



LC-Modbus-8R-D7

1. Overview

Elsay eight-way Modbus relay module is equipped with mature and stable 8-bit MCU and RS485 level communication chip. Using standard MODBUS RTU format RS485 communication protocol, it can realize 8-channel input signal detection and 8-channel relay output, which can be used for digital detection or power control occasions.

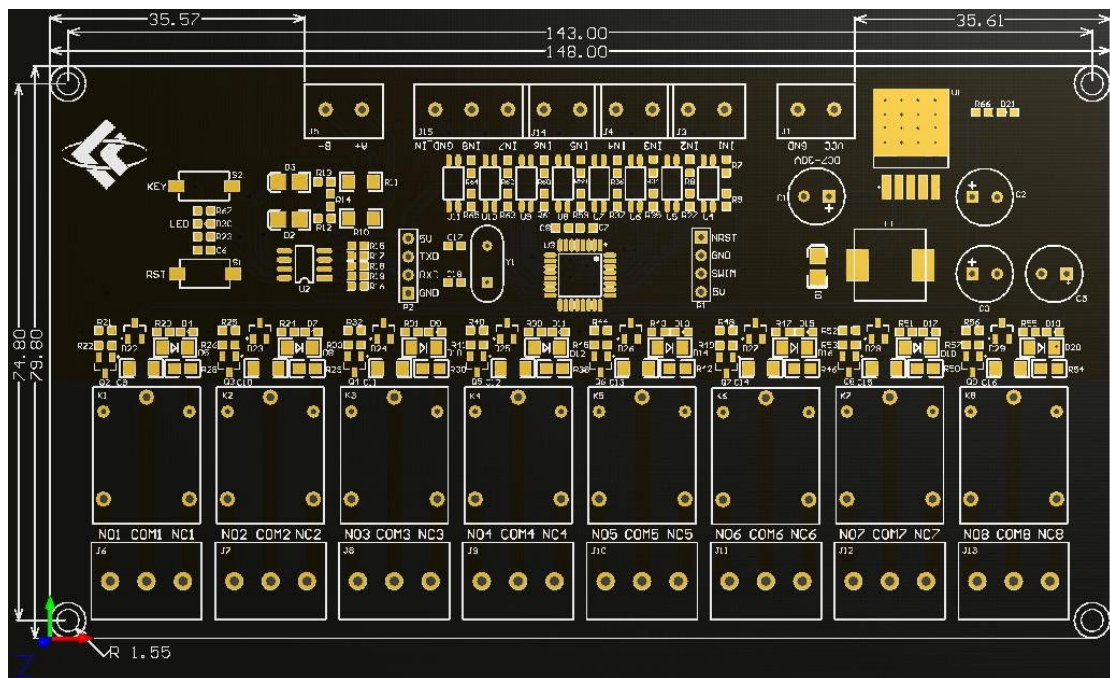
2. Features

1. Onboard mature and stable STM8S103K3T6 MCU and MAX485 level conversion chip;
2. Communication protocol: support standard Modbus RTU protocol;
3. Communication interface: TTL UART interface supporting RS485/5V level;
4. Communication baud rate: 4800/9600/19200, the default is 9600bps, and it supports power-off save;
5. Optocoupler input signal range: DC3.3-30V (this input cannot be used for relay control);
6. Output signal: relay switch signal, support manual, flash off, flash off mode, flash off/flash off delay base is 0.1S, the maximum flash off/flash off time can be set to $0xFFFF \times 0.1S = 6553.5S$;
7. Device address: range 1-255, default 255, support power-off save;
8. The baud rate, input signal, relay status, and device address can be read using software/commands;
9. There are 8 5V, 10A/250V AC 10A/30V DC relays onboard, which can be activated continuously for 100,000 times, with diode effusion protection, and short response time;

