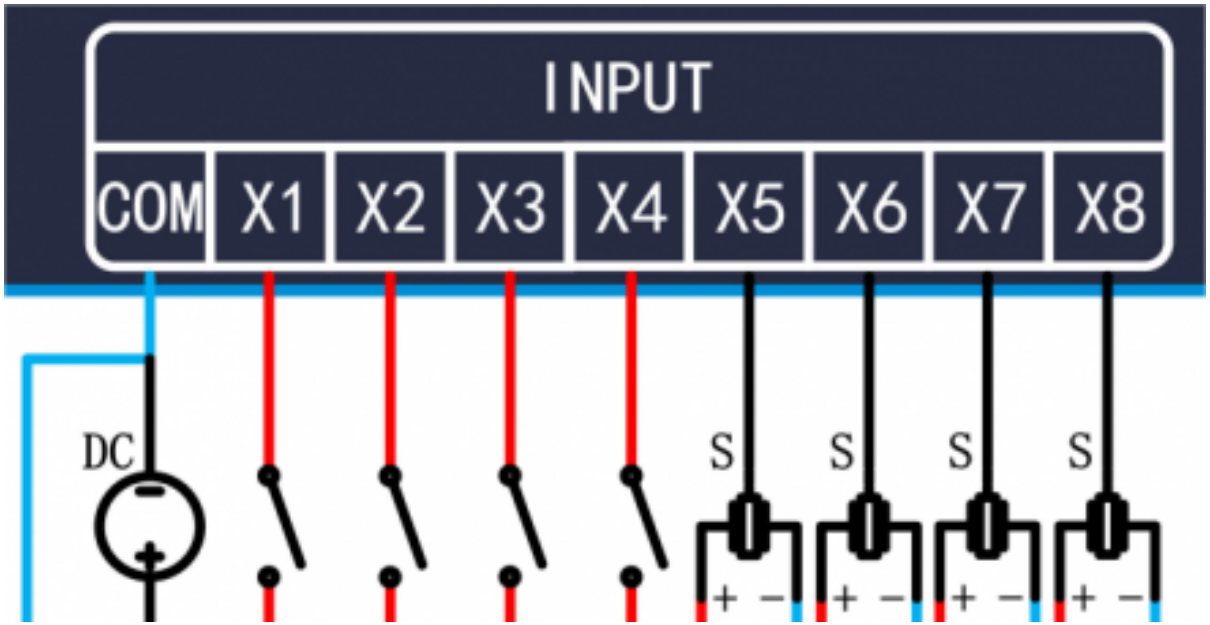


Active Wet Contact Wiring

- Active wet contact NPN input

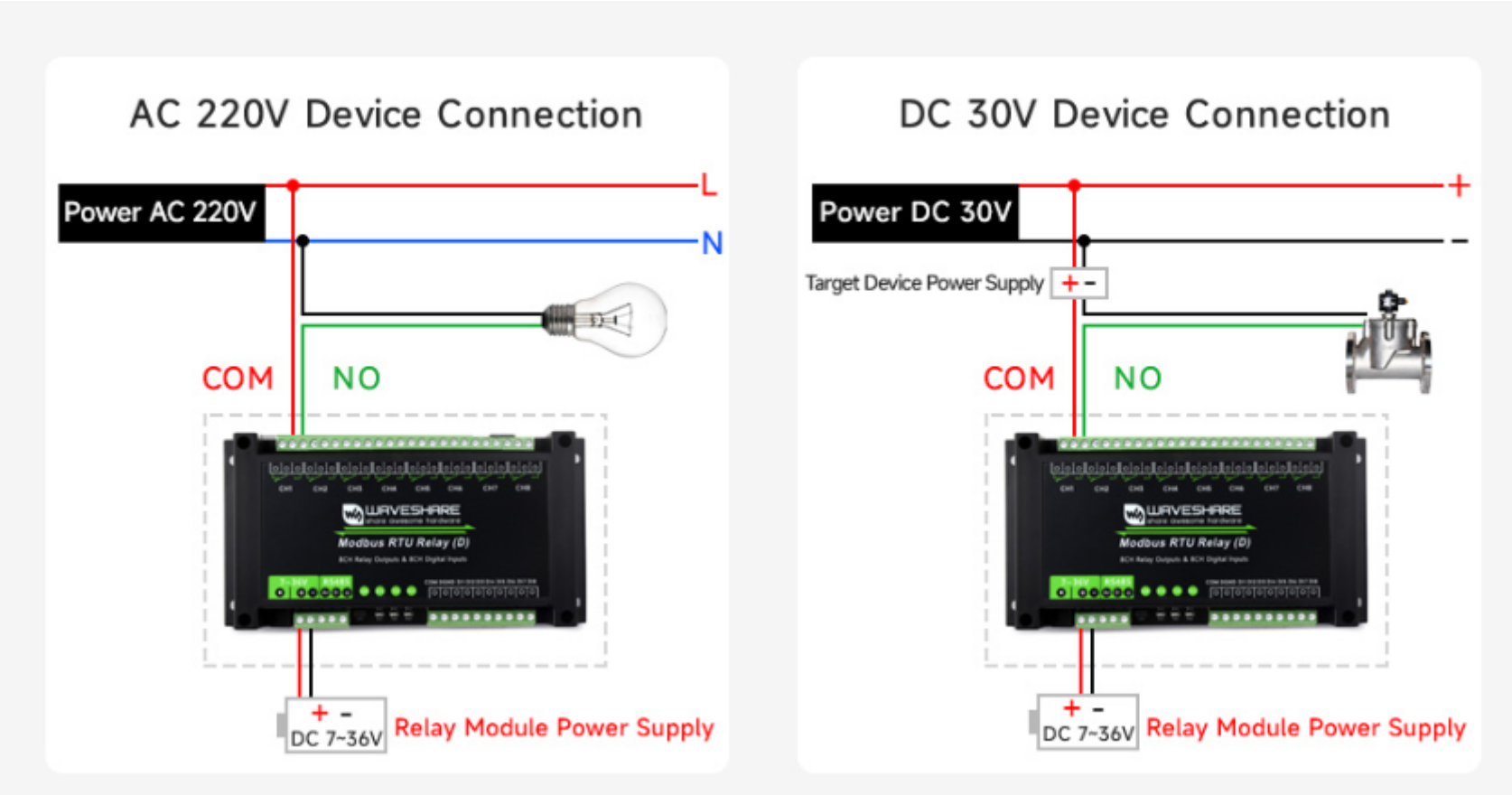


- Active wet contact PNP input



Relay Output Wiring

The device comes with a relay with a contact capacity of 10A 250V AC or 10A 30V DC, which allows direct control of household 220V AC equipment or DC equipment up to 30V.



Operation Modes Introduction

Normal Mode

Command direct control

Sending commands

Power ON/OFF control

Load

Flash ON / Flash OFF

Flash on: sending a command to turn on the output for a while and then turn it off automatically
Flash off: sending a command to turn off the output for a while and then turn it on automatically
Setting time interval by passing argument to the command

Setting flash on/off mode and time interval

Power ON/OFF control

Load

Linkage Control

The output status follows the digital input status

Digital input monitoring

Power ON/OFF control

Load

Toggle Mode

When inputs a pulse, the output status will toggle once

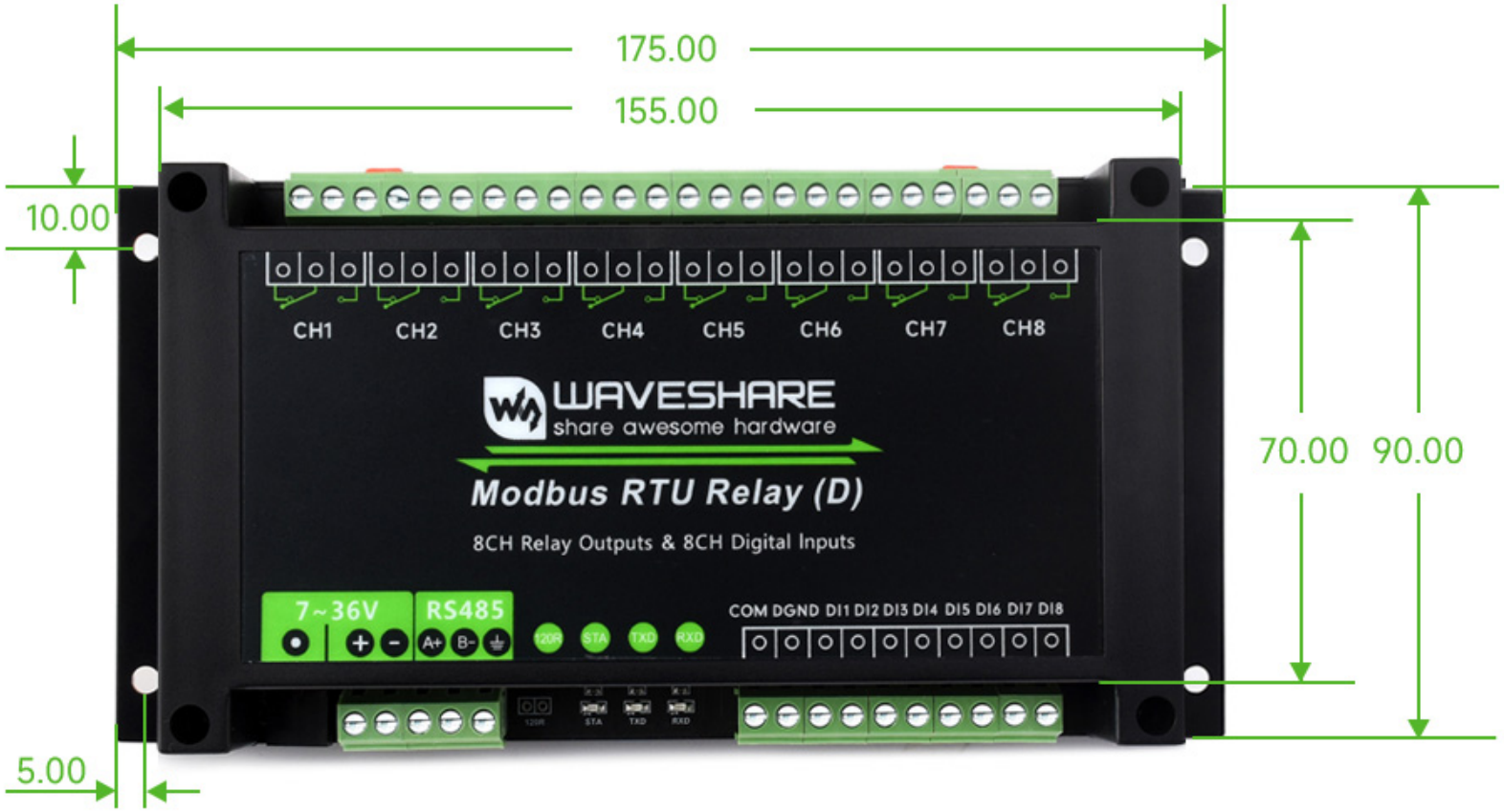
Pulse input monitoring

Power ON/OFF control

Precautions

- The default is the Normal Mode, the relay is only controlled by commands.
- Except for the Normal Mode, other modes support controlling relays via input.
- Except for the Linkage Control, other modes support controlling relays via commands.

