ZLAN6408

4-way DI/AI IO Controller/4G Gateway

CAT1 4G to RS485/4 channels DI /4 channels AI

Version information

The following modifications have been made to the document：

Change log

Date	Version number	Document number	Confidentiality level	Modify content.
2024-10-30	Rev.1	ZL DUI 20241030.1.0	Public	Release version

Ownership Information

TABLE OF CONTENTS

1. OVERVIEW 4
2. FUNCTION FEATURES 5
3. TECHNICAL SPECIFICATIONS 5
4. HARDWARE INSTRUCTIONS 7
DI/AI FUNCTION DESCRIPTION 9
1.1 Use the Vircom tool to connect the device. 9
1.2 Modbus Register 11
1.3 DI Instructions for use 12
1.4 DI Instructions for counting usage 13
1.5 DI Logical inversion 14
1.6 AI Instructions for use 14
1.7 AI High precision usage 15
1.8 DI Report voluntarily 16
1.9 AI Report voluntarily 19
1.10 DI and AI upload simultaneously. 20
1.11 DI trigger level hold for 2 seconds function 21
6. 4G CONFIGURATION METHODS 21
6.1 Serial AT commands 21
6.2 Firmware/Configuration File Mode 27
7. CONFIGURATION AND USAGE INSTRUCTIONS 32
7.1. Basic parameter settings 32
7.2. MODBUS RTU Convert to JSON configuration 32
7.3. Server Transparent Transmission Test 37
7.4. Modbus Protocol conversion test 39
7.5. MQTT Protocol testing 41
APPENDIX 1: PARAMETER SUMMARY 43
APPENDIX 2: AI CALIBRATION 48
AFTER-SALES SERVICE AND TECHNICAL SUPPORT 50

1. Overview

The ZLAN6408 is a newly launched 4G data acquisition gateway by Shanghai ZLan, which supports RS485 serial port acquisition functions. It can connect third-party RS485 acquisition devices and controllers on the RS485-4G interface, achieving remote acquisition and control. It also supports 4-way DI/AI acquisition, which includes digital input and analog input.

The DI input supports dry and wet nodes with optical isolation; the AI input supports 4-20MA current input, and the ADC precision is 12-bit. The properties of AI can be modified according to needs, such as 0-5V voltage, 0-10V voltage type, resistance type, etc.

◦

Figure 1 ZLAN6408 Gateway Appearance Diagram

The ZLAN6408 is a high-performance data transmission and processing device that supports TCP, HTTP, and MQTT protocols, ensuring the security and efficiency of data transmission. The gateway not only facilitates easy remote data acquisition

and transmission but also possesses powerful data processing capabilities. In particular, the gateway supports the conversion of serial DLT645 protocol and Modbus RTU protocol data to JSON format over the network, making data interaction between different systems more convenient.

2. Function Features

1. Supports custom Modbus RTU to JSON conversion.
2. Supports edge computing features including data limit violation alerts, data translation and scaling calculations, data change uploads, and device offline alerts.
3. Supports 4 channels of DI digital input and AI analog input. The analog input accuracy is 12 bits.
4. Supports 3 network mode options: TD-LTE, FDD-LTE, and GSM, including China Unicom 4G, 2G, China Mobile 4G, 2G, and China Telecom 4G networks.
5. Serial port supports baud rates from 1200 to 115200, supports 5 to 8 data bits, supports no parity, odd parity, even parity, and supports 1 to 2 stop bits.
6. Supports serial AT command configuration and supports viewing some parameters with ZLVircom software.
7. Firmware updates can be performed through the serial port.

3. Technical specifications

Parameter name	Parameters
Support mode	4G CAT1 supports 3 modes: B1/B3/B5/B8@FDD LTE B34/B38/B39/B40/B41@TDD-LTE B3/B8@GSM It includes China Unicom 4G, 2G, China Mobile 4G, 2G, and China Telecom 4G networks.

Transfer rate	LTE：Max 10Mbps（Downward）/Max 5 Mbps（Upward） GPRS：85.6Kbps（Downward）/Max85.6Kbps（Upward）
SIM	Voltage: 3V, 1.8V; Size: Mini Card
Antenna interface	50Ω/SMA rubber stick antenna or suction cup antenna optional
Serial Port Type	RS485*2:485-IO、485-4G
Network protocols	MODBUS TCP、JSON、HTTP、MQTT
RS485 Protocol	MODBUS RTU
Serial port parameters	Baud rate: 1200~115200bps; Data bits: 5 to 8 bits; Stop bits: 1 to 2 bits; Parity: None, Even, Odd.
Power interface	3.5mmTerminal block。
DI input	4-way dry/wet node, low level active
AI output	4-way 4-20ma/0-5V/0-10V, 12-bit precision
Input voltage	DC9V~24V
Work power	1W
Operating temperature	-40 degrees to 85 degrees
Storage temperature	-40 degrees to 120 degrees
Humidity range	0~95% Non-condensing
Product dimensions	Length × Width × Height=9.4cmx6.5cmx2.5cm

4. Hardware instructions

The front view of the data acquisition gateway is shown in Figure 2.

Figure 2 ZLAN6408 front view

The gateway features a radiation-resistant metal casing with two mounting ears on both sides, allowing for screw fixation.

Panel light:

Indicator light	Color	Note
POWER	Red	Device is powered on normally.
4G/TCP	Green/blue	Green represents 4G call connected / Blue represents link established.
DI	Green	Green represents a valid DI, with a value of 1.

Figure 3Interface diagram 1

The interface of the collection gateway is as shown in Figure 3:

Reset button: Pressing for more than 3 seconds can reset the parameters of the 4G gateway.

RS485 Interface: 2 channels of RS485 signal input, -IO, +IO can be read by MODBU RTU master for AI and DI parameters. -4G, +4G can connect to 485 instrument slave, please note not to connect power.

SIM Interface: When installing the SIM card, ensure the device is not powered. Use a pen tip or screwdriver to pop out the SIM card slot, and then push the SIM card in with the metal side down.

Antenna: The antenna interface of the acquisition gateway uses 50 Ω /SMA (male), and the external antenna must be suitable for the 4G operating frequency band.

The interfaces behind the Gathering Gateway are as shown in Figure 4:

Figure 4 Interface Diagram 2

1.Power: Terminal block input, input voltage DC +9V~ +24VDC.