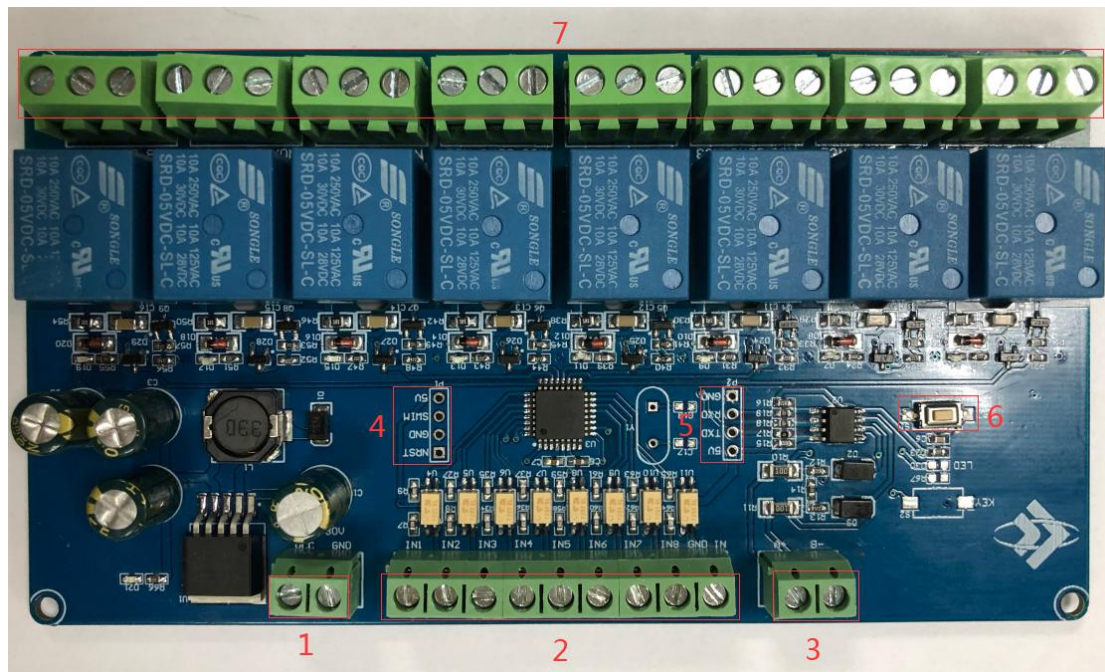
10. On-board relay switch indicator;
11. Reserve STM8 SWIM program programming port to support customers' secondary development and download their own firmware;
12. Power supply voltage: DC7-30V, using 5.08mm terminal for power supply;

3. Hardware introduction and description

3.1. Board size (without shell): 143*79.8, weight (without shell): 145g



3.2 Interface introduction



1. VCC, GND: DC7-30V 5.08mm power input terminal;

2. DC3.3-30V optocoupler signal input:

IN1-IN8: Channels 1-8 positive;

GND_IN: The negative pole of the common terminal of channels 1-8.

3. A+, B-: RS485 communication interface, A+, B- are respectively connected to A+, B- of the external control terminal;

4, 5V, SWIM, GND, NRST: STM8 SWIM program download port

5. TXD, RXD, GND: TTL level UART communication interface, TXD, RXD, GND are respectively connected to RXD, TXD, GND of the external control terminal, only supports 5V level TTL module/MCU. Recommended wiring sequence: Power on the Modbus module → Connect the Modbus module to the TTL module → Power on the TTL module;

6. RST: MCU reset button

7. Relay switch signal output:

NO: Normally open end, the relay is suspended before being closed, and shorted to COM after being closed;



COM: public end;

NC,: Normally closed, the relay is short-connected with COM before it is closed, and it is suspended after it is closed.

4. Introduction to Modbus RTU instruction

Modbus devices perform related operations by receiving Modbus RTU instructions from an external control terminal (such as host computer/MCU). A frame of instruction generally consists of device address, function code, register address, register data, and check code. The frame length and Function code is related. Generally, the first byte of each frame of data is the device address, the range of which can be set is 1-255, the default is 255 (ie 0xFF), and the last 2 bytes are the CRC check code.

Assuming the device address is 255, the commonly used Modbus RTU commands are as follows:

1. Turn on relay No. 1 (manual mode)

Send: FF 05 00 00 FF 00 99 E4

Return as it is: FF 05 00 00 FF 00 99 E4

Remarks: (1) The 3-4 bytes of the sending frame represent the relay address. The addresses of relay 1-relay 8 are 0x0000, 0x0001, 0x0002, 0x0003, 0x0004, 0x0005, 0x0006, 0x0007

(2) The 5-6 bytes of the sending frame represent data, 0xFF00 means opening the relay, 0x0000 means closing the relay