Dimensions



Hardware Connection

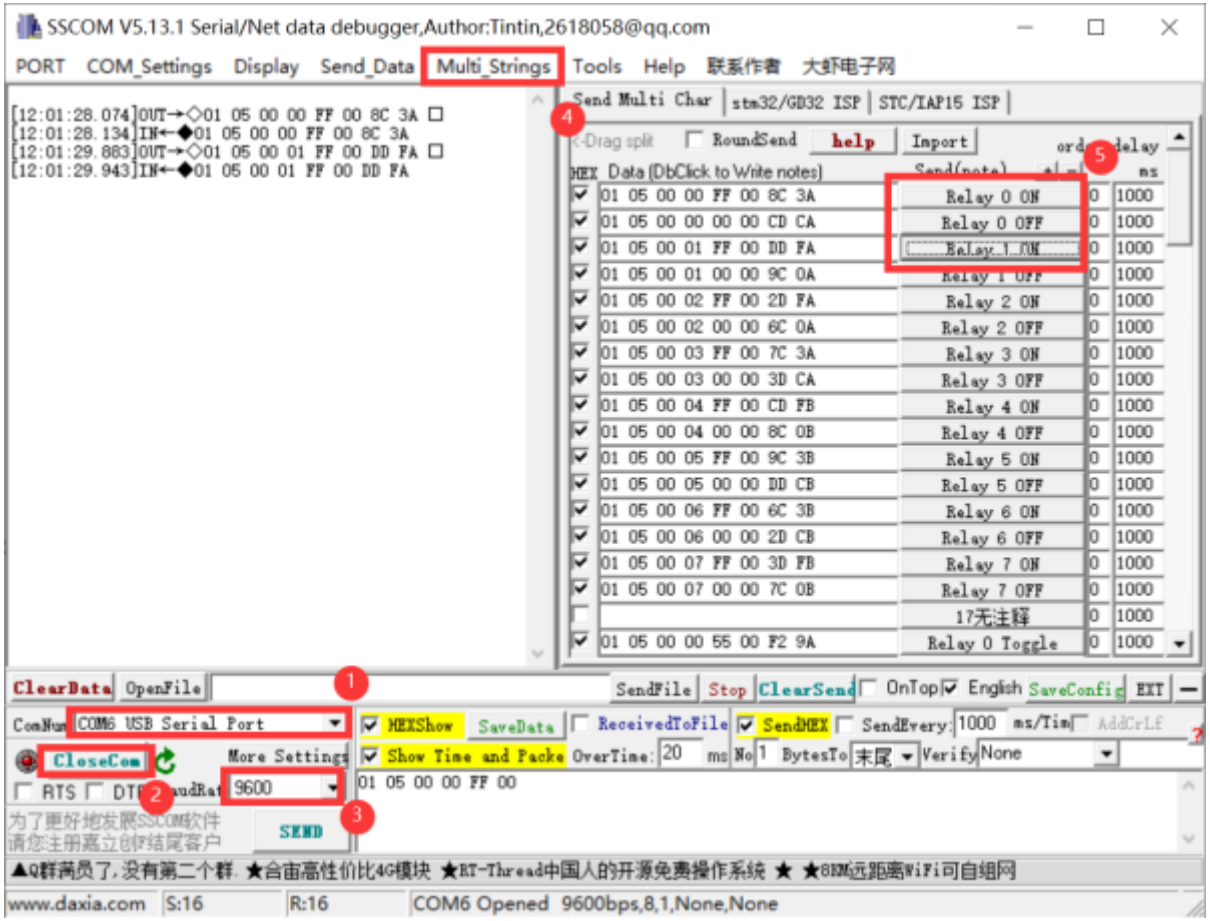
- Connect the USB TO 485 to the target boards via cables, A-A and B-B connected as shown below:

Example Demonstration

The demo shows how the following two software operate.
SSCOM serial port debugging assistant is more convenient to operate, free of installation, and more convenient for complete display and analysis of instructions, but the disadvantage is that the data is not intuitive.
Modbus Poll software is directly operated on the register, and the data display is more convenient to observe, but the disadvantage is that the instruction is not displayed completely, so you need to be familiar with the Modbus register operation.
You can test using any method. It is recommended to use the SSCOM serial port debugging assistant software for the first test.

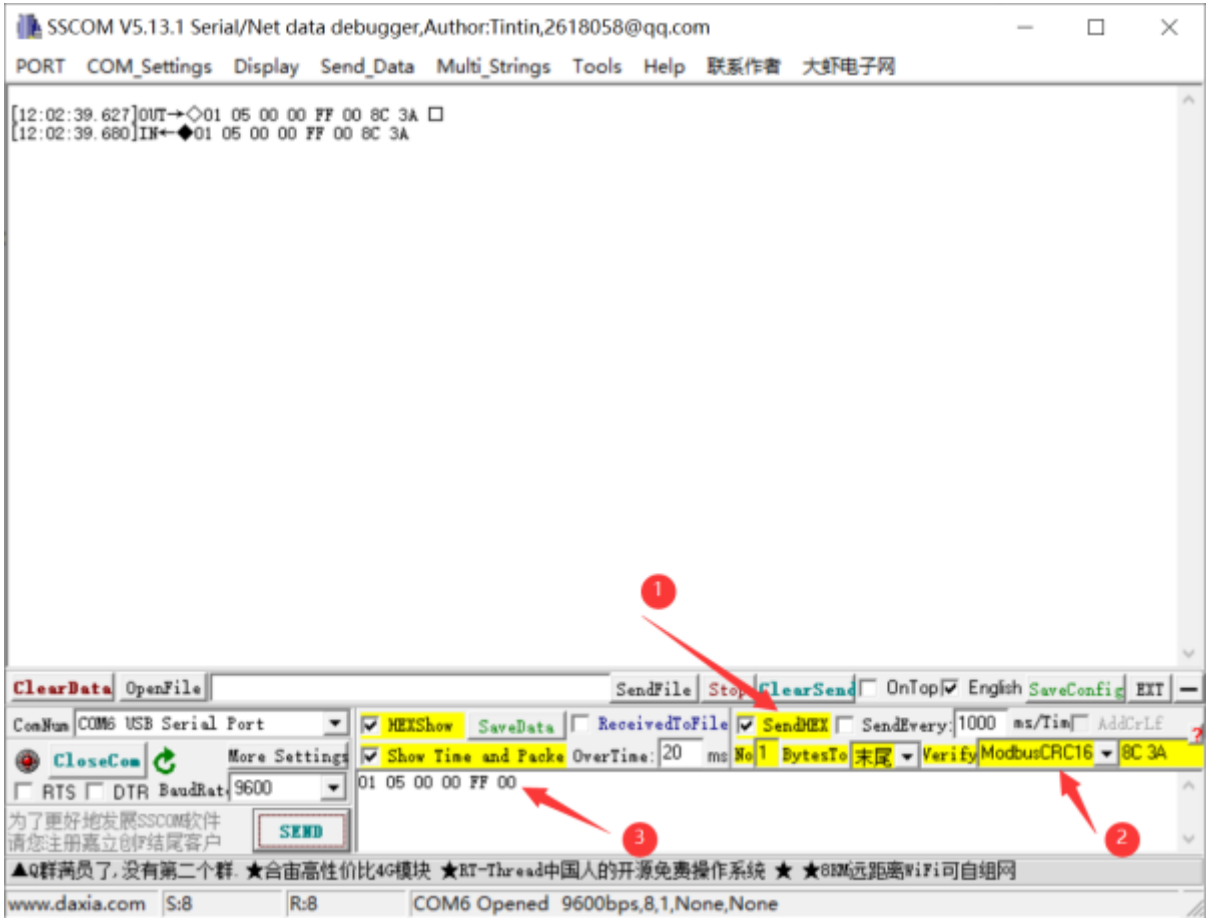
SSCOM Serial Port Debugging Assistant

- 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-Char to open the Send Multi-Char window, and click the function to send the corresponding command.



Note: The module defaults to normal mode at the factory, the relay can be directly commanded and controlled. If there is a command to return normally, but the relay does not act, the module may have been changed to other control mode, you can query it by reading the relay control mode command.

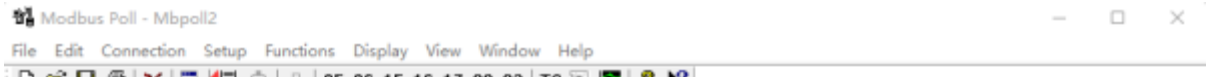
- If you need to send other commands, choose SendHEX. For checksum validation, select ModbusCRC16. After entering the first six bytes of the command, clicking SEND will automatically add the CRC check code.

