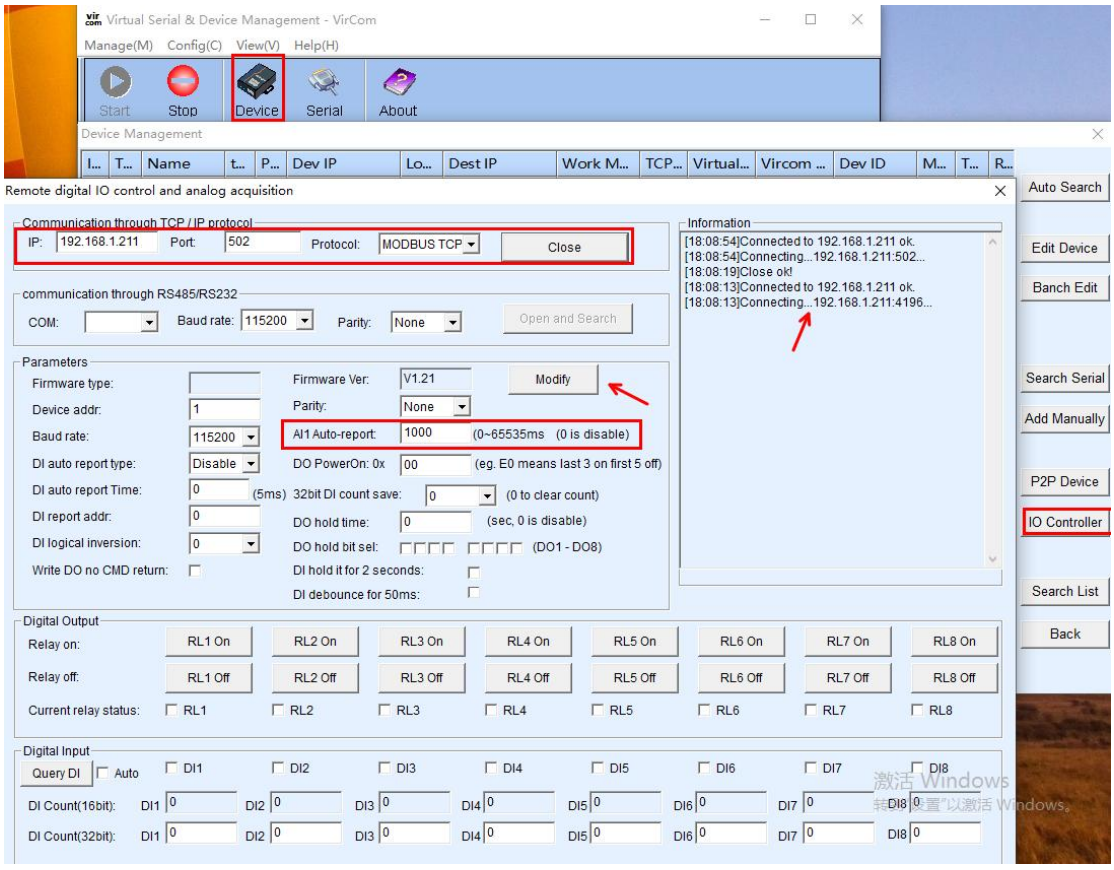
IO port advanced function application

AI/DI proactive reporting

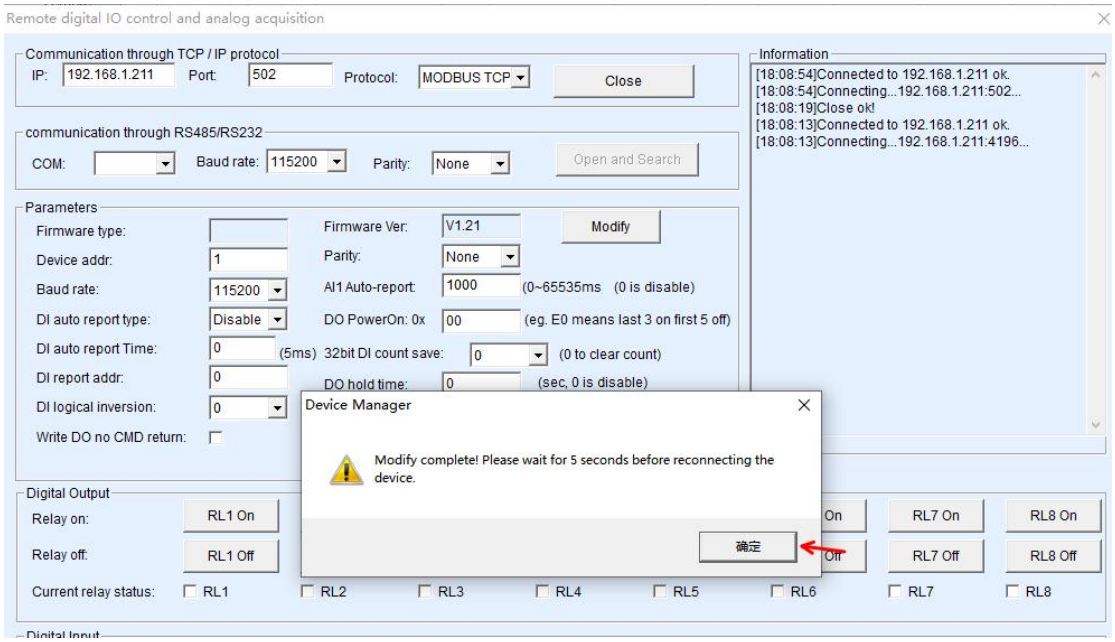
The normal logic for both analog AI input and switch DI input requires the upper computer to actively issue query instructions in order to provide real-time status feedback. If the customer's application site requires the device to have a proactive reporting mechanism logic, then this feature needs to be enabled.

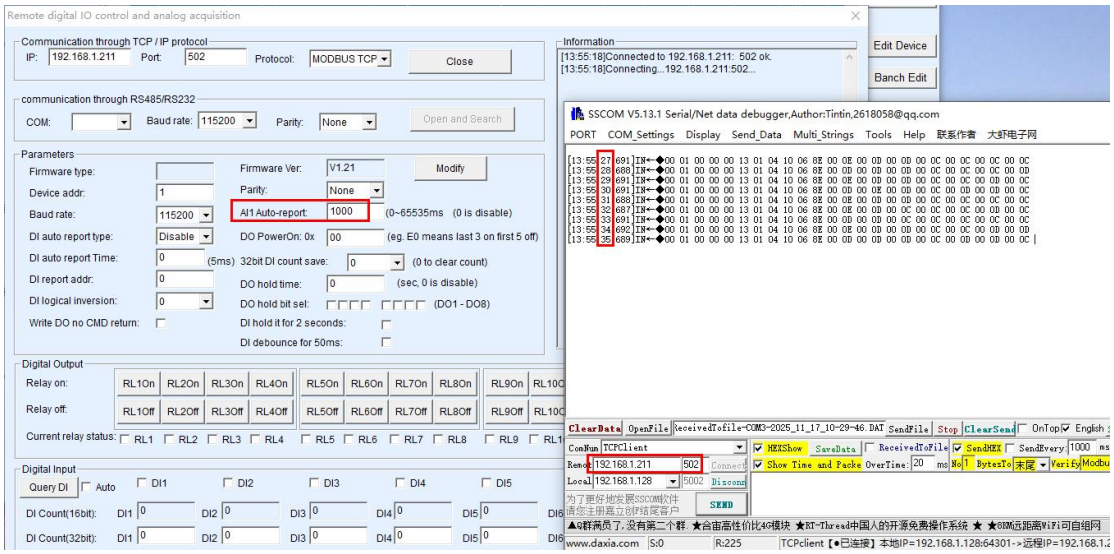
AI proactively reports

As shown in the figure, the reporting time for analog signals can be set, with a time interval ranging from 0 to 65535, in milliseconds. Setting 0 means not enabling active reporting.



After changing the settings, you will be prompted with a successful setting. Click OK, exit, and then use third-party software to connect to the receiving device to actively report the information.





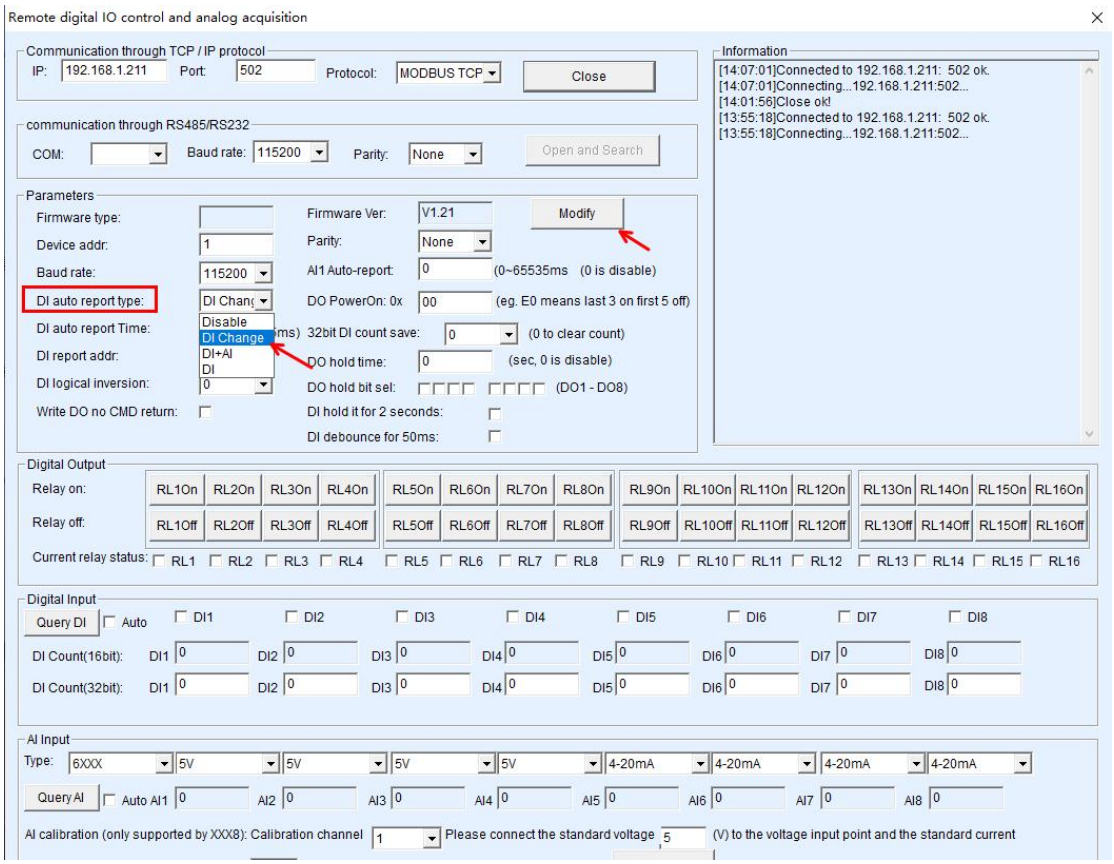
As shown in the above figure, once set up, connect the device through IP and port, and you can see that the AI status is reported every 1 second.

DI proactively reports

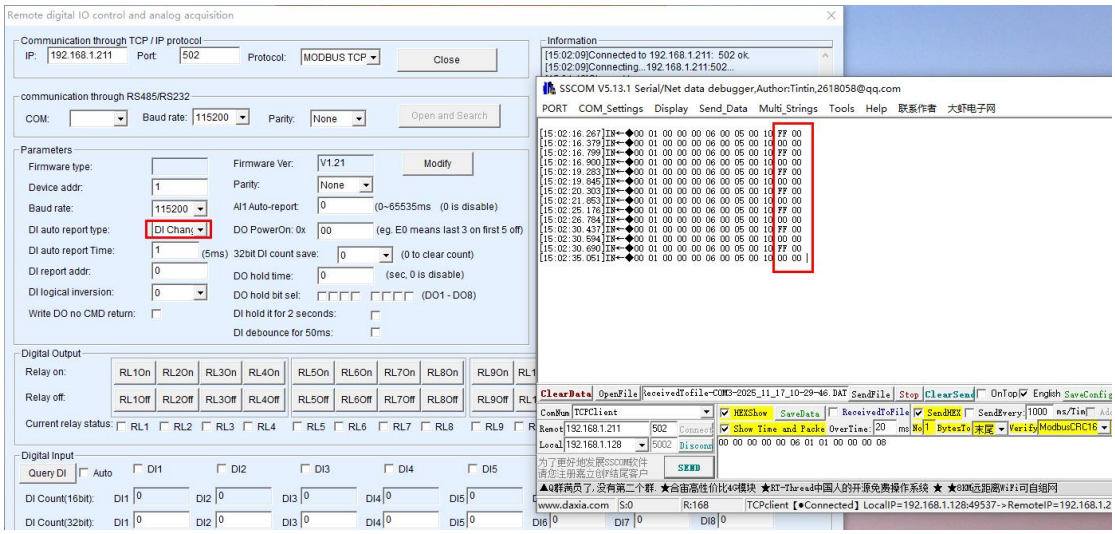
DI reporting score: 1. DI change reporting refers to reporting once when there is a change in DI status (such as from 0 to 1 or from 1 to 0); 2 DI timed reporting, report the status of DI according to the software's set time, it is recommended to set it to no less than 200ms.

DI Change Reporting

The standard MODBUS device is used in a question and answer format. For some users who want to receive feedback when there is a change in DI input, that is, the function of actively returning, the DI active reporting function is set up to achieve this. As shown in the figure, the DI reporting address should generally not be the same as the device address, otherwise the 05 command will be indistinguishable from the DO controlled return command. The default setting is 0, and customers can define their own settings.



As shown in the figure, select the DI change reporting settings, click on Settings, and the message window indicating successful settings will take effect.

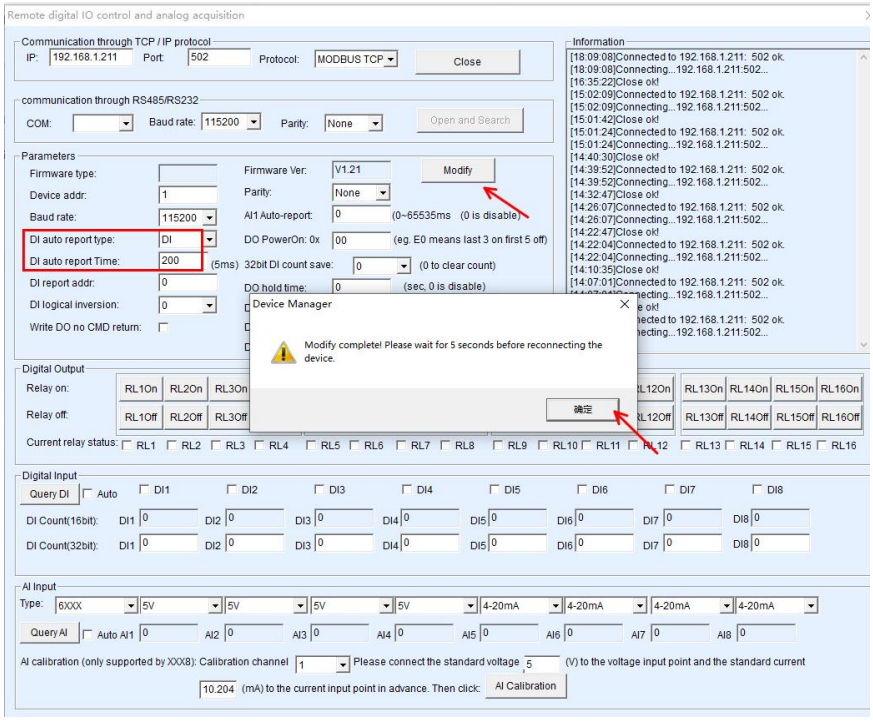


As shown in the test, when the status of DI changes, this side can receive the uploaded information status normally.

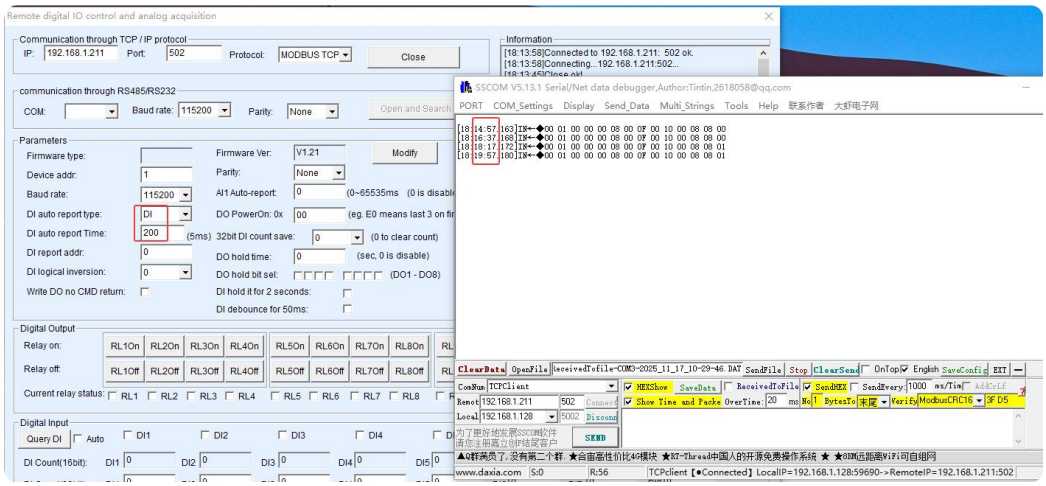
DI timed reporting

If some customers need DI regular reporting, they can set the DI reporting type to "DI", and then adjust the reporting time to 200 (that is, the reporting time is an even number other than 0). The actual time is （200-100） * 1=100s. Eight DI data will be uploaded at regular intervals.

When the DI active reporting time is set to 0, disable active reporting; Set to 1, initiate DI change active reporting. Set to other values (2-65535) and start scheduled reporting. When set to even, 8 DIs will be reported at regular intervals according to the 15 instruction; when set to odd, both DI and AI will be enabled to report simultaneously.When set to n, where n is a non-zero even number, n<100, The time for reporting DI is n* 5 milliseconds, n>100， For example, if it is 200, it is reported in (200-100) seconds.



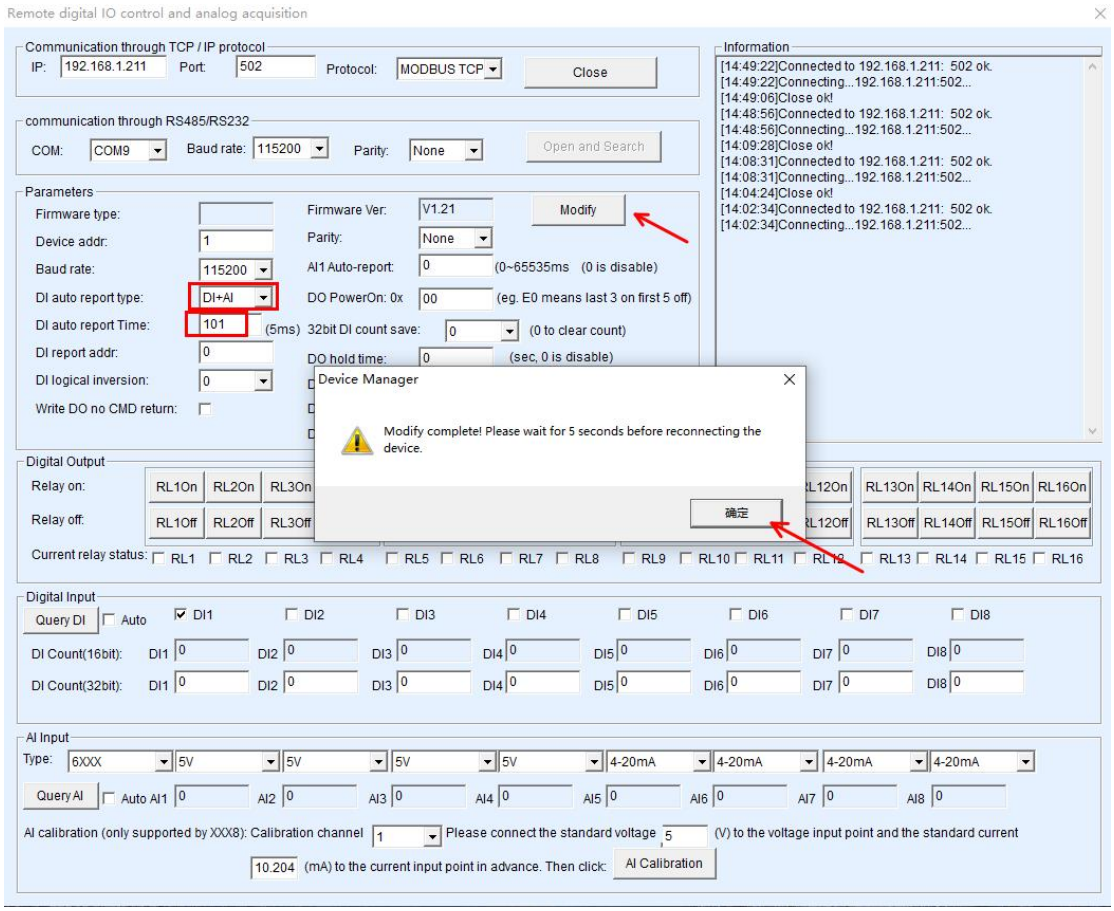
As known from the test, DI data status will be sent every 100 seconds at regular intervals.



AI+DI scheduled reporting

As shown in the figure, set the active reporting time to an odd number of 11, and the reporting type will automatically change to AI+DI. Set the AI reporting time to 1 second and click on "Set".

When n is a non-zero odd number, n<100, Report time (n-1) * 5mS; n>100, Report time (n-100) * 1 second.



As shown in the link test, the status of AI+DI is scheduled to be sent for 1 second. The first 16 bytes are the parameter parsing of AI, and the last 2 bytes are the parameter parsing of DI. State1 is the status of DI9-DI16, and State2 is the status of DI1-DI8.

Return byte format

Modbus TCP

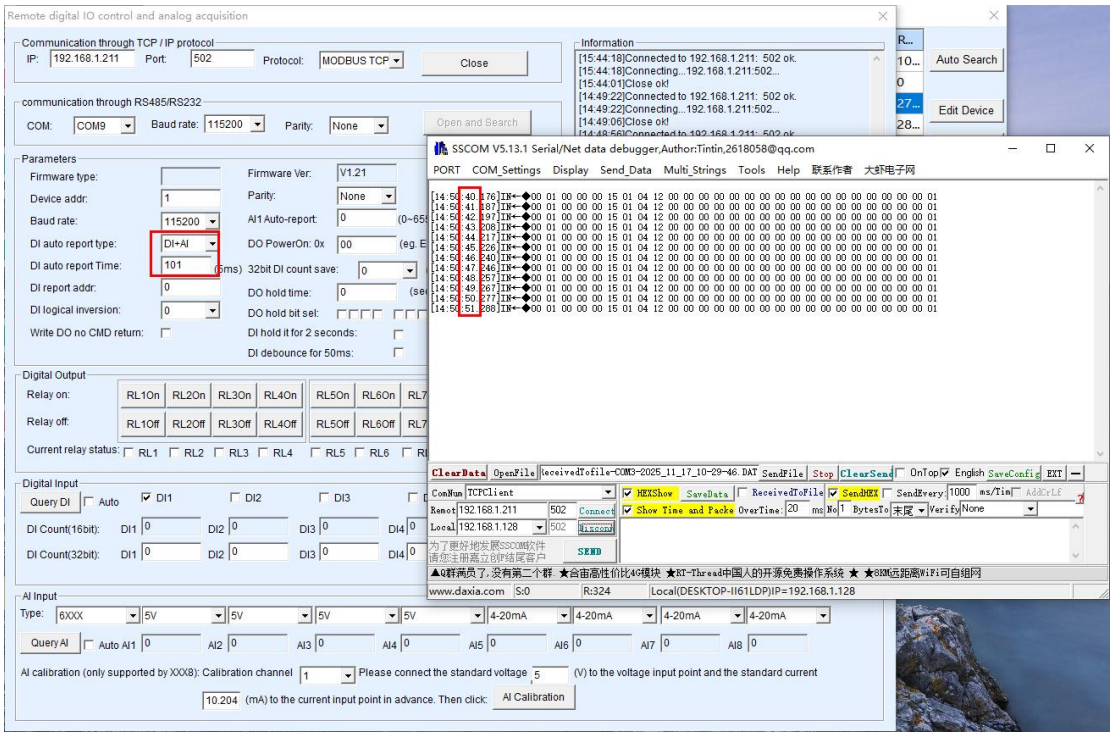
Modbus TCP prefix (6 bytes)	Station address (1 byte)	04	Data length (1 byte)	AI numerical value (16 bytes)	DI value (2 bytes)

Instructions returned as shown in the figure：00 01 00 00 00 15 01 04 12 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 01

According to the feedback instructions, all 8 AI values are 0, and DI1 has a signal input.

Modbus Rtu

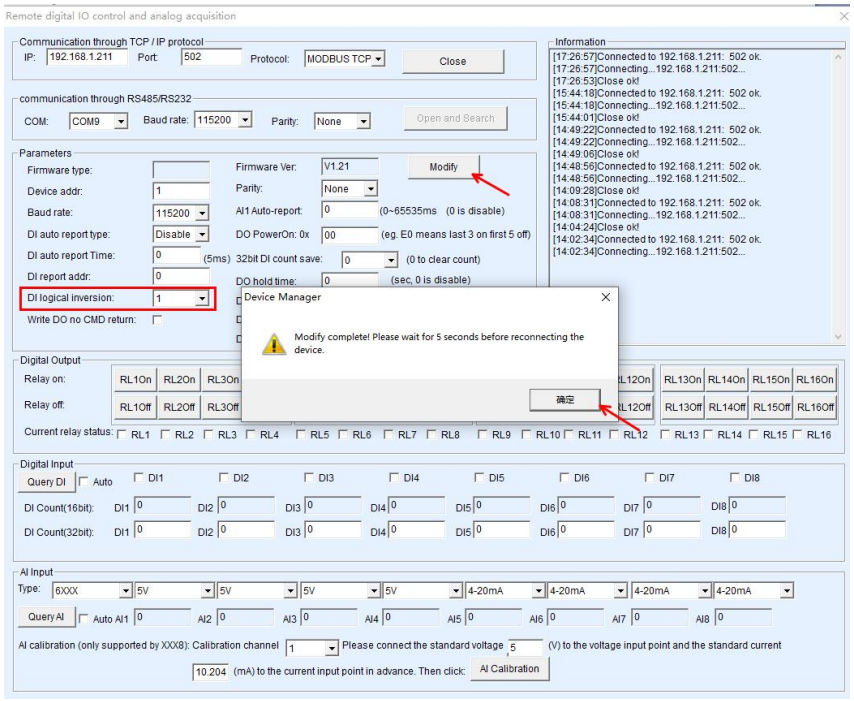
Station address (1 byte)	04	Data length (1 byte)	AI numerical value (16 bytes)	DI value (2 bytes)	CRC check (2 bytes)



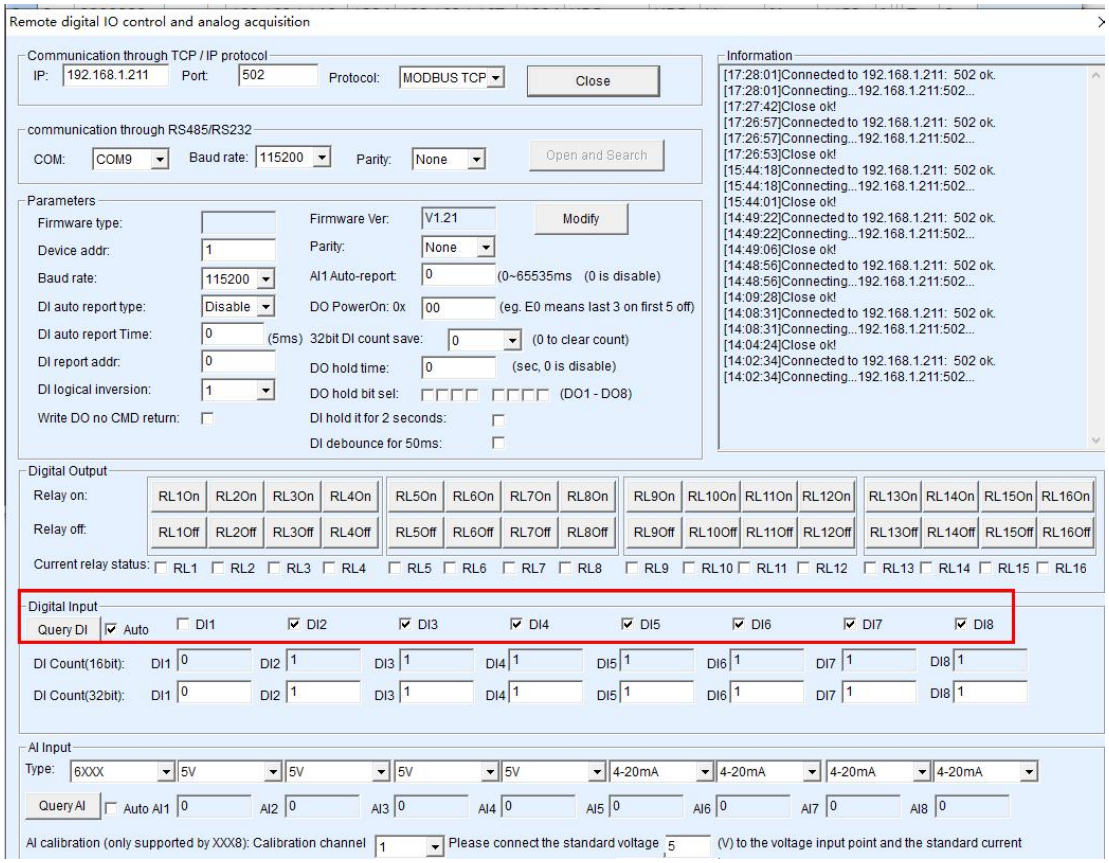
DI logic inversion

When the DI is connected to a low level, the 6808I defaults to a low DI input of 1. Normally, when the DI input is at a low level, the corresponding bit is returned as 1. If you do not want to change the connection method now and need the DI input high-level bit to be 1 and the input low-level bit to be 0, you can choose the "DI logic inversion function" at this time. DI inversion can also affect DI counting, which occurs when DI changes from 0 to 1 and increases by 1, that is, when the high level changes to the low level. If the DI logic is reversed, changing from low level to high level will increase the count by one.

The method for setting DI logic inversion is as follows.

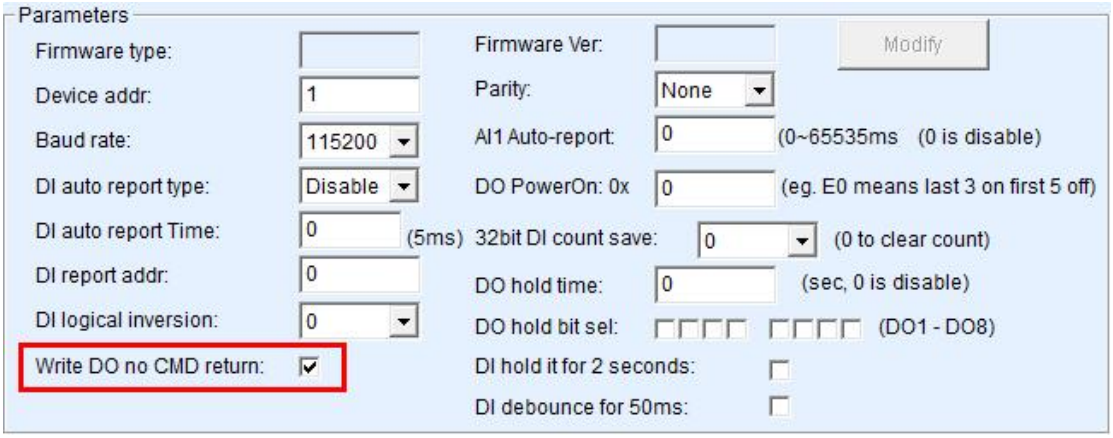


As shown in the test diagram, DI1 sends a low-level signal, and the remaining 7 channels are not connected because the logic inversion function is enabled. The original state of DI with input "1" becomes "0", and the other 7 channels become "1"



Write DO with no return instruction

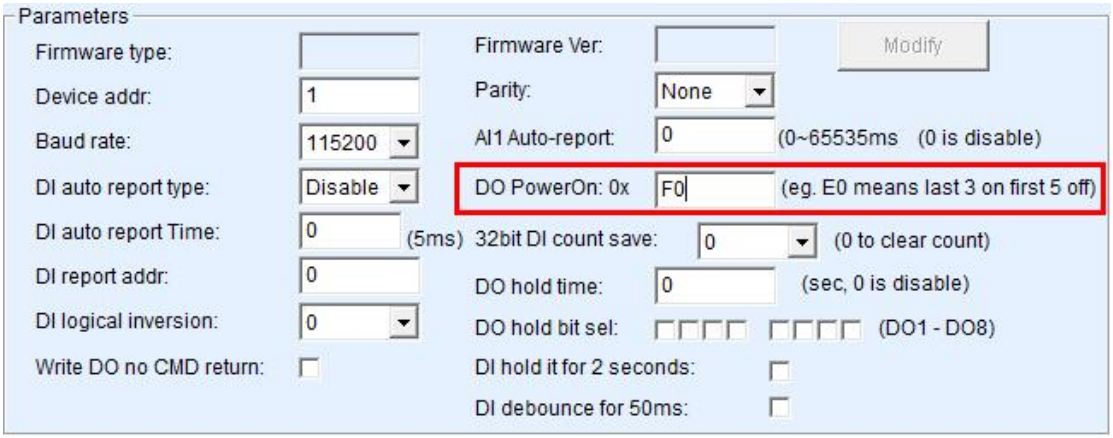
The write DO function does not have a return instruction, meaning that when a write DO command is sent, such as with function codes 05 or 0F, the device can be controlled but will not return a response.



Just check the box on the interface as shown in the figure.

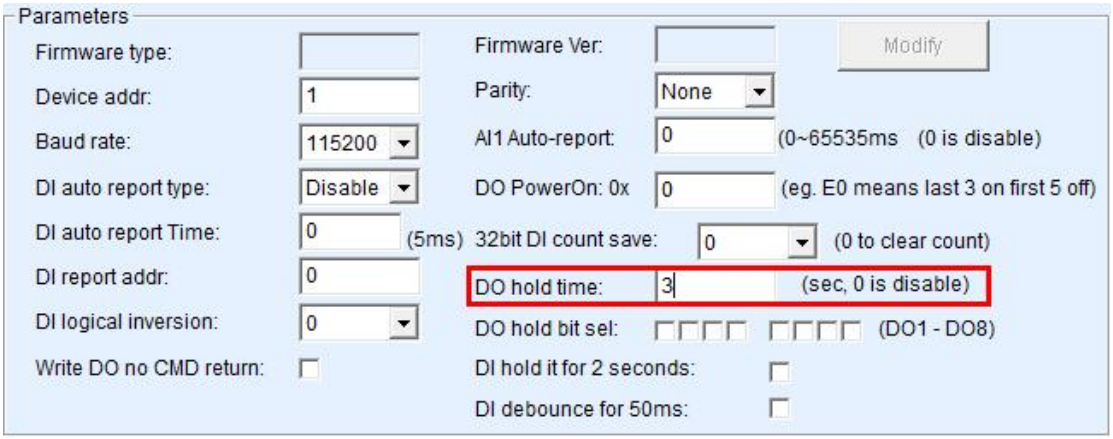
DO status after power on

In some scenarios, the IO controller needs to be in a closed or open state immediately after power on. This function can now be set through the "IO" controller dialog box. If set to F0, it means that the first 4 channels are disconnected and the last 4 channels are engaged. Among the 8 bits, each bit represents the status of a DO channel, and 1 represents suction.



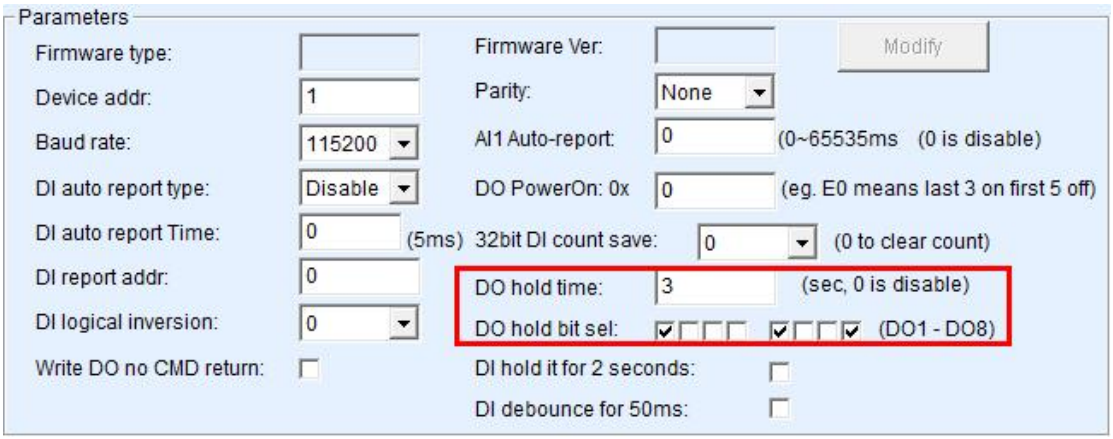
DO data retention function

DO maintains its function, that is, if DO is in a closed state, it needs to continuously give instructions set to 1. Once the instruction set to 1 is not received within a certain period of time, DO will be immediately disconnected.



As shown in the figure, set the DO holding time to 3 seconds using the IO controller software.

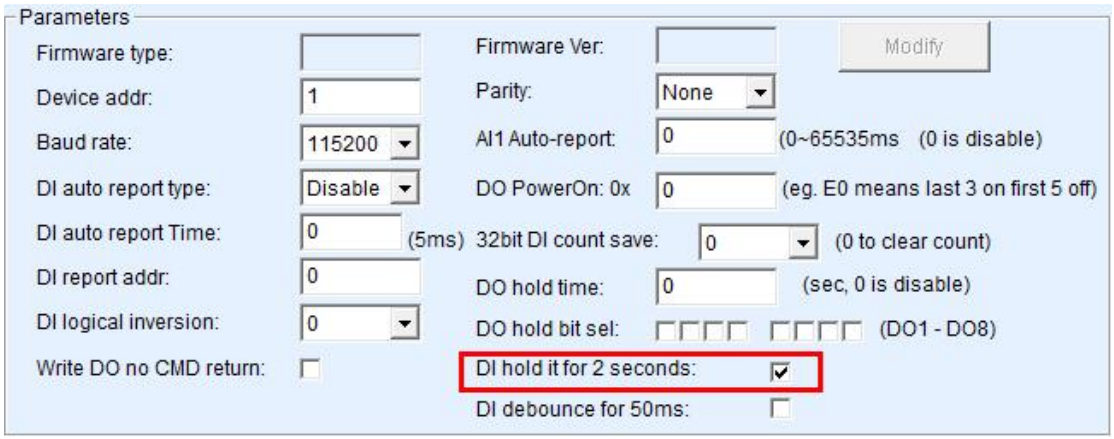
The single/multiple DO hold function (VIRCOM software version needs to be above V6.76,the program version needs to be V1.26 or above), which allows setting the hold function for a single or several DOs. Once the instruction to set to 1 is not received within a certain period of time, the DO will be immediately disconnected.



As shown in the figure, set the DO1, DO5, and DO8 circuits to close and hold for 3 seconds.

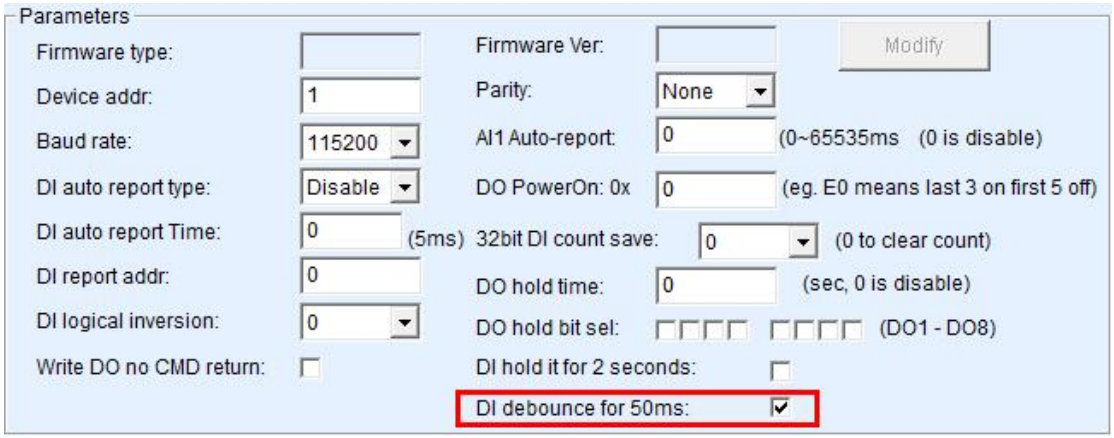
DI trigger level to maintain for 2 seconds function

Write the function of holding the DI trigger level for 2 seconds after it disappears, that is, even if the DI state has already changed to 0 after triggering DI to 1, it will still hold 1 for another 2 seconds. For example, when triggering DI for 0.5 seconds, the actual time it takes to read DI status as 1 is 2.5 seconds (0.5 seconds plus 2 seconds).



DI anti shake function

After selecting this feature, DI adds 50ms of anti shake function.



DI counting function

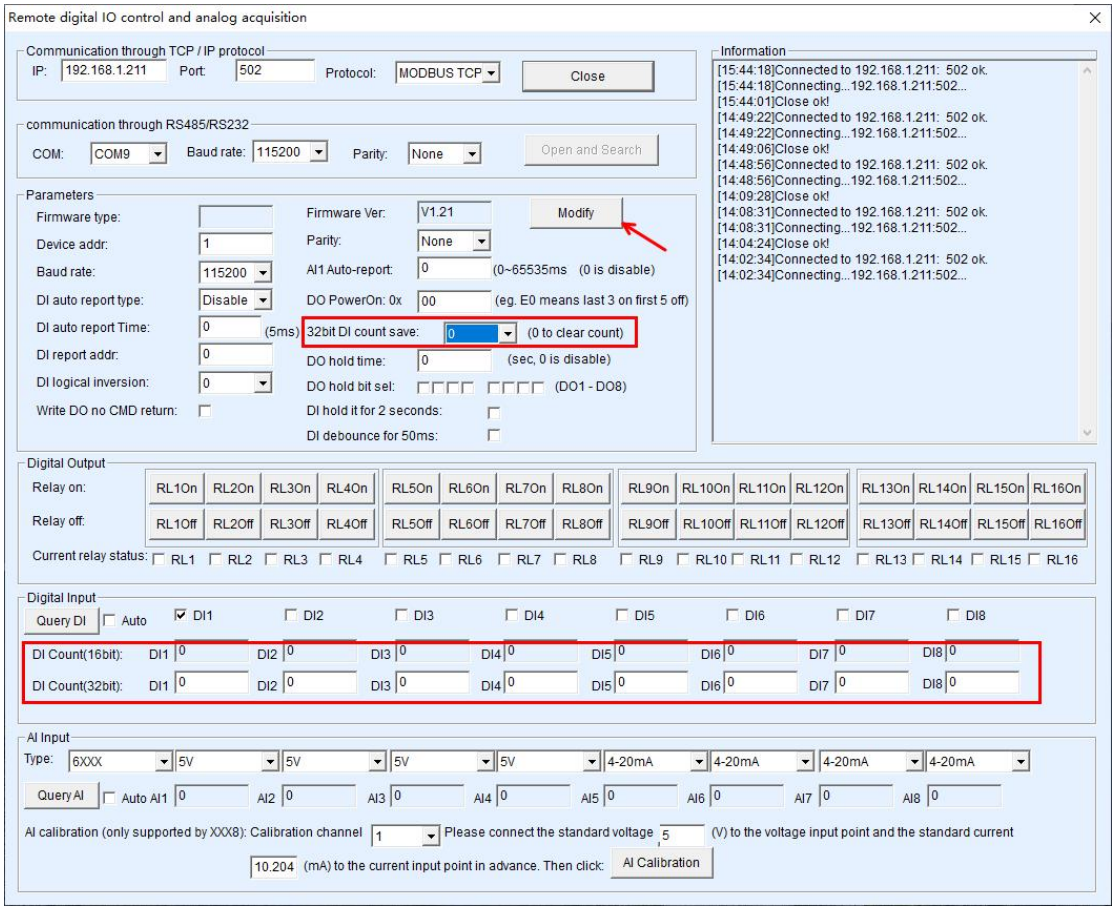
One cycle when DI changes from "0" to "1" and then back to "0" is counted as one count. DI counting is divided into three types: 16 bit no storage counting, 32-bit no storage counting, and 32-bit with storage counting. No storage starts from 0 after pointing out, and if there is storage, it will keep counting after power failure. The 32-bit non storage count and 32-bit storage count are in the same register location, but with different settings and configurations.

The DI count has automatically added debounce processing, with a debounce time of 10ms.

By using Modbus's 03 function code, reading the 0~7 register positions can result in a 16 bit no storage count, with data in big end format. By using the 03 function code, 32-bit counting can be read from positions 256 to 271, with the data bit being in the big end format.

Byte count	1	1	1	1	1	1	1	1
Name	Device address	03	High starting address	Low starting address	High length	Low length	CRC high	CRC low

For example, the Modbus RTU command to read the 16 bit count of DI8 is:
Send ->01 03 00 07 00 01 35 cb
Return ->01 03 02 01 0a 39 d3
The Modbus TCP command is:
Send ->00 00 00 00 00 06 01 03 00 07 00 01
Return ->00 00 00 00 00 05 01 03 02 01 0a
Read register 7 here, and return data with 01 0a representing the value 266.
For example, the Modbus RTU command to read the 32-bit count of DI8 is:
Send ->01 03 01 0E 00 02 a4 34
Return ->01 03 04 00 01 14 FB AC
The Modbus TCP command is:
Send ->00 00 00 00 00 06 01 03 01 0E 00 02
Return ->00 00 00 00 07 01 03 04 00 01 14
Here, 00 00 01 14 represents the value 276.



After Vircom successfully connects to the device, click on "Query DI Status" to check the DI count values, including 16 bit and 32-bit values. We found that the values of 16 bit and 32-bit are different here, because 32-bit is stored during power failure. Before this power on, the 32-bit count had already accumulated 10 values.

Use the "32-bit DI Count Save" function in the figure to set the 32-bit count to save or not save. If you want to reset the saved data to zero and start counting again. Simply set the '32-bit DI Count Save' function to 0 to clear the count.

DI controls DO function

DI controls its own DO

For the device itself, it is possible to close the corresponding DO control when DI is effective, and disconnect it when DI is not. For example, when DI1 is 1, it will control DO1 to close, and when DI1 is 0, it will control DO1 to disconnect.

Read and write through Modbus's 03/06 command, write 256 to register 72 through 06 to enable DI control of its own DO function, and write 0 to disable it. By reading the value of register 72 through the 03 instruction, the current status of the function on/off can be obtained.

Use the 06 instruction, address 72, in the following format:

Byte count	1	1	1	1	1	1	1	1
Name	Device address	06	High starting address	Low starting address	01 or 00	00	CRC high	CRC low

The Modbus RTU command is:
Open: Send ->01 06 00 48 01 00 08 4c
Return ->01 06 00 48 01 00 08 4c
Close: Send ->01 06 00 48 00 00 09 DC
Return ->01 06 00 48 00 00 09 DC
The Modbus TCP command is:
Open: Send ->00 00 00 00 00 06 01 06 00 48 01 00
Return ->00 00 00 00 00 06 01 06 00 48 01 00
Close: Send ->00 00 00 00 00 06 01 06 00 48 00 00
Return ->00 00 00 00 00 06 01 06 00 48 00 00
Use instruction 03 with an address range of 72. The format is as follows:

Byte count	1	1	1	1	1	1	1	1
Name	Device address	03	High starting address	Low starting address	High length	Low length	CRC high	CRC low

The Modbus RTU command is:
Send ->01 03 00 48 00 01 04 1C
Return ->01 03 02 01 00 B9 D4 Open
Return ->01 03 02 00 B8 44 Close
The Modbus TCP command is:
Send ->00 00 00 00 00 06 01 03 00 48 00 01
Return ->00 00 00 00 00 05 01 03 02 01 00 Open
Return ->00 00 00 00 00 05 01 03 02 00 00 Close

DI control remote end DO

In industrial control environments, there are some situations where the button switch and control equipment are far apart, and it is difficult to control the on-site wiring and cost. Therefore, it is necessary to use switch input signals to control the opening and closing of switch relays to achieve remote monitoring, which reduces labor costs and wiring troubles.

There are several application configurations for implementing DI control DO:

1) Implement DI control of DO relay output in a 1-on-1 format;

2) Implement DI in multiple forms to simultaneously control the output of multiple DO

relays;

The above application requires at least 2 devices to implement, such as DI1 of device A corresponding to DO1 of device B, and so on; Multiple transmission methods can also be considered, such as RS485, Ethernet, WiFi, Lora, 4G, etc. The appropriate solution can be selected based on the actual distance requirements and environmental conditions of the customer's site.

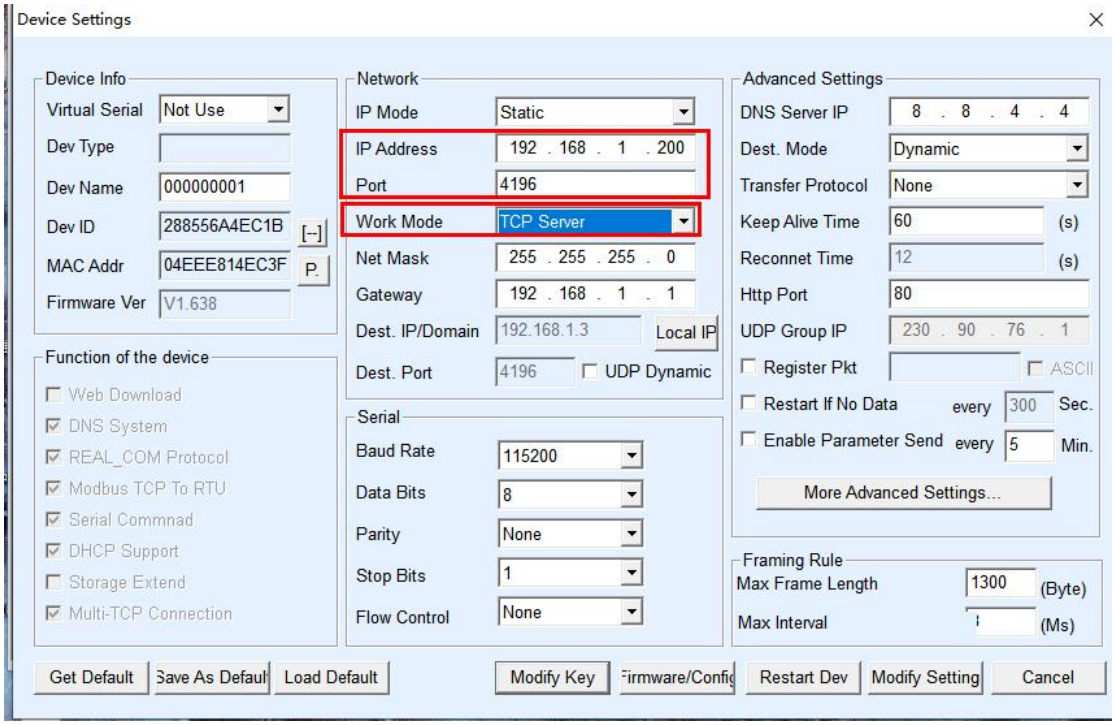
Configuration aspect (network version control scheme)

For Ethernet/WiFi classes: the network requires device pairing, where one device operates in TCP server mode and the other operates in TCP client mode, with the destination IP and port pointing to the IP and port of the server mode device. If it is in WiFi format, the parameters for connecting to the WiFi network need to be configured;

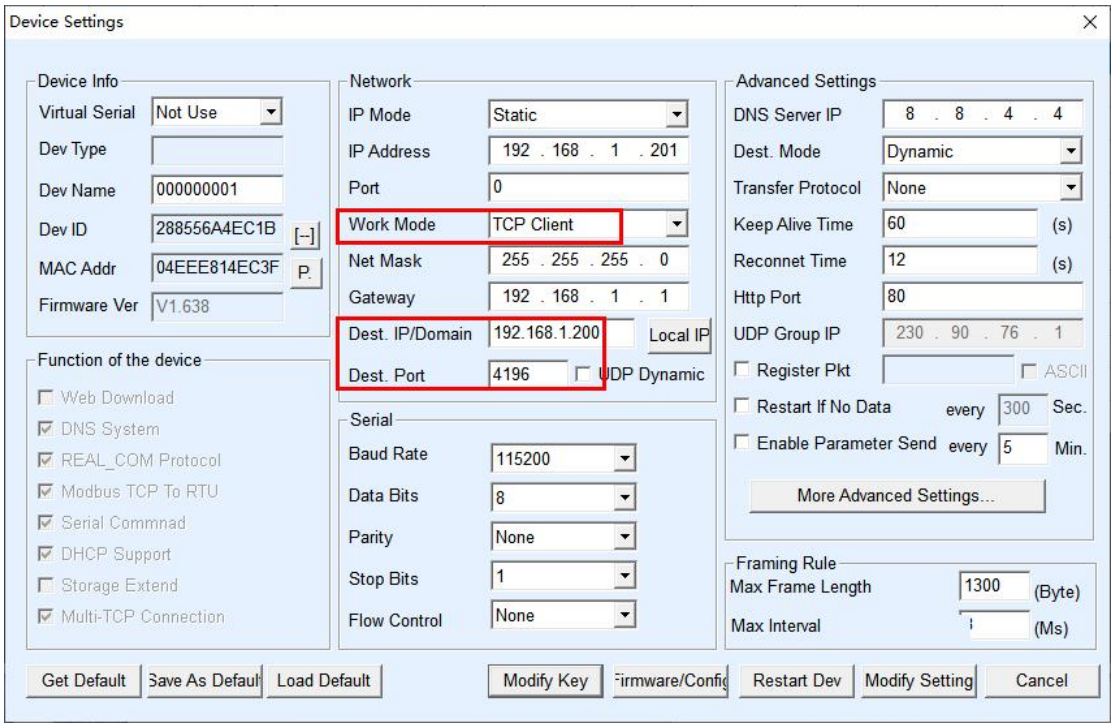
If it is in Lora form, Lora parameters need to be configured; If it is 4G, then both devices should be connected to a fixed IP public network server for cloud data M2M transfer. Specific models and configurations can also be consulted with after-sales engineers.

For Ethernet/WiFi devices: Connect the device network cable directly to the computer - Open Vircom software, click on Device Management - Auto Search - Double click to select and enter the device settings interface

Device A



Device B



The IP address needs to be consistent with the network segment of the switch/router/network port device connected to the gateway network cable, with no conflicting tail numbers, consistent gateway subnet masks, and can only be pinged

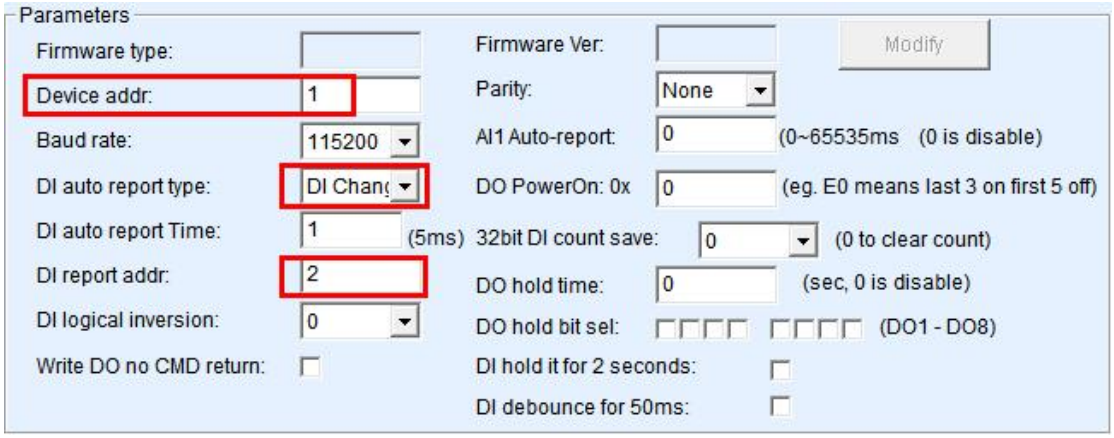
1. When clicking on automatic search - displaying TCP established status, it indicates successful network pairing; When the link lights of both devices are constantly blue, it indicates that the TCP transmission channel has been successfully established.

If it is RS485 mode, just look at the IO parameter pair set later.

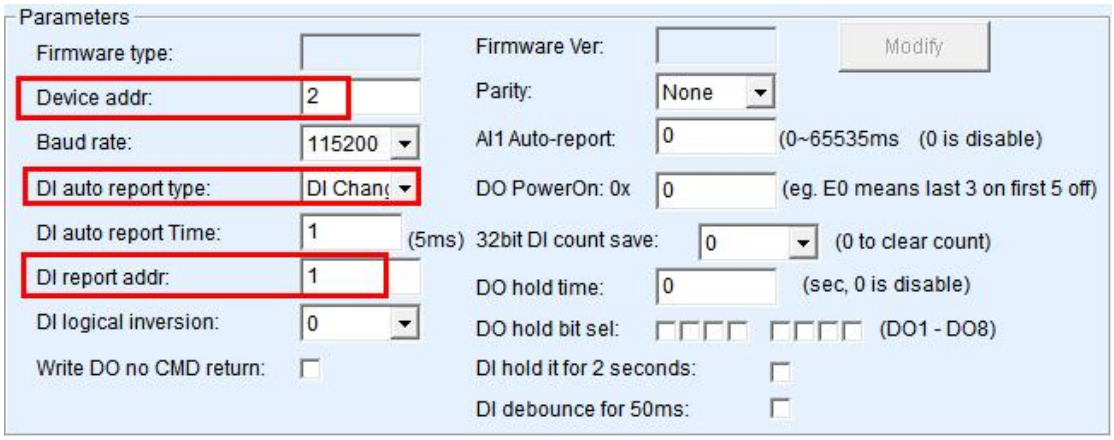
Set IO parameter pairs

Connect the device network cable directly to the computer - open the vircom software, click on Device Management - IO Controller - enter the IO Controller device settings interface; Since this can only be in client mode, it is necessary to first change all devices to TCP server mode. Enter the IP and port of the corresponding device below, click connect to enter the settings interface

Connect the A-end device to the DI switch input signal, and set it as shown in the following figure



Connect the B-end device to the DO switch relay output signal, set as shown in the following figure



If two devices A and B are needed to implement two sets of DI control DO, such as A's DI controlling B's DO, and B's DI controlling A's DO, the parameters in the red box can also be set to point to device A.

If the switch input of the A-side device DI needs to trigger the control of multiple switch outputs at the same time, it is a one to many application that only supports one-way control (A to many data flow) and cannot be controlled bidirectionally. In terms of configuration: Multiple switch outputs are configured with one device, which can be connected to the IP and port of device A in client mode. In terms of IO control, the device address is set to 2, and the DI of device A actively reports, filling in 2 for the reporting address.

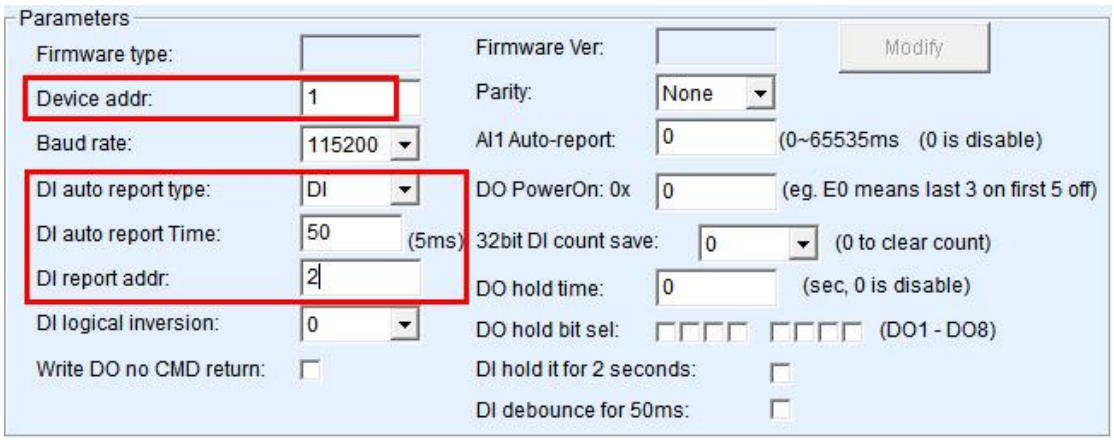
The above configuration can achieve DI triggered control of DO opening and closing, realizing remote control.

Scenario optimization plan

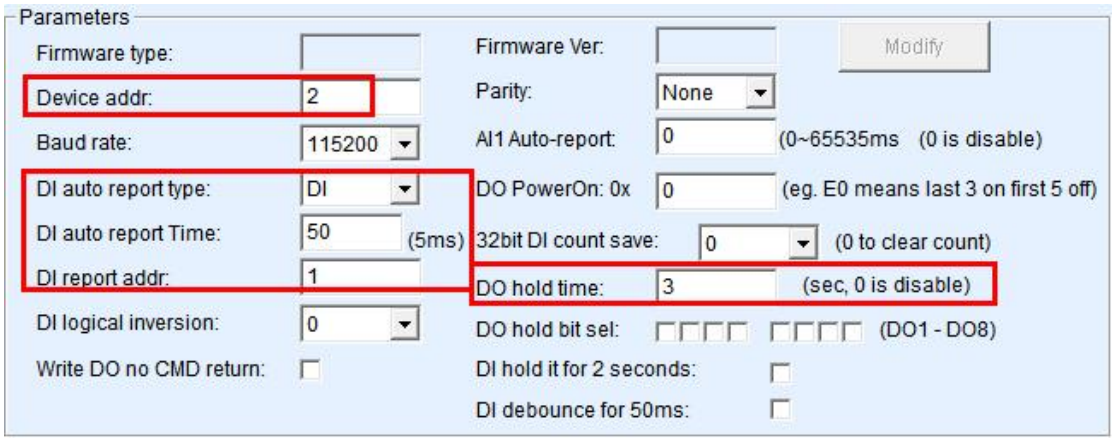
As shown above, in the application of DI controlled DO, if only one end of DO is powered off, then powering on again will result in inconsistent control.

DI collects and reports the address of DO device 2 at a fixed time of 1 second, and selects the DI reporting type as "DI" (i.e. reporting time is an even non-zero number). Adjust the

reporting time to 50, and the actual time is 50* 5=250ms. The fastest time should not be less than 100ms, which means the setting should not be less than 20.



The DO terminal is equipped with a hold function. If there is a disconnection or power failure at the DI terminal, the DO relay will be disconnected within 3 seconds, allowing for reliable DI control of DO.



AI high-precision reading

Compared with ordinary accuracy, higher precision AI numerical calculation methods do not automatically filter small fluctuations to zero voltage, and do not automatically set values with very small changes to the last collected voltage. So it can more accurately reflect the voltage value, but there may be more noise present.

By using the 04 function code to read the contents of registers 32~39 (0x20~0x27), high-precision AI values can be obtained.

The data format is the big format.

The method for calculating the input point voltage or current is:

Vx = ((Vh)/65536.0)*4.096*(Vi)+(Oi)

Among them, Vi (i=1-8) is the adjustment coefficient for each channel, which

defaults to 1.0. You can use the 03 function code to read registers starting from 0x4a~0x59 (decimal 74~89), and obtain floating-point (float) big endian format data corresponding to V1~V8. For example, the result of reading float type data of 1.063 is hexadecimal in the form of 0x3F88 1062.

Oi (i=1-8) is the adjustment offset for each channel, which defaults to 0.0. You can use the 03 function code to read registers starting from 0x103~0x112 (decimal 259~274) and obtain floating-point (float) format data corresponding to O1~O8, respectively. For example, the result of reading float data of 0.1234 is hexadecimal in the form of 0x3DFC B923.

For example, reading the adjustment coefficient of A8:

Send now ->01 03 00 58 00 02 45 d8

Return ->01 03 04 3f 80 00 00 f7 CF

The 3f 80 00 00 represents 1.0.

Read the adjustment offset of A8:

Send now ->01 03 0111 00 02 95 f2

Return ->01 03 04 3DFCB92305E6

The 3D FC B9 23 represents 0.1234.

Read the Vh of the 8th channel again:

Send now ->01 04 00 27 00 01 81 c1

Return ->01 04 02 4d c60c 32

Among them, 4d c6 represents 19910, and by substituting it into the formula, the voltage can be obtained as:

((19910)/65536)*4.096*(1.0)+0.1234= 1.367775V。

The current value is

Vi and Oi are calibrated after leaving the factory, ensuring the accuracy of the calculated values of the product.

The values of Vi and Oi are different for different AI signal readings, with 0-5V coefficients generally around 2 and 4-20mA coefficients around 10. Just read and calculate the signal type correctly according to the purchasing requirements of the order.

AI Voltage and Current Floating Point Value Reading

Compared to ordinary accuracy, it does not require any calculations to read real collected data.

By using the 04 function code to read the contents of registers 96~111 (0x60~0x6F), AI floating-point values can be obtained.

The data format is big endian format, 32-bit standard IEEE-754 floating-point number, and the parsing order is ABCD. This is

For example, reading the voltage value of AI1:

Send now ->01 04 00 60 00 02 71 d5
Return ->01 04 04 3f 80 00 00 f6 78
Where 3f 80 00 00 represents 1.0, the voltage obtained is 1.0V

AI voltage and current raw floating point value reading

The raw floating-point value reading of AI is mainly used for recalibration when there are problems with AI calibration.
By using the 04 function code to read the contents of registers 160~175 (0xA0~0xAF), the AI raw floating-point value can be obtained. The data format is big endian format, 32-bit standard IEEE-754 floating-point number, and the parsing order is ABCD.
For example, reading the raw voltage value of AI1:
Send now ->01 04 00 A0 00 02 71 E9
Return ->01 04 04 3f 80 00 00 f6 78
Where 3f 80 00 00 represents 1.0, the voltage obtained is 1.0V

AI engineering quantity floating-point value reading

AI engineering quantity floating-point value reading is mainly used for instruments and other equipment. For example, if the current value is 4-20mA and the corresponding engineering quantity is -20-120 degrees Celsius, then after setting the mode of the AI channel and the upper and lower limits of the engineering quantity (please refer to the parameter table for register positions), use the 04 function code to read the contents of registers 224~239 (0xE0~0xEF) to obtain the floating point value of the AI engineering quantity. The data format is big endian format, 32-bit standard IEEE-754 floating-point number, and the parsing order is ABCD.
For example, if the working mode of AI1-8 is first set to 0-20MA, then each channel can be set to 02 (refer to the parameter table):
Sent: 0110 01430004 08020202 02 02 02 02 4b 22
Return: 01 10 01 43 00 04 31 e2
Set the lower limit of AI1's engineering quantity to -20.123 degrees Celsius and hexadecimal to C1 A0 FB 96
Sent: 01 10 01 53 00 02 04C1 A0 FB 96 45 96
Return: 01 10 01 53 00 02 b0 25
Set the upper limit of AI1's engineering quantity to 120.123 degrees Celsius and

hexadecimal to 42 F0 3E FA
Sent: 01 10 01 93 00 02 04 42 F0 3E FA 33 2e
Return: 01 10 01 93 00 02 b0 19
Read the engineering quantity value of AI1:
Instant send ->01 04 00 E0 00 02 70 3d
Return ->01 04 42 dc3e f9 fe24
Among them, 42 dc 3e f9 represents a work quantity of 110.123

AI Register Summary

The AI high-precision floating-point series is designed to support versions including V21 and above. Currently, the mainstream existing ones are 8-channel, and products with 16 or more channels can be customized and implemented.

Function code (decimal/hexadecimal)	**Function**	**Channel**	**Address range of Route 8/16**	**Remark**
01/02（0x01/ 0x02）	Read DI status	8	0~7	
		16	0~15	
	Read DO status	8	16~23	
		16	16~31	
05 （0x05）	Set up single channel DO	8	16~23	
		16	16~31	
15 （0x0F）	Set up multiple DOs	8	16~23	
		16	16~31	
04 （0x04）	Read AI basic values	8	0~7	Only Vircom is effective, users do not need to read this register, there is a loss of accuracy
		16	0~15	
		32	0~31	
	Read the raw collected values of AI	8	32~39	The final voltage or current value can only be obtained by combining coefficients and analog quantities according to the formula
		16	32~47	
		32	32~63	
	Read the value	8	64~71	

	calculated by AI translation and scaling	16	64~79	
		32	64~95	
	Read the calculated values of AI coefficients and offsets	8	96~111	32-bit standard IEEE-754 floating-point number, with parsing order ABCD, can be used to directly read current and voltage values.
		16	96~127	
		32	96~159	
	Read the value of AI without coefficient and offset calculation	8	160~175	32-bit standard IEEE-754 floating-point number, parsing order ABCD. Can be used to read calibration coefficients and offsets.
		16	160~191	
		32	160~223	
	Read the floating point number of the engineering quantity	8	224~239	32-bit standard IEEE-754 floating-point number, parsing order ABCD, only meaningful after setting the upper and lower limits of the engineering quantity.
		16	224~255	
		32	224~287	

Serial Modbus command setting parameters

This content mainly involves the technical details of parameter setting and reading. And it helps users to configure and modify parameters with their own software. For regular applications, this section can be skipped.

List the separate columns related to parameter reading and setting from the

register table as follows

Function code	Function	Address range
03（0x03）	Read basic parameters	63~67
	Read extension parameters	68~162
	Read DI 16 bit count	0~7
	Read DI 32-bit count	256~271
06（0x06）	Set basic parameters	63~67
	Set extension parameters	68~162
	Set DI 16 bit counter	0~7
	Set DI 32-bit counter	256~271
16（0x10）	Set multiple basic parameters	63~67
	Set multiple extension parameters	68~162
	Set up multi DI 16 bit counting	0~7
	Set up multi DI 32-bit counting	256~271

Basic parameter register

Register address	Parameter name	Length (in bytes)	Functional Description
63(0x3F)	Addr/device address	1	High byte of register value
	UpLoad/Enable DI Active Reporting	1	The low byte of the register value, 1 indicates enabled, and 2-255 indicates periodic transmission.
64(0x40)	Dstaddr/DI reporting address	1	High byte of register value
	Baud/device baud rate	1	The low byte of the register value is only set to the baud rate of 12000 for the RS485 interface of 485-IO; 2400 1； 4800 2； 9600 3； 19200 4； 38400 5； 57600 6； 115200 7.
65(0x41)	Ver/firmware version	1	High byte of register value, read-only

	Composite parameter settings	1	The low byte of the register value. Bit0: 32-bit DI count save, 1 indicates save; Bit1: DI logic inversion, where 1 represents inversion; Bit2: DI delay function, after DI changes to 1, it remains at 1 for 2 seconds after the DI input changes to 0, meaning that DI can still be read as 1 within 2 seconds. Bit3: After selecting this feature, DI adds 50ms of anti shake function.
66(0x42)	A1UpLoadH/AI reporting cycle high byte	1	High byte of register value
	A1UpLoadL/AI reporting cycle low byte	1	Low byte of register value
67(0x43)	A2UpLoadH/AI reporting cycle high byte	1	The high byte of the register value should be set to the same value as A1UpLoadH
	A2UpLoadL/AI reporting cycle low byte	1	The low byte of the register value, please coordinate with A1UpLoadL is set to the same value

AI reporting time, in units of MS, time 1 register 0x42, time 2 register 0x43, both ranging from 0-65535ms. The specific reporting time is based on the larger of the two. If the time of 0x42 is 1000 and the time of 0x43 is 2000, then the AI reporting time is 2000ms once.

Extended parameter register

Register address	Parameter name	Length (in bytes)	Functional Description
68(0x44)	Dostate/DO configuration after power on	1	The high byte of the register value, 0xF0 indicates that the last 4 paths are pulled in

	Checkb/checksum	1	Low byte of register value, 0: no checksum; 1: Odd verification; 2: Occasional verification; 3: Marking; 4: Space.
69(0x45)	Baud rate of network communication and 485 Net	1	The high byte of the register value is currently read-only and adapts through the network module without the need for setting.
	Datab/data bit	1	The low byte of the register value is reserved for future expansion.
70(0x46)	Stopb/stop position	1	The high byte of the register value is reserved for future expansion.
	TCP_LINKFLAG/Reserved	1	The low byte of the register value is reserved for future expansion.
71(0x47)	Firmware Type/Device Type	1	The high byte of the register value, of type 10:485; 11: Ethernet type.
	DO holding time	1	The low byte of the register value and the duration for which the DO state is maintained.
72(0x48)	DI controls its own DO	1	The first bit of the high byte of the register value (Bit0): 1: enabled; 0: Close.
	Maintain the DO state before the last power outage after powering on	1	The second bit of the high byte of the register value (Bit1): 1: enabled; 0: Close.

	AI actively reports to report high-precision values instead	1	The third bit of the high byte of the register value (Bit2): 1: enabled; 0: Close.
	Write DO with no return instruction	1	The fourth bit of the high byte of the register value (Bit3): 1: enabled; 0: Close.
	Reserve/Reservation	1	The low byte of the register value is reserved for future expansion.
73(0x49)	Reserve/Reservation	2	No parameters
74~89(0x4a~0x59)	V1~V8 are the adjustment coefficients for each AI channel	32	For large format data, please refer to the "AI High Precision" chapter for details.
90(0x5a)	AI calibration status	2	1 indicates that it is in AI calibration state
91~106(0x5b~0x6a)	32-bit counting	32	There are a total of 16 registers, 8 DIs, each with 2 registers.
107(0x6b)	Single/multiple DO hold	2	Set whether DO1-DO8 single/multiple channels are maintained (valid for versions V1.26 and above, reserved for DO9-16)
108~123(0x6c~0x7b)	Adjustment coefficient for each AO channel, float type	32	External use table
124~130(0x7c~0x7b)	Reserve	14	There are a total of 7 registers
131~162(0x83~0xa2)	DI combination for DO logic control	32	There are a total of 16 registers.

163~210(0xa3~0xd2)	AI translation and scaling	48	Support translation (subtraction, 2 registers), scaling (multiplication, 2 registers), current voltage coefficient (2 registers).

Contact Us

Phone/WhatsApp: +86 18321985506
Web: www.valtoris.com
Email: support@valtoris.com