MODBUS RTU Communication Protocol

This communication protocol uses the standard ModBus protocol in RTU (hexadecimal number) transmission mode. The ModBus protocol is a master-slave protocol.
Here the host computer is the master The rectifier answers as a slave

Rectifier communication address setting instructions:

1.The rectifier communication address setting machine needs to power off and then press and hold the setting button, and then the machine is powered on.

2.At this time, the voltage digital tube (upper) displays the setting mark 00, and the current digital tube (lower) displays the setting parameters 01~32 Power supply communication address.

3.Pulse potentiometer to set the parameter corresponding to the mark.

4.After setting, press manual/automatic key to save the setting. After protection, the digital tube displays.

5.After power failure, re-power on to enable the new setting parameters.

Address: 1-32
Function code: Contains read and write multiple registers.
Data: Transmitted in binary code.
CRC16: Cyclic Redundancy Check, checksum from slave address to last byte of data area, calculates polynomial code as A001(hex).

1.Communication Port Setup

Communication Mode RS-485
Asynchronous Serial Communications Interface
Baud Rate 19200bps

2.Byte Data Format HEX

One-bit start bit
Eight data bits
One stop bit
Even calibration

1	*	*	*	*	*	*	*	*	1	*
Start bit					Data bits (low to high)				Stop bit	Checksum bit

3 Examples of Message Formats (Read and Write Functions are Defined From the Master's Point of View)

Read Registers

Slave address	Function code	First Register Address	Number of registers N	CRC16
1 byte	1 byte	2 bytes	2 bytes	2 bytes
01H	03H	0000H	0002H	CrcL, CrcH

Read Registers and Return

Slave Address	Function code	Byte Count	Register Data	CRC16
1 byte	1 byte	1 byte	N*2 bytes	2 bytes
01H	03H	04H	0000H 0000H	CrcL, CrcH

Write Registers

Slave Address	Function Code	First Register Address	Number of Registers N	Byte Count	Register Data	CRC16
1 byte	1 byte	2 bytes	2 bytes	1 byte	N*2 bytes	2 bytes
01H	10H	0005H	0002 H	04H	0000H 0000H	CrcL, CrcH

Write Register Return

Slave Address	Function Code	First Register Address	Number of Registers N	CRC16
1 byte	1 byte	2 bytes	2 bytes	2 bytes
01H	10H	0005H	0002H	CrcL, CrcH

Function Code List

Function Code	ModBus Name	Functional Name	Public	Quantity
03H	Read Holding Registers	Read N register values	No	3
10H	Write Multiple Registers	Write N register values	No	2

4 Register address table (read and write registers)

Corresponding Data Address of the Upper Computer	Parameter Name	Message Address	Type	Numerical Range	Note
400001	Voltage display value	0	read and write	0-rated (decimal)	Includes decimal places e.g. 12.00V rectified 0-1200 corresponds to 0-12.00V
400002	Current display value	1		0-Rated value (decimal)	Includes decimal places For example, a 100.0A rectifier 1-1000 corresponds to 1-100.0V.
400003	status value	2		0-3 (decimal)	0 - Normal 1 - Overheating 2 - Overcurrent 3 - Remaining faults.

400004	communication counter	3		0	The value is accumulated by the host computer at regular intervals. If the value is greater than the set value, a communication interruption is displayed.
400005	given value	4		0 - Rated value (decimal)	Set the voltage to include decimal places when regulating e.g. 12.00V rectified 1-1200 corresponds to 1-12.00V
					Set the current to include decimal places when stabilizing the current For example, 100.0A rectified 1-1000 corresponds to 1-100.0V
400006	switching mode	5		0000H-0103H (hexadecimal)	See Control Bit Variables for details 400006.0: 0-regulated current 1-regulated voltage 400006.1: 0 - forward 1 - reverse 400006.3: 0-no alarm 1-alarm 400006.8: 0-stop 1-start

Control Bit Variable

Figure	8	7	6	5	4	3	2	1	0
1	activate (aplan)					give a warning		opposite direction	stable voltage
0	cessation					no alarm		forward	stable flow

Register Address Table Description: 1. All data types are unsigned integer (two bytes).

2. All data with decimal point in the communication transmission are replaced by integers, e.g. 1.000 instead of 1000.(ignore the decimal point) 29.1 instead of 291.

3、All register data is expressed in hexadecimal number during transmission, first pass the high byte, then pass the low byte, such as transmitting 291, first pass 01H, then pass 23H.