2. Turn off the No. 1 relay (manual mode)

Send: FF 05 00 00 00 00 D8 14

Return as it is: FF 05 00 00 00 00 D8 14

3. Turn on the No. 2 relay (manual mode)

Send: FF 05 00 01 FF 00 C8 24

Return as it is: FF 05 00 01 FF 00 C8 24



4. Turn off the No. 2 relay (manual mode)

Send: FF 05 00 01 00 00 89 D4

Return as it is: FF 05 00 01 00 00 89 D4

1. Turn on all relays

Send: FF 0F 00 00 00 08 01 FF 30 1D

Return: FF 0F 00 00 00 08 41 D3

2. Turn off all relays

Send: FF 0F 00 00 00 08 01 00 70 5D

Return: FF 0F 00 00 00 08 41 D3

3. Set the device address to 1

Send: 00 10 00 00 00 01 02 00 01 6A 00

Return as it is: 00 10 00 00 00 01 02 00 01 6A 00

Remark: The 9th byte 0x01 of the sending frame is the written device address

4. Set the device address to 255

Send: 00 10 00 00 00 01 02 00 FF EB 80

Return as it is: 00 10 00 00 00 01 02 00 FF EB 80

Remarks: The 9th byte of the sending frame, 0xFF, is the written device address

9. Read the device address

Send: 00 03 00 00 00 01 85 DB

Returns: 00 03 02 00 FF C5 C4

Remarks: The 5th byte of the return frame, 0xFF, is the device address read



10. Read the relay status

Send: FF 01 00 00 00 08 28 12

Returns: FF 01 01 01 A1 A0

Remarks: Bit0--Bit7 of the 4th byte of the return frame 0x01 represent relay 1--relay 8, 0 is closed, 1 is open

11. Read optocoupler input status

Send: FF 02 00 00 00 08 6C 12

Returns: FF 02 01 01 51 A0

Remarks: IN1--IN8 of the 4th byte of the return frame 0x01 represents the input signal of optocoupler 1--optocoupler 8, 0 represents low level, and 1 represents high level

12. Set the baud rate to 4800

Send: FF 10 03 E9 00 01 02 00 02 4A 0C

Returns: FF 10 03 E9 00 01 C5 A7

Remarks: The 9th byte of the sending frame is the baud rate setting value, 0x02, 0x03, x04 represent 4800, 9600, 19200 respectively

13. Set the baud rate to 9600

Send: FF 10 03 E9 00 01 02 00 03 8B CC

Returns: FF 10 03 E9 00 01 C5 A7

14. Set the baud rate to 19200

Send: FF 10 03 E9 00 01 02 00 04 CA 0E

Returns: FF 10 03 E9 00 01 C5 A7



15, read the baud rate

Send: FF 03 03 E8 00 01 11 A4

Returns: FF 03 02 00 04 90 53

Remarks: The 5th byte of the return frame represents the baud rate read, 0x02, 0x03, 0x04 represent 4800, 9600, 19200 respectively

16, open the No. 1 relay (flash close mode 2S)

Send: FF 10 00 03 00 02 04 00 04 00 14 C5 9F

Returns: FF 10 00 03 00 02 A4 16

Remarks: (1) The 3-4 bytes of the sending frame represent the relay address. The addresses of relay 1-relay 8 are 0x0003, 0x0008, 0x000D, 0x0012, 0x0017, 0x001C, 0x0021, 0x0026

(2) The 10th-11th byte of the sent frame represents the delay setting value, the delay base is 0.1S, so the delay time is $0x0014 \times 0.1 = 20 \times 0.1S = 2S$, and the relay is turned off automatically after 2S.

17. Turn off relay No. 1 (flash mode 3S)

Send: FF 10 00 03 00 02 04 00 02 00 1E A5 99

Returns: FF 10 00 03 00 02 A4 16

Remarks: (1) The 3-4 bytes of the sending frame represent the relay address. The addresses of relay 1-relay 8 are 0x0003, 0x0008, 0x000D, 0x0012, 0x0017, 0x001C, 0x0021, 0x0026