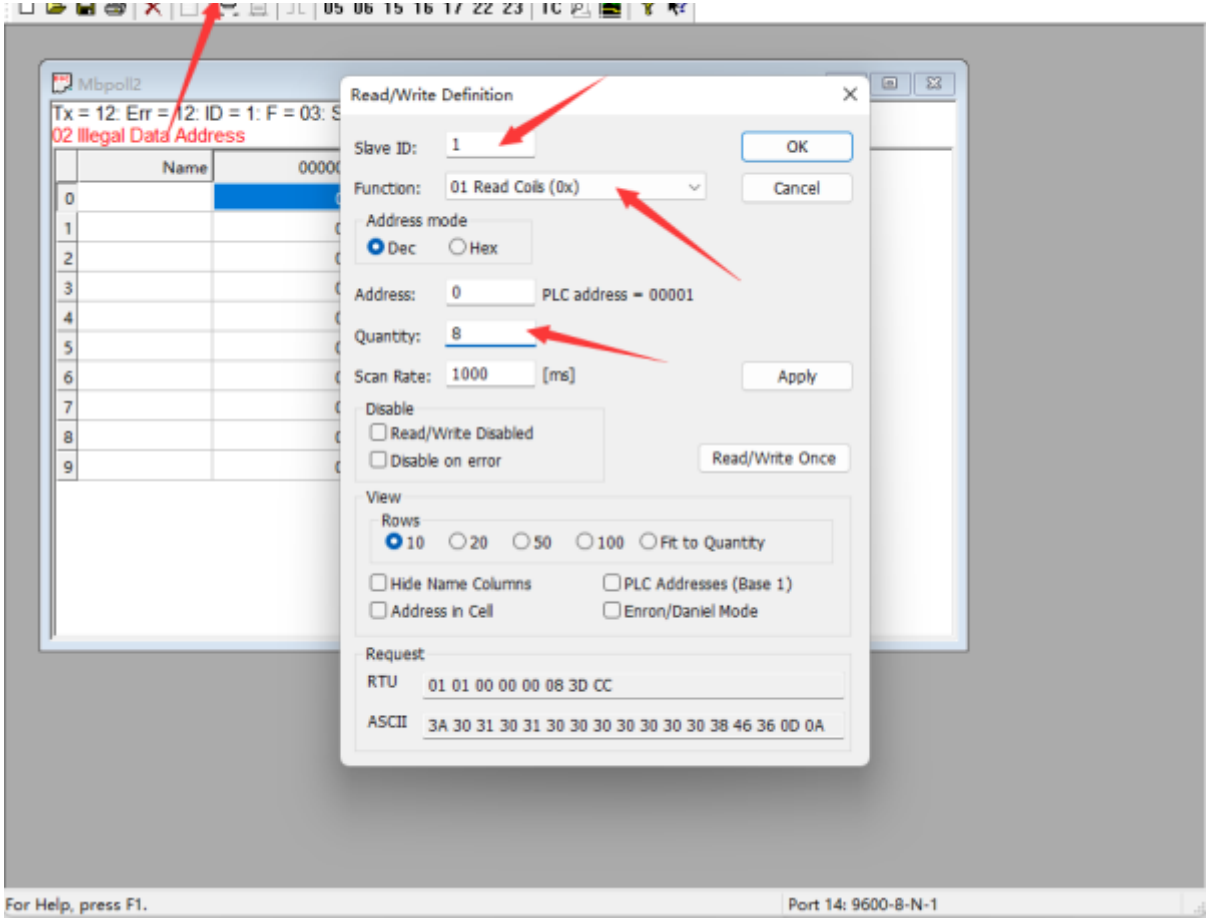


- For detailed control commands, please see the development protocol.

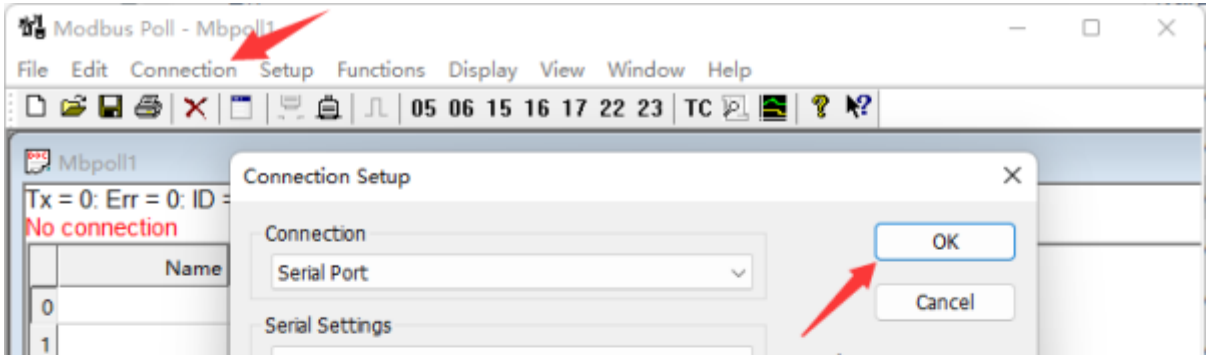
Modbus Poll Software

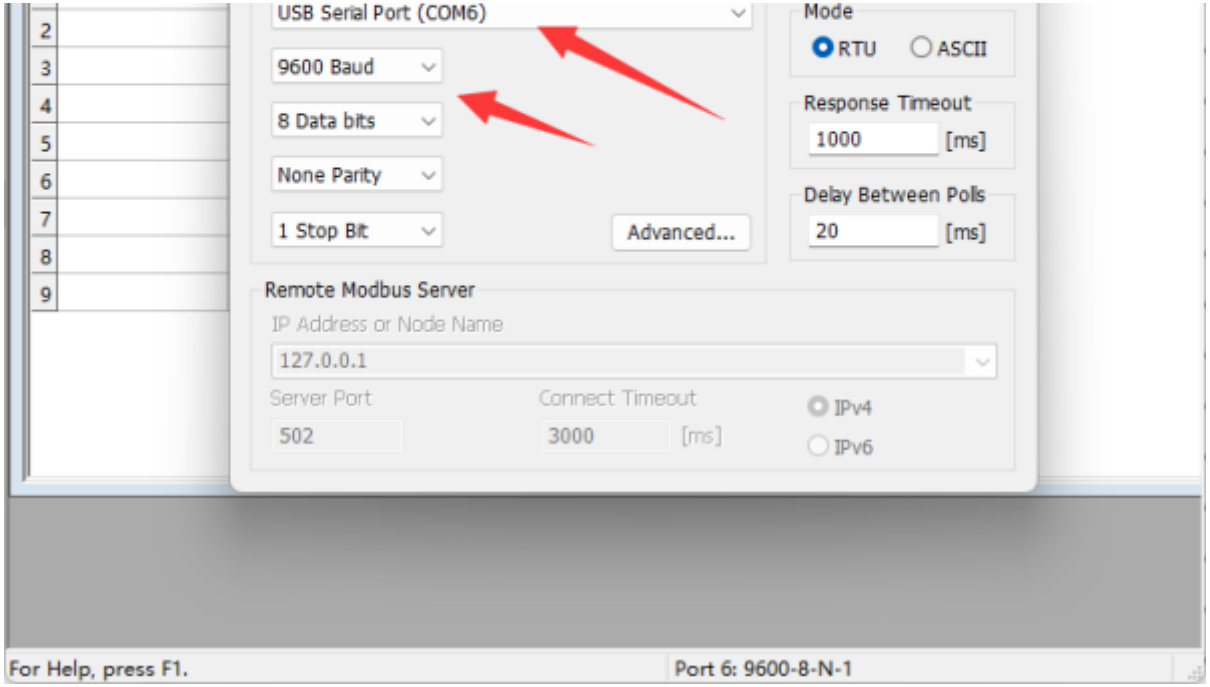
- The serial port software is not convenient to observe the data, you can choose Modbus Poll software to read the data. Download and install the Modbus Poll software.
- Open the software, select Setup->Read/Write Definition, select the actual device address for Slave ID, select 01 Read Coils function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



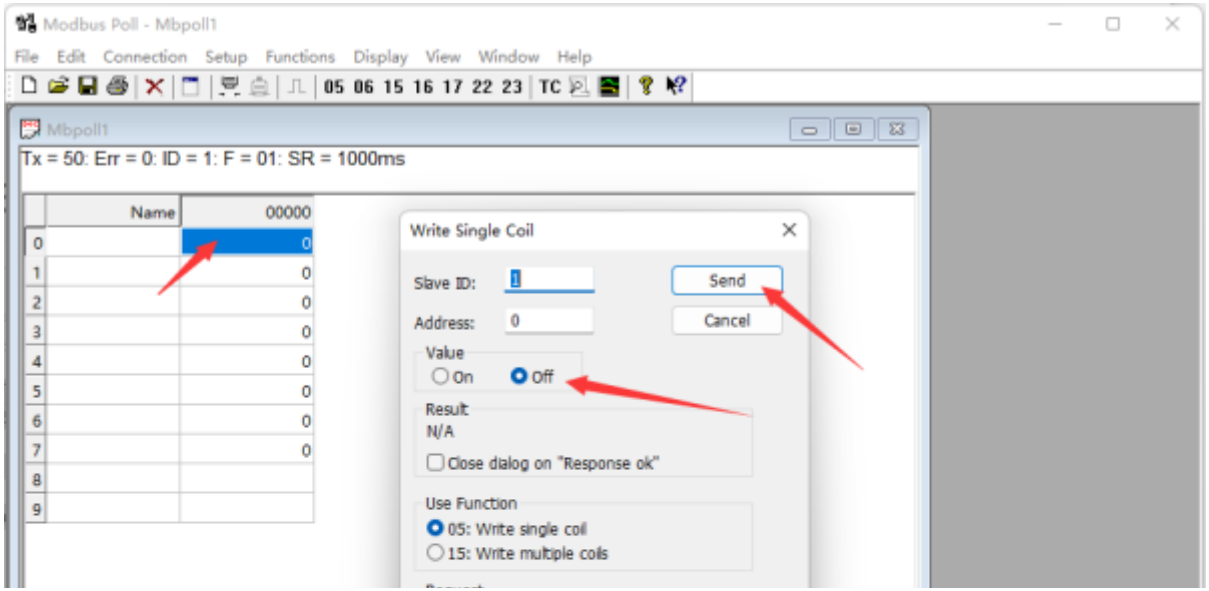


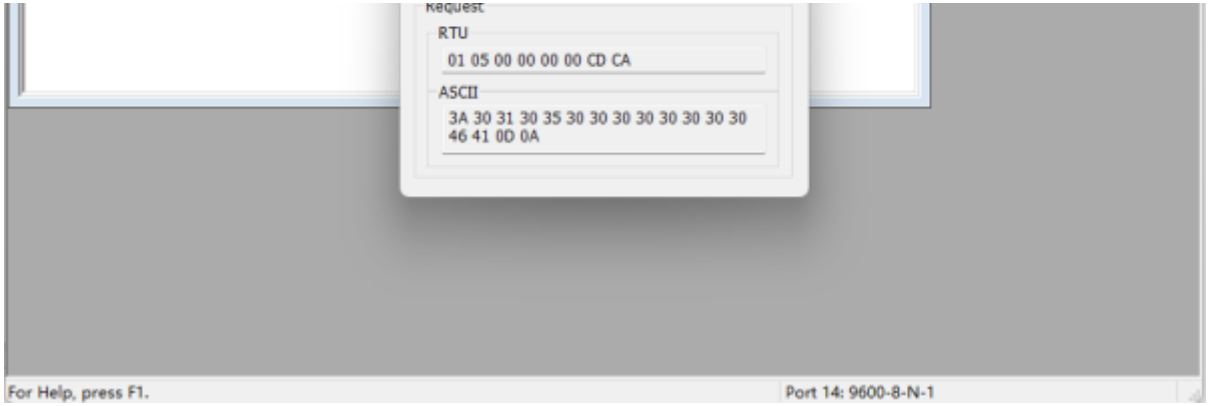
- Select Connection->Connect..., choose the corresponding serial port, set the baud rate to 9600, and select 8 Data bits and None Parity. Click OK to connect.