2.GND: Connected to the negative pole of the power supply. When using dry contact input, connect this terminal across switches with DI1~DI4 to collect the switch status.

3.COM: Common terminal, internal interface, not connected externally.

4.DI1~DI4: 4 channels of digital input.

5.AI1~AI4: 4 channels of analog input.

1. 4-way digital inputDI1~DI4。

Supports passive switch (dry contact) and active level (wet contact). For dry contacts, simply short it with GND to collect a 1 signal. For wet contacts, the range of the active level difference with GND is as follows.：

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

2. 4-channel analog input: Precision is 12 bits, default is 4-20mA analog input (customization required)：
   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, etc.

   Both voltage and current are relative to GND.

5. DI/AI Function DescriptionUse the Vircom tool to connect the device.

The gateway configures IO parameters through the RS485 interface. Power on the device and connect the -IO and +IO interfaces in RS485. Use VIRCOM software to open the main interface device management and click the "IO Controller" button directly.

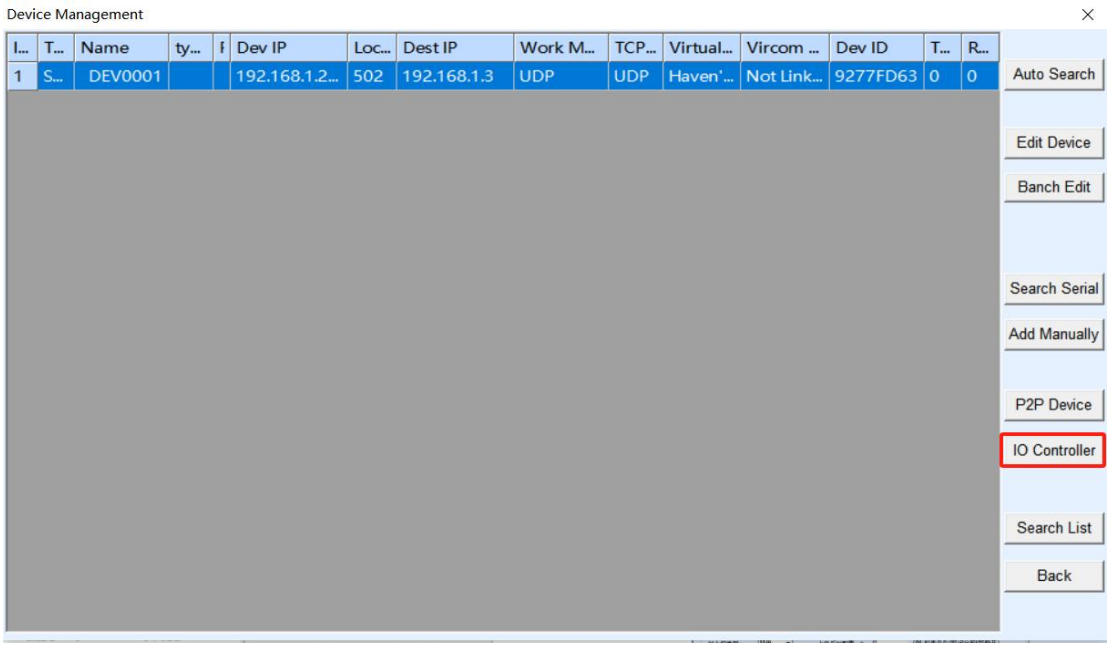


Figure 5 How to enter the IO controller dialog box

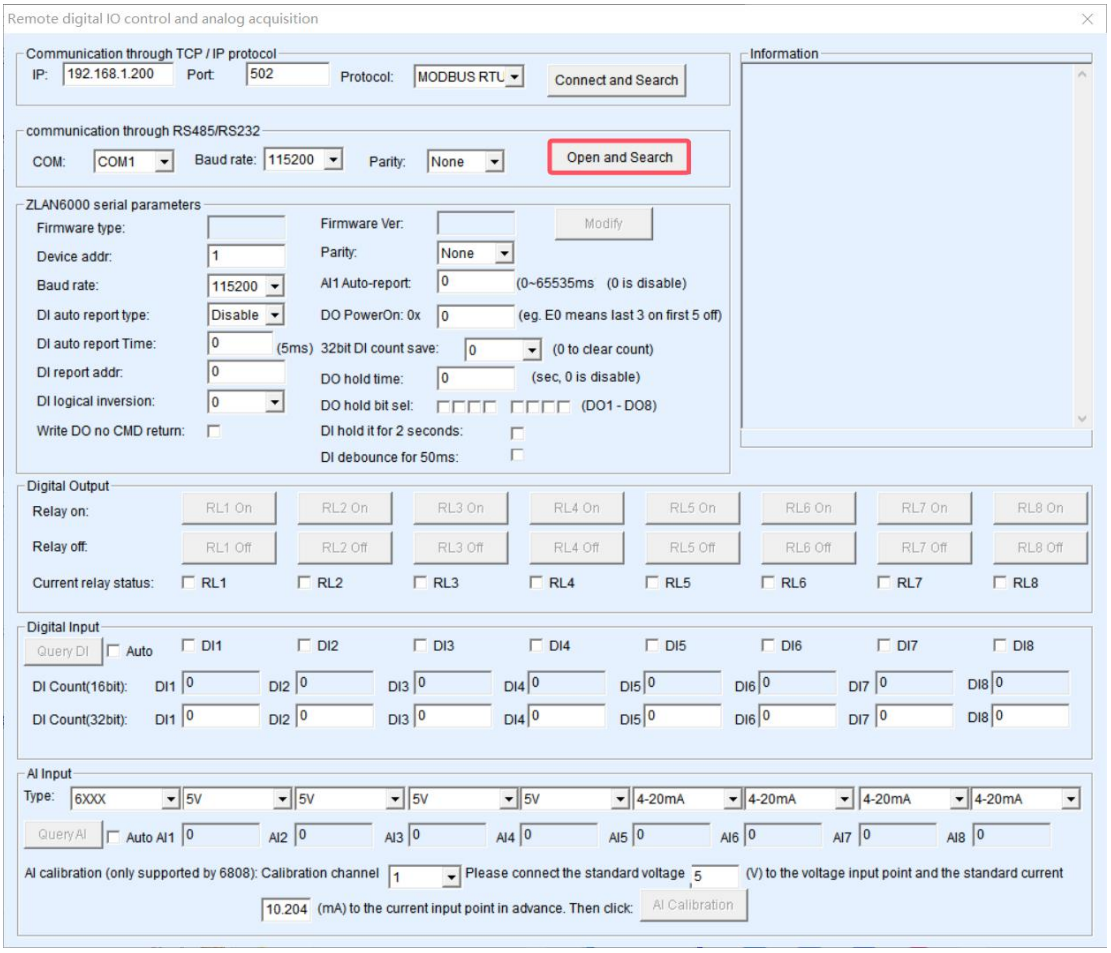


Figure 6 IO Controller Dialog Box

Select the correct COM port and click "Open and Search" to communicate with the device, without needing to choose the baud rate. If you have set the parity bit before, please select the corresponding parity bit and then click "Open and Search". After opening the COM port, obtain the device parameters through the software Modbus RTU command.

Once the device parameters are obtained, they will be displayed in the dialog box. You can then proceed to modify the parameters, read DI, read AI, and perform other tests.

1.2 Modbus Register

Serial port supports Modbus RTU commands. Specific registers and address ranges are as follows.：

Table 1. Modbus Register Summary

Function code	Function	Address range
01/02	Read DI	0~3
04	Read AI	0~3
04	Read AI high precision value	32~35
03	Read the basic parameters.	63~67
03	Read extended parameters	68~162
03	Read DI 16-bit count	0~3
03	Resd DI 32-bit counting	256~263
06	Set parameters	63~67
06	Set the extended parameters	68~162
06	Set DI 16-bit count	0~3

06	Set DI 32-bit counter	256~263
16	Set multi DI 16-bit count	0~3
16	Set multi DI 32-bit count	256~263
16	Set the basic parameters.	63~67
16	Set the extended parameters	68~162

1.3 DI Instructions for use

Read DI using 01 instruction, address range 0~3, corresponding to DI1 to DI4. The instruction format is as follows:

Byte count	1	1	1	1	1	1	1	1
Name	Device address	01	Starting address high	Starting address low	High length	Low length	CRC Tall	CRC Low

For example, the Modbus RTU command to read 4 DI is:

Send-> 01 01 00 00 00 04 3D C9

Return-> 01 01 01 01 90 48

When the DI input is at a low level (note that when the device supply voltage is above 12V, a 5V voltage input is considered a low level), the corresponding bit returns as 1, and the fourth byte in the return instruction is set to 0x01 indicating that the first channel is in a closed (low level) state.

