8CH-IO-ETH
8CH-IO-LTE
User Manual

Figure 1 8CH-IO-ETH

Contents

1. Overview

They are 8-channel remote I/O controller supports 8 channels of DI/DO/AI, namely digital inputs, relay outputs, and analog inputs (including voltage and current).

It also supports serial server functionality, allowing connection of third-party RS485 data acquisition devices and controllers to the RS485 interface for remote control.

The DI inputs support both dry and wet contacts with optocoupler isolation; the DO outputs are relay outputs with a control capability of 5A 250V AC or 5A 30V DC; the first four AI inputs support 0-5V voltage inputs, and the last four support 4-20mA current inputs, with a 12-bit ADC accuracy. The AI attributes can be modified as needed, such as 5V voltage, 10V voltage, current type, or resistive type.

Model	Communication	Network interface	Support protocol	Instructions
8CH-IO-ETH	Ethernet TCP UDP/RS485	Ethernet	Modbus TCP/ Modbus RTU/ JSON/ MQTT	
8CH-IO-LTE	4G CAT1/GPRS/RS 485	4G CAT1	Modbus TCP/ Modbus RTU/ JSON/ MQTT	Support 4G CAT1 communication or 2G communication

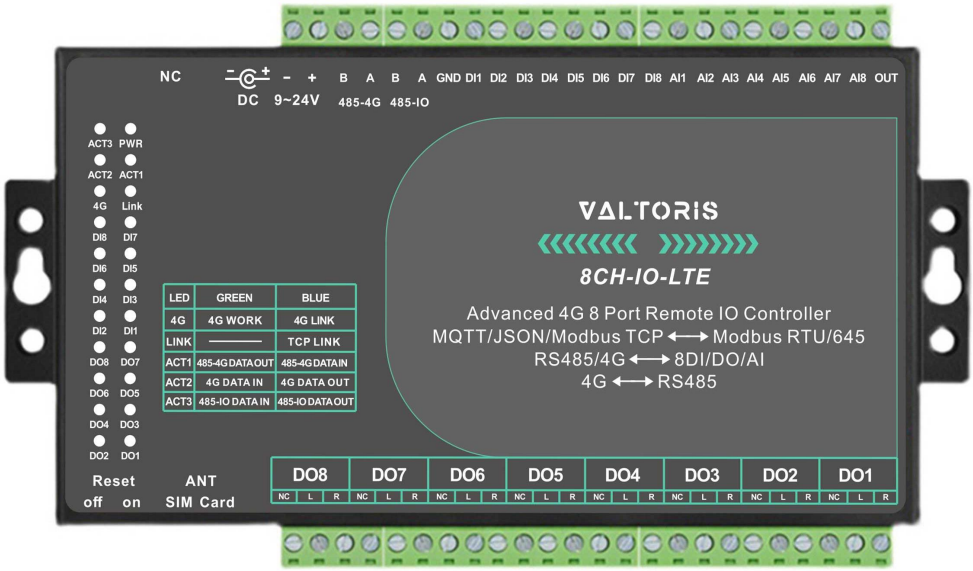

Figure 2 8CH-IO-LTE front view

8-channel remote I/O controller is divided into 4 kinds of external interfaces:

1. **485-IO**: This is an RS485 port through which DI/DO/AI can be read, written, and controlled. Through it to achieve local RS485 control, communication protocol support Modbus RTU protocol. This interface can be searched and configured through Vircom's "IO Controller" dialog.
2. **Network interface**: This interface is a remote control communication mode can be 4G CAT1, Ethernet etc.
3. **485-4G**: RS485 interface, all data from the network interface will be sent to this serial port output. Instead, the serial data received from this interface is forwarded to the network. In addition to the remote IO control function, the controller also supports the serial port server function, which can be connected to various collection and control devices on the 485-4G interface. This interface can configure parameters of the communication module through Vircom's "serial search" function.
4. **DI/DO/AI**: This is an external control interface that can be controlled by 485-IO and network interfaces, but cannot be controlled by 485-4G.

8CH-IO-LTE is the 4G version, with a special watchdog circuit, which can ensure the stable operation of 4G modules for a long time. The default baud rate of the model is 115200bps.

8CH-IO-ETH / 8CH-IO-LTE can be used in:

- Building/access control/security control system;
- Industrial automation System;
- Internet of Things, remote meter reading, information collection, etc.

8CH-IO-ETH (Ethernet interface) is used as an example. Figure 3 shows the typical application connection. Connect the field input and output devices to the **8CH-IO-ETH**, and then connect the **8CH-IO-ETH** to the network via a network cable. Then the upper computer can send data to **8CH-IO-ETH** through Modbus TCP protocol to realize the query input device and control output device.

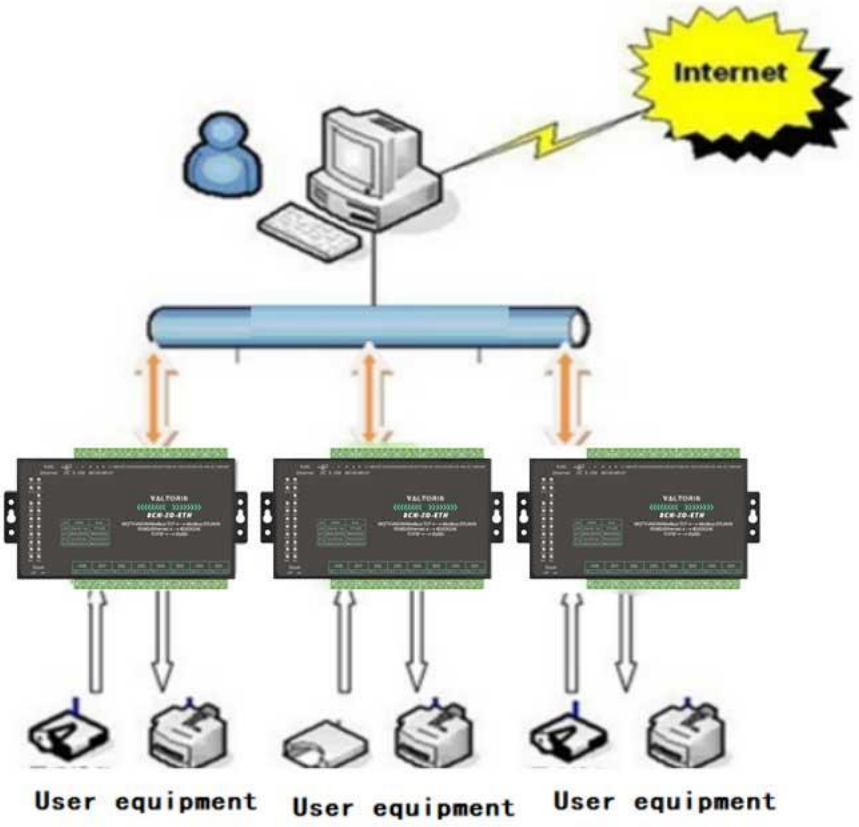

Figure 3 8CH-IO-ETH Connection case

2. Feature

1. Supports eight DI/DO/AI channels and can be controlled remotely or locally.
2. AI supports 12-bit accuracy, and the data is adjusted to ensure accuracy.
3. Supports the serial port server function to control external third-party RS485 devices over the network.
4. Support DI control DO function, using a pair of controller through 4G and other communication methods can control each other, easy to use.
5. Sub-models support 4G/CAT1/RS485/ Ethernet and other communication media.
6. Support Modbus TCP, Modbus RTU, MQTT, JSON, HTTP and other communication modes.
7. Connect to various public clouds, send data in JSON format, and control delivery in JSON format.
8. Rich indicators: display DI, DO status, network status, data flow status, etc.
9. Provide IO controller dialog box or Remote IO of Vircom control demonstration software through RS485 or TCP/IP control, which can demonstrate IO control and AI data acquisition of equipment.
10. It can provide complete RS485 control instructions and Modbus RTU instructions,

which is convenient for engineers to integrate development.

11. You can restore factory Settings with one click, including baud rate, station address, network configuration of communication module, etc.

3. Technical Parameters

Table 1 Technical Parameters

External interface	
dimension：	L x W x H =9.2cm×19.7cm×2.5cm
Serial port parameter	
485-IO Baud rate: The default baud rate is 115200bps, which can be changed by using the Remote IO software or commands. 485-4G baud rate: 115200bps. Data bit: 8 bits. Check bit: No check, odd check, even check. Stop bit: 1 bit	
software	
Network protocol:	MODBUS TCP/MQTT/JSON/HTTP
RS485 protocol:	MODBUS RTU
AI input form	
Current input: 4~20mA Voltage input: 0~5V, 0~10V (need to customize) Resistance input: 0~10K, resistance type temperature and humidity sensor, etc. (need to be customized)	
Power consumption (relay non-draw state)	
Running stable state: 30mA@12V 4G dial status: 60mA@12V DO relay closed, DI input closed (maximum power consumption) : 300mA@12V	
8CH-IO-LTE (4G CAT1) Parameters	
Transmission rate	LTE：Max 10Mbps（down）/Max 5 Mbps（up） GPRS：85.6Kbps（down）/Max85.6Kbps（up）
Support band	B1/B3/B5/B8@FDD LTE B34/B38/B39/B40/B41@TDD-LTE B3/B8@GSM
SIM	Voltage: 3V, 1.8V; Size: large card (small card can be purchased to use)

Antenna interface	50Ω/SMA glue stick antenna or suction cup antenna is optional.
8CH-IO-ETH (Ethernet) parameters	
Ethernet	10/100M adaptive Ethernet can be connected
Environmental requirement	
Operating temperature:	-40~85℃
Storage temperature:	-45~165℃
Humidity range:	5~95%Relative humidity

4. Instruction

4.1 Hardware specifications

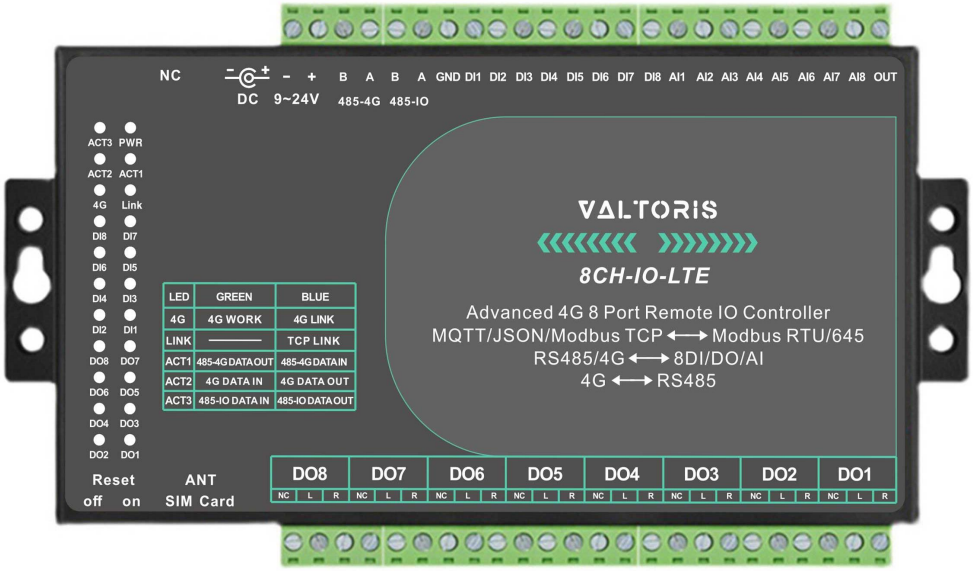

Figure 4 8CH-IO-LTE

The ports on the top of 8CH-IO-LTE are shown as follows:

Table 2 Ports on the upper side

Terminal	Feature
RJ45/NC	8CH-IO-ETH (Ethernet version) : 10M/100M Ethernet interface for remote IO control over TCP/IP. 8CH-IO-LTE: NC, invalid.
DC	DC plug type power input, supply voltage 9 ~ 24V
Power terminal	Terminal type power input, power supply voltage 9 ~ 24V, and DC terminal can be selected to intervene the power supply.

485-4G	RS485 port for transparent transmission of network and serial port, realizing the function of serial port server.
485-IO	RS485 port used to control device I/O and collect DI and AI information.
GND	When entering a dry node, switch the jumper between this terminal and DI1 to DI8 to collect the switch status.
DI1～DI8	8 switch inputs
AI1～AI4	Four 0 to 5V voltage inputs
AI5～AI8	Four 4 to 20mA current inputs
OUT	Test output point, can output 5V level, generally not used.

Bottom ports:

Table 3 Ports on the lower side

Interface	function
ANT	8CH-IO-LTE(4G) : The antenna interface adopts 50Ω/SMA (female head), and the external antenna must use an antenna suitable for 4G operating band. we can provide 4G Suction antenna.
Reset	After you dial ON, the TCP indicator blinks, and then dial back. The device restores to the default Settings. The default baud rate are 115200bps.
SIM Card	When installing the SIM card, ensure that the device is not powered on. Use a pen tip or screwdriver to push the SIM card out of the slot and push the SIM card face down into the slot.
DO8～DO1	R and L represent the 2 contacts of the relay respectively, where 8 relay outputs are represented. The NC is not connected.

1. 8 Digital Input DI1 to DI8.

Passive switching (dry nodes) and active levels (wet nodes) are supported. The dry node only needs to short-circuit it with GND to collect the 1 signal. When the node is wet, the range of difference between active level and GND is as follows:

VCC voltage	Low level range	High level range
24V	0～17V	17～24V
9V	0～3V	3～9V

2. 8 Digital output DO1 to DO8.

The output type is relay output (5A@AC250V/DC30V). Setting 1 indicates that the relay is closed.

3. **8 analog inputs**

The accuracy is 12 bits. By default, the first four inputs are 0 ~ 5V voltage inputs, and the last four inputs are 4~20mA. Any path can be modified in the following way (need to be customized) :

1) Current signal input: 4~20mA.
2) Voltage signal input: 0~5V.
3) Voltage signal input: 0~10V.
4) Resistance impedance input: such as 0~10k or resistance type temperature and humidity sensors.

4. Both voltage and current are relative to GND.

Panel light

Table 4. Indicators

Indicator	Indicator name	green	blue
PWR	Power indicator light		
ACT3	IO communication light	485-IO Interface data input	485-IO interface data is returned, indicating that the sent IO control instruction was correctly identified.
ACT2	Network/telecommunication indicator	The network end (such as 4G) receives data	The network side (such as 4G) sends data; This indicator blinks during initialization, indicating that during initialization, the indicator is turned off after initialization.
ACT1	Serial communication indicator	485-4G Data output of the RS485 port	485-4G RS485 port Data input
4G	4G connection indicator	8CH-IO-LTE (4G) : Green nonsense.	8CH-IO-LTE (4G) : Blinking blue indicates that the dial is in progress. Steady blue indicates that the dial is successful. The dial starts 5 seconds

			after the system is powered on.
LINK	TCP connection Indicator	8CH-IO-LTE (4G) : Green nonsense 8CH-IO-ETH (Ethernet) : indicates that the network cable is properly connected.	Steady on when the TCP connection is established. The Reset button of the device is in the reset state. The reset succeeds and blinks blue for 3 seconds.
DI1～DI8	DI indicator light	On indicates that the input is low or closed.	
DO1～DO8		On: The relay is closed.	

4.2 DI/DO/AI Function description

4.2.1 Connecting Devices Using Vircom

8CH-IO-ETH are configured through Ethernet interfaces.

8CH-IO-LTE are configured through RS485 interfaces.

Power on the device and connect the 485-IO port and network port.

Open the main screen device management, if the Ethernet interface model, click the "automatic search" button to find the device, click one of the devices, and then click "IO Controller". For an RS485 port device, tap IO Controller.

Figure 5 How do I go to the IO Controller dialog box

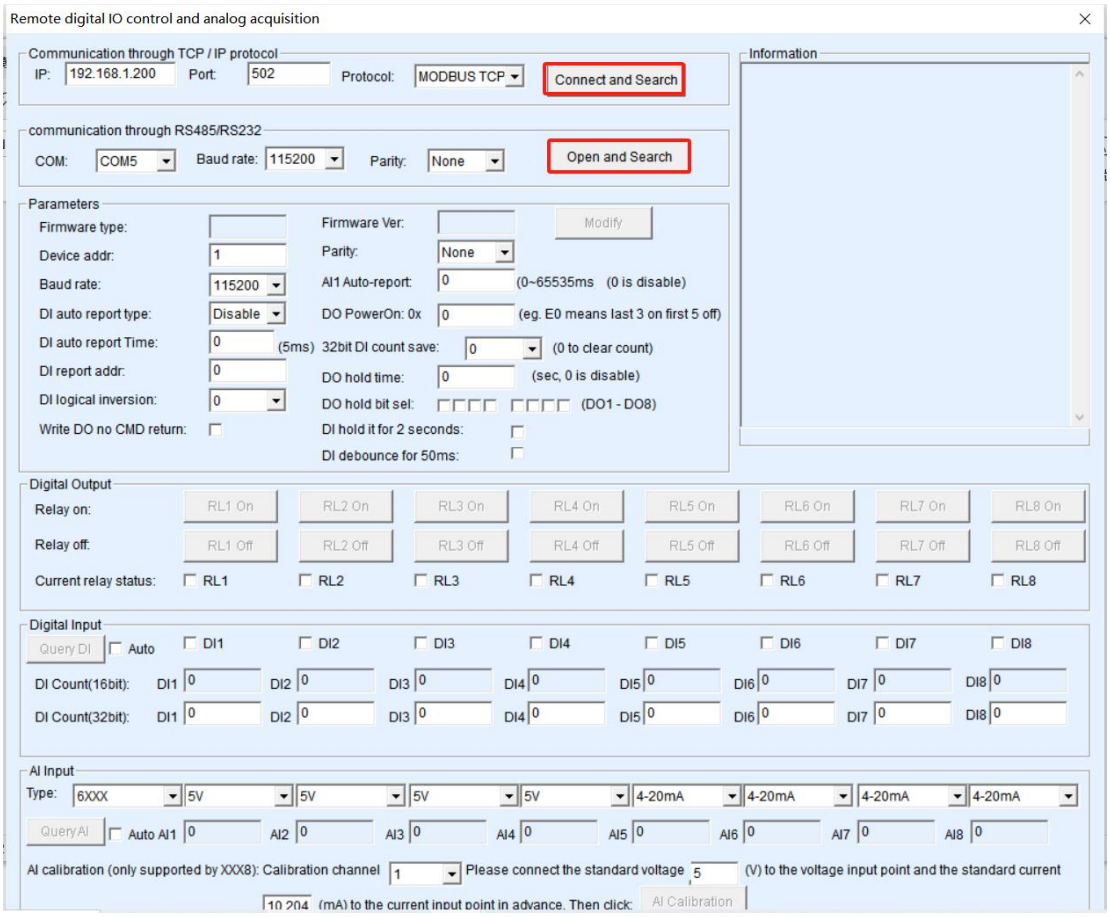

Figure 6 IO Controller dialog box

If it is a network type device, you can connect the device through the "Connect and Search" or "Open and Search" button. It corresponds to the communication in network mode and RS485 mode. For devices in serial port mode, you can only open and search

the device in serial port communication mode.

For the network mode, the IP address and port | conversion protocol are already obtained when you select the device, just click "Connect and search". When the TCP connection is established, Vircom obtains the parameters of the device by sending Modbus TCP instructions. In some applications, you can also set the Modbus RTU protocol to communicate through the network port. At this time, you need to double-click the network device in the previous dialog box and change the "conversion protocol" to "none" to support Modbus RTU mode network communication.

For RS485 mode, only need to select the corresponding USB to 485 com port (connected to the serial cable on the computer in advance), do not need to select the baud rate. If the parity bit has been set before, select the corresponding parity bit. Then click "Open and search". After com port is opened, the parameters of the device are obtained by software Modbus RTU command.

In either case, the device gets the parameters and displays them in a dialog box. Later, you can modify parameters, DO control, DI read, AI read and other tests.

4.2.2 General Table of Modbus registers

Network interfaces support Modbus TCP commands, and serial ports support Modbus RTU commands. The specific registers and address ranges are as follows:

Table 5. Summary of Modbus registers

Function code	Function	Address range
01/02	Read DI	0 to 7 (corresponding to DI1 to DI8)
01/02	Read DO	16~23
05	Set DO	16~23
15	Set multiple DO	16~23
04	Read AI	0~7
04	Read AI high precision values	32~39
03	Read base parameter	63~67
03	Read spread parameter	68~162
03	Read DI 16 bits count	0~7
03	Read DI 32 bits	256~271

03	Multi-do Settings with masks	512
03	Read the meter parameters	1535
03	Read time	1023
06	Set parameters	63~67
06	Set extension parameters	68~162
06	Set the DI 16-bit count	0~7
06	Set the DI 32-bit count	256~271
16	Set the multi-DI 16-bit count	0~7
16	Set the multi-DI 32-bit count	256~271
16	Set basic parameters	63~67
16	Set extension parameters	68~162
0	Month day hour minute second week DO 0/1 32	544~547

The specific usage is introduced later.

4.2.3 DO Usage Instructions

DO is the control relay, through Modbus 05/15 instruction (force single coil instruction), write 1 to 16~23 register to pull the relay, write 0 to disconnect the relay. By reading the values of registers 16 to 23 with the 01 instruction, the current DO state can be obtained.

05 Command format is as follows:

Number of bytes	1	1	1	1	1	1	1	1
Name	Device address	05	Start address high	Start address low	Ff or 00	00	CRC high	CRC low

For example, the Modbus RTU command that sets DO1 to be on is:

send-> 01 05 00 10 **ff 00** 8d ff

Back-> 01 05 00 10 **ff 00** 8d ff

The Modbus TCP command is:

send-> 00 00 00 00 00 06 01 05 00 10 **ff 00**

Back-> 00 00 00 00 00 06 01 05 00 10 **ff 00**

For example, the Modbus RTU command that sets DO1 to be off is:

send-> 01 05 00 10 00 00 cc 0f

Back-> 01 05 00 10 00 00 cc 0f

The Modbus TCP command is:

send-> 00 00 00 00 00 06 01 05 00 10 00 00

Back-> 00 00 00 00 00 06 01 05 00 10 00 00

Other instructions are listed below:

- On DO2 01 05 00 11 ff 00 dc 3f
- Off DO2 01 05 00 11 00 00 9d cf
- On DO3 01 05 00 12 ff 00 2c 3f
- Off DO3 01 05 00 12 00 00 6d cf
- On DO4 01 05 00 13 ff 00 7d ff
- Off DO4 01 05 00 13 00 00 3c 0f
- On DO5 01 05 00 14 ff 00 cc 3e
- Off DO5 01 05 00 14 00 00 8d ce
- On DO6 01 05 00 15 ff 00 9d fe
- Off DO6 01 05 00 15 00 00 dc 0e
- On DO7 01 05 00 16 ff 00 6d fe
- Off DO7 01 05 00 16 00 00 2c 0e
- On DO8 01 05 00 17 ff 00 3c 3e
- Off DO8 01 05 00 17 00 00 7d ce

15 Simultaneously set the multi-coil command format as follows:

Number of bytes	1	1	1	1	1	1	1	1	1	1
Name	Device address	0x0F	Start address high	Low start address	High quantity	Low quantity	Number of bytes	Value (low bit on the right)	CRC HIGH	CRC LOW

For example, the Modbus RTU command with the first four channels on and then four channels off is as follows:

send-> 01 0F 00 10 00 04 01 0F bf 51

Back-> 01 0f 00 10 00 04 55 cd

The Modbus TCP command is:

send-> 00 00 00 00 00 08 01 0F 00 10 00 04 01 0F

Back-> 00 00 00 00 00 06 01 0f 00 10 00 04

01 Read the DO status command

Number of bytes	1	1	1	1	1	1	1	1
Name	Device address	01	Start address high	Low start address	Length height	Low length	CRC HIGH	CRC LOW

For example, the Modbus RTU instruction for reading 8 DO states is:

send-> 01 01 00 10 00 08 3c 09

Back-> 01 01 01 **0f** 11 8c

The Modbus TCP command is:

send-> 00 00 00 00 00 06 01 01 00 10 00 08

Back-> 00 00 00 00 00 04 01 01 01 **0f**

Here, 0F indicates that the first four channels are closed.

IO Controller dialog control demo:

Figure 7 DO control in the IO controller dialog box

After Vircom successfully connects the device, click RLx to turn on the relay. At the same time, the corresponding DO indicator light of the device is lit, and RL1 is ticked. The function of the RL1 selection box is to obtain the current relay state, because the TCP connection disconnection does not change the current relay state of the device, so when the first communication with the device is established, you can obtain the DO state of the device and then decide whether to close or disconnect.

Note: If there are more than 8CH IO controllers in the same use environment, please configure different station addresses, otherwise the return instruction of the DO control will be used as the control instruction of another device, and then it will return a same instruction, so repeatedly oscillating.

4.2.4 DI Usage Instructions

If the read DI is used, the 01 command is used. The address range is 0 to 7, corresponding to DI1 to DI8. The instruction format is as follows:

Number of bytes	1	1	1	1	1	1	1	1
Name	Device address	01	Start address high	Low start address	Length height	Low length	CRC high	CRC low

For example, the Modbus RTU instruction for reading 8 DI is:

send-> 01 01 00 00 00 08 3d cc

Back-> 01 01 01 **80** 50 28

The Modbus TCP command is:

send-> 00 00 00 00 00 06 01 01 00 00 00 08

back-> 00 00 00 00 00 04 01 01 01 **80**

When the DI input is low (note that when the power supply voltage of the device is above 12V, the 5V voltage input is considered low), the corresponding bit returned is 1, and the fourth byte in the return command is 0x80, indicating that the eighth circuit is closed (low).

IO Controller dialog control demo:

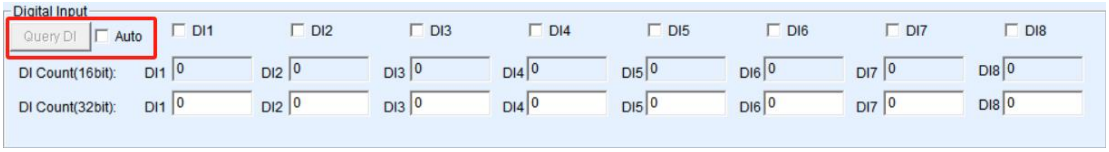

Figure 8 DI read in the IO controller dialog

After Vircom successfully connects to the device, click Query DI Status to query the DI status. When DI is low, the corresponding indicator is on and the corresponding bit returned is 1. Tick DI8 as shown in the figure, indicating that DI8 is in a low level state. Click the "Automatic" selection box to automatically query the DI status every 1 second and display it.

DI Counting Instructions

A period when DI changes from high to low and back to high is counted as a count. DI counts are divided into three types: 16-bit count without storage, 32-bit count without storage, and 32-bit count with storage. If no storage device starts from 0 after a pointer is dropped, it keeps counting after a pointer is powered off. Among them, 32-bit no storage count and 32-bit stored count are the same register location, but the Settings are different.

The DI count has been automatically added to the buffeting process, and the buffeting time is 10ms.

Through the Modbus 03 function code, you can read the 16-bit non-storage count by reading the register positions from 0 to 7, and the data is in big-endian format. Through the 03 function code, read 256~271 positions can read 32-bit count, data bit big-endian format.

Number of bytes	1	1	1	1	1	1	1	1
Name	Device Address	03	Start address high	Low start address	Length height	Low length	High CRC	Low CRC

For example, the Modbus RTU instruction for reading the 16-bit count of DI8 is:

send-> 01 03 00 07 00 01 35 cb

Back-> 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

Back-> 00 00 00 00 00 05 01 03 02 01 0a

Here register 7 is read and 01 0a of the returned data represents the value 266.

For example, the Modbus RTU instruction for reading the 32-bit count of DI8 is:

send-> 01 03 01 0E 00 02 a4 34

Back-> 01 03 04 00 00 01 14 fb ac

The Modbus TCP command is:

send-> 00 00 00 00 00 06 01 03 01 0E 00 02

Back-> 00 00 00 00 00 07 01 03 04 00 00 01 14

Here 00 00 01 14 represents the value 276.

IO Controller dialog control demo:

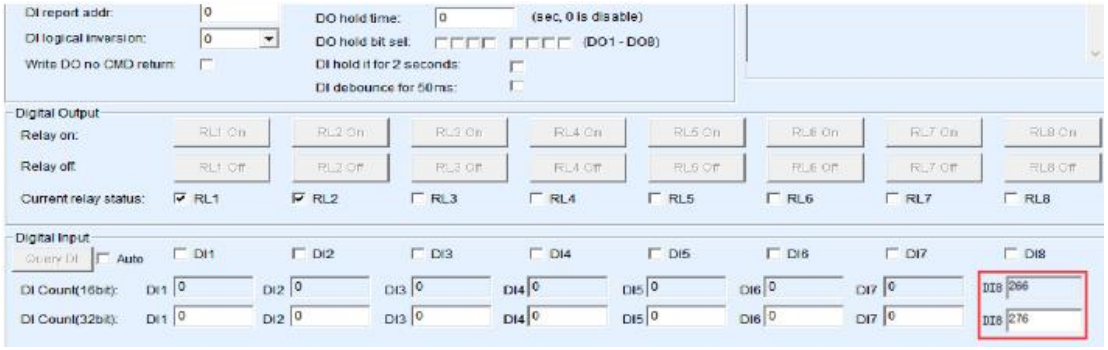

Figure 9 DI count read in the IO controller dialog

After Vircom successfully connects to the device, you can click "Query DI Status" to query the DI count value, including 16-bit and 32-bit values. It is found that the 16-bit and 32-bit values are different, because the 32-bit is stored in the power failure, and the 32-bit count has accumulated 10 values before the power on.

Use the "32-bit DI count save" function in the figure to save or not save the 32-bit count. If you want to clear the saved data, start counting again. You only need to set the

"32 Save for DI count" function to 0 to clear the count.

DI Logical Inversion

In normal condition, when the DI input is low, the corresponding bit is 1. The default DI input is high and low is valid. If the DI input is high, the default bit is 1. If the DI input is low, the default bit is 0. In this case, you can select Logical DI Reversal.

DI reversal also affects the DI count, which is when DI changes from 0 to 1, that is, the high level changes to the low level. If the DI logic is reversed, the count is increased by one when the level is changed from low to high.

The following table describes how to set DI logical inversion.

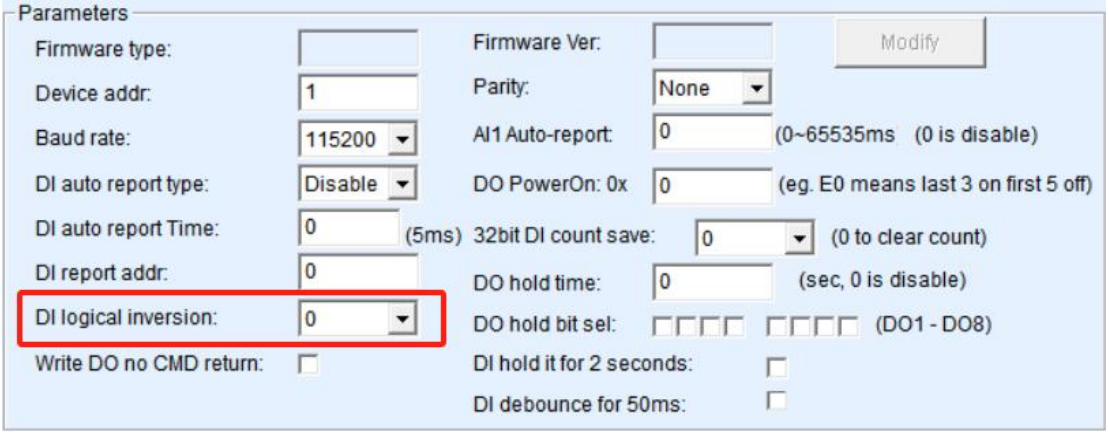

Figure 10 DI reversal Settings in the IO controller dialog box

4.2.5 AI Usage Instructions

AI can collect analog values of 0~5V, 0~10V, 4~20mA and other types. Which interface corresponds to which type is determined by the hardware at the factory. The above types of AI interfaces are defined as 5V, 10V, and 4 to 20mA respectively.

At present, standard products are divided into the following categories for AI, and the corresponding different types of AI are as follows:

Table 6 Different types of AI

Product model	Option	AI 1~AI 4	AI 5~AI 8
8CH-IO-LTE 8CH-IO-ETH	1	0-5V	4~20mA
	2	0-5V	0-5V
	3	0-10V	0-10V
	4	4~20mA	4~20mA
	5	Customization	Customization

Use Modbus 04 instruction to read the value of register 0~7, you can get the value of AI1~AI8. Data is stored in big-end format.

Number of bytes	1	1	1	1	1	1	1	1
name	Device address	04	Start address high	Low start address	Length height	Low length	CRC High	CRC low

For example, the Modbus RTU instruction to read the value of AI8 is:

send-> 01 04 00 07 00 01 80 0b

back-> 01 04 02 01 82 38 c1

The Modbus TCP command is:

send-> 00 00 00 00 00 06 01 04 00 07 00 01

back-> 00 00 00 00 00 05 01 04 02 01 82

The specific use of the returned data 01 82 depends on the type of AI. If 01 82 is converted to decimal, it is Vin/Ain=386. For different AI types, the formula is as follows:

- 5V：True voltage value = (Vin/ 1024)*5=1.8848;
- 10V：True voltage value = (Vin/ 1024)*10=3.7695;
- 4~20mA：True current = (Ain / 1024)*5/200*1000=9.4238;

IO Controller dialog control demo:

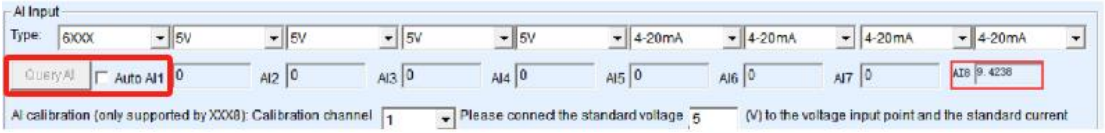

Figure 11 AI read in the IO controller dialog

After Vircom successfully connects the device, you can click "Query AI status" to query the AI value, or click "Automatic" to query once a second. Before the query, you need to select the purchased model. After selecting the model, the analog interface type of AI1~AI8 is automatically configured according to the standard configuration, so that the real voltage or current value of the interface can be displayed in the numerical dialog box.

AI uses with high precision

Our IO controller provides a higher precision AI numerical calculation method. Compared with the ordinary accuracy, no small fluctuations are automatically filtered to 0 voltage, and no small changes in the value are automatically set to the last collection voltage. So the voltage value can be more realistic, but there may be more noise.

Read the contents of 32~39(0x20~0x27) registers using the 04 function code to obtain AI high precision values. The data format is big-endian. This is a 12-bit effective precision value Vh.

The method of calculating the input point voltage is as follows:

Vi = ((((Vh)/1024)-1.0)*(Vri)*2.0)

The calculated input point current is:

Ii=((((Vh)/1024)-1.0)*(Vri)*2.0)/200

Vi (i=1 to 8) is the adjustment coefficient of each route. The default value is 1.0. Registers starting from 0x4a to 0x59 (74 to 89 decimal) can be read using the 03 function code to obtain floating-point (float) large-endian data corresponding to V1 to V8, respectively. For example, float data of 1.063 reads the result as 0x3F88 1062 hexadecimal.

For example, read the adjustment factor of A8:

Send -> 01 03 00 58 00 02 45 d8

Back -> 01 03 04 3f 80 00 00 f7 cf

Where 3f 80 00 00 means 1.0.

Read the Vh of route 8 again:

Send -> 01 04 00 27 00 01 81 c1

Back -> 01 04 02 07 c7 fa 92

Where 07 c7 represents 1991, the voltage obtained by bringing into the formula is:

((((1991)/1024)-1.0)*(1.0)*2.0)= 1.8887。

Vi adjustment coefficient is calibrated after the factory, which can ensure the accuracy of the calculated value of the product.

4.2.6 The DI is reported automatically

Our IO controller is a standard MODBUS device and is used in a question-and-answer format, but some users want to get feedback as soon as the DI input changes, that is, the active return function. This section describes the IO controller active reporting function. As shown in the figure, set Enable active DI reporting to 1 to enable active DI reporting. The IP address reported by the DI must not be the same as the device IP address. Otherwise, the 05 command is indistinguishable from the return command controlled by the DO.

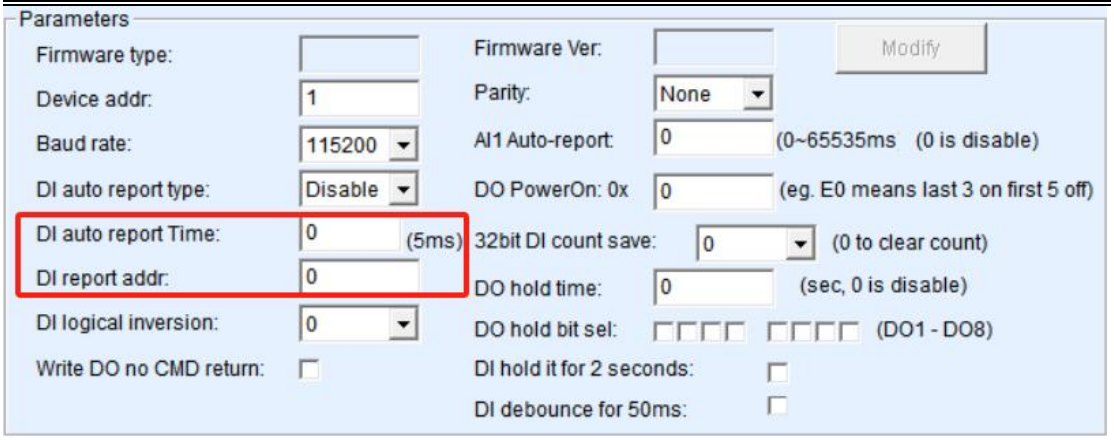

Figure 12 DI actively reports Settings

When the DI status changes, the DI sends the 05 command after active reporting is enabled. The 05 command can realize the function of the change of DI to control the trigger of the DO of another Modbus device.

Number of bytes	1	1	1	1	1	1	1	1
Name	DI report address	05	Start address high	Low start address	Ff or 00	00	CRC high	CRC low

Examples are as follows:

DI1 becomes a high level input

00 05 00 10 00 00 CD 2E

DI1 changes to low input

00 05 00 10 ff 00 8C 2E

DI2becomes a high level input

00 05 00 11 00 00 9C 1E

DI2 changes to low input

00 05 00 11 ff 00 DD EE

DI3 becomes a high level input

00 05 00 12 00 00 6C 1E

DI3 changes to low input

00 05 00 12 ff 00 2D EE

DI4 becomes a high level input

00 05 00 13 00 00 3D DE

DI4 changes to low input

00 05 00 13 ff 00 7C 2E

When Vircom is used to test, the DI is actively reported to update the current DI status. The DI initiative report is sent to both 485-IO and the network (Ethernet, 4G).

When the active reporting time is set to 0, the active reporting time is disabled. When the active reporting time is set to 1, the active reporting of DI changes is enabled. If the value is set to another value, it will be reported periodically. If the value is set to an even number, eight DI's are reported periodically based on 15 commands. If the value is set to an odd number, the DI and AI report at the same time. For details, see the following section in this chapter. If this parameter is set to n and n is a non-zero even number, the DI report time is (n-1) x 5 milliseconds. For example, configure the first four DI lines to be short connected to GND and the last four lines to be suspended to send the DI to the GND.

Send -> 01 0F 00 10 00 04 01 0F bf 51

4.2.7 AI's active reporting

The active reporting function of AI is to enable the collected analog quantity to be automatically sent to the upper computer. This method does not need Modbus instruction query on the host computer, and is very useful for network analog monitoring based on Internet.

The value ranges from 0 to 65535. The unit is ms. If the value is set to 0, active reporting is disabled. You can directly set this parameter in the IO controller dialog box.

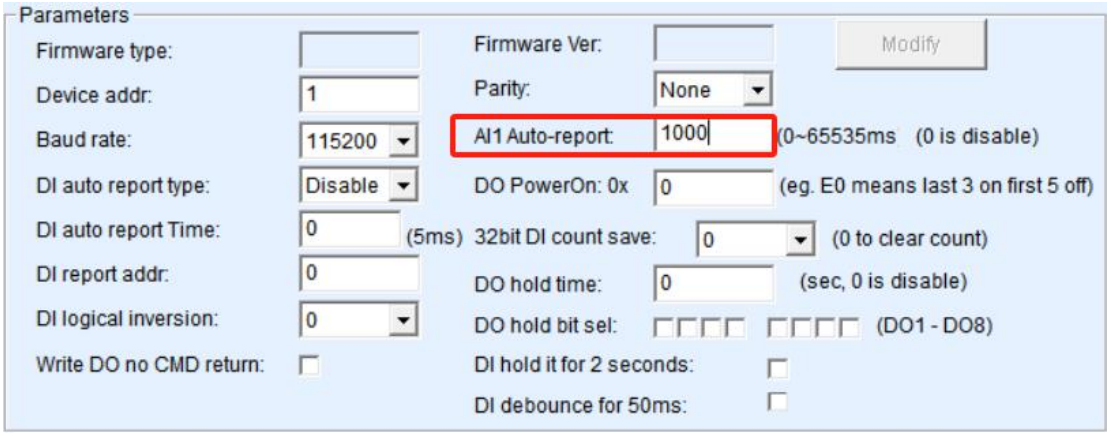

Figure 13 Setting the AI active reporting time in the IO controller

The instructions actively uploaded by AI are:

- When converting protocol to Modbus RTU:01 04 10 H1 L1 H2 L2 H3 L3 H4 L4

H5 L15 H6 L6 H7 L7 H8 L8 C1 C2

- When converting protocol to Modbus TCP:00 00 00 00 00 13 01 04 10 H1 L1 H2 L2 H3 L3 H4 L4 H5 L15 H6 L6 H7 L7 H8 L8

Here H1 L1 represents the collection amount of A1, H2 L2 represents the collection amount of A2, and so on, in big-endian format. C1 and C2 are CRC.

If there is a device parameter search before the AI initiative report, the AI initiative report will pause for 5 seconds, which can prevent the AI initiative report and parameter search conflict.

4.2.8 Uploading DI and AI at the Same Time

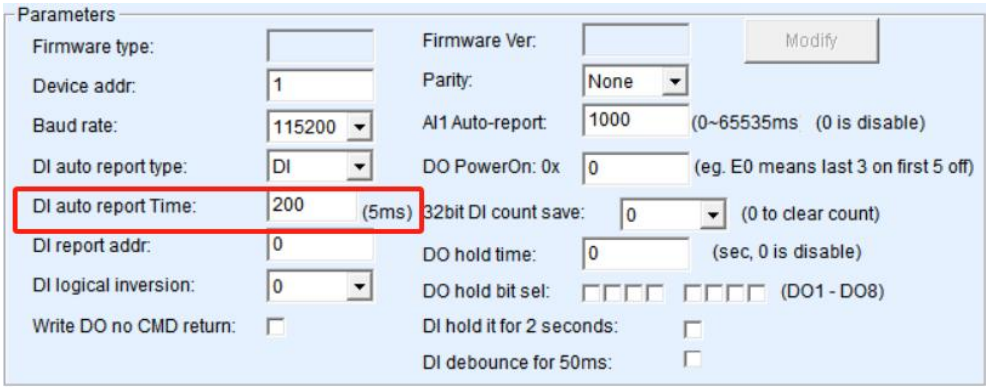

Figure 14 DI and AI actively report Settings at the same time

In the software, if the value of DI active reporting is set to greater than 1 (2 to 255), the value -1 multiplied by 5 is the period for reporting AI and DI. For example, if the value is set to 201, the reporting period is (201-1)*5=1000ms.

This function allows the current AI and DI values to be reported at the same time. The Modbus RTU format is as follows:

00 04 12 03 01 00 00 00 00 00 00 00 00 00 03 07 03 08 00 08 c3 83

The first 00 is set for the DI report address. The 04 function code is used to report eight AI registers and eight DI data. 03 01 indicates the data of AI1, and 03 08 indicates the data of AI8. 08 indicates the state of eight DI's. 08 indicates that route 4 is 1.

When the AI and DI report at the same time, the data of the AI and DI can be viewed on the IO controller page at the same time. In this case, you do not need to click Automatic to query the data. AI and DI actively report to 485-IO and network (including Ethernet, 4G) at the same time.

If a device parameter search is performed before the DI and AI report, the DI and AI report will be paused for 5 seconds to prevent a conflict between the DI and AI report and

parameter search.

4.2.9 DO Status After Power-on

Sometimes you want the IO controller to be in the on or off state immediately after powering on. Now you can set this function through the IO Controller dialog box.

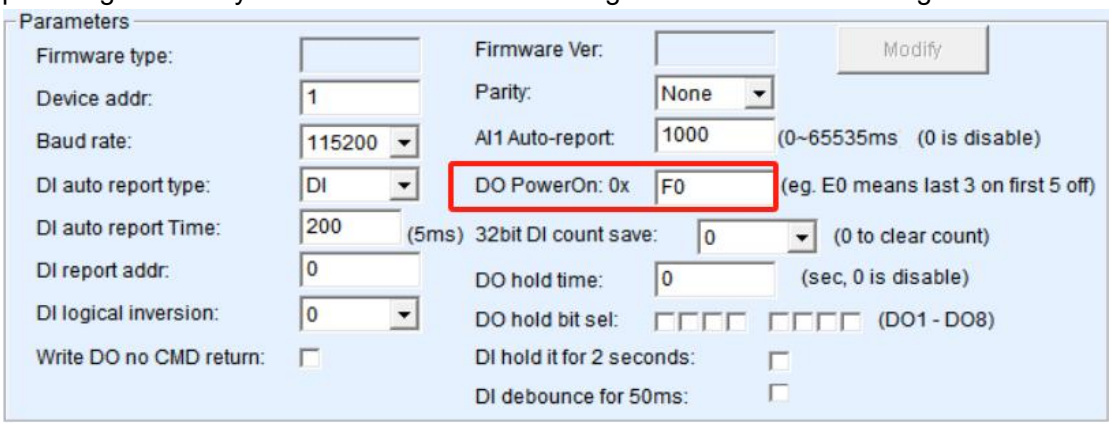

Figure 15 Setting DO configuration after power-on

If the value is set to F0, the front four channels are disconnected and the back four are closed. Each of the eight bits indicates the status of a DO line, and 1 indicates a pull-in.

4.2.10 DI Controls the DO

Considering that the user needs to control the DO output through the DI input, but the DI input device and the DO output device are far apart, here we take the Ethernet version as an example, we can connect the two IO controller through the Ethernet network to achieve DI remote control of the DO output.

Since DI active reporting is reported when changes are made, this can be used to send control instructions. The control instruction can set the station address of the controlled device /DO device through "DI report address", and the DI report instruction is exactly 05 set instruction, and the register address will be changed to the address corresponding to DO. Therefore, the DI input can control the DO of another device through active reporting.

For example, DI1 of device 1 controls DO1 of device 2, DI2 of device 1 controls DO2 of device 2, and so on.

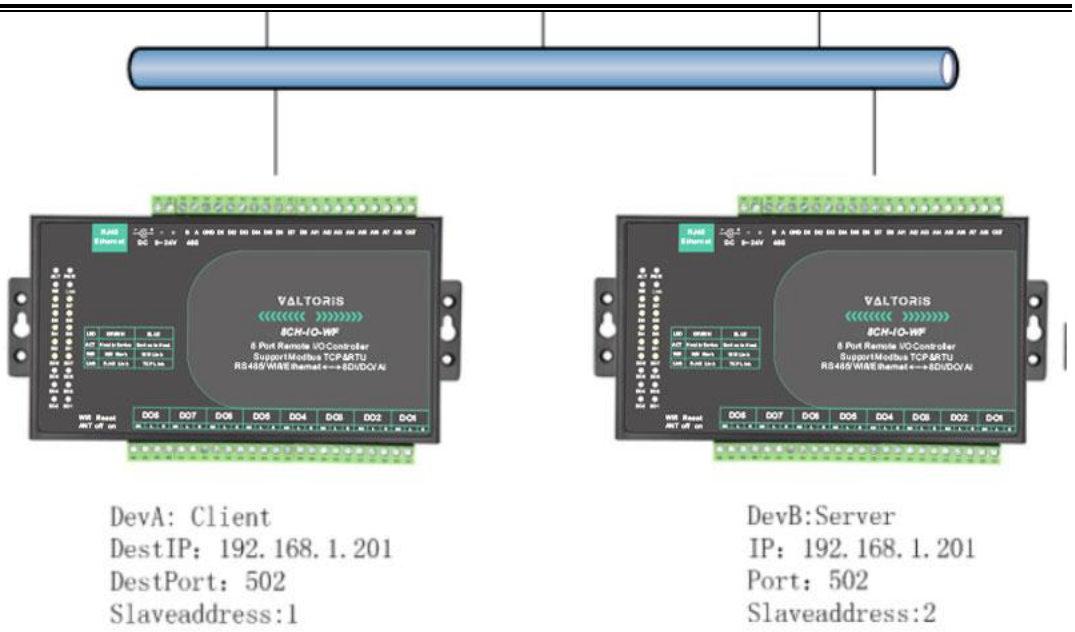

Figure 16 IO controller Pair control

As shown in the figure, two IO controllers are connected together via Ethernet. First, you need to set the two IO controllers parameters, including the IP address and whether to report.

Connect device DevA and search for it in the IO controller dialog box. Set device address to 1. Enter 1 in "Report or Not" to enable this function. DevA Settings are complete.

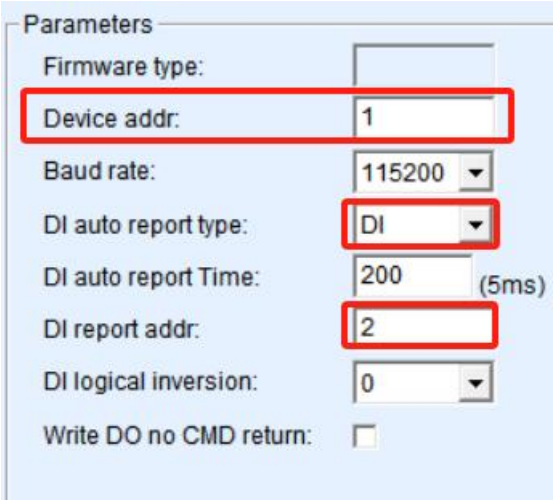

Figure 17 DevA configuration

Then connect the device DevB, search for and set the device address to 2, report whether to set to 1, and report the address to 1 (DevA). With this setup, DevA sends a control DO command to DevB when the DI changes. Similarly, DevB sends a control command to DevA for changes in DI.

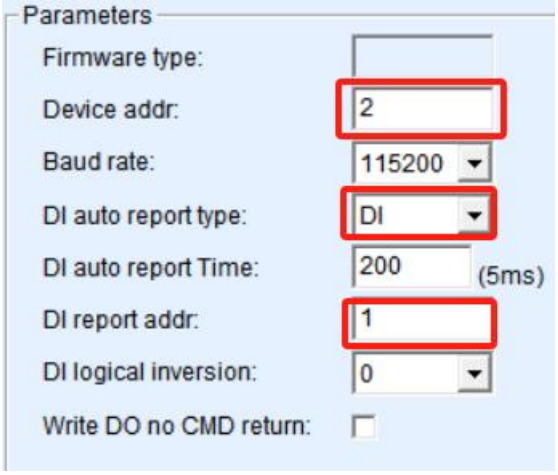

Figure 18 DevB configuration

If you are communicating over a network, configure the DevA and DevB network parameters to establish a TCP connection. DevB works in server mode and sets the working IP address and port. DevA acts as client mode and sets the destination IP address and port of DevA to the IP address and port of DevB.

If the communication is over RS485, you only need to connect the 485-IO serial ports of DevA and DevB.

4.2.11 DI Controls the DO

IO controller supports DI control of its own DO, that is, when DI is valid, the corresponding DO is closed, and otherwise disconnected. For example, when DI1 is 1, the control DO1 is closed, and when DI1 is 0, the control DO1 is disconnected.

Read and write by Modbus 03/06 instruction, and write 256 to 72 register through 06 to enable DI control of its own DO function, write 0 to shut down. By reading the value of the 72 register with the 03 instruction, you can get the status of the current function on/off.

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

Number of bytes	1	1	1	1	1	1	1	1
name	Device address	06	Start address high	Low start address	01 or 00	00	CRC high	CRC low

The Modbus RTU command is:

On：send->01 06 00 48 01 00 08 4c

back->01 06 00 48 01 00 08 4c

Off：send->01 06 00 48 00 00 09 dc

Back->01 06 00 48 00 00 09 dc

The Modbus TCP command is:

on：send->00 00 00 00 00 06 01 06 00 48 01 00

Back->00 00 00 00 00 06 01 06 00 48 01 00

Off：send->00 00 00 00 00 06 01 06 00 48 00 00

Back->00 00 00 00 00 06 01 06 00 48 00 00

Use 03 instruction, address range 72. The format is as follows:

Number of bytes	1	1	1	1	1	1	1	1
name	Device address	03	Start address high	Low start address	Length height	Low length	CRC high	CRC low

The Modbus RTU command is:

send->01 03 00 48 00 01 04 1C

back->01 03 02 01 00 B9 D4 ON

back->01 03 02 00 00 B8 44 OFF

The Modbus TCP command is:

send->00 00 00 00 00 06 01 03 00 48 00 01 04 1C

back->00 00 00 00 00 05 01 03 02 01 00 ON

back->00 00 00 00 00 05 01 03 02 00 00 OFF

4.2.12 DO Data Retention Function

The V16 start version supports the DO hold function, that is, if the DO is in the closed state, it needs to continue to give the instruction set to 1, and once the instruction set to 1 is not received within a certain period of time, it immediately disconnects the DO.

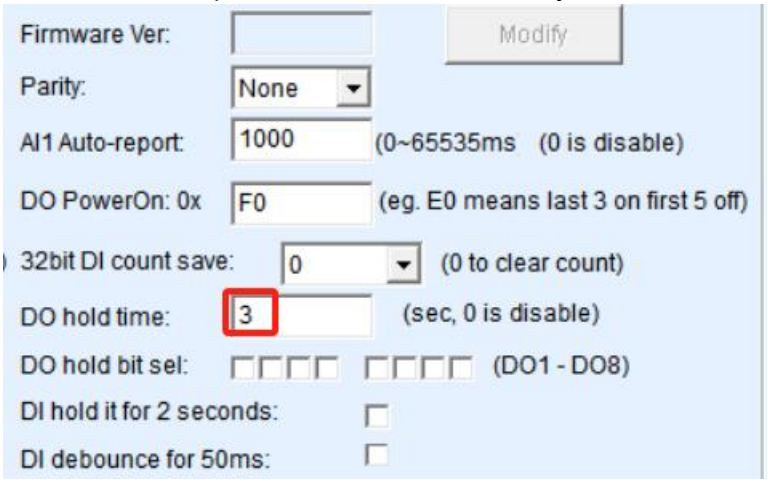

Figure 19 DO hold time

As shown in the figure, if the IO controller software is used, the DO hold time is set to 3 seconds.

The V26 starting version supports single/multi-way DO hold function (VIROM

software version needs to be above V6.76), that is, a single or several of the DO Settings can be set to hold function, once the instruction set to 1 is not received within a certain period of time, the DO is immediately disconnected.

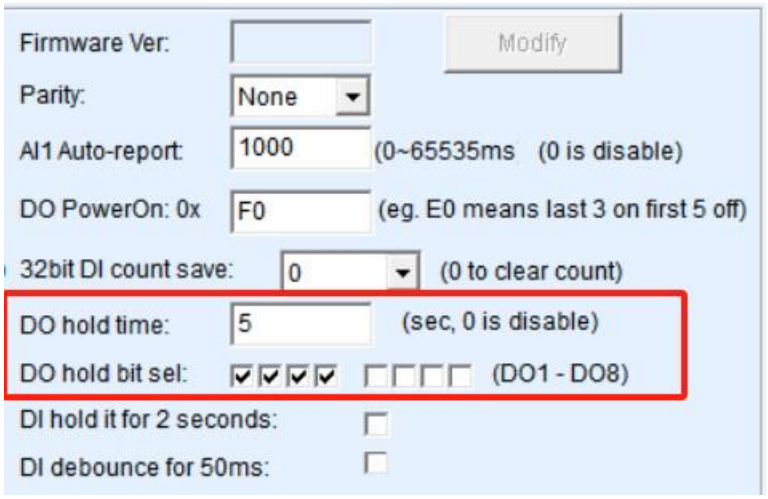

Figure 20 Single/multiple DO hold time

Active DI reporting and DO holding time work together to implement reliable DI control DO. The DO terminal is shown in the figure above. Set the address of the station to 2. The DI terminal is set as follows:

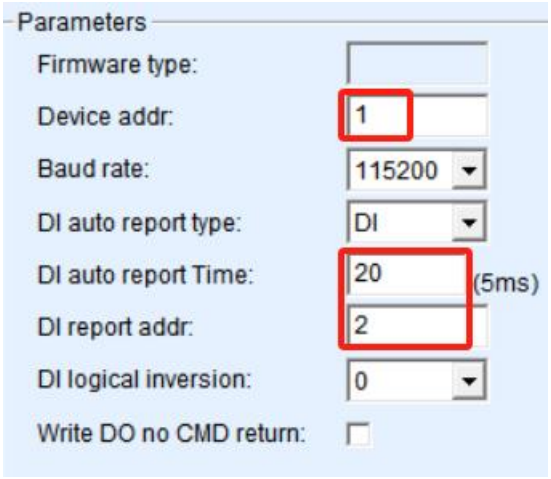

Figure 21 Reliable DI control DO

The site address of the DI device is set to an address other than 2. The DI automatically reports the address as the address of station 2. Set the same baud rate and set the DI report type to DI (that is, the report time is an even number that is not 0). Then adjust the reporting time to 20, and the actual time is 20*5=100ms.

According to the section "DI Report", after the DI report type is set to "DI", eight DI data are uploaded at intervals to implement the DO corresponding to DI control. In this case, the value is 100ms. In this way, the DO end can receive the corresponding instruction that the DO is set to 1. If the DI end is offline or powered off, the DO end will

disconnect the DO relay within 3 seconds.

4.3 Set serial port parameters

Current serial port parameters include baud rate and parity. Set this parameter in the IO Controller dialog box.

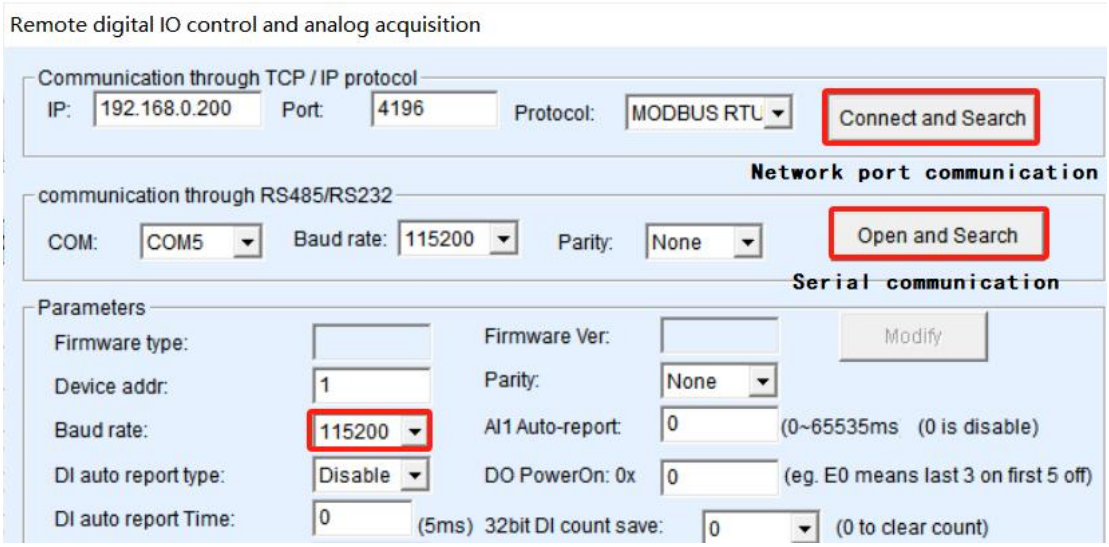

Figure 22 Configuration of serial port parameters in the I/O controller

The baud rate affects only the 485-IO RS485 interface. The baud rate of the network interface and 485-4G is determined by the baud rate set by the network module, 4G module do not limited by this baud rate.

When communicating through a serial port, it is not necessary to select the appropriate baud rate, because the software will automatically search for all baud rates.

However, the setting of the parity bit can affect the 485-IO serial port, 485-4G serial port, and network module. That is, when the parameter is set to parity (not parity), the parity bit of the network module needs to be changed accordingly. Otherwise, the "Open" button of "Network Communication" cannot be opened successfully. You can modify the serial port check bit of the network module in the Edit Device dialog box. As shown in the following picture.

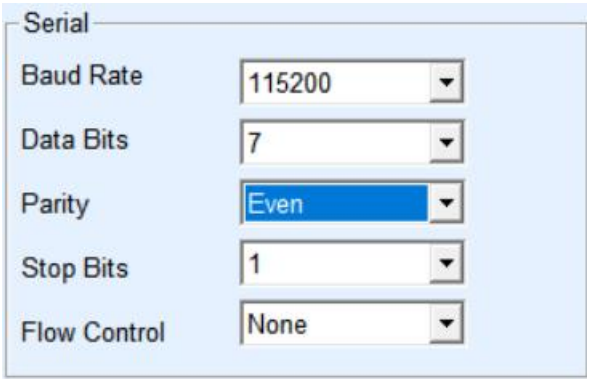

Figure 23 Check bit Settings of the network module

After the parity bit is changed, the parity bit of the 485-IO control device and the 485-4G serial port will be changed accordingly.

Note: When the verification mode is set to Non-None, the verification mode must be selected when the serial port is opened to search for devices. Otherwise, the corresponding device cannot be found. Otherwise, if the device is in No verification mode, you need to search for the serial port in No verification mode. That is, serial port search does not support automatic check bit search. You must specify a check mode.

4.4 Network-to-serial port function

The baud rate of RS485-4G is adaptive through the serial port parameters of the network module, and is not set. Currently, the baud rate ranges from 1200 to 115200bps and various parity bits are supported.

Users can connect RS485 devices such as meters to the RS485-4G interface. Data can be read and written to the instrument through the network.

4.5 How to use

4.5.1 Configuring Network Modules

Our IO controller uses 485-4G RS485 interface to configure the remote communication module, and the upper computer software used in the configuration is VirCOM.

Click device management and select serial port search, as shown in Figure 24, the interface for selecting serial port parameters will pop up, as shown in Figure 25, select serial port number, here is COM20, baud rate is 115200, 115200 here is the factory default setting, if the user previously set to other baud rate (such as 9600), you can also search.

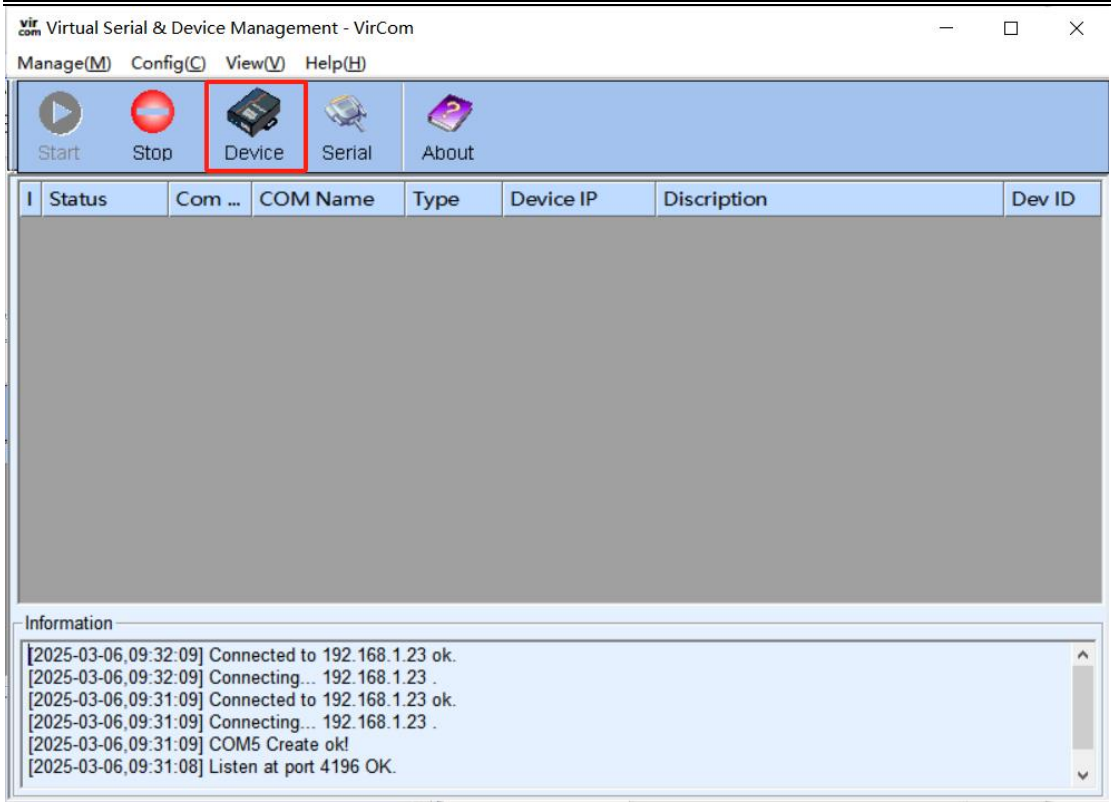

Figure 24 Configuration tool main page

Figure 25 Serial search page

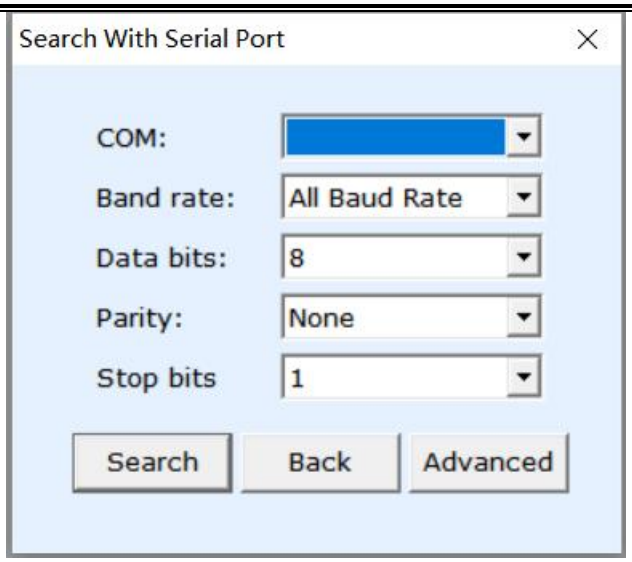

Figure 26 Serial port parameter page

4.5.2 8CH-IO-LTE

1. Configuration Method

First install the SIM card and 4G antenna. Then connect the 485 to USB cable to the 485 interface RS485-4G. Click Search. In this case, the configuration tool attempts to communicate with the device. If the communication succeeds, the configuration page is displayed. As shown in Figure 27 below:

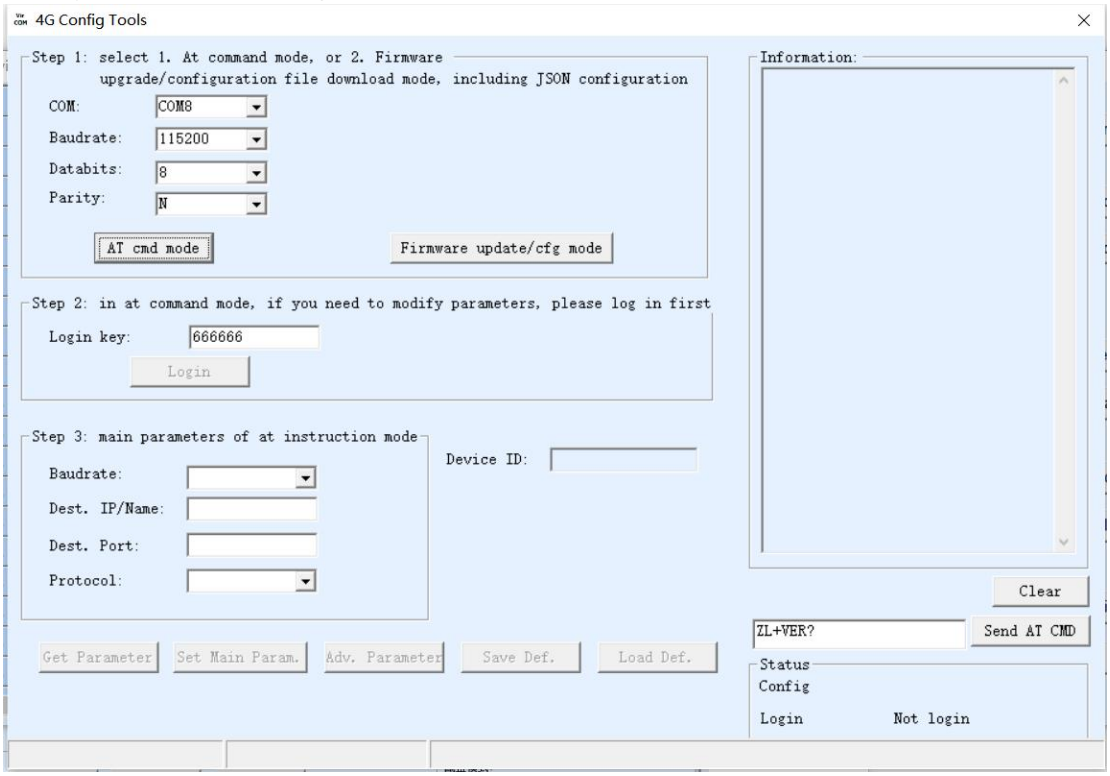

Figure 27 Conf Tool interface

Click to enter the AT command mode, the configuration tool will try to communicate with the device, the communication is successful, the AT command return information will be displayed on the right side, and the configuration mode will be displayed as having entered the configuration mode, as shown in Figure 28 below:

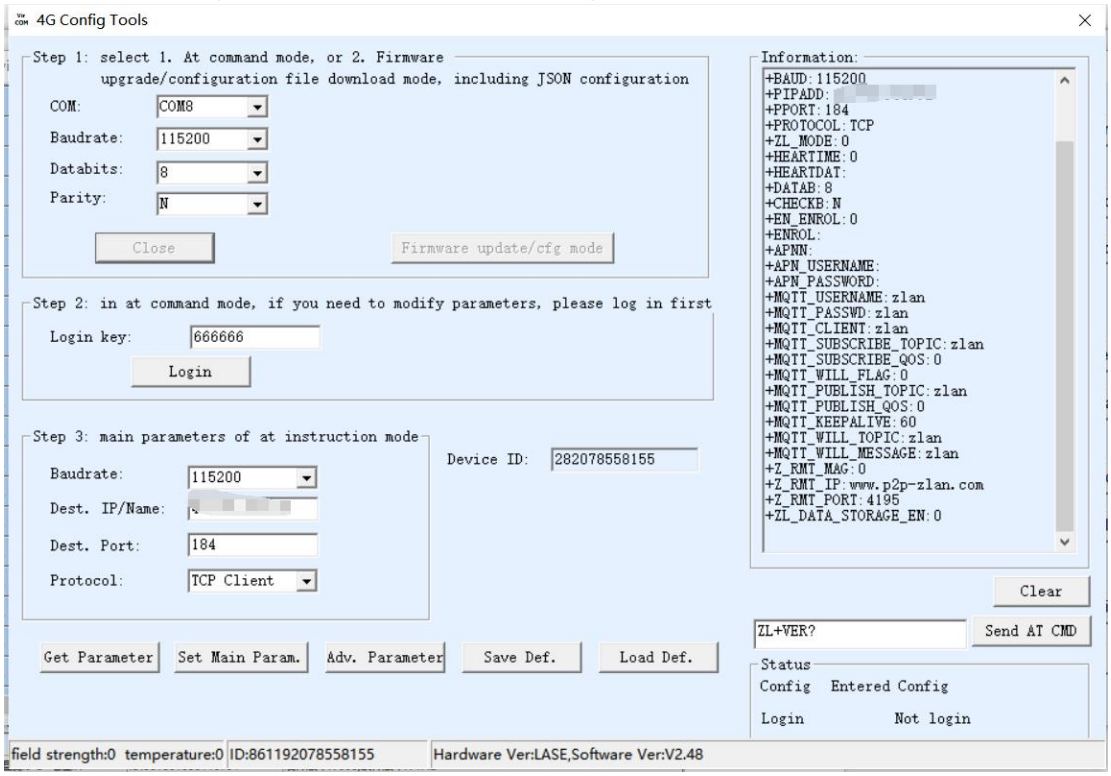

Figure 28 The Configuration mode page is displayed

The default login password is 666666. Before you click Log In, the parameters are read-only and cannot be set or modified. Click the "Login button" :

After login, the login status changes to Logged In, as shown in Figure 29.

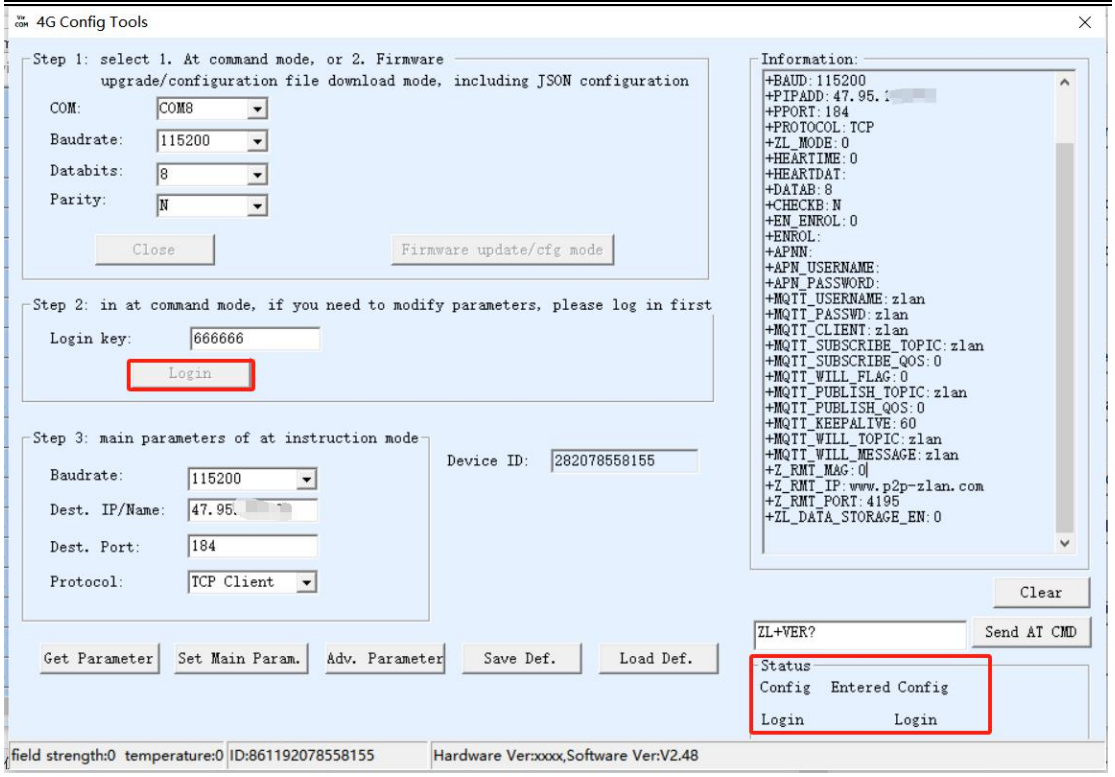

Figure 29 Login page

The main parameters of the AT command mode include the baud rate, destination IP address, destination port, and protocol. Protocol TCP or UDP is supported. After modifying the corresponding parameters, click "Set parameters" to set the new parameters to the device, and the device will return the parameters successfully set, as shown in Figure 30.

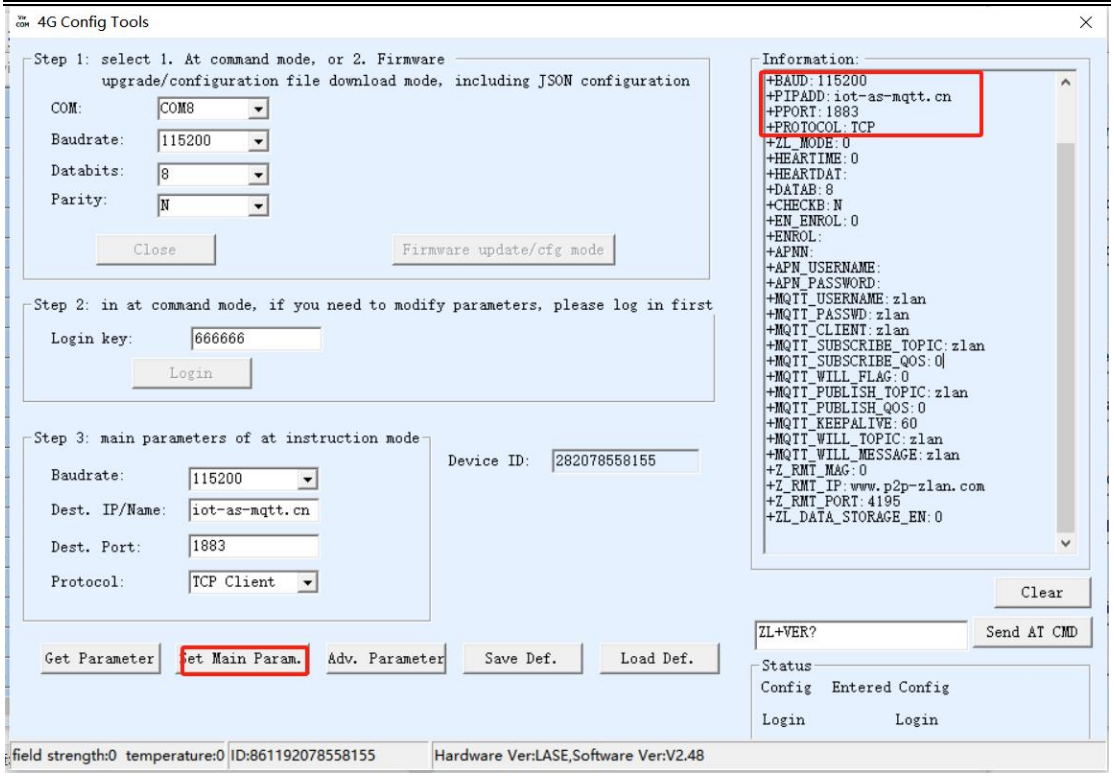

Figure 30 Setting parameters

The "Get Parameters" button can obtain the parameters of the current device, which is obtained by sending the AT instruction. The returned data of the AT instruction is listed on the right. For AT directives, refer to the other sections of this article. Because the "Get parameters" will be automatically executed once after the "open" is successful, it is generally not necessary to click the "Get parameters" button.

Click "Advanced Parameters", and the advanced parameters box is shown in Figure 31. Commonly used parameters are:

1. Heartbeat interval: You can set the heartbeat packet interval to 15s.
2. Heartbeat content: Set the heartbeat packet content.
3. Serial port data bit
4. Serial port verification bit
5. Enable the registration package: Whether to enable the registration package.
6. Registration package content: The content of the registration package sent after connecting to the server.
7. APN: indicates the APN access point name.
8. APN User name
9. APN password
10. MQTT parameters: Set parameters for accessing the MQTT server
11. Remote device management: Connects devices with the remote management

function to the remote server

After selecting the parameters, click the button of "Effective Advanced Parameters" and observe the information bar on the right to check whether the Settings returned by the device are consistent with the information filled in, as shown in Figure 32.

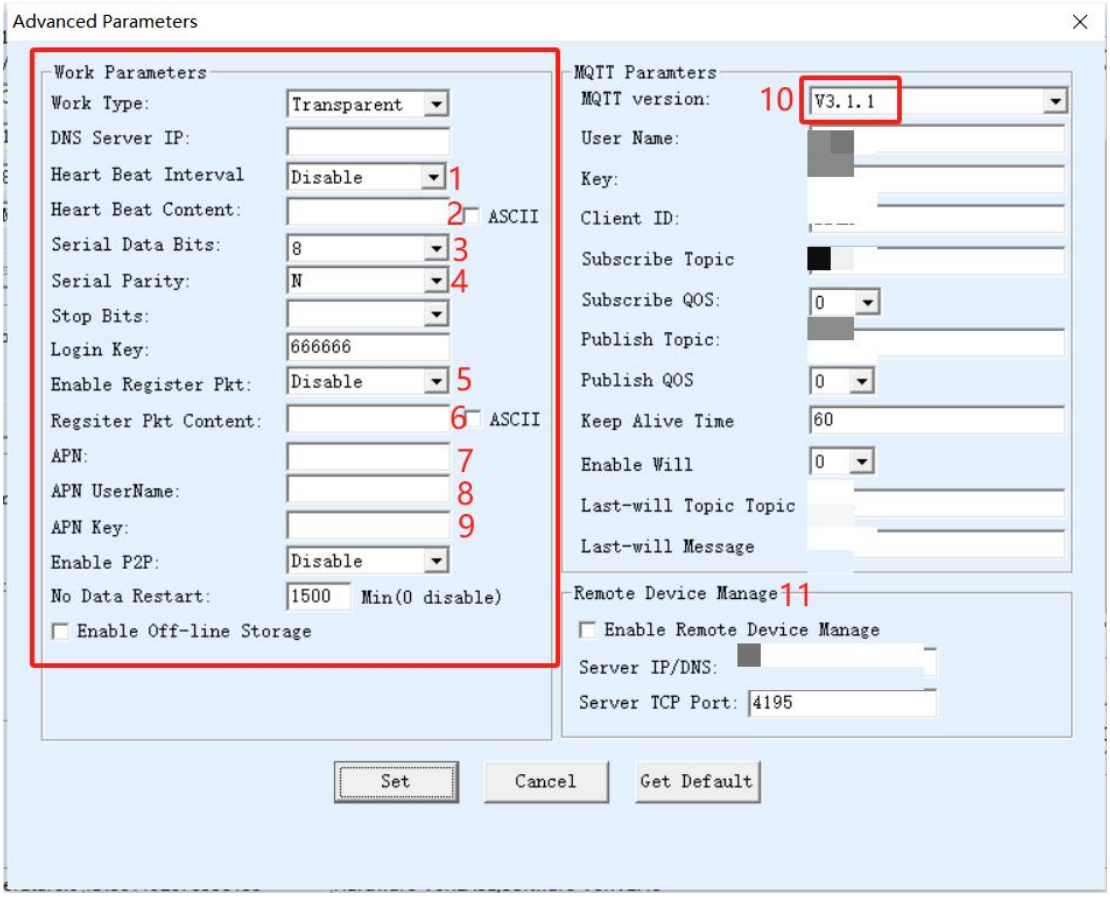

Figure 31 Advanced parameters

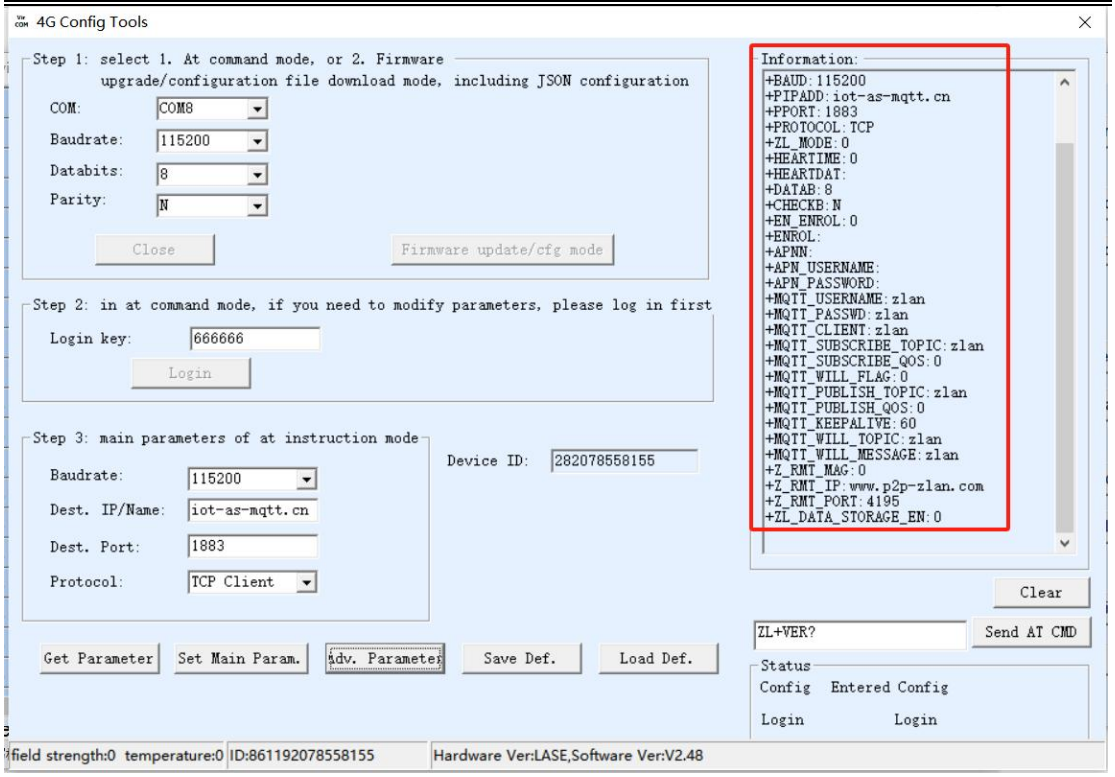

Figure 32 Setting Advanced parameters Return information

2. Server transparent transmission Test

Suppose there is the following networking structure as shown in the figure below, configured to connect to the *** port of the server ***.***.***.***. Please configure it through the methods in the "Serial Port Configuration" section. After the configuration is completed and the power is restored, it takes 20 to 40 seconds to connect to the server.

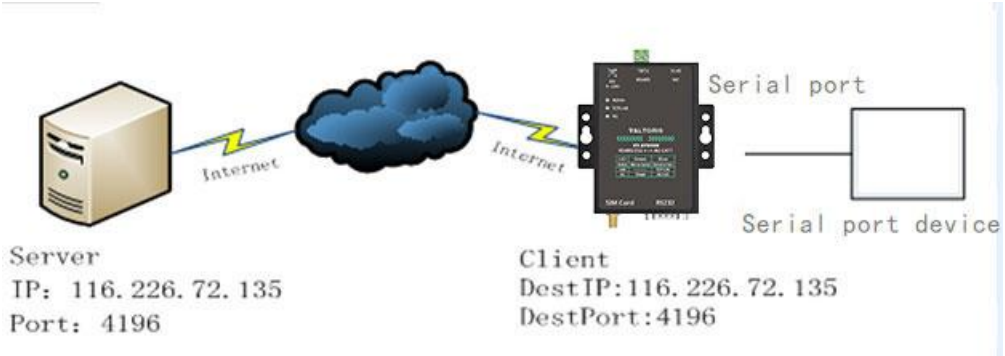

Figure 33 Networking structure diagram

We run the TCP tool SocketDlgTest on the server.

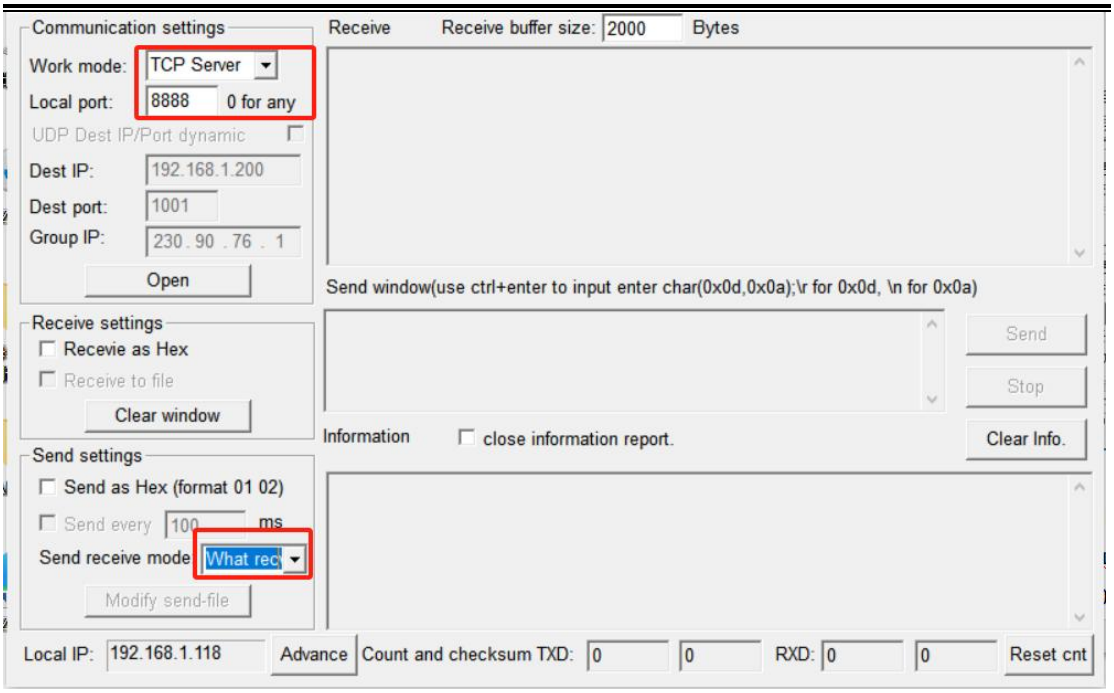

Figure 34 Server-side tools

As shown in the figure, select the local port as 4196 (note that if you are running the Vircom tool, you need to change the port), and then click the "Open" button. When The device is connected to the server, it will display "The NO..." is accepted!" The information.

Now connect the device's serial port to a USB-to-485 serial port cable, and open the serial port debugging tool, as well as the correct COM port.

Now, when data is sent through the serial port, the server side will reply with the corresponding data. Similarly, when the device receives the message reply from the server and outputs it through the serial port, the serial port tool will receive the same data here. This demonstrates the two-way communication from the serial port to 4G network, as shown in Figure 35 below:

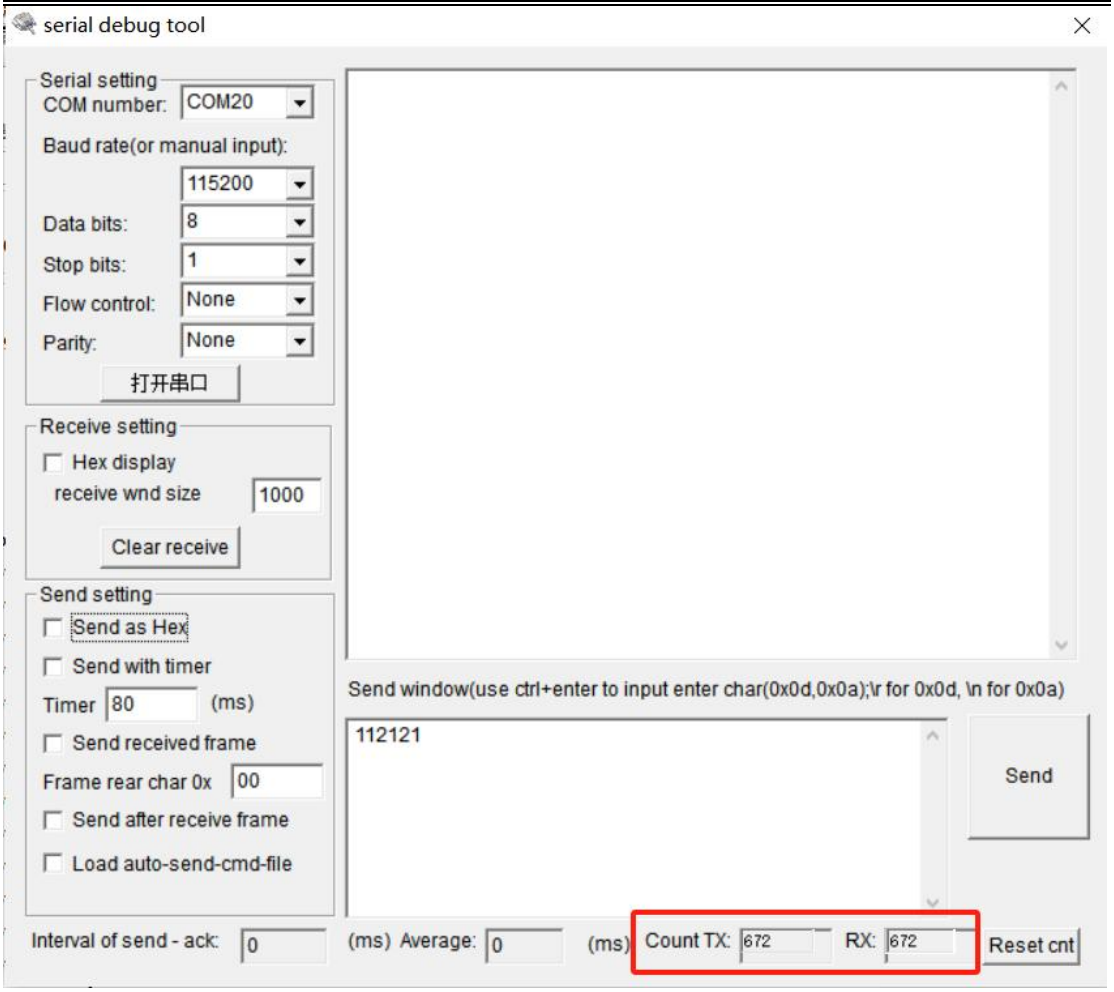

Figure 35 Serial port debugging tool on the device

3. Modbus Protocol Conversion Test

The configuration parameters are basically the same as the no-protocol pass-through test; you only need to change the transformation protocol to the Modbus protocol. The serial port Modbus RTU protocol can be converted to the network Modbus TCP protocol, and the network Modbus TCP protocol can be converted to the serial port Modbus RTU protocol.

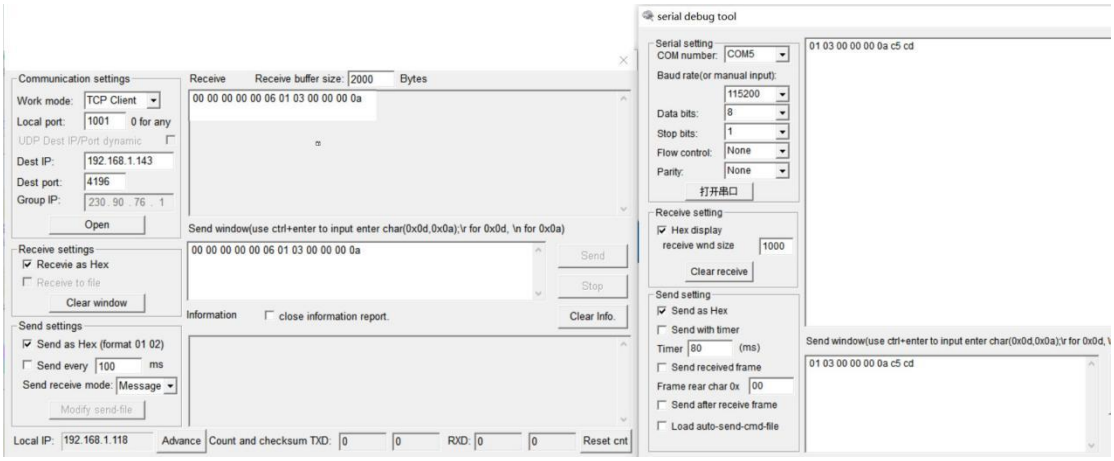

Figure 36 Modbus protocol conversion test

4. MQTT protocol testing

This test is for connecting to Alibaba Cloud. Create a subscription topic named "test" and a publishing topic named "1" on Alibaba Cloud, as shown in Figure 38. According to the configuration instructions in step 5, first fill in the IP and port configuration of the MQTT server, save the parameters, and the parameter filling is shown in Figure 39. Then, on the advanced parameter page, fill in the MQTT ID, username, password, as well as the subscription and publication topic and the retention time. The parameter filling is shown in Figure 40. Please note to select the working mode as MQTT mode.

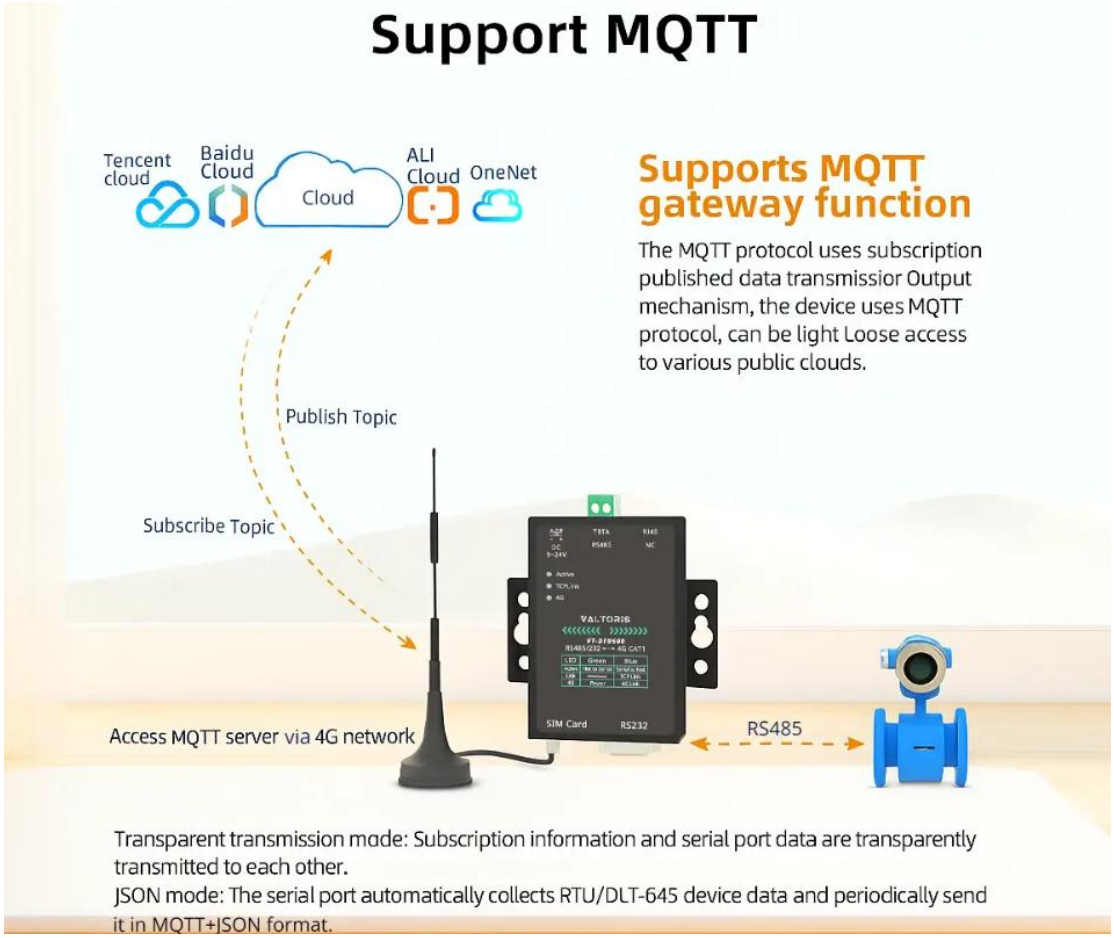

Figure 37 Schematic diagram of MQTT

Topic	Operation permission
/a1WSVHIXkDh/${deviceName}/user/test	Subscribe
/a1WSVHIXkDh/${deviceName}/user/1	Publish

Figure 38 Adds a theme to Alibaba Cloud

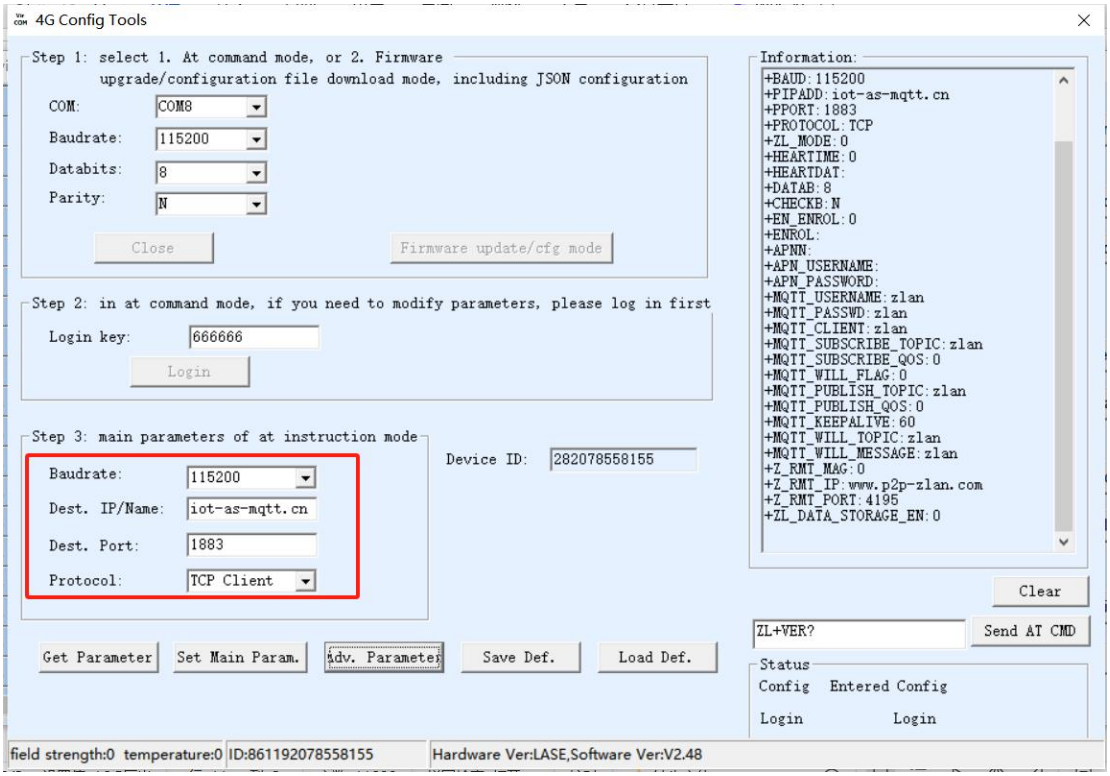

Figure 39 Aricloud IP address and port number

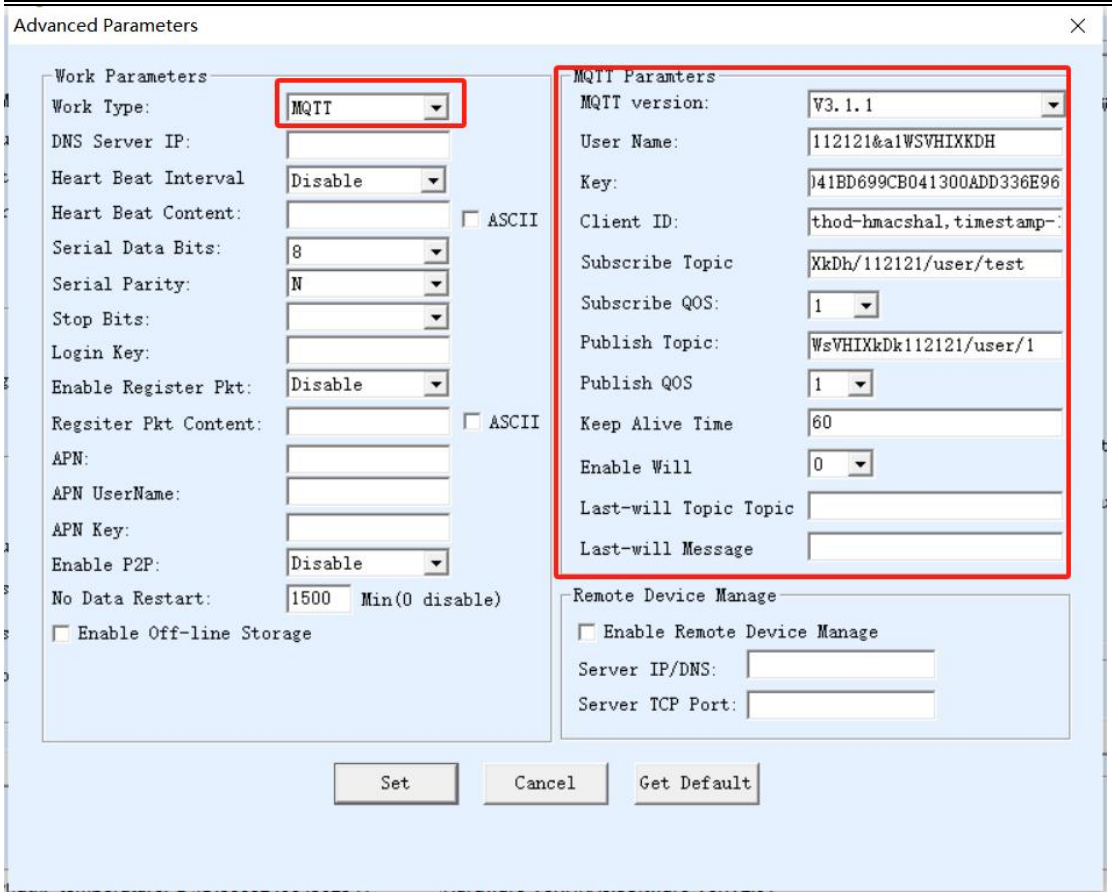

Figure 40 Configuration of Alibaba Cloud MQTT

After the Settings are completed, open the Alibaba Cloud Device Management interface and enter the log service page to view the information sent on the device, as shown in Figure 41. Data is sent through the serial port of the device. A message (" MTEST ") is sent to the MQTT server of Alibaba Cloud via the subject of 1. Alibaba Cloud receives the data as shown in Figure 42. The Alibaba Cloud server then sends a message (" ALI_send ") to the serial port of the device via the test subject as shown in Figure 30. This completes the MQTT sending and receiving test.

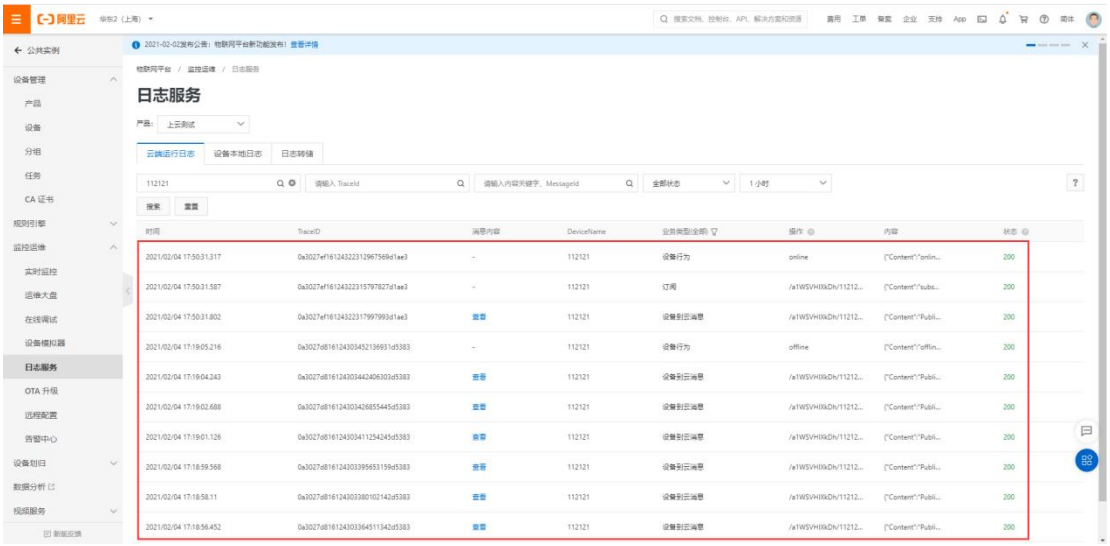

Figure 41 Alibaba Cloud Log Service

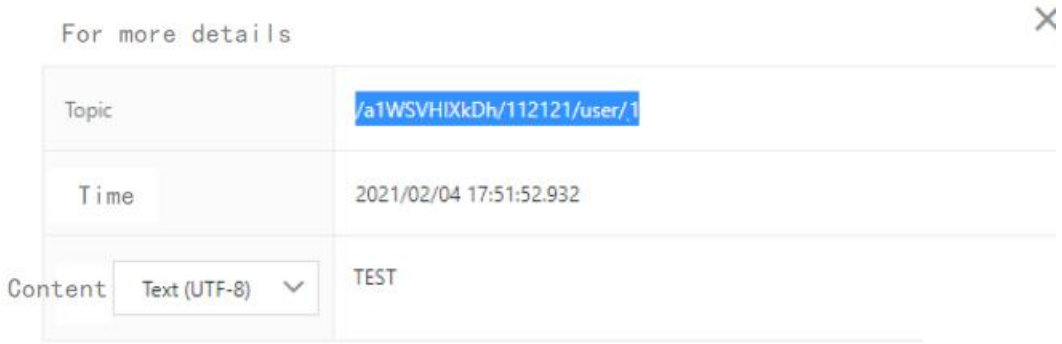

Figure 42: Alibaba Cloud receives serial port data

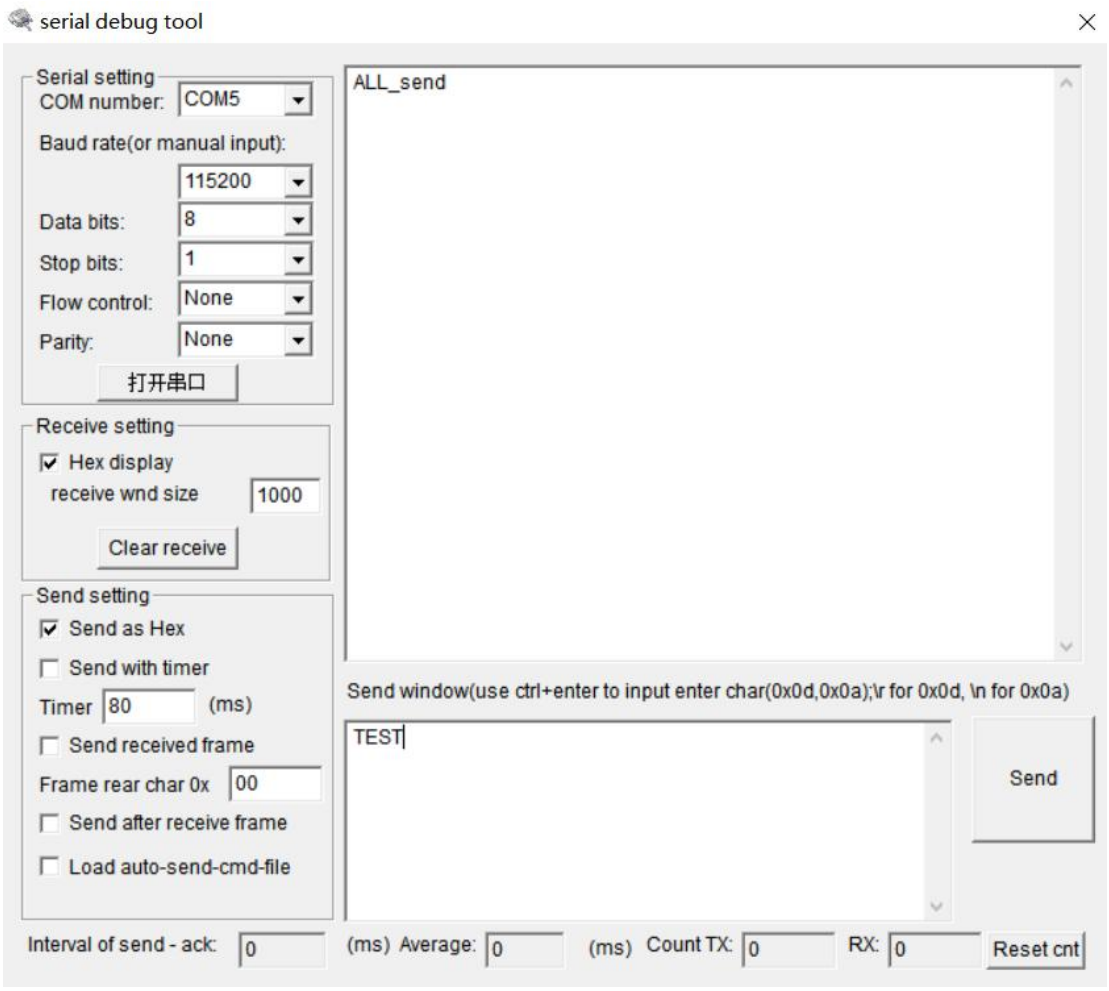

Figure 43 shows the serial port receiving data from Alibaba Cloud

5. Configure JSON for Posting

Through the above part: Modbus protocol conversion test, configure a simple JSON upload template. The configuration process is shown in Figure 44, 45, 46 and 47, and the data of some MODBUS nodes is collected and converted into JSON format for upload.

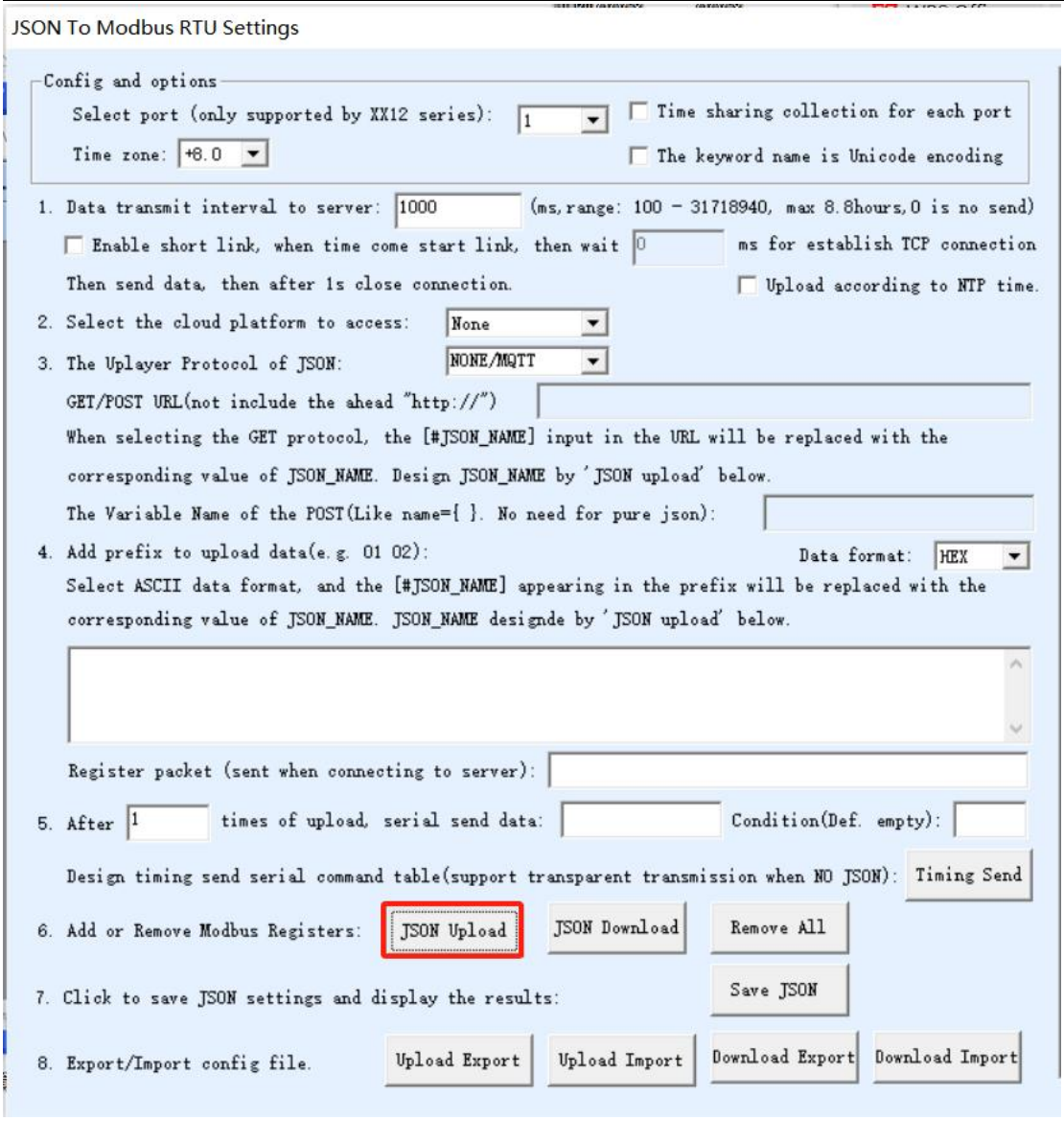

Figure 44 Configuration JSON upload

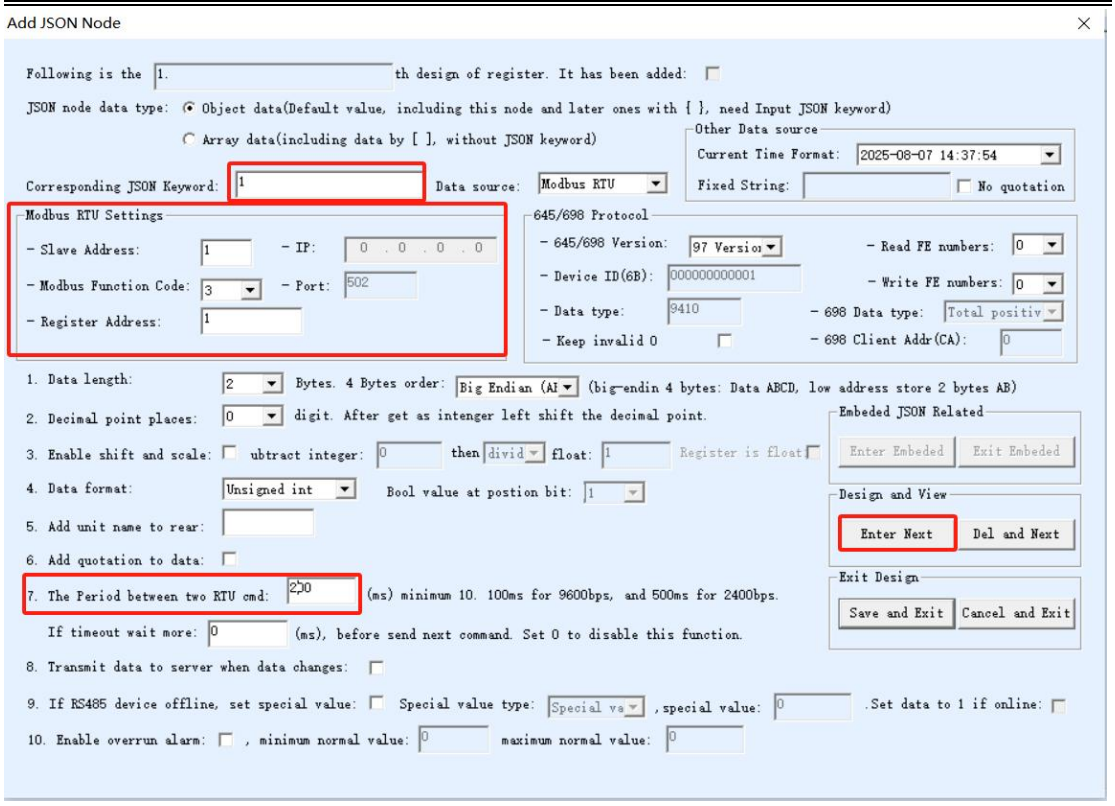

Figure 45 Configure the keywords for collection, register addresses, and collection intervals

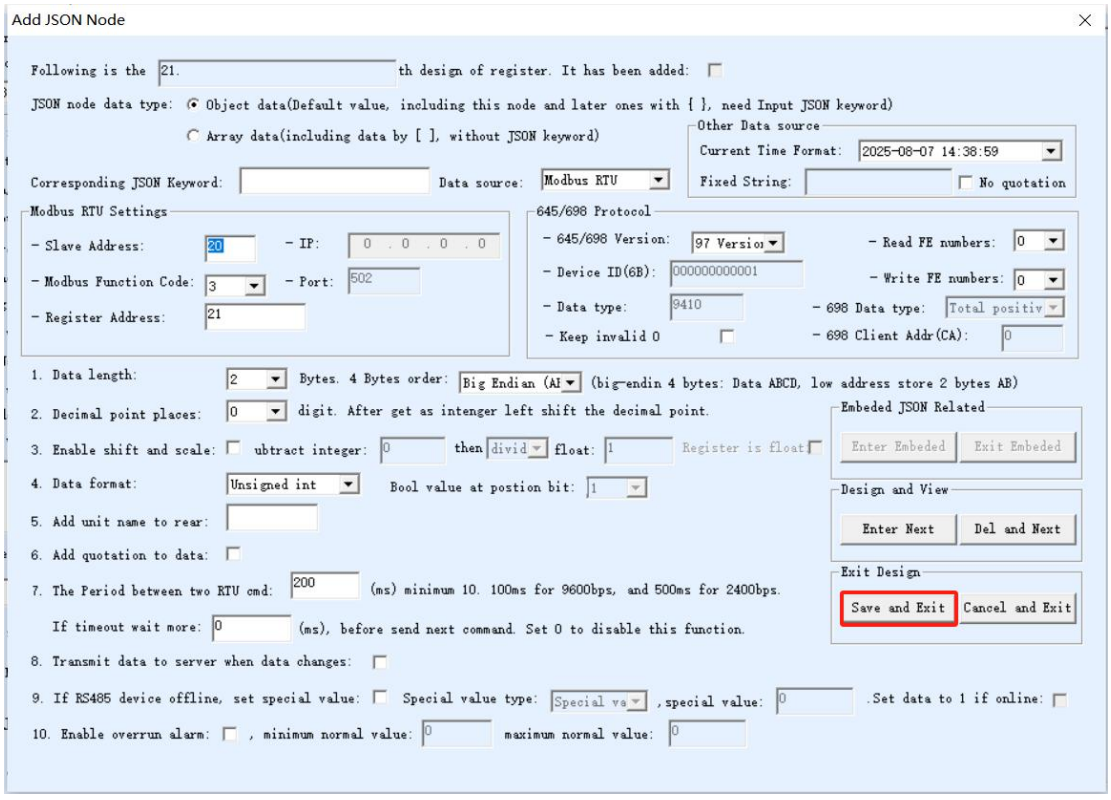

Figure 46 Save and exit after the configuration is completed

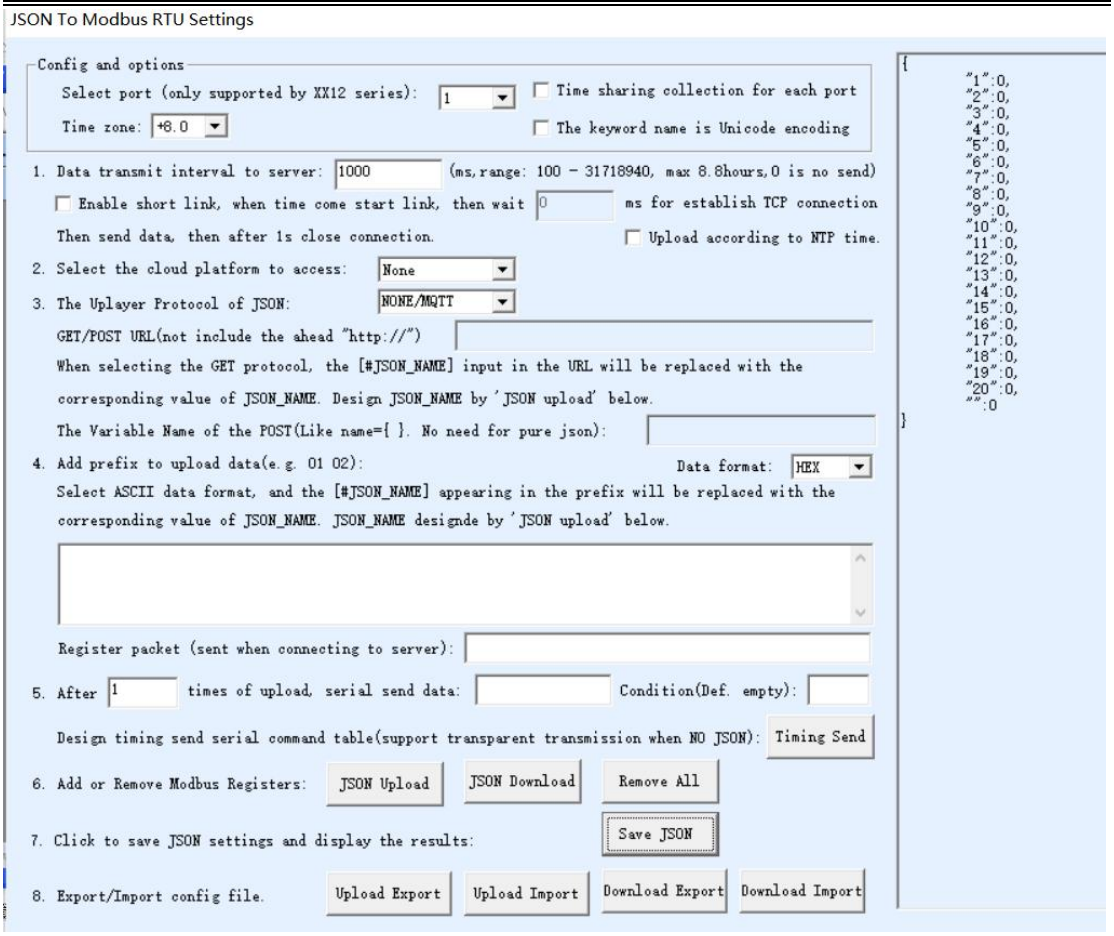

Figure 47 saves the JSON Settings and views the preview of the JSON format

6. Configure the MODBUS RTU simulation device

Simulate the MODEBUS Slave device through the Modbus Slave software, connect the device to the computer via the serial port cable, and open the connection of Modbus Slave. The configuration of Modbus Slave is shown in Figure 48.

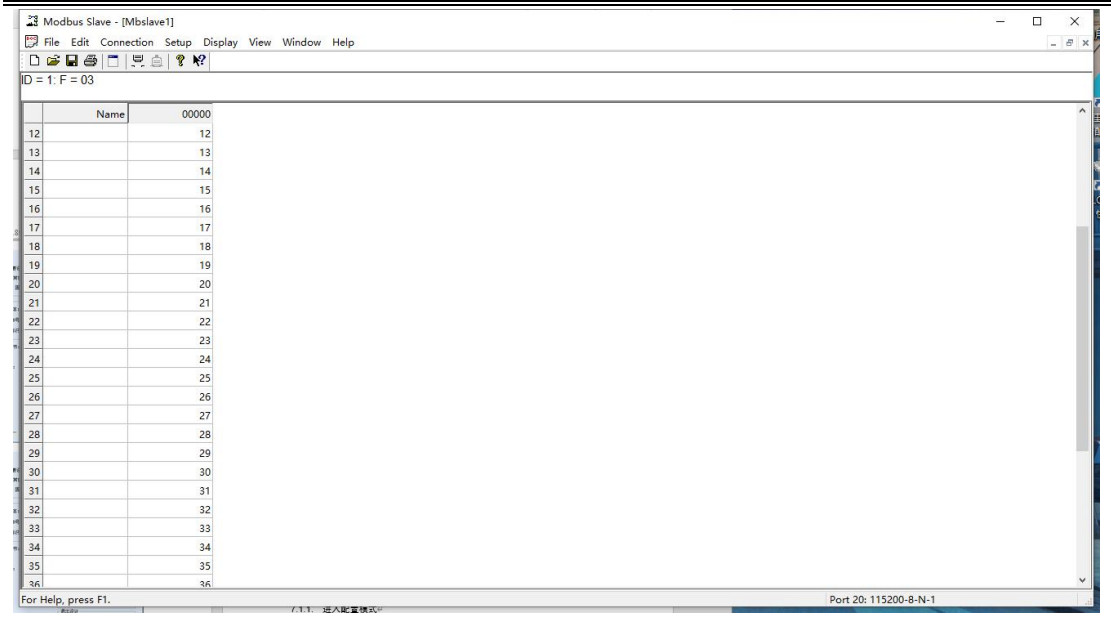

Figure 48: Modbus Slave fills in the simulation data

7. View the JSON posted above

By using the Alibaba Cloud log service to view the uploaded JSON data, it can be observed that the collected data is consistent with the data configured by Modbus Slave. This completes the simple MODBUS to JSON conversion test.

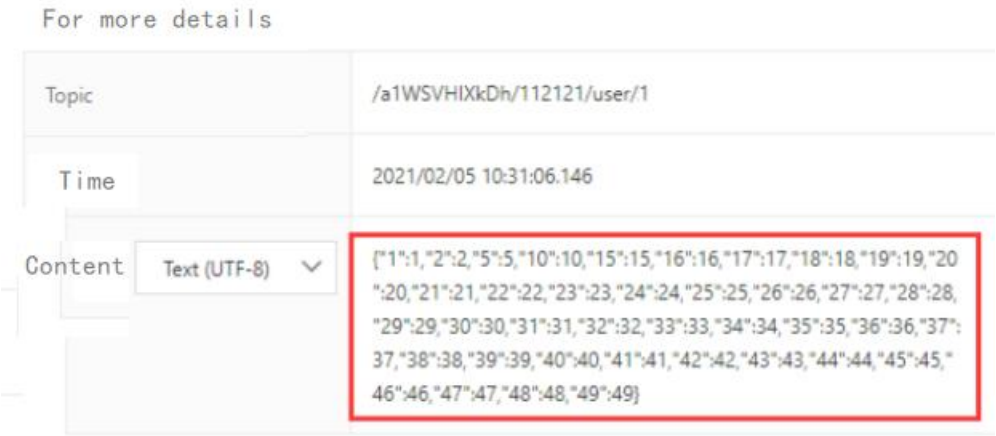

Figure 49 shows the serial port receiving data from Alibaba Cloud

4.5.3 8CH-IO-ETH

1. Configuration Method

After clicking the search button, the page shown in Figure 50 is displayed.

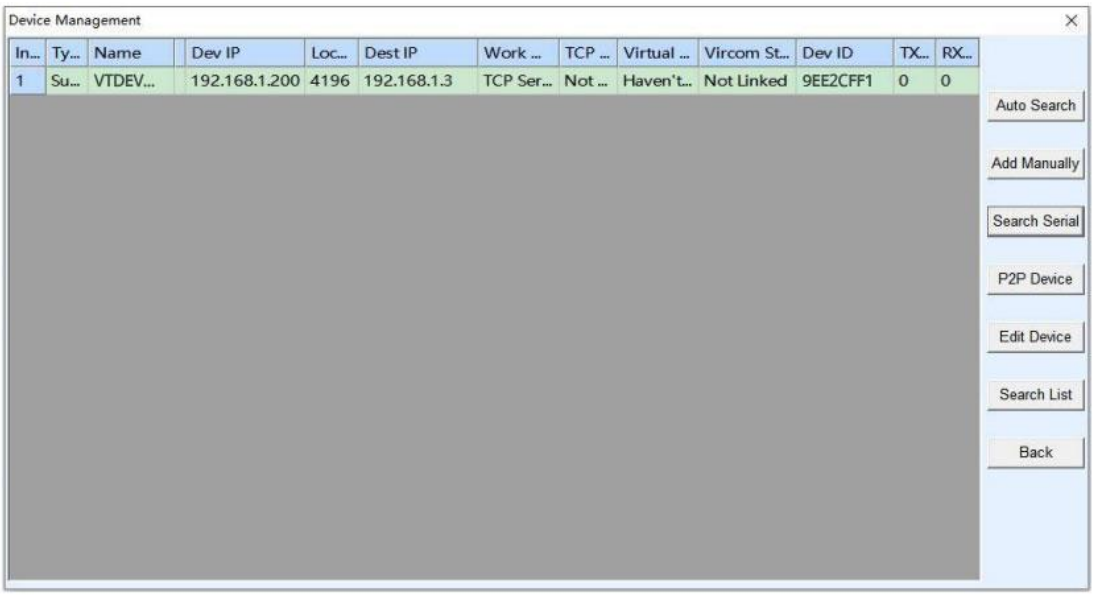

Figure 50 Ethernet configuration page

You can see all the currently online devices from the device list. Click "Edit Device" to configure the parameters.

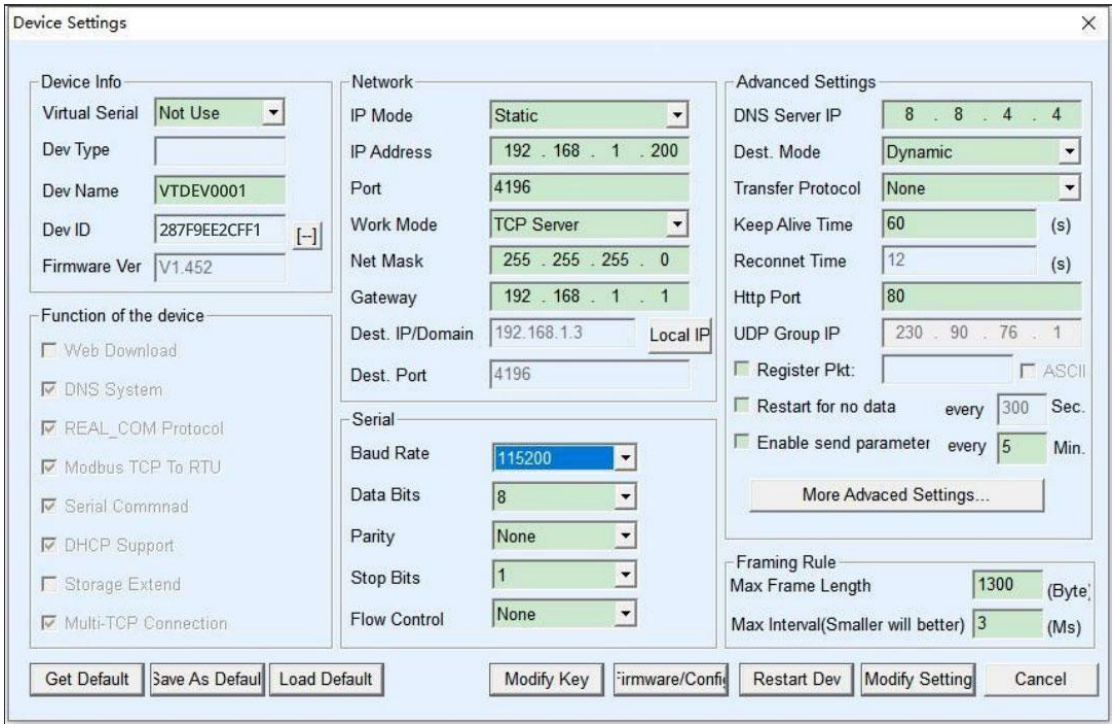

Figure 51 Device Parameters

In this interface, the user can set the parameters of the device, and then click "Modify Settings", the parameters will be set to the device's flash and will not be lost when the power is off. At the same time, the device will automatically restart.

The main parameters configured here are: baud rate, data bit, and check bit in the serial port settings; IP address, subnet mask, and gateway in the network settings; sometimes, according to the computer software, it is also necessary to configure the working mode of the serial port server.

The detailed meanings of other parameters are as follows:

Table 7 Parameter meanings

Parameter	Value Range	Meaning
Virtual Serial	Port Unused, created virtual serial port	You can bind the current device to a created virtual serial port. Please add a COM port in "Serial Port Management" on the main interface first.
Device model		Only display the core module model
Device name	Any	You can give the device an easy-to-read name with a maximum length of 9 bytes.
Device ID		Factory unique ID, cannot be modified.
Firmware Version		Core module firmware version
Supporting Function		Refer to Table 6 for the functions supported by the device.
IP Mode	Static, DHCP	Users can choose static or DHCP (dynamic IP)
IP Address		IP address of the serial device server

Port	0～65535	The listening port of the serial port server when it is in TCP Server or UDP mode. When acting as a client, it is best to specify port 0, which is beneficial to improve the connection speed. When using port 0, the system will randomly assign a local port. The difference between this and non-zero ports is: (1) When the local port is 0, a new TCP connection is established with the PC when the module restarts. The old TCP connection may not be closed, and there may be multiple false connections with the device. Generally, the host computer hopes to close the old connection when the module restarts; specifying a non-zero port will close the old connection. (2) When the local port is 0, the TCP re-connection time is faster. When the serial port server is in TCP client mode, it also acts as a TCP server to listen for connections on the port. At the same time, the local port number used by the TCP client to connect to the server is "port + 1".
Working Mode	TCP server mode, TCP client mode, UDP mode, UDP multicast	When set as a TCP server, the serial device server waits for the computer to connect; when set as a TCP client, the serial device server actively initiates a connection to the network server specified by the destination IP.
Sub-net Mask	Eg: 255.255.255.0	Must be the same as the subnet mask of the local area network.
Gateway	Eg: 192.168.1.1	Must be the same as the local LAN gateway.
Destination IP or domain name		In TCP client or UDP mode, data will be sent to the computer indicated by the destination IP or domain name.
Destination Port		In TCP client or UDP mode, data will be sent to the destination port of the destination IP.
Baud Rate	1200、2400、4800、7200、9600、14400、19200、28800、38400、57600、76800、115200、230400、460800、921600	Serial port baud rate
Data Bits	5、 6、7、8、 9	
Check digit	None, Even, Odd, Mark, Space	
Stop Bits	1、2	

Flow Control	No flow control, hard flow control CTS/RTS, hard flow control DTR/DCR, soft flow control XON/XOFF	Only valid for RS232 serial port
DNS Server		When the destination IP is described by domain name, you need to fill in the DNS server IP. When the IP mode is DHCP, you do not need to specify the DNS server, it will be automatically obtained from the DHCP server.
Destination mode	Static, Dynamic	In TCP client mode: When using static destination mode, the device will automatically restart after 5 consecutive failures in connecting to the server.
Conversion agreement	NONE、Modbus TCP、Real_COM	NONE means that data forwarding from the serial port to the network is transparent; Modbus TCP will directly convert the Modbus TCP protocol into the RTU (ASCII) protocol to facilitate coordination with the Modbus TCP protocol; RealCOM is designed to be compatible with the old version of the REAL_COM protocol and is a virtual serial port protocol. However, when using a virtual serial port, it is not necessary to select the RealCom protocol.
Keep-alive time	0～255	Heartbeat interval. (1) When set to 1-255, if the device is in TCP client mode, it will automatically send TCP heartbeats every "keep alive time". This can ensure the TCP validity of the link. When set to 0, there will be no TCP heartbeat. (2) When set to 0-254, when the conversion protocol is selected as REAL_COM protocol, every keep alive time, the device will send a data with a length of 1 and content of 0 to implement the heartbeat mechanism in the Realcom protocol. When set to 255, there will be no realcom heartbeat. (3) When set to 0-254, if the device is working in TCP client mode, the device will send device parameters to the destination computer every keep alive time. When set to 255, there will be no parameter sending function, which can realize remote device management.

Disconnection re-connection time	0～255	In TCP client mode, when the connection is not successful, each "disconnection re-connection time" will re-initiate a TCP connection to the computer. It can be 0 to 254 seconds. If it is set to 255, it means that it will never reconnect. Note that the first TCP connection (such as hardware power-on, device restart through Vircom software, no data light) will generally be made immediately. Only after the first connection fails will it wait for the "disconnection re-connection time" and try again, so the "disconnection re-connection time" will not affect the connection establishment time under normal circumstances between the network and the server.
Web access port	1～65535	The default is 80
Multicast address		Used for UDP multicast
Enable registration package		When the TCP connection is established, the registration packet is sent to the computer. After enabling the registration packet, the realcom protocol must be selected. Supports TCP server and TCP client modes.
Data packet length	1～1400	One of the serial port framing rules. After receiving data of this length, the serial port server sends the received data as a frame to the network.
Data packet interval	0～255	Serial port framing rule 2: When the data received by the serial port of the serial server pauses and the pause time is greater than this time, the received data will be sent to the network as a frame.

2. Usage Method

Power on the device and connect it to the network using a network cable. If Modbus TCP is used, select Modbus TCP as the conversion protocol. Otherwise, select None. The network module of the IO controller works as the TCP server mode and the port is 502.

The user software connects to this IP and port 502 to control the device.

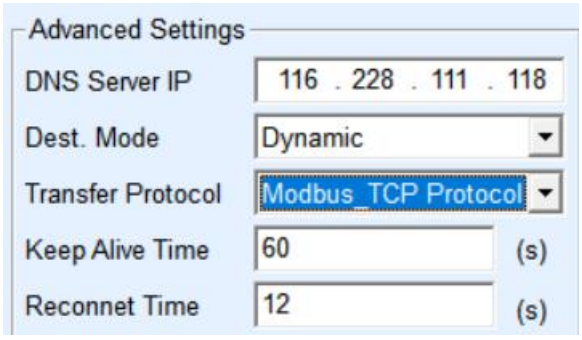

Figure 52 Enabling MODBUS TCP

If the Modbus TCP software/device of the user serves as the Slave station, you need to convert the protocol to Modbus TCP, change the working mode to the client, change the destination IP address to the IP address of the Modbus TCP software/device, and set the destination port to 502, as shown in Figure 53.

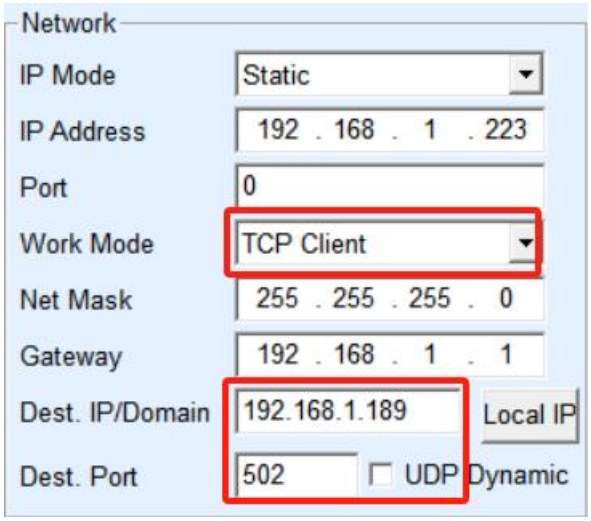

Figure 53 MODBUS TCP as the client

After the modification is complete, power on the device again to work properly.

Appendix 1: Summary of parameters

This chapter mainly covers the technical details of parameter setting and reading. It also helps users to configure and modify parameters with their own software. For common applications, you can skip this section.

Separate the parameters read and set from the register master table as follows.

Table 8. Parameter related read operations

Function code	Feature	Address range
03	Read base parameter	63~67
03	Read spread parameter	68~162
06	Set parameters	63~67
06	Set extension parameters	69~162
16	Set basic parameters	63~67
16	Set extension parameters	68~162

As can be seen from the table, parameters are read using 03 function code and set using 06 and 16 instructions. The parameters are divided into basic parameters and extended parameters, corresponding to registers 63~67 and 68~162 respectively.

Table 9. Base parameter register

Register address	Parameter name	Length (bytes)	Instructions
63(0x3F)	addr/Device address	1	The high byte of the register value
63(0x3F)	upLoad/The DI active reporting function is enabled	1	The low byte of the register value, 1 indicates that it is enabled, and 2 to 255 indicates that it is sent periodically.
64(0x40)	dst_addr/DI report address	1	The high byte of the register value
64(0x40)	baud/Device baud rate	1	The low byte of the register value sets only the baud rate of the 485-IO RS485 interface. 1200 0; 2400 1; 4800 2 9600 3; 19200 4; 38400 5; 57600 6; 115200 7

65(0x41)	ver/Firmware version	1	High byte of the register value, read only
65(0x41)	Compound parameter setting	1	The low byte of the register value. Bit1:32-bit DI count save, 1 indicates save Bit2: DI logical inversion. 1 indicates inversion Bit3: DI delay function. After DI changes to 1, it keeps the value of 1 for 2 seconds after DI input changes to 0, that is, it can still read DI as 1 within 2 seconds.
66(0x42)	A1UpLoadH/AI Description The report period is high in bytes	1	The high byte of the register value
66(0x42)	A1UpLoadL/ AIDescription The reporting period is low bytes	1	The low byte of the register value
67(0x43)	A2UpLoadH/AI Description The report period is high in bytes	1	The register value must be the same as the value of A1UpLoadH
67(0x43)	A2UpLoadL/ AIDescription The reporting period is low bytes	1	The low byte of the register value must be the same as the value of A1UpLoadL

Table 10. Extend parameter register

Register address	Parameter name	Length (bytes)	Instructions
68(0x44)	dostate/Configure the DO after power-on	1	The higher byte of the register value, 0xF0 indicates the last four aspirates
68(0x44)	checkb	1	The low byte of the register value.

			0: no check 1: odd check 2: parity check 3: Mark 4: space
69(0x45)	baud_UART_0_2/ baud rate for network communication and 485-4G	1	The high byte register value, currently read-only, is adaptive through the network module and does not need to be set
69(0x45)	datab	1	The low byte of the register value. Leave for further expansion.
70(0x46)	stopb	1	The high byte of the register value is left for later expansion
70(0x46)	TCP_LINK_FLAG/reserver	1	The low byte of the register value. Leave for further expansion.
71(0x47)	FirmwareType	1	The high byte of the register value. 3
71(0x47)	DO hold time	1	The low byte of the register value. DO status retention duration.
72(0x48)	DI controls its own DO	1	The first byte of the high register value (Bit0). 1: Enable 0: Off
72(0x48)	reserver	1	The low byte of the register value. Leave for further expansion.
73(0x49)	reserver	2	
74~89 (0x4a~0x59)	V1 to V8 is the adjustment factor of each AI route	32	Large-end format data, specifically refer to the "AI high-precision Use chapter."
90（0x5a）	AI calibration status	2	1 indicates that the AI is in the calibration state
91~106 (0x5b~6a)	32-bit count	32	Total 16 registers, 8 DI, 2 registers each.
107	Single/multiple DO hold	2	Set DO1-DO8 single/multiple

（0x6b）			channels.
108~130 (0x6c~82)	reserve	46	A total of 23 registers
131~162 (0x83~a2)	The DI combination controls the DO logic	32	A total of 16 registers

Appendix 2: AI calibration

Procedure: The following uses RS485-IO serial port communication as an example

1. Send 01 06 00 5a 00 01 68 19 and set AI Calibration Status to 1 to enter the calibration mode.
2. Send 01 04 00 00 00 08 f1 cc to query the data of the 8-way AI. For the received data 01 04 10 02 81 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 95 2D, calculate the value of each channel V1~V8. For example, for the first route.
   a) If the value is 02 81, the value is Vin=641. The input voltage is calculated according to the formula in "AI Usage Instructions" as follows: Vi =(Vin/ 1024)*5, where Vin is 641 and Vi is the known voltage, for example, 3.3V. V1 = Vi/such adjustment coefficient ((Vin / 1024) * 5) = 3.3 / ((641/1024) * 5) = 1.0543525.
   b) b) V1 is represented as float data and converted to HEX big-endian format 0x3F86 F506.
   c) c) Write 0x3F86 to the first register 0x4a corresponding to V1 and 0xF506 to the second register 0x4b corresponding to V1. Send 01 06 00 4a 3f 86 38 4e and 01 06 00 4b f5 06 3e 8e.
3. Send 01 06 00 5a 00 00 a9 d9 to exit the calibration mode.

Using the "AI Calibration Function" of Vircom's "IO Controller" dialog box, users can calibrate themselves. However, each IO controller device has been professionally calibrated after the factory, if not necessary, the user does not need to calibrate. The calibration steps are as follows:In the model, please select the correct product submodel: Only if you select the correct model, you can determine the AI type of each route is 5V, 10V, 4~20mA. To calibrate.

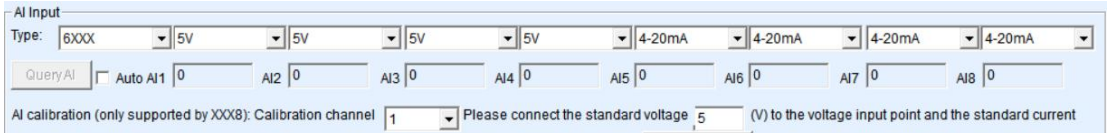

Figure 54 AI calibration

1. Select the path to be calibrated from the number of paths. Because users may not be able to connect eight test points at the same time, it is easier to adjust along the way.
2. Connect the OUT pin of IO controller to the corresponding path number, and the OUT pin is next to the AI8. By default, this OUT provides a reference voltage of 5.0V or a reference current of 10.204mA. If you prepare the standard voltage source and current source by yourself, enter the values in the corresponding input boxes.
3. Click the "AI Calibration" button to start the system calibration. After calibration, the AI value is more accurate. After calibration, the system automatically saves the calibration parameters without restarting.

Appendix 3: Dimensional drawing

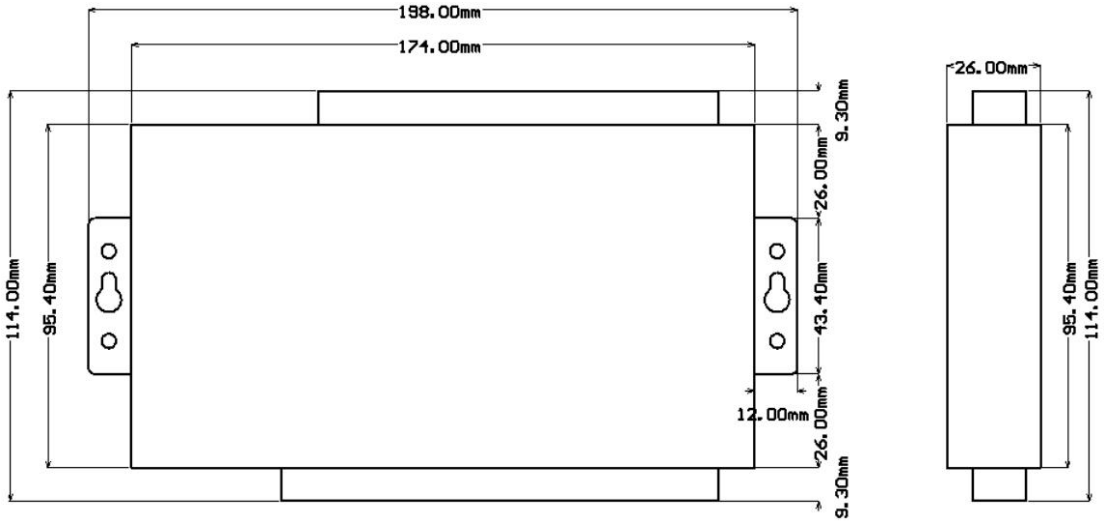

Figure 55 dimensions

5. After-sales service and Technical Support

Phone/WhatsApp: +86 18321985506

Web: www.valtoris.com

Email: support@valtoris.com