Modbus RTU Analog Output 8CH (B)

From Waveshare Wiki
Jump to: navigation, search

Overview

Hardware Description

- AO1~AO8 is the analog output, AGND is the analog ground, and the output range is 0-10V.

Modbus RTU Analog Output 8CH (B)

(https://www.waveshare.com/modbus-rtu-analog-output-8ch.htm)

Voltage Analog Output Module, RS485

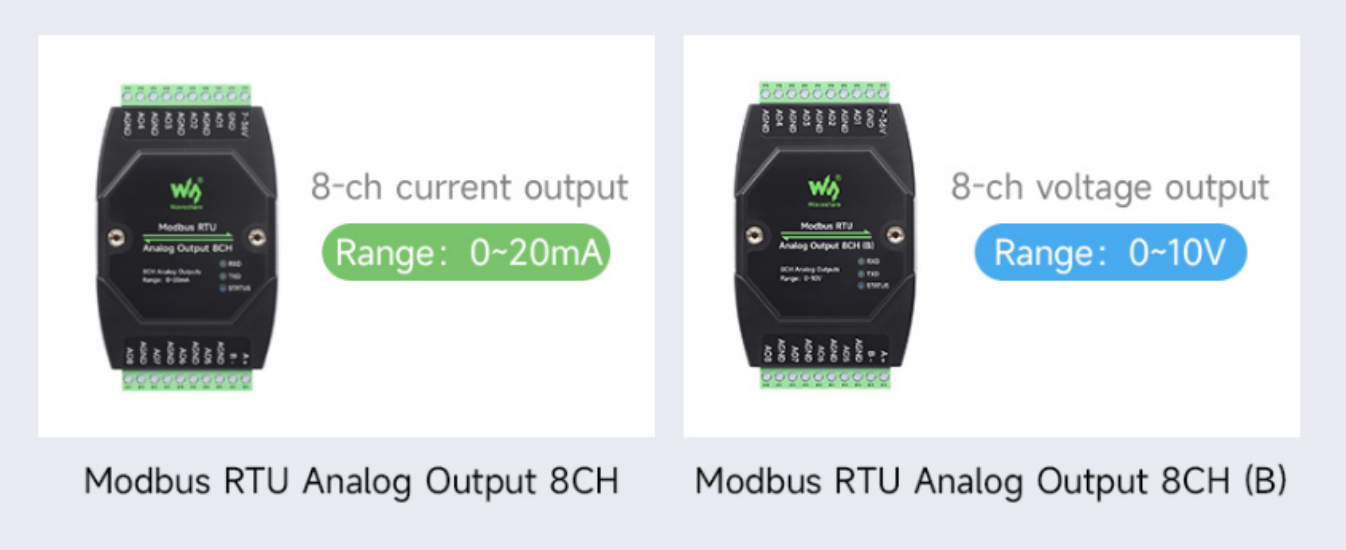
(/wiki/File:Modbus_RTU_Analog_Output_8CH-b-01.jpg)

Note: The input voltage must be greater than the output voltage. Insufficient input voltage may result in insufficient output voltage. For example, to output 10V, the input voltage must be at least 15V.

Version Comparision

Currently, the analog output has two versions: the current output version and the voltage output.

- Modbus RTU Analog Output 8CH outputs the current with the range of 0-20mA.
- Modbus RTU Analog Output 8CH (B) outputs the voltage with the range of 0-10V.



(/wiki/File:Modbus_RTU_Analog_Output_8CH-02.png)

Hardware Connection

- Connect the USB TO 485 and the target board through the wires, connects A --> A and B --> B as shown below:

(/wiki/File:Modbus_RTU_Analog_Output_8CH-03.jpg)

Software Test

SSCOM

- Download SSCOM (https://files.waveshare.com/upload/b/b3/Sscom5.13.1.zip) and open it on the PC, open the corresponding port, set the baud rate of 9600, and select hex to send and receive.

Send the following commands, channel 1 will then output 1V.