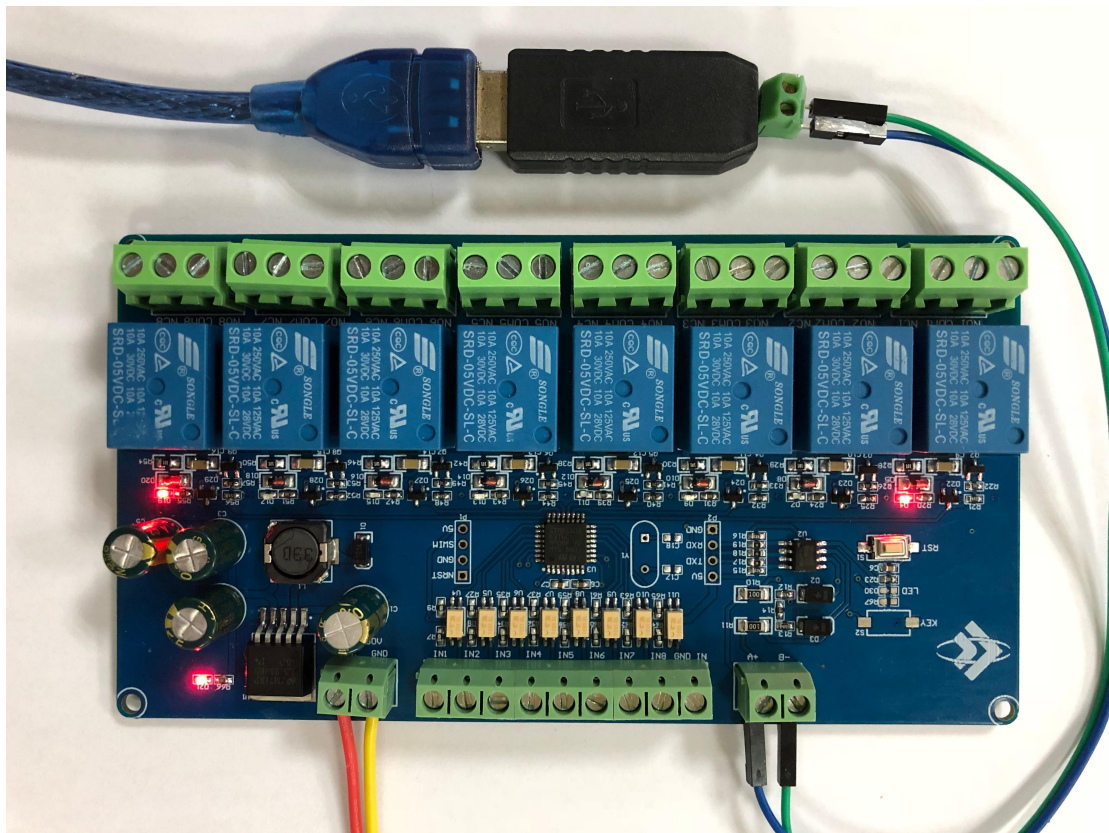
(2) The 10th-11th byte of the sent frame represents the delay setting value, the delay base is 0.1S, so the delay time is $0x001E \times 0.1 = 30 \times 0.1S = 3S$, and the relay is turned off automatically after 3S



5. Simple instructions

The Modbus relay module can receive Modbus RTU commands from the host computer/MCU via the RS485/TTL UART interface to perform related operations. The following is an example of using the host computer software to open relays 1 and 8 (manual mode) through the RS485 interface. Assuming the device address is 255 and the baud rate is 9600, the steps are as follows:

1. Connect the VCC and GND of 5.08mm terminal to the power supply;
2. A+ and B-respectively USB to A+ and B- of the output end of the RS485 module;
3. Open the host computer software "ModbusRTU configuration tool", select the correct port number, select 9600 for the baud rate, set the address to 255, and click "open serial port";
4. Then click "JD1 open" and "JD8 open" to open relays 1 and 8, and the relay indicator lights at the same time. As shown below:





6. How to generate check code

When the Modbus RTU command is sent through the ready-made host computer software (such as: Modbus RTU configuration tool), the CRC check code is automatically generated. If you want to use the serial port debugging software (such as SSCOM) to test the Modbus relay module, you need to manually generate the CRC. The check code is placed at the end of the sending frame, such as opening the first relay (manual mode):

1. The composition of the sending frame to turn on/off the relay (manual mode) is:

Device address (1Byte) + function code (1Byte) + register address (2Byte) + register data (2Byte) + CRC check code (2Byte)

2. Assuming the device address is 0xFF, the first 6 bytes of the sending frame are:

FF 05 00 00 FF 00

3. Use the CRC check tool to find the check code for these 6 bytes:

<http://www.ip33.com/crc.html>



CRC (循环冗余校验) 在线计算

☒ Hex ☐ Ascii

需要校验的数据:

输入的数据为16进制, 例如: 31 32 33 34

参数模型 NAME:

宽度 WIDTH:

多项式 POLY (Hex): 例如: 3D65

初始值 INIT (Hex): 例如: FFFF

结果异或值 XOROUT (Hex): 例如: 0000

☒ 输入数据反转 (REFIN) ☒ 输出数据反转 (REFOUT)

校验计算结果 (Hex):

高位在左低位在右, 使用时请注意高低位顺序!!!

4. After exchanging the high and low byte positions of the check calculation result E499, the CRC check code 99E4 is obtained, and the complete transmission frame: FF 05 00 00 FF 00 99 E4

5. Send the sending frame to the Modbus relay module through the serial debugging software SSCOM V5.13.1 to open the first relay (manual mode), as follows:

