

GRD9L-S Modbus-RTU communication protocol V1.21

1. Summary:

- Protocol type: MODBUS-RTU
- Transmission mode: RS485
Slave address: 3
Baud rate: 9600
- Frame format: 1 bit start bits, 8 bit data bits, 1 bit stops.

2. A packet format:

Address code	Function code	Data code	CRC code
8Bits	8Bits	N*8Bits	16Bits

- **Address code:**

The address field is composed of 1 bytes in the beginning part of the frame, indicating the terminal address specified by the user. The address of each terminal device is unique, and only the addressed terminal device exchanges data with the host.

- **Functional code:**

The function code tells the addressable terminal what functions to perform. The following table lists all the functional codes of this protector, their meanings and their initial functions.

Function code	Meaning	Behavior
03H	Read the data	Get the current data from one to five registers.
06H	Preset register	Write a set of binary data to a register.

- **Data code:**

The data code contains the data needed by the terminal to perform specific functions or the data collected by the terminal when it responds to queries. The contents of these data may be numerical values, reference addresses or limit values. For example, the function code tells the terminal to read a register, and the data code needs to indicate which register to start with and how many data to read.

- **Check code:**

Provides the basis for checking errors in the transmission process by the host

and terminal. Error checking can ensure that the host or terminal does not respond to the error data in the transmission process, and improve the security and reliability of the system data. Error checking uses a 16 bit cyclic redundancy (CRC) method.

3. cyclic redundancy check (CRC) calculation method:

CRC takes up two bytes, its value is calculated by the transmitting device, then attached to the last of the data code and sent out. After receiving the data, the receiving device recalculates the valid data check code except the CRC code, and then compares it with the received CRC check code. If the two values are not equal, the data will be sent out. An error occurred in the transmission.

The process of generating a CRC check code:

- A. preset a 16 bit register as 0FFFFH, called CRC register.
- B. XOR operations are performed on the first byte data in the data package and the low bytes in the CRC register, and the results are stored back in the CRC register.
- C. moves the CRC register to the right one, the top level is filled with 0, the lowest bit is removed and detected.
- D. if the lowest position is 0, repeat C (next shift). If the lowest bit is 1, the XOR operation between CRC register and 0A00H is performed.
- E. repeat C and D until 8 times.
- F. repeats B to E to process the next byte data until all data bytes are processed.
- G. exchanges the high and low byte of the CRC register (low byte before, high byte after).
- H. the value of the final CRC register is the value of CRC.

4. Examples:

● Read the data (Function code 03H)

This function allows the user to get the read-only parameters and read and write parameters of the protector on the host.

Message format sent by host:

Host sending	Bytes	Message sent	Information description
Slave address	1	03H	Slave for sending messages to 03 addresses
Function code	1	03H	Read register
Starting address	2	00H	The starting address of the parameter is 0000H.
Variable number	2	05H	Read 5 registers (a total of 10 bytes).
CRC code	2	842BH	Calculated by host

Slave address	Function code	Starting address	Variable number	CRC code
03H	03H	00H	05H	842BH

Message format returned from slave response:

Slave response	Bytes	Returned information	Information description
Slave address	1	03H	Slave 03 address slave
Function code	1	03H	Read register
Read bytes	1	0AH	Read 5 registers with a total of 10 bytes.
Register 1	2	00H	The address is the contents of the 0000H register.
Register 2	2	03H	The address is the contents of the 0002H register.
Register 3	2	00H	The address is the contents of the 0004H register.
Register 4	2	00H	The address is the contents of the 0006H register.
Register 5	2	08H	The address is the contents of the 0008H register.
CRC code	2	1132H	Calculated from slave

Slave address	Function code	Read bytes	Reg1	Reg2	Reg3	Reg4	Reg5	CRC code
03H	03H	0AH	00H	03H	00H	00H	08H	1132H

● Write single register (function code 06H)

This function allows users to set up a single read and write parameter on the host computer.

Message format sent by host(Switch off):

Host sending	Bytes	Message sent	Information description
Slave address	1	03H	Slave for sending messages to 03 addresses
Function code	1	06H	Write control
Address	2	02H	Corresponding control address
Control command	2	01H	OFF
CRC code	2	E828H	Calculated by host

Slave address	Function code	Starting address	Variable number	CRC code
03H	06H	02H	01H	E828H

Message format sent by host(Switch on):

Host sending	Bytes	Message sent	Information description
Slave address	1	03H	Slave for sending messages to 03 addresses
Function code	1	06H	Write control
Address	2	02H	Corresponding control address
Control command	2	02H	ON
CRC code	2	A829H	Calculated by host

Slave address	Function code	Starting address	Variable number	CRC code
03H	06H	02H	02H	A829H

The message format returned from the machine response is exactly the same as the message format and content sent by the host.

● **Parameter**

Name	Address	Set range	Unit	Property	Remarks
MCB state	0000	0: OFF 1: ON	—	R	
Communication address setting	0001	1~255	—	R/W	The default is 3
MCB control command	0002	0: NC 1: OFF 2: ON	—	W	The read value is 0
The number of ON controlled by instruction	0003	0~65535	—	R/W	Write 0 can be cleared
The number of OFF controlled by instruction	0004	0~65535	—	R/W	Write 0 can be cleared
Number of manual closing	0005	0~65535	—	R/W	Write 0 can be cleared
Number of trips due to failure	0006	0~65535	—	R/W	Including manual switching., write 0 can be cleared
Baud rate	0007	0: 9600 1: 19200	Bps	R/W	
Restore factory settings	0008	0: NC 1: Restore factory settings	—	W	The read value is 0

Auto/Manu state	0009	0: MANU 1: AUTO		R	
-----------------	------	--------------------	--	---	--