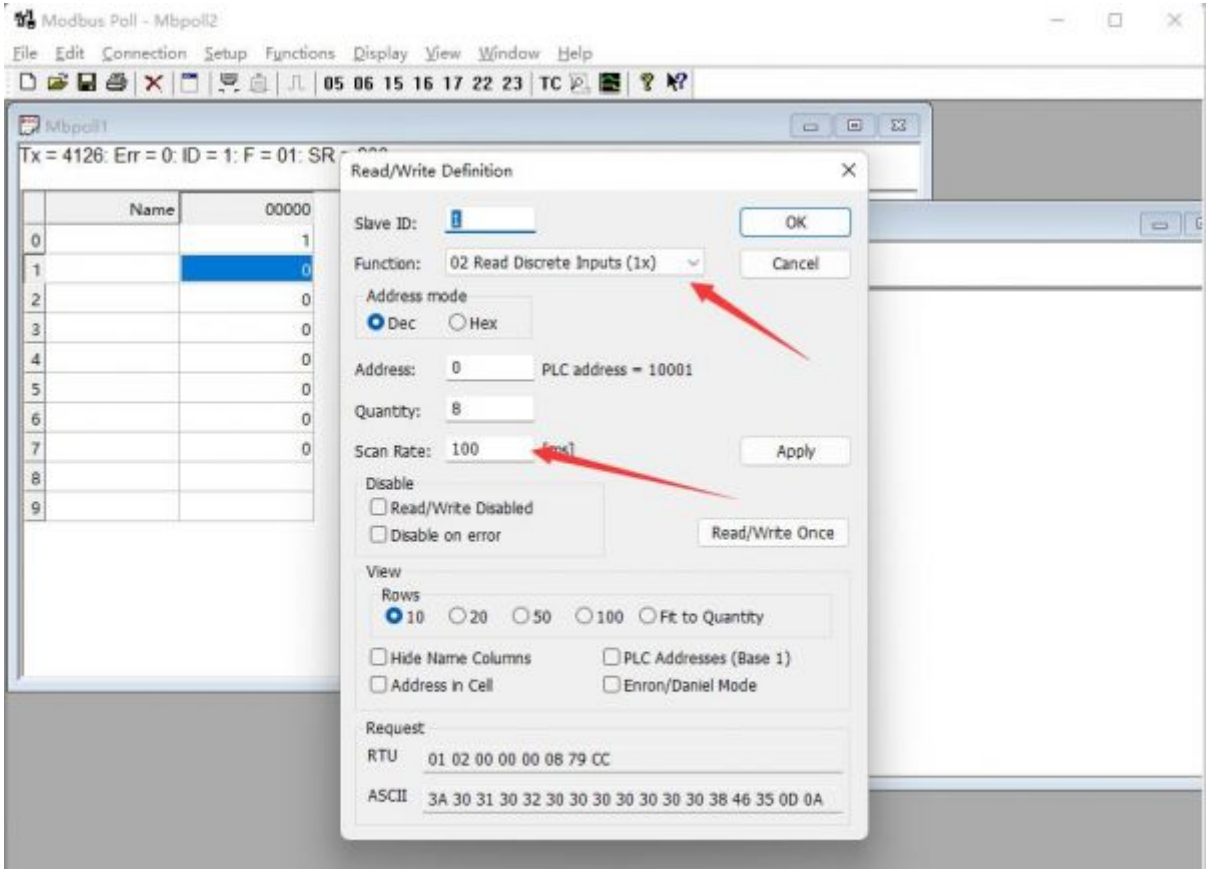


- After the connection is normal, you can check the current relay status. Select the corresponding channel, then double-click the status value to pop up the send page. Choose On or Off, then Click Send to control the relay opening and closing.

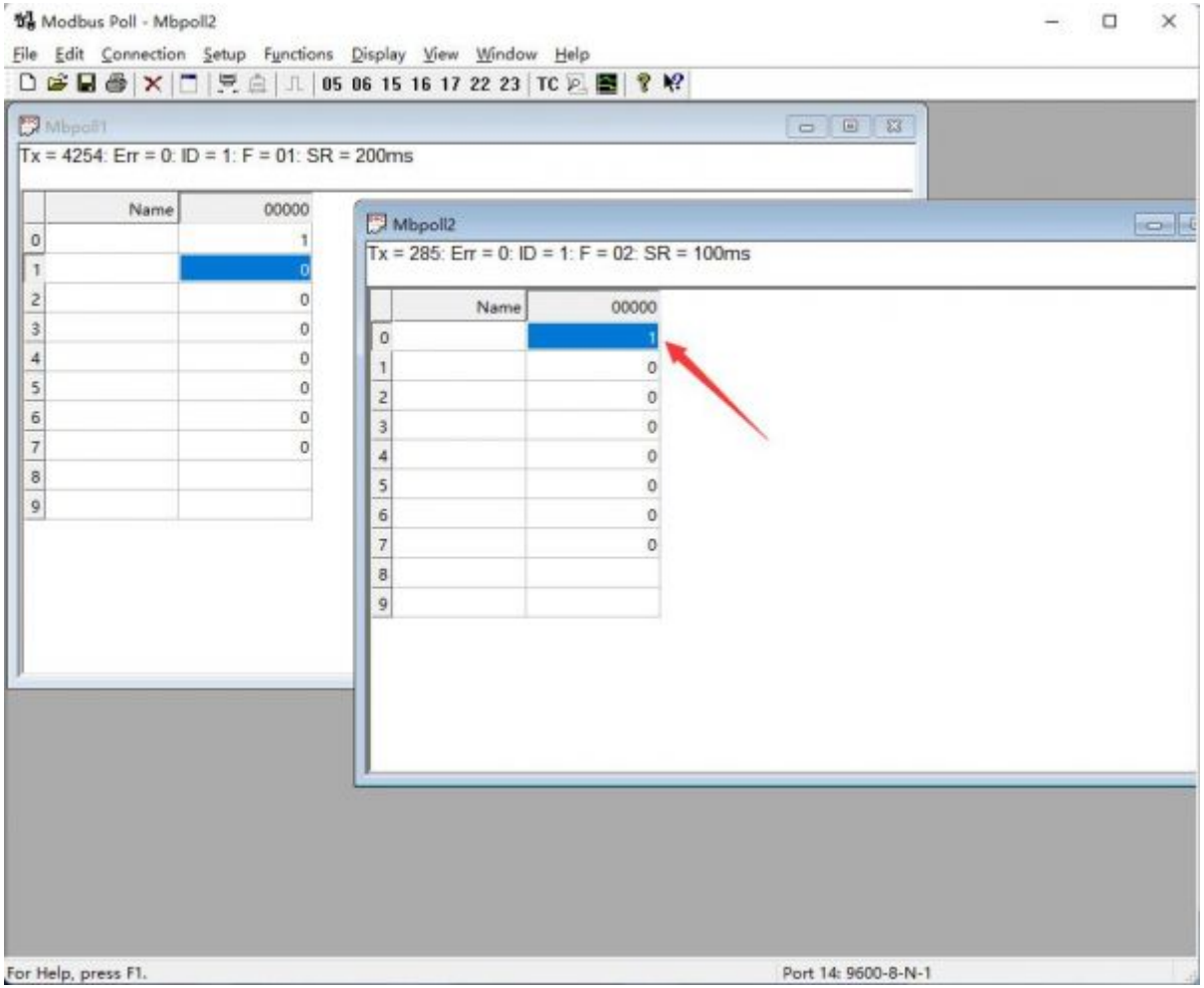




- Choose File->New to create a new window, select Setup->Read/Write Definition, choose the actual device address for Slave ID, select 02 for Function, set Address to 0, set Quantity to 8 channels, and change the Scan Rate to 100ms. Click OK to confirm.



- The new window 2 can display the current input status, change the input interface level, and the corresponding values will also change.



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 Port, select Yes to open the hardware serial port

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:

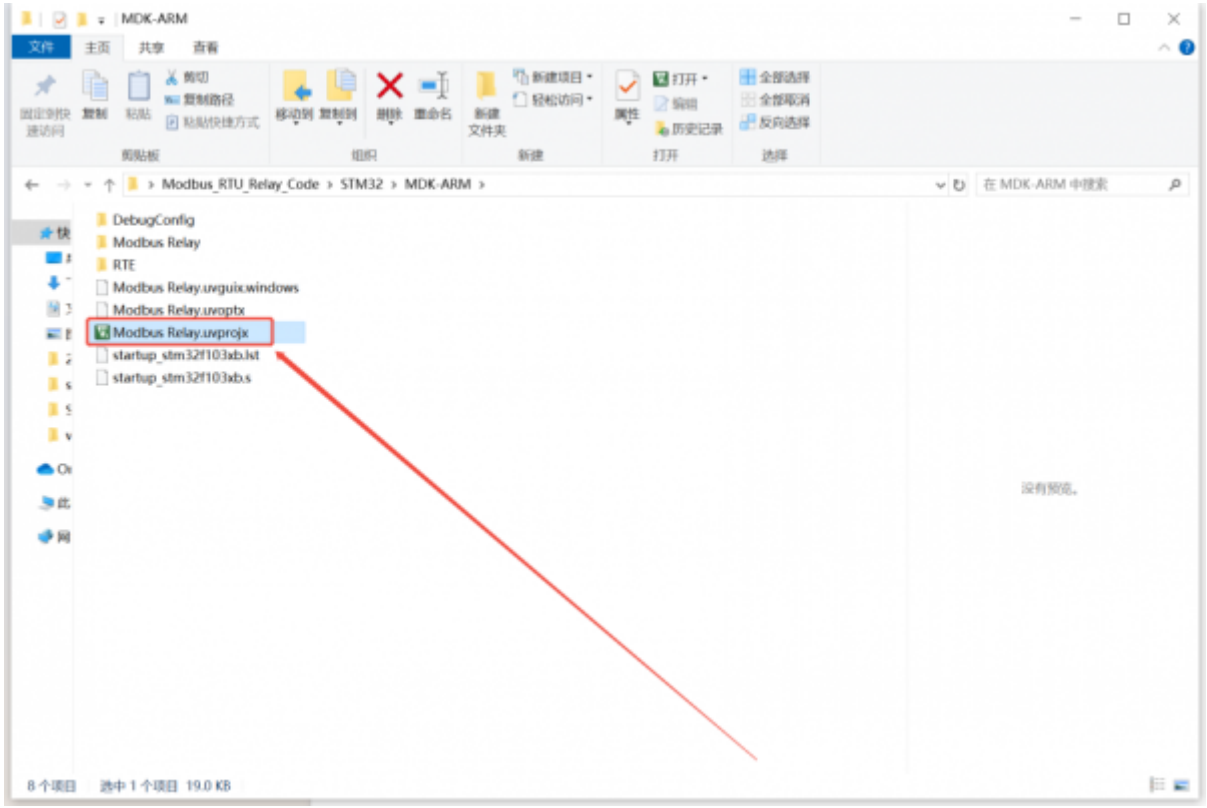
sudo apt-get install unzip
wget wget https://files.waveshare.com/wiki/Modbus-RTU-Relay-D/Modbus_RTU_Relay_D_Code.zip
unzip Modbus_RTU_Relay_D_Code.zip
cd Modbus_RTU_IO_Code/Python3
sudo python3 main.py

After the demo normally runs, each channel can be opened and closed sequentially, and finally the current input status is displayed.

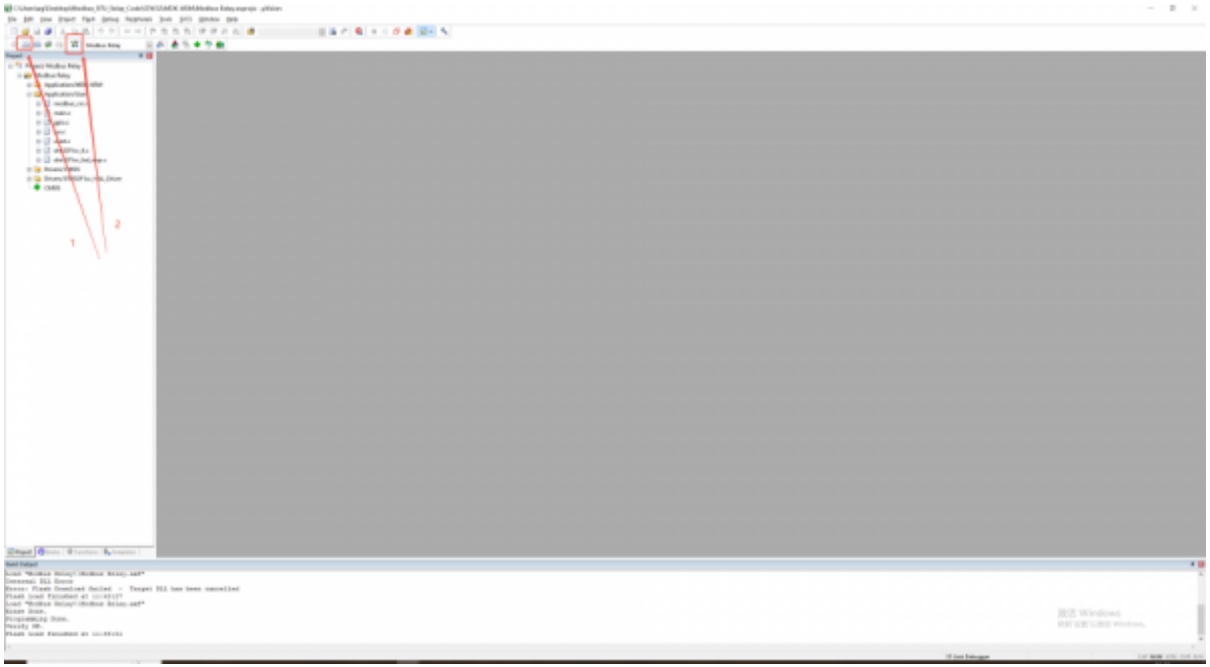
STM32

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_D_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.

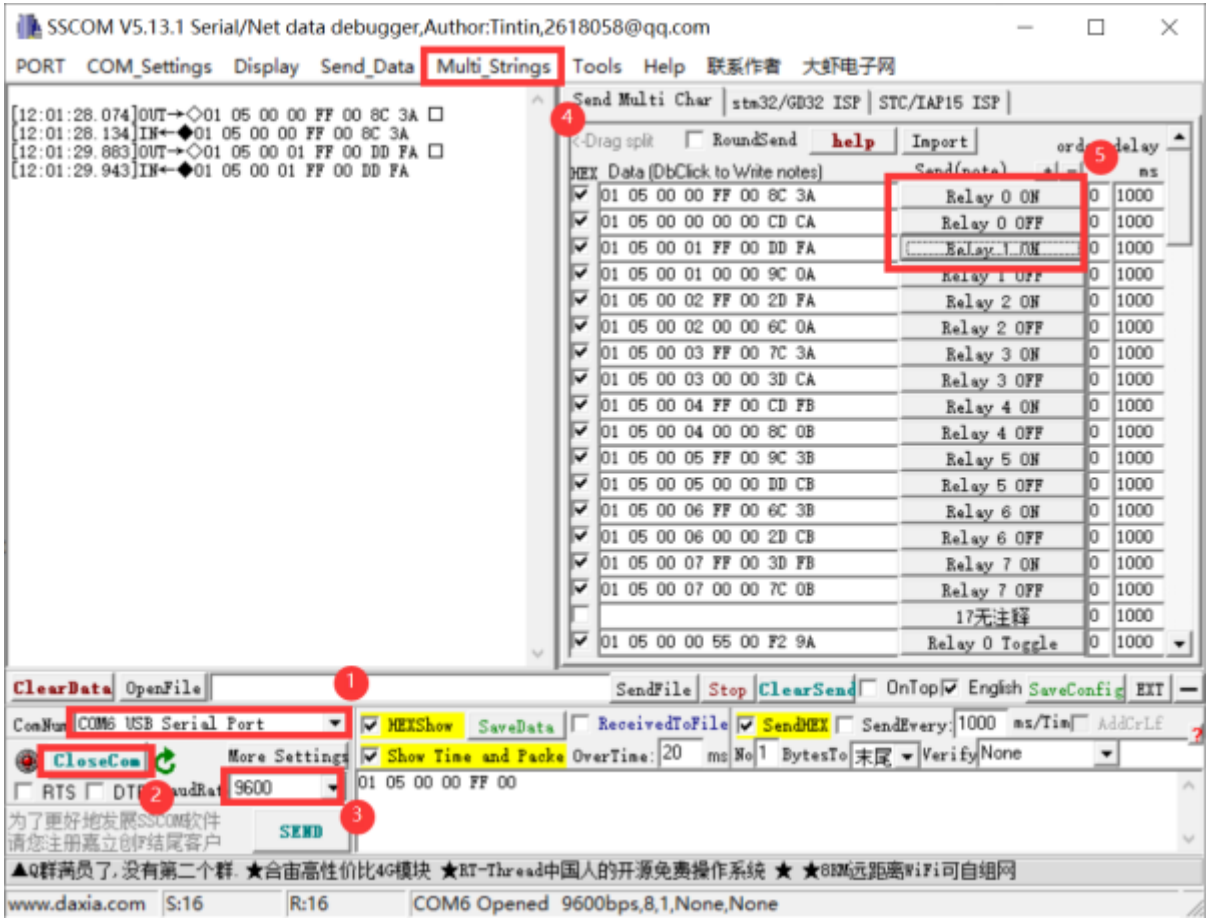


2. Connect the STM32 to a computer via the STM32 download and debug probe. Compile and download the program to the development board.



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.

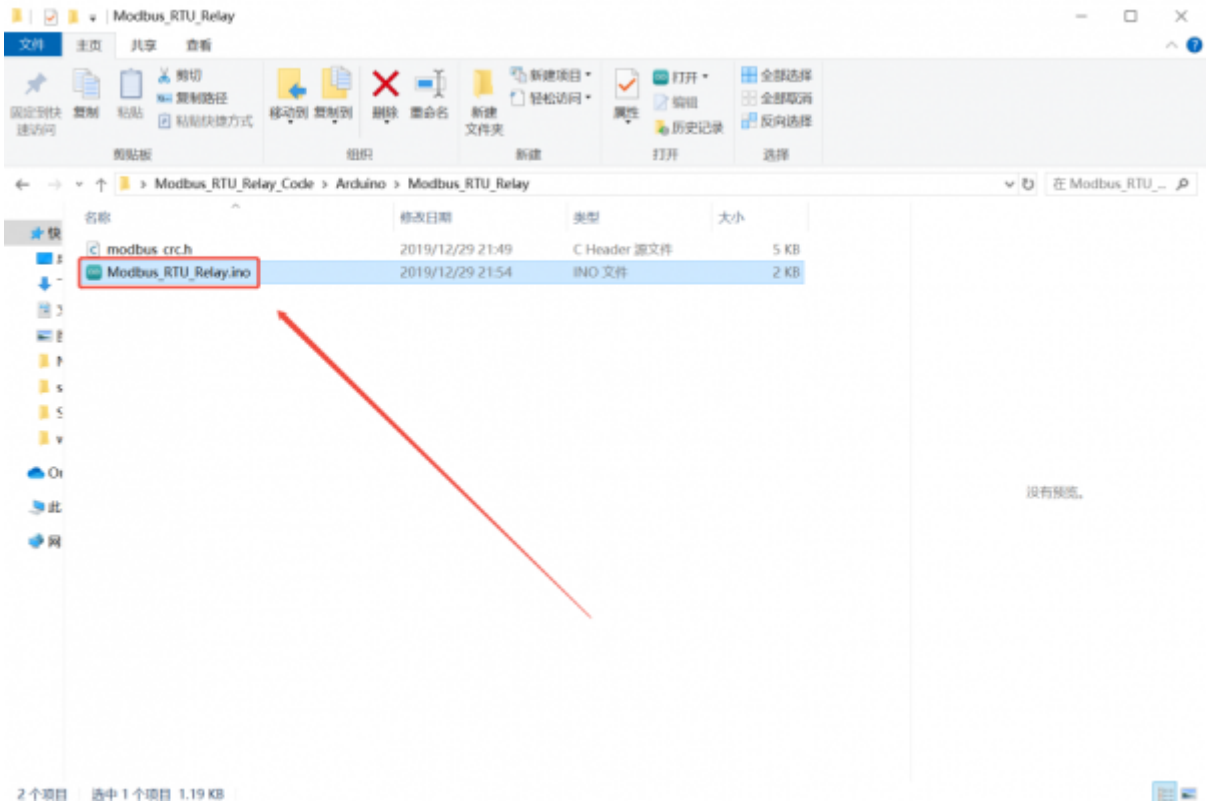
4. After powering on, the serial port will output the transmitted commands, which can be observed using a serial port assistant. Meanwhile, the relays will sequentially turn on (e.g., 1→2→3→4). Once all relays are fully activated, they will then sequentially turn off (e.g., 1→2→3→4). You can observe whether the relay is normally engaged through the LED indicator light.