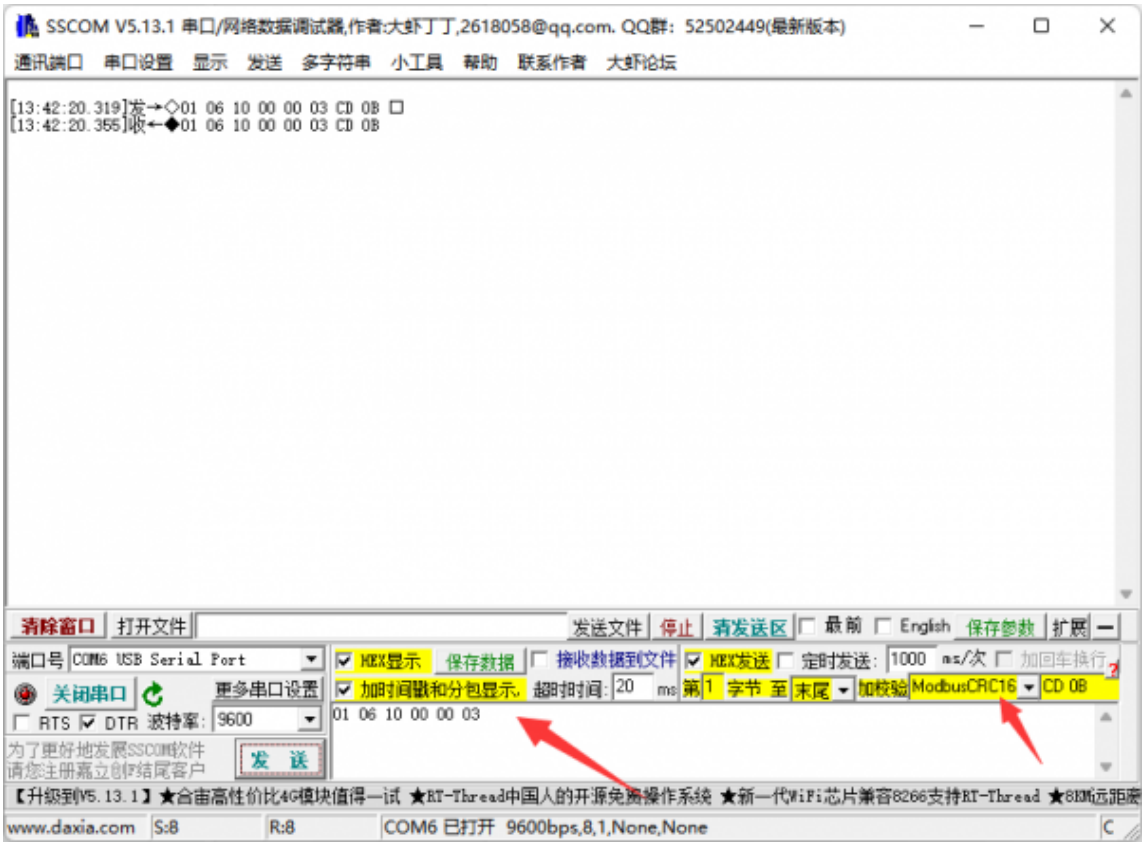
```
01 04 00 00 00 08 F1 CC
```

(/wiki/File:Modbus_RTU_Analog_Output_8CH-b-SSCOM.png)

- If you need to send other commands then select HEX to send, plus check select ModbusCRC16 checksum, enter the first six bytes of the command, and click send then the CRC checksum will be added automatically.

For example, send the following command to set channel 1 to output 5V.

```
01 06 10 00 00 03
```

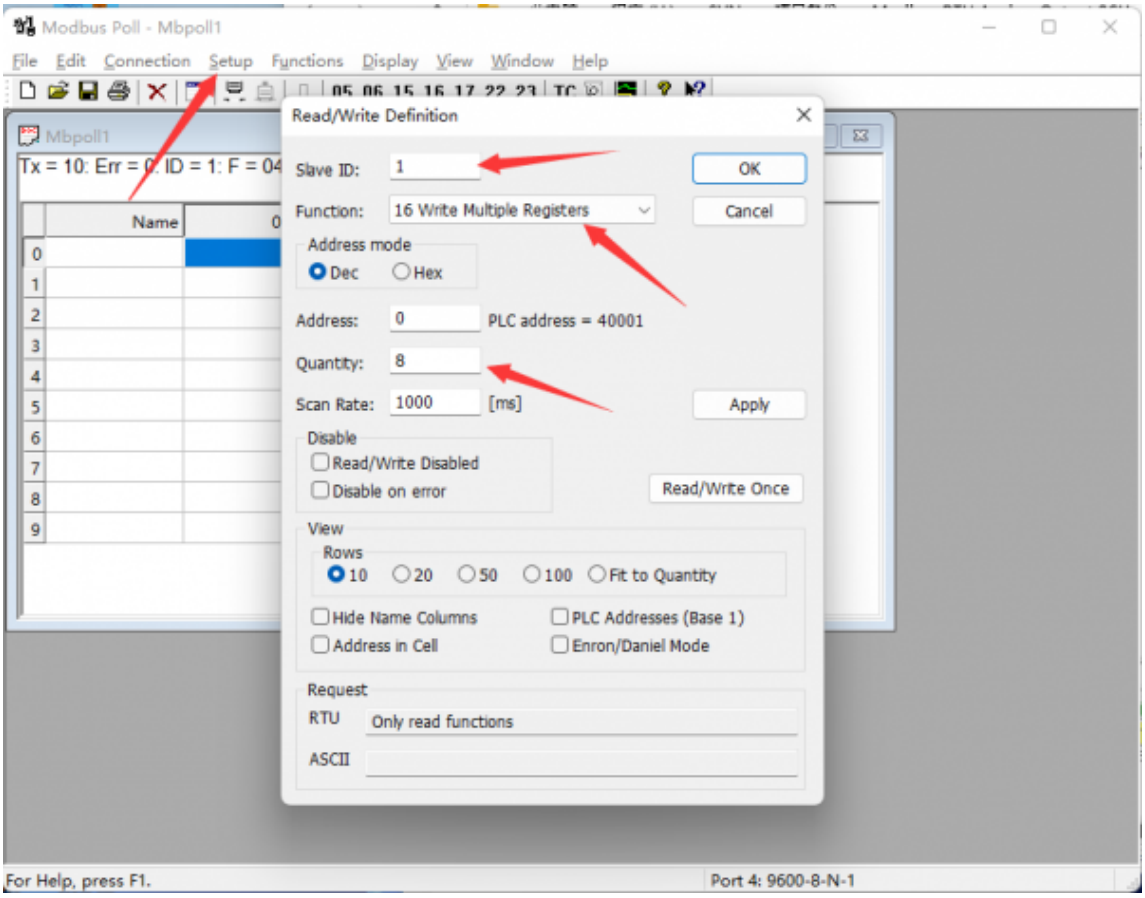


(/wiki/File:Modbus_RTU_Analog_Output_8CH-b-SSCOM02.png)

- For more details, you can see the development protocol.

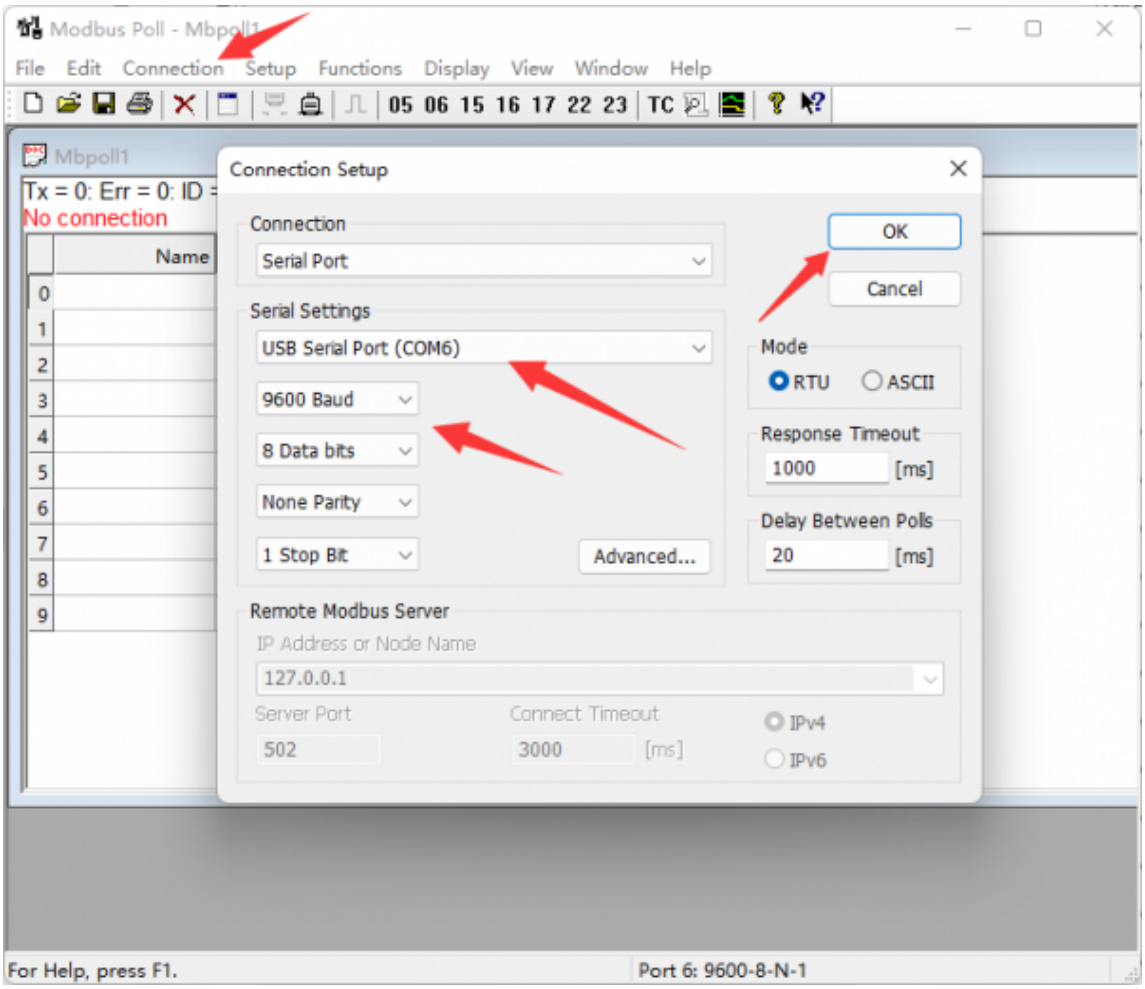
Modbus Poll Software

- It is not convenient to observe data through the SSCOM software. You can use Modbus Poll software to read data. Download and install Modbus Poll software (https://www.modbustools.com/download.html).
- Open the software, and select 'Setup -> Read/Write Definition'. Choose the actual device address for Slave ID, select function code 16, and change the Quantity to 8 channels. Click "OK" to confirm.



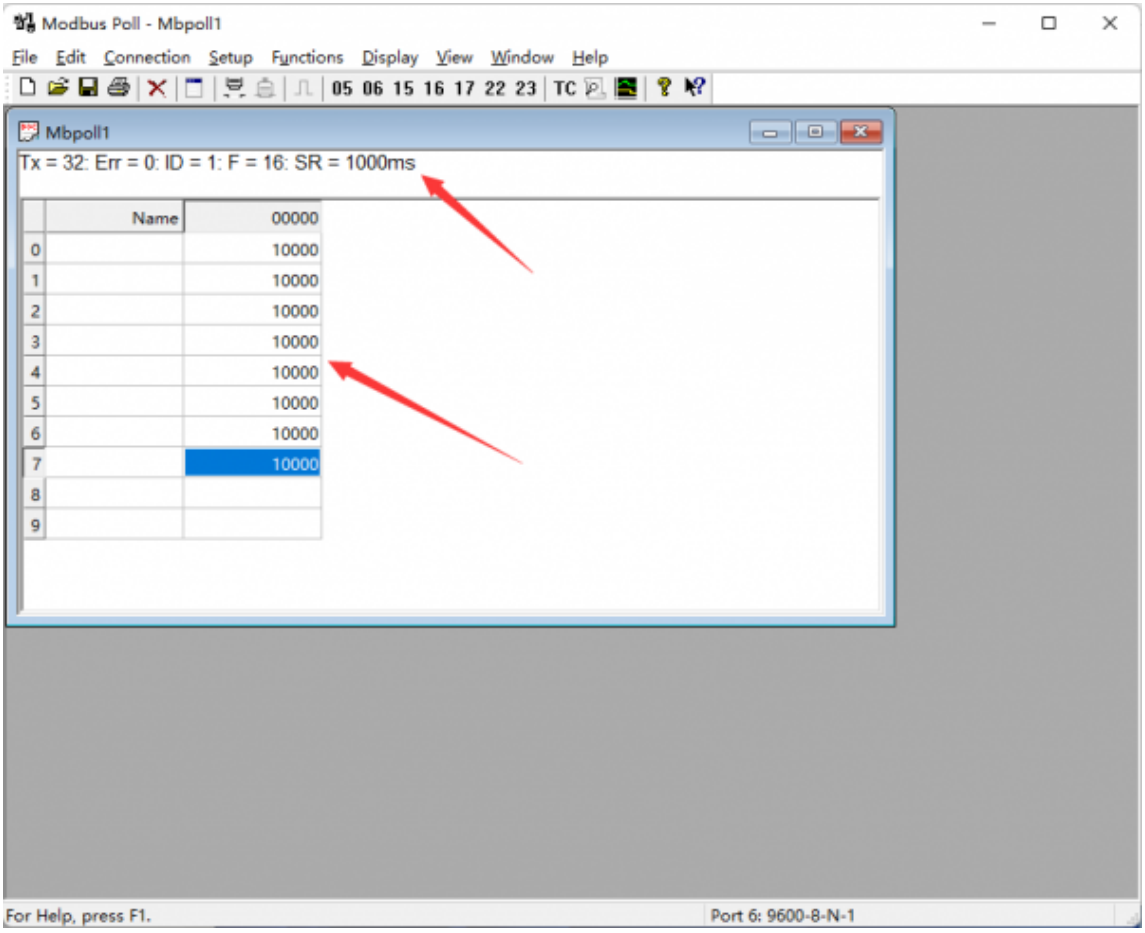
(/wiki/File:Modbus_RTU_Analog_Output_8CH-Poll01.png)

- Select Connection -> Connect..., and choose the corresponding serial ports, the baud rate as 9600, 8 Data bits, none parity mode. Click "OK" to connect.



(/wiki/File:Modbus_Poll-2.png)

- After the connection is normal, you can set 1-8 channels analog output voltage, unit in mV, with the range of 0-100000uA, or 0-10V.



(/wiki/File:Modbus_RTU_Analog_Output_8CH-Poll03.png)

Demo Test

Note: RS485 cannot be directly connected to the Raspberry Pi's serial port as it may damage the devices. It requires a 485 level converter. It's advisable to use the RS485 CAN HAT with the Raspberry Pi. For NUCLEO-F103RB and Arduino, it's recommended to use the RS485 CAN Shield module.

Raspberry Pi

Enter the Raspberry Pi terminal and input the following commands to enter the configure interface:

```
sudo raspi-config
Select Interfacing Options -> Serial Port, Select Yes, and Open the hardware serial port
```

And then reboot the Raspberry Pi:

```
sudo reboot
```

Insert the RS485 CAN HAT module into the Raspberry Pi, and connect the Modbus RTU Relay module to the RS485 CAN HAT module corresponding to the A and B connections. If you are using other 485 devices, make sure the connection is A-A, B-B.

Run the following command to run the demo:

```
sudo apt-get install unzip
sudo apt-get install python3-pip
pip install modbus_tk
wget https://files.waveshare.com/wiki/Modbus-RTU-Analog-Output-8CH/Modbus_RTU_Analog_Outp
ut_Code.zip
unzip Modbus_RTU_Analog_Output_Code.zip
cd Modbus_RTU_Analog_Output_Code/Python3
sudo python3 modbus.py
```

All channels will output 5mA after the demo runs normally.

STM32

STM32 demo is based on the NUCLEO-F103RB and the RS485 CAN Shield module.
Find the STM32 demo file directory and open the STM32 project. Note that before use make sure the computer has installed keil5 software, and download the demo to the development board.
After the demo runs normally, all channels will output 5V current.

Arduino

The Arduino demo is based on NUO PLUS and RS485 CAN Shield module.
Use Arduino IDE to open the demo, select the corresponding development board, and then download the demo.
All channels will output 5V after the demo runs normally.

Development Protocol

Function Code

Function Code	Note
03	Read Single Holding Register
06	Write Single Holding Register
10	Write Multiple Holding Registers

Register Address Introduction

Address (HEX)	Address Storage Content	Register Value	Permission	Modbus Function Code

4x0000 ..... 4x0007	Channel 1 ~ 8 Output Data	Unsigned hexadecimal	Read/Write	0x03,0x06,0x10
4x2000	Serial port parameters	High 8-bit is the parity mode: 0x00~0x02 Low 8-bit is the baud rate mode: 0x00~0x07	Read/Write	0x03,0x06
4x4000	Device address	Direct store Modbus addresses Device address: 0x0001-0x00FF	Read/Write	0x03,0x06
4x8000	Software version	Convert to decimal and then shift the decimal point two places to the left to represent the software version 0x0064 = 100 = V1.00	Read	0x03

Set Single-channel Output Command

Send code: 01 06 00 00 03 E8 89 74

Fields	Description	Note
01	Device Address	0x00 indicates the broadcast address, 0x01-0xFF indicates the device address
06	06 Command	Write a single register command
00 00	Register Address	0x0000 - 0x0007 corresponds to 1~8 channel output data
03 E8	Value Setting	Set the value to hexadecimal, unit in uA, high bits in the front, low bits at the end 0x03E8 = 1000mV, or 1V
89 74	CRC16	CRC16 checksum of the first 6 bytes of data

Receive code: 01 06 00 00 03 E8 89 74

Fields	Description	Note
01	Device Address	0x00 indicates the broadcast address, 0x01-0xFF indicates the device address
06	06 Command	Write a single register command
00 00	Register Address	0x0000 - 0x0007 corresponds to 1~8 input channel data type
03 E8	Value Setting	Set the value to hexadecimal, unit in uA, high bits in the front, low bits at the end 0x03E8 = 1000mV, or 1V
89 74	CRC16	CRC16 checksum of the first 6 bytes of data

For example:

[No.1 address device]

Set channel 1 data type as 1V : 01 06 00 00 03 E8 89 74

Set channel 2 data type as 5V : 01 06 00 01 13 88 D5 5C

Set Multiple-channel Output Command

Send code: 01 10 00 00 00 08 10 03 E8 03 E8 03 E8 03 E8 03 E8 03 E8 03 E8 03 E8 3C 05

Fields	Description	Note
01	Device Address	0x00 indicates the broadcast address, 0x01-0xFF indicates the device address
10	10 Command	Write multi-register command
00 00	Register Initial Address	0x0000 - 0x0007 corresponds to channel 1~8 output
00 08	Register Numbers	Set the number of registers, up to 8 channels
10	Bytes Numbers	Set the output bytes
03 E8 ..... 03 E8	Command	Channel 1 analog output ..... Channel 8 analog output Set the value in hexadecimal, unit in mV, high bits in the front, low bits in the end 0x03E8 = 1000mV, or 1V
3C 05	CRC16	CRC16 checksum of the first 6 bytes of data

Receive code: 01 10 00 00 00 08 C1 CF

Fields	Description	Note
01	Device Address	0x00 indicates the broadcast address, 0x01-0xFF indicates the device address
10	10 Command	Write multiple-register output command
00 00	Register Initial Address	0x0000 - 0x0007 corresponds to channel 1~8 output
00 08	Register Numbers	Set the number of registers, up to 8 channels
C1 CF	CRC16	CRC16 checksum of the first 6 bytes of data

For example:

[No.1 address device]

Set channel 1-8 input 1V : 01 10 00 00 00 08 10 03 E8 03 E8 03 E8 03 E8 03 E8 03 E8 03 E8 03 E8 3C 05

Set channel 3-5 output 2V : 01 10 00 02 00 03 06 07 D0 07 D0 07 D0 84 0E

Read Channel Output Command

Send code: 01 03 10 00 00 08 40 CC

Fields	Description	Note
01	Device Address	0x00 indicates the broadcast address, 0x01-0xFF indicates the device address
03	03 Command	Read holding register
00 00	Register Initial Address	0x0000 - 0x0007 corresponds to channel 1~8 input
00 08	Register Numbers	Read the number of registers, up to 8 channels
40 CC	CRC16	CRC16 checksum of the first 6 bytes of data

Receive code: 01 03 10 03 E8 03 E8 03 E8 03 E8 03 E8 03 E8 03 E8 03 E8 C1 91

Fields	Description	Note
01	Device Address	0x00 indicates the broadcast address, 0x01-0xFF indicates the device address
03	03 Command	Read the holding register
10	Bytes Numbers	Returns all bytes of the status message
03 E8 …… 03 E8	Output channel	Channel 1 analog output …… Channel 8 analog output Set the value in hexadecimal, unit in mV, high bits in the front, low bits in the end 0x03E8 = 1000mV, or 1V
C1 91	CRC16	CRC16 checksum of the first 6 bytes of data

For example:

[No.1 address device]

Read channel 1-8 data type: 01 03 00 00 00 08 44 0C

Read channel 1 data type: 01 03 00 00 00 01 84 0A

Read channel 2 data type: 01 03 00 01 00 01 D5 CA

Read channel 3-5 data type: 01 03 00 02 00 03 A4 0B

Set Baudrate Command

Send code: 00 06 20 00 00 05 43 D8

Fields	Description	Note
00	Device Address	0x00 indicates the broadcast address, 0x01-0xFF indicates the device address
06	06 Command	Set the baud rate and the device address
20 00	Command Register	0x2000 for setting baudrate, 0x4000 for setting device addresses
00	Parity Method	0x00 is no parity, 0x01 is even parity, 0x02 is odd parity
05	Baudrate Value	0x00: 4800 0x01: 9600 0x02: 19200 0x03: 38400 0x04: 57600 0x05: 115200 0x06: 128000 0x07: 256000
43 D8	CRC16	CRC16 checksum of the first 6 bytes of data

Receive code: 00 06 20 00 00 05 43 D8

Fields	Description	Note
00	Device Address	0x00 indicates the broadcast address; 0x01-0xFF indicates device address
06	06 Command	Set the baud rate and device addresses
20 00	Command Register	0x2000 is setting the baud rate, 0x4000 is setting the device address
00	Parity Method	0x00 is no parity, 0x01 is even parity, 0x02 is odd parity
05	Baudrate	0x00: 4800 0x01: 9600 0x02: 19200 0x03: 38400 0x04: 57600 0x05: 115200 0x06: 128000 0x07: 256000
43 D8	CRC16	CRC16 checksum of the first 6 bytes of data

For example:

[No.1 address device]

Set baud rate 4800: 00 06 20 00 00 00 83 DB

Set baud rate 9600: 00 06 20 00 00 01 42 1B

Set baud rate 115200: 00 06 20 00 00 05 43 D8

Set Device Address Command

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

Fields	Description	Note
00	Device Address	0x00 indicates the broadcast address; 0x01-0xFF indicates device address
06	06 Command	Set the baud rate and device addresses
40 00	Command Register	0x2000 is setting baudrate, 0x4000 is setting the device address
00 01	Device Address	Set the device address: 0x0001-0x00FF
5C 1B	CRC16	CRC16 checksum of the first 6 bytes of data

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

Fields	Description	Note
00	Device Address	0x00 indicates the broadcast address; 0x01-0xFF indicates device address
06	06 Command	Set the baud rate and device addresses
40 00	Command Register	0x2000 is setting baudrate, 0x4000 is setting the device address
00 01	Device Address	Set the device address: 0x0001-0x00FF
5C 1B	CRC16	CRC16 checksum of the first 6 bytes of data

For example:

[No.1 address 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

Fields	Description	Note
00	Device Address	0x00 indicates the broadcast address; 0x01-0xFF indicates device address
03	03 Command	Read device address command
40 00	Command Register	0x0200 is reading software version, 0x0040 is reading device address
00 01	Byte Numbers	Fixed 0x0001
90 1B	CRC16	CRC16 checksum of the first 6 bytes of data

Receive code: 01 03 02 00 01 79 84

Fields	Description	Note
00	Device Address	0x00 indicates the broadcast address; 0x01-0xFF indicates device address
03	03 Command	the command for reading the software version and the device address
02	Byte Numbers	Return byte numbers
00 01	Device Address	Set the device address: 0x0001-0x00FF
79 84	CRC16	CRC16 checksum of the first 6 bytes of data

For example:

[No.1 address device]

Send: 00 03 40 00 00 01 90 1B

Receive: 01 03 02 00 01 79 84 //Address 0x01

[No.2 address device]

Send: 00 03 40 00 00 01 90 1B

Receive: 02 03 02 00 02 7D 85 //Address 0x02

[No.3 address device]

Send: 00 03 40 00 00 01 90 1B

Receive: 03 03 02 00 03 81 85 //Address 0x03

Read Software Version Command

Send code: 00 03 80 00 00 01 AC 1B

Fields	Description	Note
01	Device Address	0x00 indicates the broadcast address; 0x01-0xFF indicates device address
03	03 Command	the command for reading the software version and the device address
80 00	Command Register	0x4000 is reading software version, 0x8000 is reading device address
00 01	Byte Numbers	Fixed 0x0001
8F CA	CRC16	CRC16 checksum of the first 6 bytes of data

Receive code: 01 03 02 00 64 B9 AF

Fields	Description	Note
01	Device Address	0x00 indicates the broadcast address; 0x01-0xFF indicates device address
03	03 Command	the command for reading the software version and the device address
02	Byte Numbers	Receive byte numbers
00 64	Software Version	Convert to decimal and then shift the decimal point two places to the left to represent the software version. 0x0064 = 100 = V1.00
B9 AF	CRC16	CRC16 checksum of the first 6 bytes of data

For example:

[No.1 address device]

Send: 00 03 80 00 00 01 AC 1B

Receive: 01 03 02 00 64 B9 AF //0x0064 = 100 =V1.00

Resource

Demo

- Demo code (https://files.waveshare.com/wiki/Modbus-RTU-Analog-Output-8CH/Modbus_RTU_Analog_Output_Code.zip)

Software

- Sscom5.13.1 (https://files.waveshare.com/upload/b/b3/Sscom5.13.1.zip)
- Modbus Poll Software (https://www.modbustools.com/download.html)

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 AM GMT+8 (Monday to Friday)

Submit Now (https://service.waveshare.com/)