Arduino

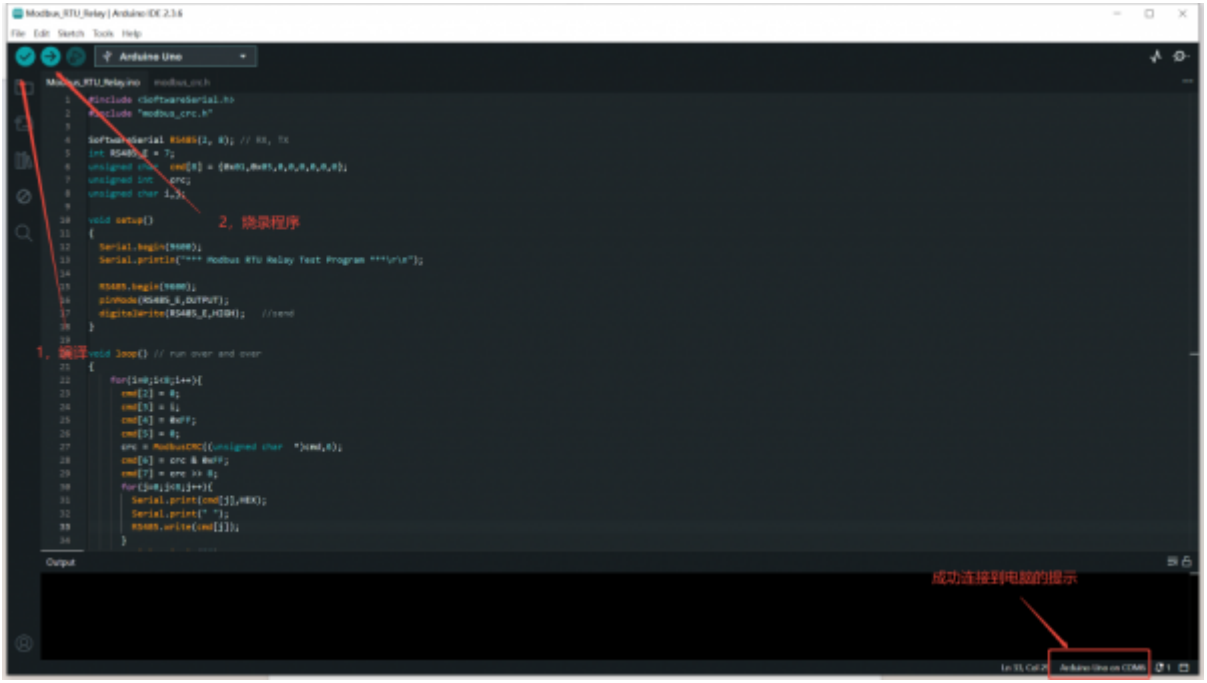
Note: The Arduino demo is based on the UNO PLUS and RS485 CAN Shield module.

1. Download Demo, find the Arduino project file Modbus_RTU_Relay.ino in the path Modbus_RTU_Relay_D_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.



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.

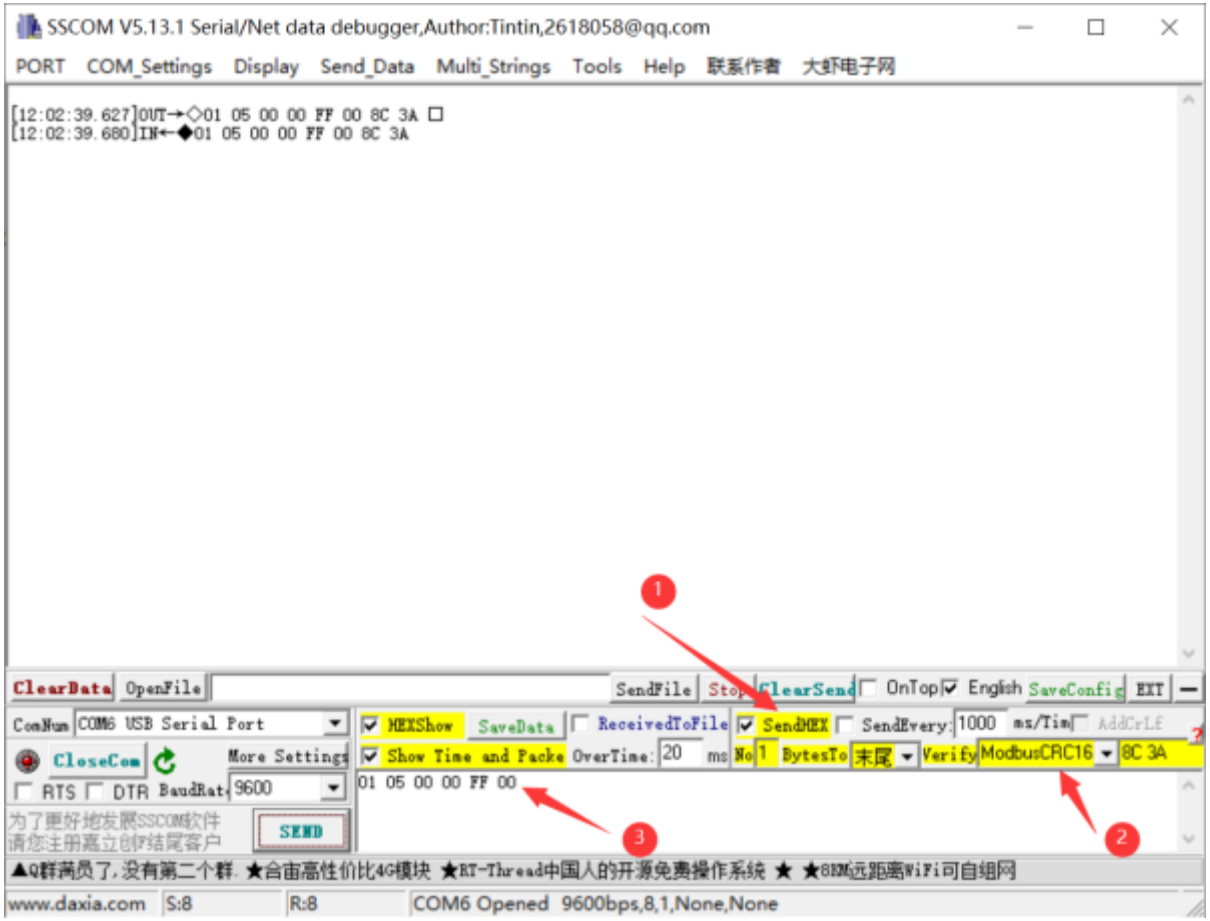
3. After seeing the prompt to connect to the computer in the lower right corner, click to compile and flash the program, and wait for the flashing to complete.



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 transmitted commands, which can be observed using a serial port assistant. Meanwhile, the relays will sequentially turn on (e.g., 1→2→3→4). Once all relays are fully activated, they will then sequentially turn off (e.g., 1→2→3→4). You can observe whether the relay is normally engaged through the LED indicator light.

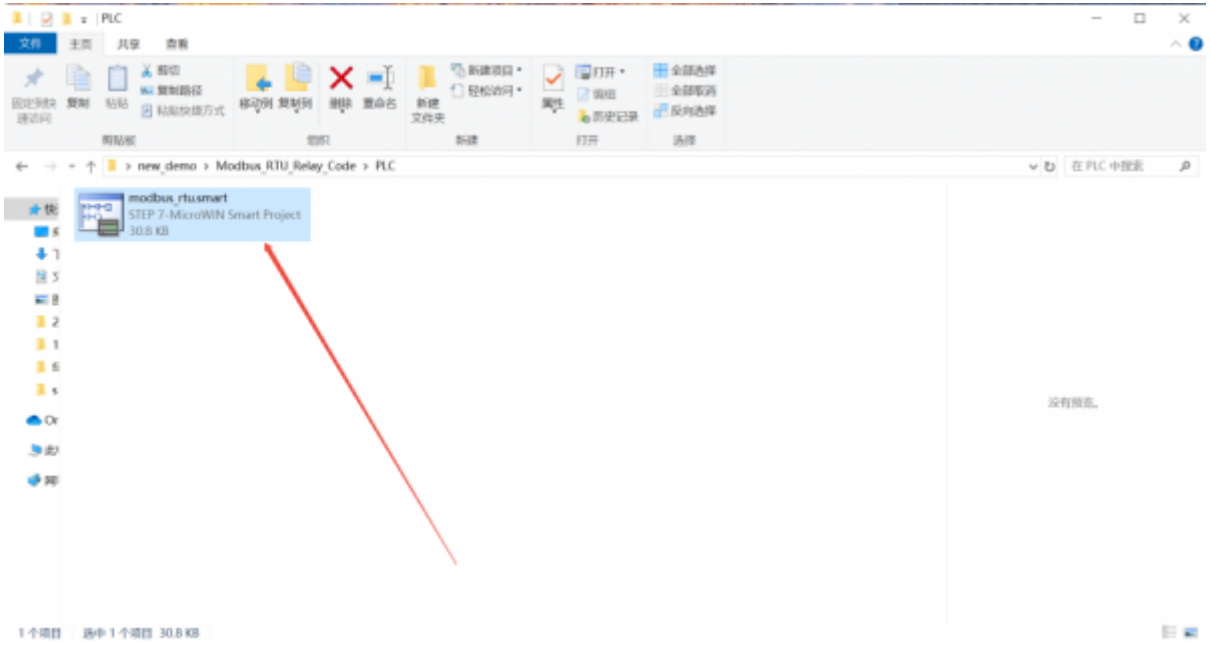
Note: The data in the figure below is in ASCII code containing spaces and line breaks, and is not the actual data transmitted.



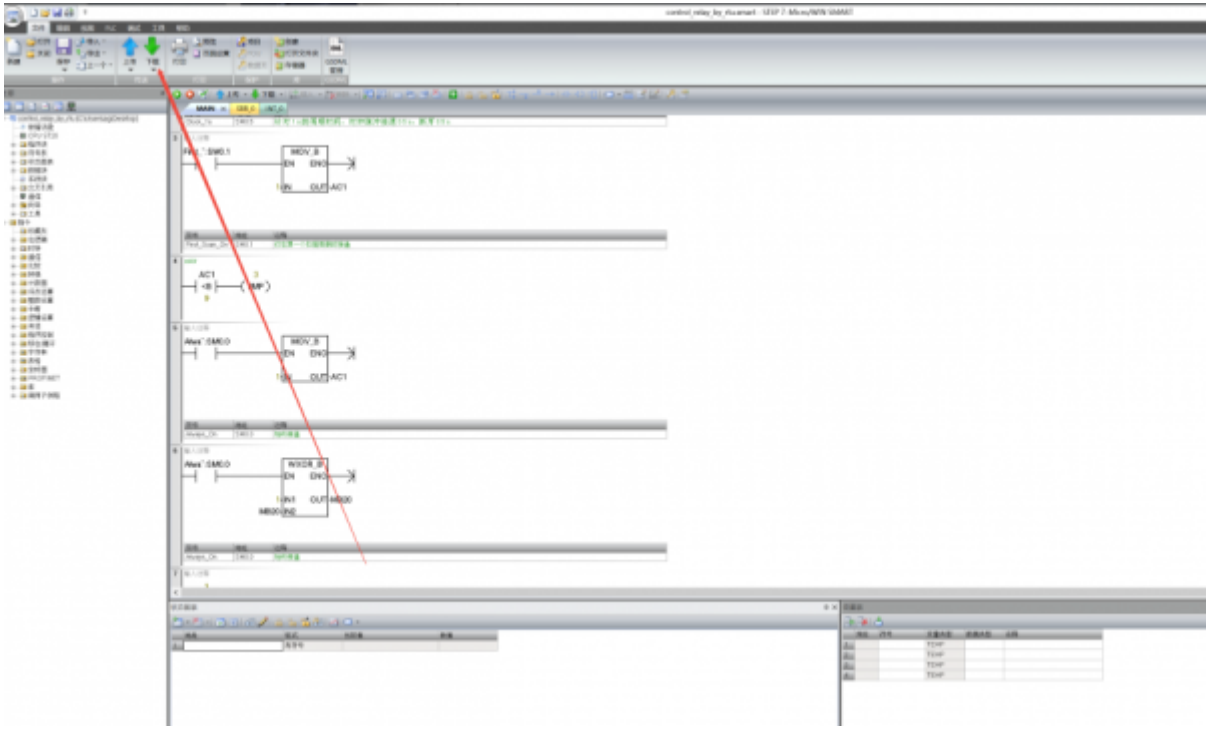
PLC

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

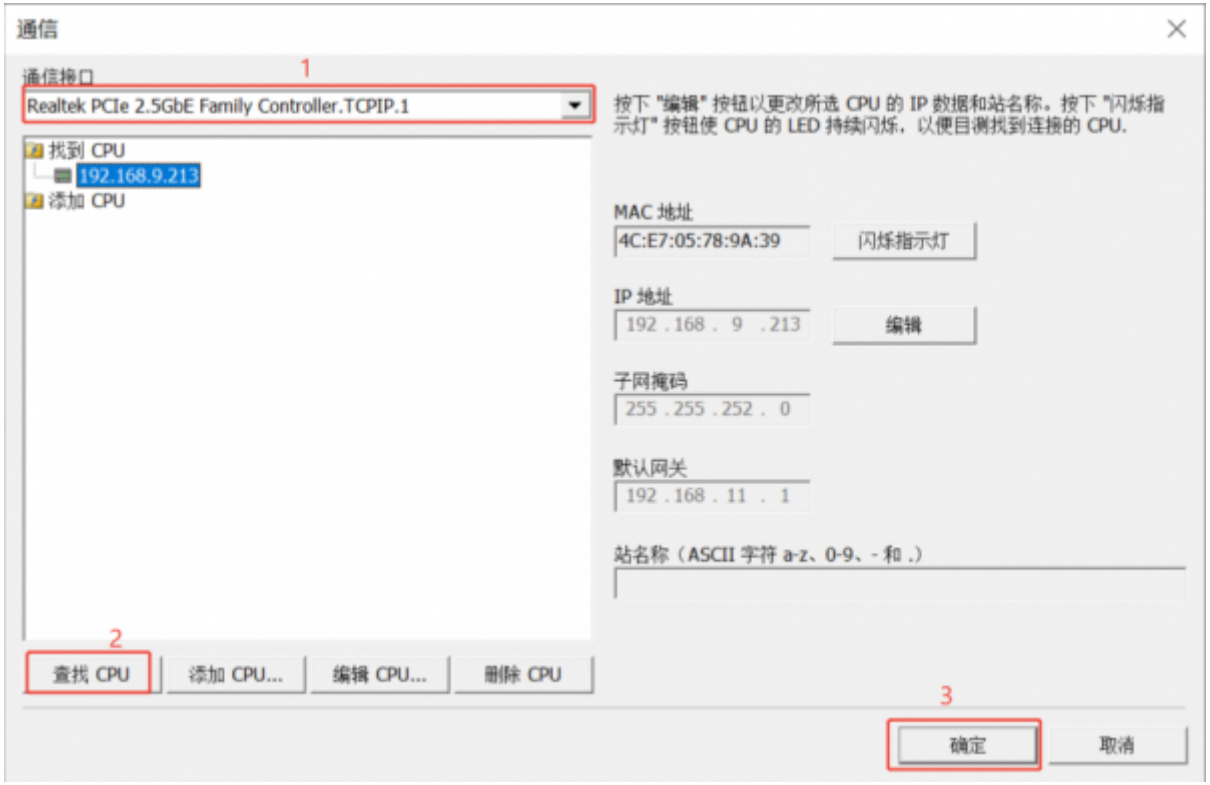
1. Download Demo, find the PLC project file modbus_rtu.smart in the path Modbus_RTU_Relay_Code\PLC, and double-click to open the PLC project file. Note that you should ensure STEP 7-MicroWIN SMART software is installed on your computer before using it.



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



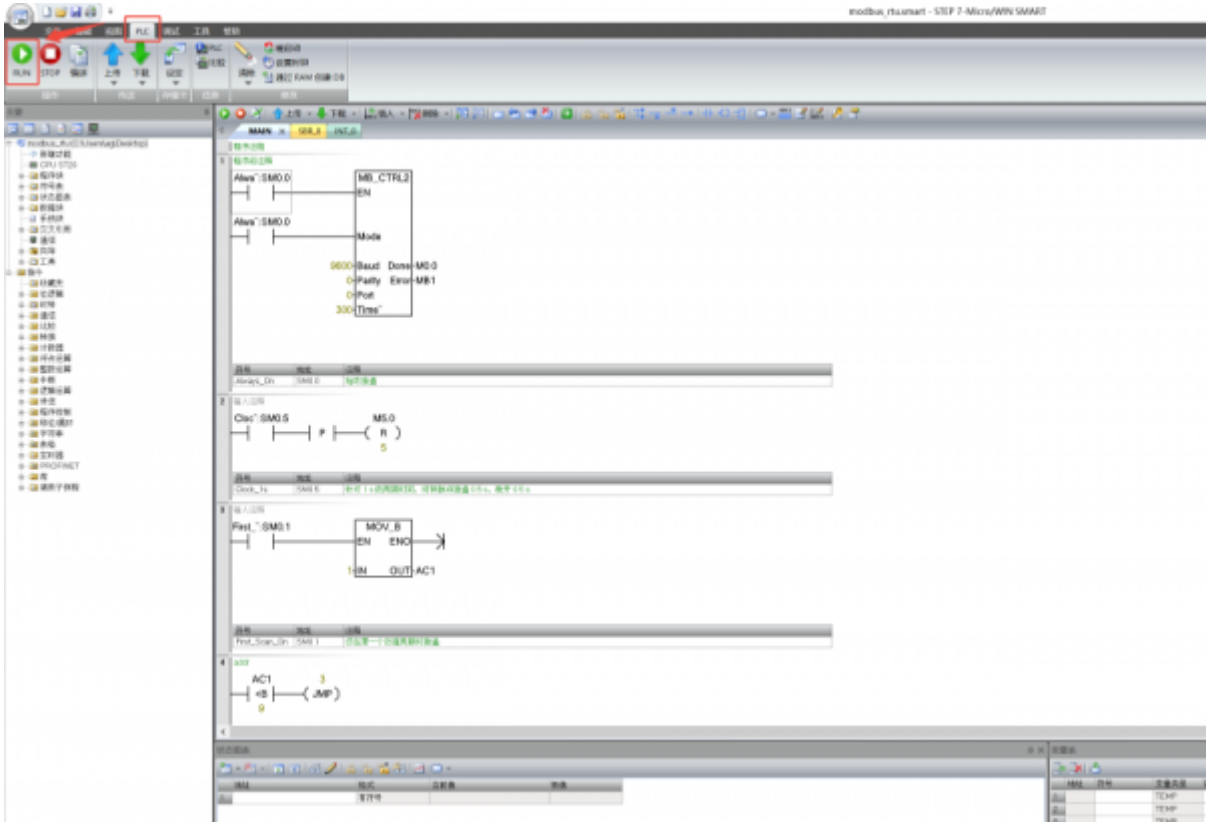
3. Select the communication interface in the communication popup, find your device, and click Confirm



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

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 sequentially turn on (e.g., 1→2→3→4). Once all relays are fully activated, they will then sequentially turn off (e.g., 1→2→3→4). You can observe whether the relay is normally engaged through the LED indicator light.

Function Code Introduction

Function Code	Description
01	Read relay status
02	Read input status
03	Read save register
05	Write single relay

06	Set single register
0F	Write multiple relays
10	Set multiple registers

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 Registers	0xFF00: all relays on 0x0000: all relays off 0x5500: all relays toggle	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
1x0000 …… 1x0007	Channels 1~8 input addresses	Indicates channel 1~8 input channel status	Read	0x02
4x1000 …… 4x1007	Channels 1~8 relay control mode	0x0000~0x0003 four control modes	Read/Write	0x03, 0x06, 0x10

4x2000	UART Parameters	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

Operation 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 01 94 0B

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	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	Command	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	Command	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

Read Input Status

Send code: 01 02 00 00 00 08 79 CC

Field	Description	Note
01	Device Address	0x00 indicates the broadcast address, 0x01-0xFF indicates the device address
02	02 Command	Read input status
00 00	Input Start Address	The address of the input register, 0x0000 - 0x0007
00 08	Register Number	The number of the input channels to be read, which must not exceed the maximum number of the input
79 CC	CRC16	The CRC16 checksum of the first 6 bytes of data

Receive code: 01 02 01 00 A1 88

Field	Description	Note
01	Device Address	0x00 indicates the broadcast address, 0x01-0xFF indicates the device address
02	02 Command	Read input status
01	Byte Number	The number of all bytes of the returned status information

00	Query status	Received input channel status Bit0: the first channel status; Bit1: the second channel status; And so on, with the idle high bit being zero
A1 88	CRC16	The CRC16 checksum of the first 6 bytes of data

For example: [Address 1 device]

Send: 01 02 00 00 00 08 79 CC
Receive: 01 01 01 00 51 88 //Inputs are all untriggered
Send: 01 02 00 00 00 08 79 CC
Receive: 01 02 01 01 60 48 //Channel 1 input is triggered, and the rest of channels are not triggered
Send: 01 02 00 00 00 08 79 CC
Receive: 01 02 01 41 61 B8 //Channel 1 and 7 input are triggered, and the rest of channels are not triggered

Read Relay Control Mode

Send code: 01 03 10 00 00 08 40 CC

Field	Description	Note
01	Device Address	0x00 indicates the broadcast address, 0x01-0xFF indicates the device address
03	03 Command	Read Holding Register
10 00	Register Start Address	0x1000 - 0x1007 corresponds to 1~8 input channels
00 08	Register Number	The number of the registers to be read, which must not exceed the maximum number of the input channels
40 CC	CRC16	The CRC16 checksum of the first 6 bytes of data

Receive code: 01 03 10 00 00 00 00 00 00 00 00 00 00 00 00 00 00 E4 59

Field	Description	Note

01	Device Address	0x00 indicates the broadcast address, 0x01-0xFF indicates the device address
03	03 Command	Read Holding Register
10	Byte Number	The number of all bytes of the returned status information
00 00 …… 00 00	Control Mode	Indicates relay 1 - 8 control mode, 0x0000~0x0003 indicate four control modes 0x0000: **`Normal mode`**, the relay is directly controlled by commands; 0x0001: **`Linkage mode`**, relay status is the same as the corresponding input channel status; 0x0002: **`Toggle mode`**, the corresponding relay toggles once when the input channel inputs a pulse; 0x0003: **`Edge Trigger Mode`**, the corresponding relay status changes once when the input channel level changes once **Note:** All modes except Linkage mode support relay control by command.
E4 59	CRC16	The CRC16 checksum of the first 6 bytes of data

For example: [Address 1 device]

Read relay 1-8 control mode: 01 03 10 00 00 08 40 CC
Read relay 1 control mode: 01 03 10 00 00 01 80 CA
Read relay 2 control mode: 01 03 10 01 00 01 D1 0A
Read relay 3-5 control mode: 01 03 10 02 00 03 A0 CB

Set Single Relay Control Mode

Send code: 01 06 10 00 00 01 4C CA

Field	Description	Note
01	Device Address	0x00 indicates the broadcast address, 0x01-0xFF indicates the device address
06	06 Command	Write single register

10 00	Register Start Address	0x1000 - 0x1007 corresponds to relay control modes for channels 1~8
00 01	Control Mode	Indicates relay 1 - 8 control mode, 0x0000~0x0003 indicate four control modes 0x0000: **`Normal mode`**, the relay is directly controlled by commands; 0x0001: **`Linkage mode`**, relay status is the same as the corresponding input channel status; 0x0002: **`Toggle mode`**, the corresponding relay toggles once when the input channel inputs a pulse; 0x0003: **`Edge Trigger Mode`**, the corresponding relay status changes once when the input channel level changes once **Note:** All modes except Linkage mode support relay control by command.
4C CA	CRC16	The CRC16 checksum of the first 6 bytes of data

Return code: 01 06 10 00 00 01 4C CA

Field	Description	Note
01	Device Address	0x00 indicates the broadcast address, 0x01-0xFF indicates the device address
06	06 Command	Write single register
10 00	Register Start Address	0x1000 - 0x1007 correspond to relay control modes for channels 1~8
00 01	Control Mode	Relay control modes, 0x0000~0x0003 indicate four control modes
4C CA	CRC16	The CRC16 checksum of the first 6 bytes of data

For example: [Address 1 device]

Set relay 1 as Linkage mode: 01 06 10 00 00 01 4C CA

Set relay 2 as toggle mode: 01 06 10 01 00 02 5D 0B

Set Multiple Relay Control Mode

Send code: 01 10 10 00 00 08 10 00 01 00 01 00 01 00 01 00 01 00 01 00 01 00 01 7C B1

Field	Description	Note
01	Device Address	0x00 indicates the broadcast address, 0x01-0xFF indicates the device address
10	10 Command	Write multiple registers
10 00	Register Start Address	0x1000 - 0x1007 correspond to relay control modes for channels 1~8
00 08	Register Number	Set register number, which must not exceed the maximum number of the input channels
10	Byte Number	Set the byte number of the input
00 01 …… 00 01	Control Mode	Indicates relay 1 - 8 control mode, 0x0000~0x0003 indicate four control modes 0x0000: **`Normal mode`**, the relay is directly controlled by commands; 0x0001: **`Linkage mode`**, relay status is the same as the corresponding input channel status; 0x0002: **`Toggle mode`**, the corresponding relay toggles once when the input channel inputs a pulse; 0x0003: **`Edge Trigger Mode`**, the corresponding relay status changes once when the input channel level changes once **Note:** All modes except Linkage mode support relay control by command.
7C B1	CRC16	The CRC16 checksum of the first 6 bytes of data

Return code: 01 10 10 00 00 08 C5 0F

Field	Description	Note
01	Device Address	0x00 indicates the broadcast address, 0x01-0xFF indicates the device address
10	10 Command	Write multiple registers
10 00	Register Start Address	0x1000 - 0x1007 correspond to relay control modes for channels 1~8

00 08	Register Number	Set register number, up to 8 channels
C5 0F	CRC16	The CRC16 checksum of the first 6 bytes of data

For example: [Address 1 device]

Set channel 1-8 relay as Linkage mode: 01 10 10 00 00 08 10 00 01 00 01 00 01 00 01 00 01 00 01 00 01 00 01 7C B1

Set channel 3-5 relay as Toggle mode: 01 10 10 02 00 03 06 00 02 00 02 00 02 4A 4B

Set Baudrate Command

Send code: 00 06 20 00 00 05 43 D8

Field	Description	Note
00	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	Command Register	0x2000: set the baud rate; 0x4000: set the device address, 0x8000: read software version
00	Parity Method	0x00: no parity, 0x01: even parity; 0x02: odd parity
05	Baud Rate Value	Corresponding baud rate 0x00: 4800 0x01: 9600 0x02: 19200 0x03: 38400 0x04: 57600 0x05: 115200 0x06: 128000 0x07: 256000
43 D8	CRC16	The CRC16 checksum of the first 6 bytes of data

Receive code: 00 06 20 00 00 05 43 D8

Field	Description	Note
00	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	Command Register	0x2000: set the baud rate; 0x4000: set the device address, 0x8000: read software version
00	Parity Method	0x00: no parity, 0x01: odd parity; 0x02: even parity
05	Baud Rate	Corresponding baud rate 0x00: 4800 0x01: 9600 0x02: 19200 0x03: 38400 0x04: 57600 0x05: 115200 0x06: 128000 0x07: 256000
43 D8	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: 00 06 40 00 00 01 5C 1B

Field	Description	Note

00	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	Command Register	0x2000: set the baud rate; 0x4000: set the device address, 0x8000: read software version
00 01	Device Address	Set the device address, 0x0001-0x00FF
5C 1B	CRC16	The CRC16 checksum of the first 6 bytes of data

Receive code: 00 06 40 00 00 01 5C 1B

Field	Description	Note
00	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	Command Register	0x2000: set the baud rate; 0x4000: set the device address, 0x8000: read software version
00 01	Device Address	Set the device address, 0x0001-0x00FF
5C 1B	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	Command Register	0x2000: set the baud rate; 0x4000: set the device address, 0x8000: read software version
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 02 04 45

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 02	Device Address	Set the device address, 0x0001-0x00FF
04 45	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: 00 03 80 00 00 01 AC 1B

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
80 00	Command Register	0x2000: set the baud rate; 0x4000: set the device address, 0x8000: read software version

00 01	Byte Number	Fixed 0x0001
AC 1B	CRC16	The CRC16 checksum of the first 6 bytes of data

Receive code: 00 03 02 00 C8 84 12

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 C8	Software Version	Converting to decimal and then shifting the decimal point two places to the left will represent the software version 0x00C8 = 200 = V2.00
84 12	CRC16	The CRC16 checksum of the first 6 bytes of data

For example:

Send: 00 03 80 00 00 01 AC 1B

Receive: 00 03 02 00 C8 84 12 //0x00C8 = 200 =V2.00

Exception Function Code

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

Receive: 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

FAQ

Question: After the flash-on and flash-off parameters are set, does it need to be powered off and restarted to take effect?

Answer:

No, after sending a flash-on and flash-off command and running it, you can directly send a second command without restarting

Support

Technical Support

If you need technical support or have any feedback/review, please click the **Submit Now** button to submit a ticket, Our support team will check and reply to you within 1 to 2 working days. Please be patient as we make every effort to help you to resolve the issue.
Working Time: 9 AM - 6 PM GMT+8 (Monday to Friday)