IO Controller Dialog Control Demonstration：

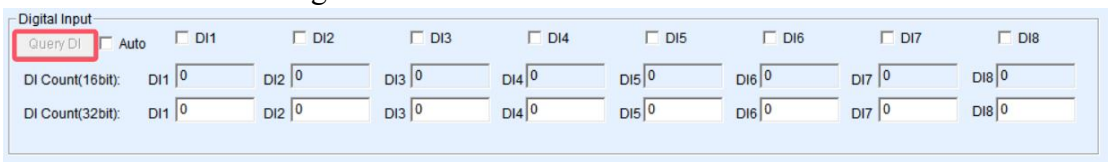


Figure 7 DI Read in IO Controller Dialog Box

After Vircom successfully connects to the device, click "Query DI Status" to check the status of the DI. When the DI is at a low level, the corresponding indicator

light turns on, and the returned bit is 1. As shown in the figure, if DI4 is checked, it indicates that DI4 is in a low level state.

Clicking the "Auto" checkbox allows for automatic querying of the DI status every 1 second and displays the result.

1.4 DI Instructions for counting usage

When the DI changes from high to low and back to high again, it counts as one cycle. DI counting is divided into three types: 16-bit non-storage counting, 32-bit non-storage counting, and 32-bit storage counting. Non-storage means it starts from zero after a power-off, while storage will retain the count after a power failure. The 32-bit non-storage counting and 32-bit storage counting share the same register location but are configured differently.

DI counting already includes debouncing processing, with a debouncing time of 10ms.

By using Modbus function code 03, reading register positions 0 to 3 will give you the 16-bit non-storage count, with the data in big-endian format. By using function code 03, reading positions 256 to 263 will allow you to read the 32-bit count, also with the data in big-endian format.

Byte count	1	1	1	1	1	1	1	1
名称	设备地址	03	起始地址高	起始地址低	长度高	长度低	CRC 高	CRC 低

For example, the Modbus RTU command to read the 16-bit count of DI4 is:

Send-> 01 03 00 03 00 01 74 0a

Return-> 01 03 02 01 0a 39 d3

Modbus TCP command is:：

Send-> 00 00 00 00 00 06 01 03 00 03 00 01

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

Here, register 3 is read, and the returned data 01 0a represents the value 266.

For example, the Modbus RTU instruction for reading a 32-bit count from DI4 is:

Send-> 01 03 01 06 00 02 25 F6

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

Modbus TCP 指令为：

Send-> 00 00 00 00 00 06 01 03 01 06 00 02

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

Here, 00 00 01 14 represents the number 276.

By using the "32-bit DI count save" function in the ZLVircom configuration software, you can set the 32-bit counter to save or not save the count. If you want to clear the saved data and start counting again, simply set the "32-bit DI count save" function to 0 to reset the count.

1.5 DI Logical inversion

When the DI input is normally at a low level, the corresponding bit returns as 1. The default DI input is high, with low level being active. If now it is required that the DI input is active on high level, meaning the default bit is 1, and if a low level is connected, then the bit is 0. In this case, the "DI Logic Inversion Function" can be selected.

The DI inversion also affects DI counting, where DI counting increases by 1 when DI changes from 0 to 1, that is, when the high level becomes low level. If DI logic is inverted, then it changes to counting when the low level becomes high level.

The method for setting DI logic inversion is as follows。

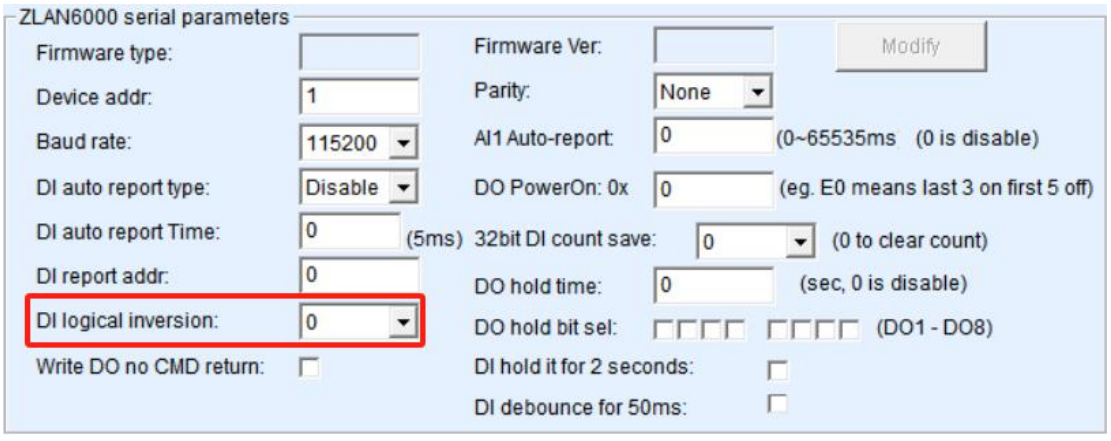


Figure 8 DI Inversion Setting in the IO Controller Dialog Box

1.6 AI Instructions for use

Using the Modbus 04 command to read the values from registers 0 to 3, you can

obtain the values of AI1 to AI4. The data is stored in big-endian format.

Byte count	1	1	1	1	1	1	1	1
Name	Device address	04	Starting address high	Starting address low	High length	Low length	CRC Tall	CRC Low

For example, the Modbus RTU command to read the value of AI1 is:

Send-> 01 04 00 00 00 01 31 ca

Return-> 01 04 02 01 82 38 c1

Modbus TCPThe instruction is：

Send-> 00 00 00 00 00 06 01 04 00 00 00 01

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

The specific usage of the data 01 82 is related to the type of AI. Converting 01 82 to decimal gives Vin=386. The calculation formula for different AI types is as follows:

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

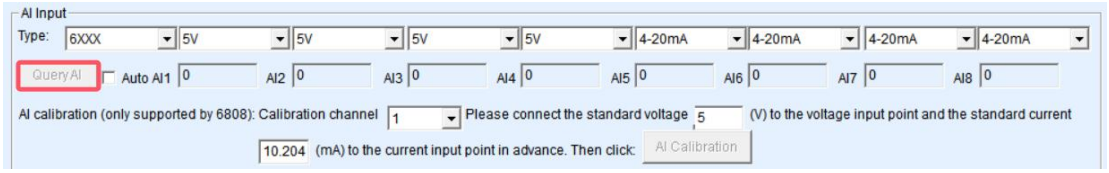


Figure 9 AI Read in the IO Controller Dialog Box,

After Vircom successfully connects to the device, click "Query AI Status" to check the AI values, or click "Auto" to query once per second. Before querying, you need to select the model based on the one you purchased. After selecting the model, the AI1~AI4 analog interface types will be automatically configured according to the standard settings, allowing the real current values of the interface to be displayed in the value dialog box.

1.7 AI High precision usage

The data acquisition gateway provides a more precise AI numerical calculation method. Compared to ordinary precision, it does not automatically filter small

fluctuations to 0 voltage, nor does it automatically set very small changes in values to the voltage of the last acquisition. Therefore, it can more truly reflect the voltage value, but there may be more noise.

To obtain the AI high-precision value, use the 04 function code to read the contents of registers 32 to 35 (0x20 to 0x23). The data format is big-endian. This is a value Vh with 12-bit effective precision.

The calculation for the input point current is:

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

Where Vi (i=1~4) is the adjustment coefficient for each channel, defaulting to 1.0. You can use the 03 function code to read registers starting at 0x4a to 0x51 (decimal 74 to 81) to obtain the corresponding float (float) type big-endian format data for V1 to V4. For example, the float data reading result for 1.063 is in the hexadecimal form of 0x3F88 1062.

For example, to read the adjustment coefficient for A1:

Send -> 01 03 00 4a 00 02 e5 dd

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

Where 3f 80 00 00 represents 1.0.

Then read the Vh for the first channel:

Send -> 01 04 00 20 00 01 30 00

Return -> 01 04 02 07 c7 fa 92

Where 07 c7 represents 1991, and plugging this into the formula gives a voltage of:

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

The Vi adjustment coefficient is calibrated after leaving the factory, which can ensure the accuracy of the product's calculated values.

1.8 DI Report voluntarily

ZLAN6408 is a standard MODBUS device that operates in a question-and-answer format. However, some users wish to receive feedback immediately upon any change in DI input, which is the function of active reporting. Here we introduce the active reporting feature of 6408. The settings are as shown in

the figure, set "Enable DI Active Reporting" to 1 to activate the active reporting function. The DI reporting address generally should not be the same as the device address.

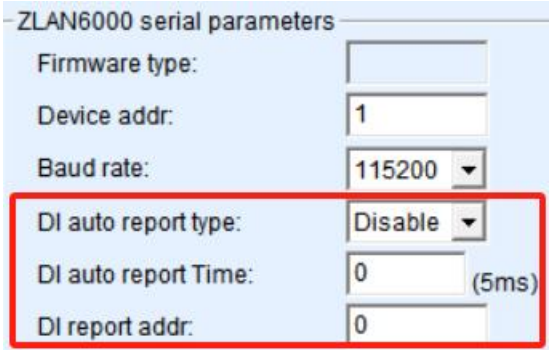


Figure 10 DI Active Reporting Settings

When the status of DI changes, after enabling active reporting for DI, the 05 command will be sent. The 05 command can achieve the function of triggering the DO of another Modbus device by controlling the change of DI.

Byte count	1	1	1	1	1	1	1	1
Name	DI report address	05	Starting address high	Low starting address	Ff or 00	00	CRC height	CRC LOW

The example is as follows:

DI1 Become a high-level input

00 05 00 10 00 00 CD 2E

DI1 Become low-level input

00 05 00 10 ff 00 8C 2E

DI2 Become a high-level input

00 05 00 11 00 00 9C 1E

DI2 Become low-level input

00 05 00 11 ff 00 DD EE

DI3 Become a high-level input

00 05 00 12 00 00 6C 1E

DI3 Become low-level input

00 05 00 12 ff 00 2D EE

DI4 Become a high-level input

00 05 00 13 00 00 3D DE

DI4 Become low-level input

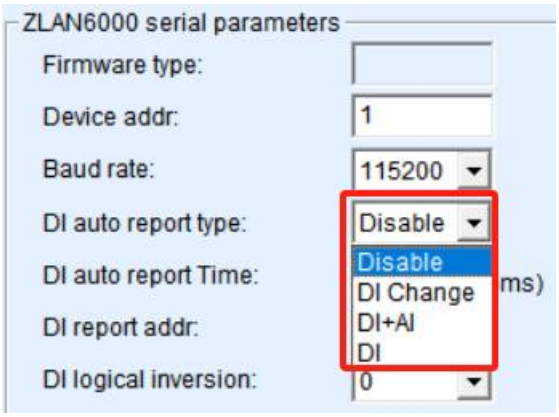
00 05 00 13 ff 00 7C 2E

When using ZLVircom for testing, the active reporting of DI will update the current status value of DI. Active reporting will be sent to both 485-IO and 4G simultaneously.

When the active reporting time is set to 0, active reporting is disabled. When set to 1, it starts the active reporting of DI changes. When set to any other number, it will report according to a scheduled interval. If set to an even number, it will report 8 DIs every 15 instructions at regular intervals. If set to an odd number, it will enable simultaneous reporting of DI and AI, refer to the later content of this chapter for details. When set to n, where n is a non-zero even number, the reporting time for DI is (n-1)*5 milliseconds. For example, if the first 4 DIs are short-circuited to GND and the last 4 are left floating, sending DI active reporting:

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

With the new version of Vircom, direct configuration is possible and does not require odd/even settings.



✳ For firmware versions below V28, the active reporting time for DI is X multiplied by 5ms. For example, if X is 200, then the reporting time is 1000ms. For firmware versions V28 and above, if the active reporting time for DI is greater than 100, such as 108, then the reporting is (108-100) seconds.

1.9 AI Report voluntarily

The active reporting function of AI allows the collected analog data to be automatically sent to the host computer. This method does not require the host computer to perform Modbus instruction queries, which is very useful for network analog monitoring based on the Internet.

The reporting time for analog data can be set, with the time interval ranging from 0 to 65535, in milliseconds. If set to 0, it indicates that active reporting is not enabled. This can be directly set in the IO controller dialog box.

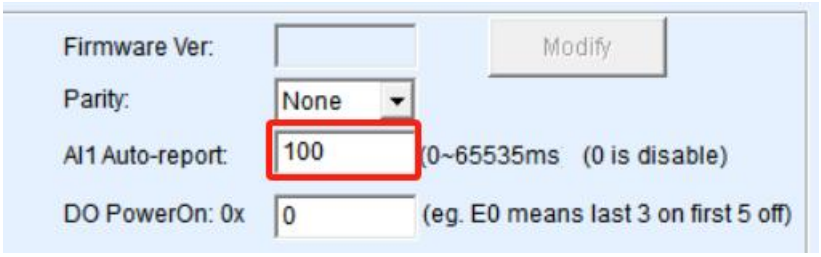


Figure 11 Set AI active reporting time in the IO controller dialog box

The AI-initiated command is:

- When converting the protocol to Modbus RTU：01 04 08 H1 L1 H2 L2 H3 L3 H4 L4 C1 C2。
- When converting the protocol to Modbus TCP：：00 00 00 00 00 0B 01 04 08 H1 L1 H2 L2 H3 L3 H4 L4

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, C2 are CRCs.

Before AI proactively reports, if there is a device parameter search, AI proactive reporting will be paused for 5 seconds, which can prevent conflicts between AI proactive reporting and parameter search.

1.10 DI and AI upload simultaneously.

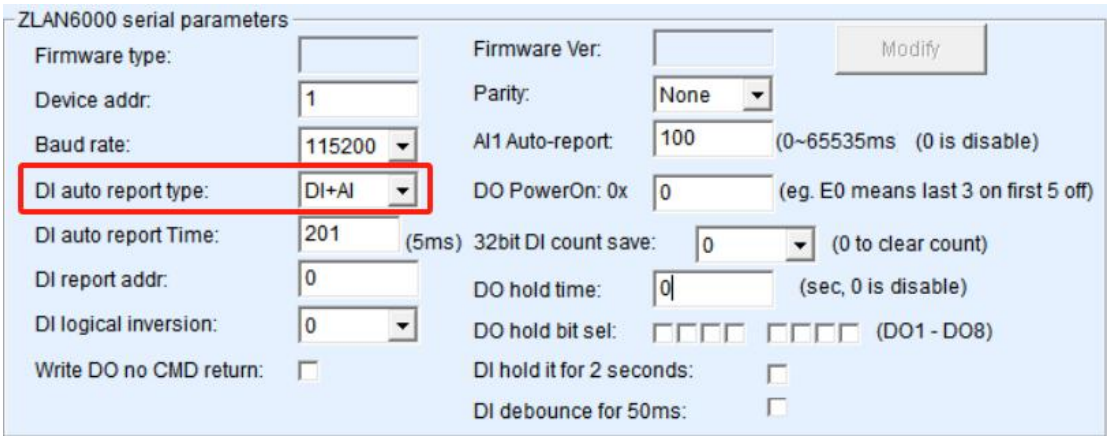


Figure 12: Simultaneous active reporting settings for DI and AI

In the software, when the DI active reporting setting is enabled and set to a value greater than 1 (2 to 255), the reporting period for AI and DI is calculated by subtracting 1 from the set value and then multiplying by 5. For example, if the setting is 201, the reporting period would be (201-1)*5=1000ms.

For versions V28 and above, when the active reporting time is greater than 100, such as 108, the reporting is (108-100) seconds.

This function allows the current values of both AI (Analog Input) and DI (Digital Input) to be reported simultaneously. The Modbus RTU format for the transmission is as follows:

00 04 0A 03 01 00 00 00 00 00 00 00 08 2f a5

The first 00 is set for the DI reporting address, using function code 04 to report four AI registers and four DI data. The 03 01 indicates the data from AI1, 08 represents the status of four DI channels, and the 08 signifies that the fourth channel is active.

When AI and DI are reported simultaneously, both AI data and DI data can be seen at the same time on the IO controller interface, without needing to click "Auto" to query the data. When AI and DI actively report, the data will be sent to both -IO, +IO, and 4G simultaneously.

Before DI and AI actively report, if there is a device parameter search, DI and AI active reporting will be paused for 5 seconds to avoid conflicts between active

reporting and parameter searching.

1.11 DI trigger level hold for 2 seconds function

The V30 start version of ZLAN6408 supports the function of maintaining the DI trigger level for 2 seconds after it disappears, meaning that even if the DI state has already become 0 after the trigger DI is 1, it will still keep the 1 for another 2 seconds. For example, if the trigger DI is 0.5 seconds, the actual time the DI state is read as 1 will be 2.5 seconds (0.5 seconds plus 2 seconds)

Figure 13 DI hold for 2 seconds function

6. 4G configuration methods

The gateway can be configured through the serial port.

6.1 Serial AT commands

Download the ZLVircom configuration tool, which can configure the data acquisition gateway via the serial port.

Connect the USB to RS485 cable to the RS485 -4G and +4G interfaces of the data acquisition gateway, and power on the gateway. Open ZLVircom (hereinafter referred to as the configuration tool), and enter the main interface of the configuration tool as shown in Figure 14.

Click on Device Management, select Serial Port Search, as shown in Figure 15, which will bring up the Serial Port Parameter Selection interface, as shown in Figure 16. Select the serial port, which in this case is COM15, with a baud rate of 115200. The 115200 is the factory default setting, but if the user has previously set the gateway to a different baud rate (such as 9600), it can still be searched for.

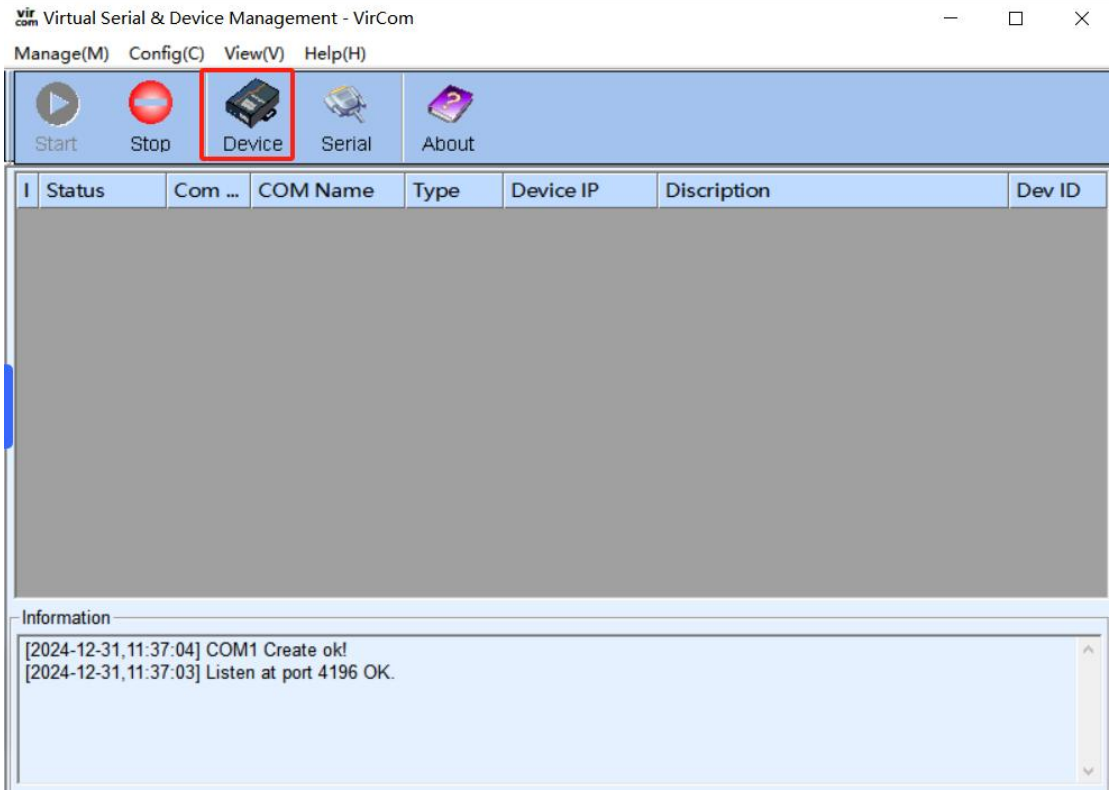


Figure 14 Main Interface of the Configuration Tool

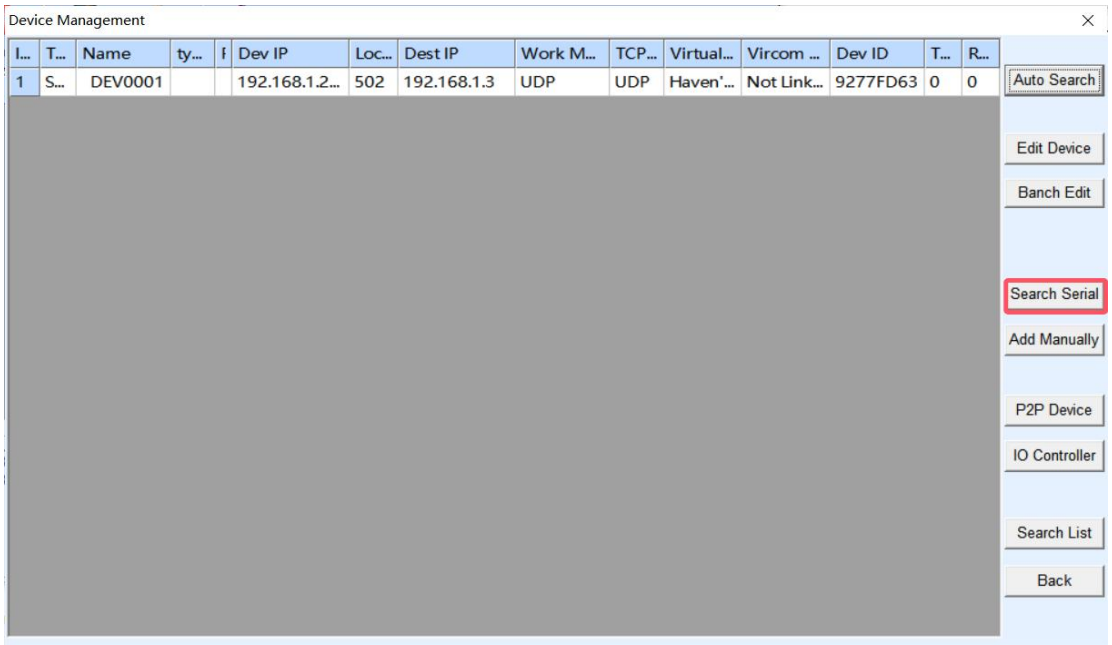


Figure 15 Serial Port Search Interface

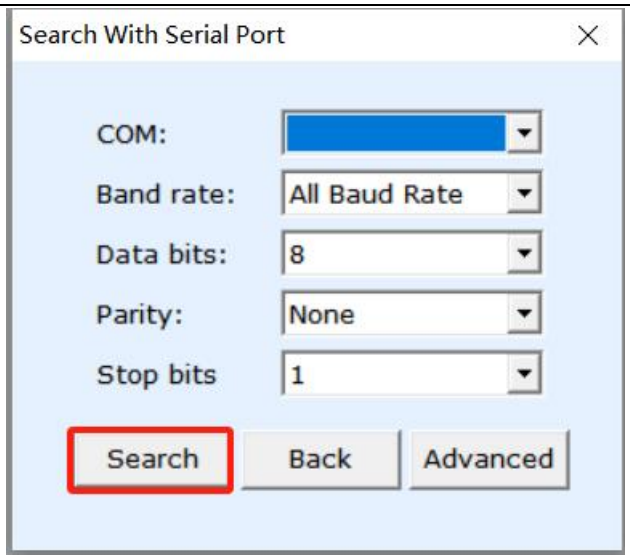


Figure 16 Serial Port Parameter Settings

After powering on, wait for 15 to 20 seconds until the dialing light starts to blink, then select and click on the "Search" serial port. At this point, the configuration tool will attempt to communicate with the device. If successful, it will enter the ConfTool interface. As shown in the following figure 17:

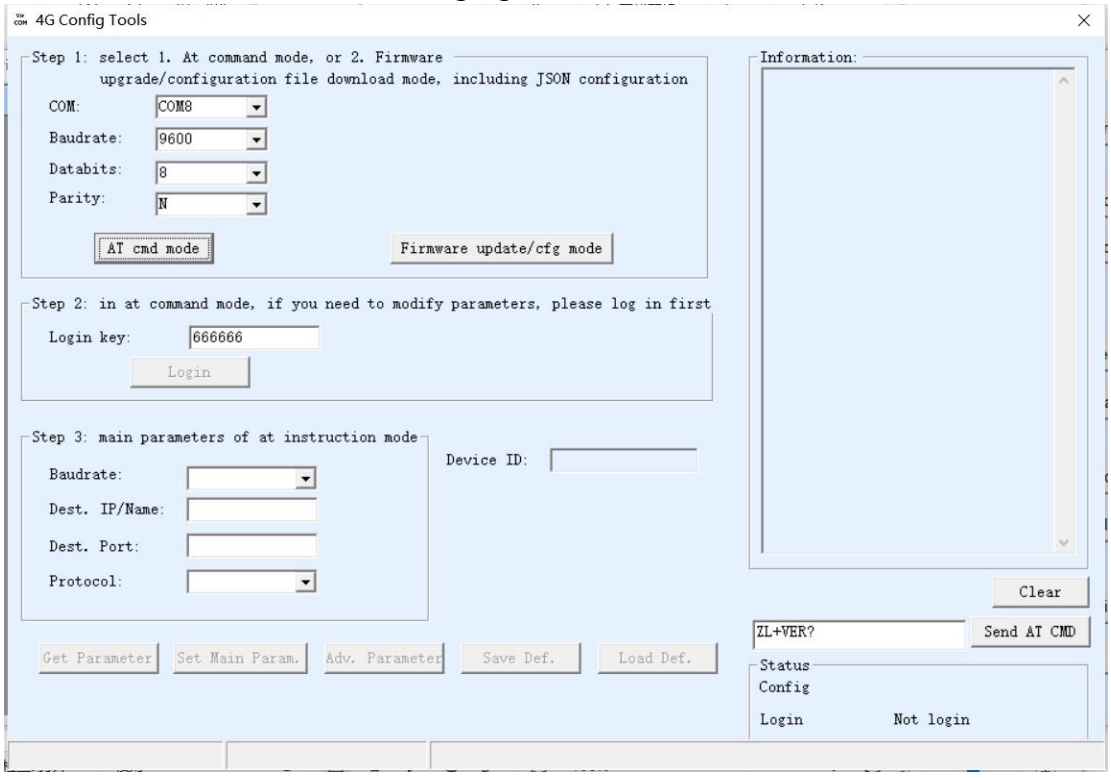


Figure 17 ConfTool Interface

Click to enter AT command mode, the configuration tool will attempt to communicate with the device. If communication is successful, the AT command return information will be displayed on the right side, and the configuration mode will show that it has entered configuration mode, as shown in the figure below 18:

Figure 18 enter the configuration mode interface

The default login password is 666666, and before clicking "Login", the parameters are read-only and cannot be set or modified. After clicking the "Login button":

You can see that after logging in, the login status changes to "Logged in", as shown in Figure 19.

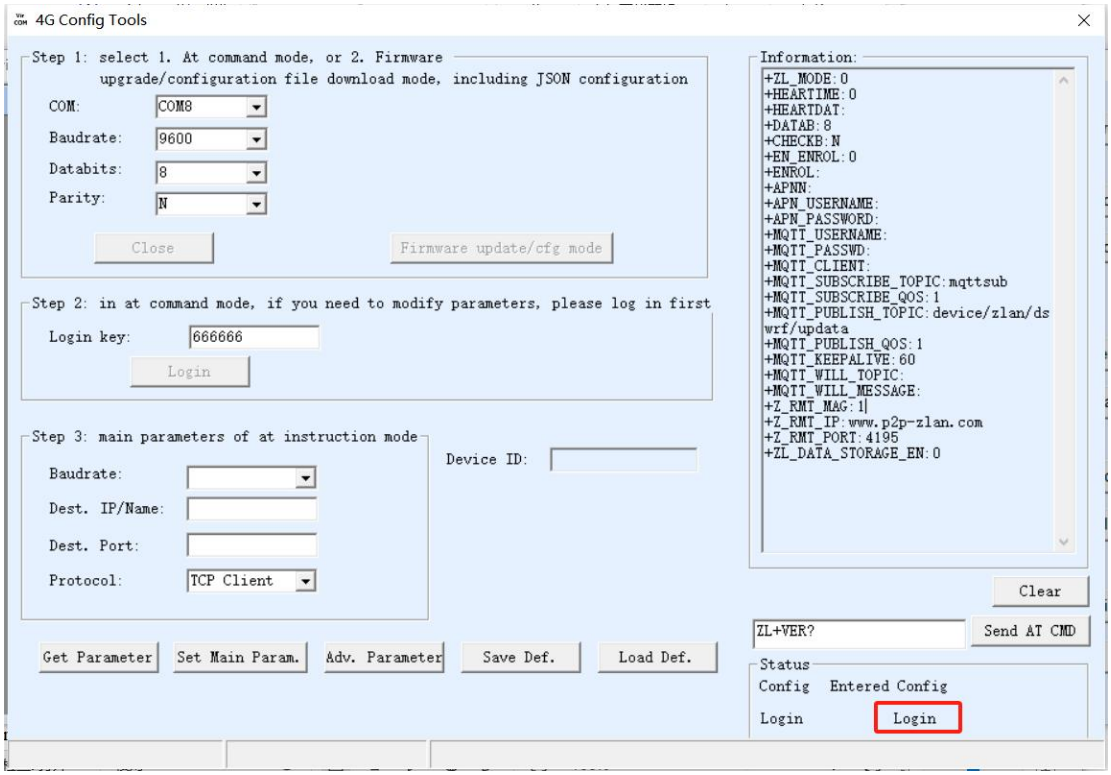


Figure 19 Login Interface

The main parameters of the AT command mode include baud rate, destination IP, destination port, and protocol. The protocol supports either TCP or UDP. After modifying the corresponding parameters, clicking "Set Parameters" will apply the new settings to the device, and the device will return the successfully set parameters,

as shown in Figure 20.

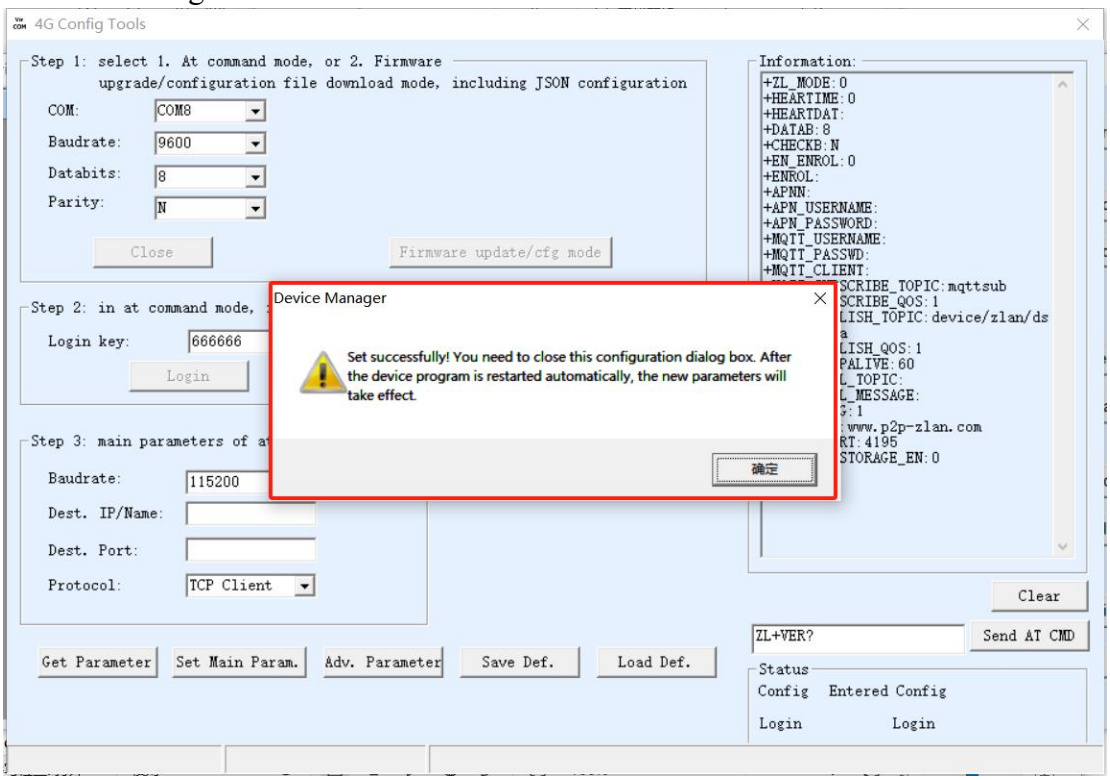


Figure 20 Set parameters

The "Get Parameters" button can obtain the current parameters of the device. This is achieved by sending AT commands to retrieve the parameters, and the data returned by the AT commands is listed on the right. For more information on AT commands, refer to other sections of this article. Since "Open" successfully triggers the "Get Parameters" command automatically, it is generally unnecessary to click the "Get Parameters" button.

Clicking the "Advanced Parameters" button will display the advanced parameters box as shown in Figure 21. Commonly used parameters are:

1. Heartbeat interval: The heartbeat packet interval can be set to 15 seconds.
2. Heartbeat content: Set the content of the heartbeat packet.
3. Serial port data bits
4. Serial port parity bit
5. Enable registration packet: Whether to enable the registration packet.
6. Registration packet content: The content of the registration packet sent after

connecting to the server.

7. APN: The access point name of the APN.
8. APN username
9. APN password
10. MQTT parameters: Parameters used to set up access to the MQTT server.
11. Device remote management: Used for devices with remote management features to access the remote server.

After selecting the parameters, click the "Apply Advanced Parameters" button, and observe the information bar on the right to check if the settings information returned by the device matches the information entered, as shown in Figure 22.

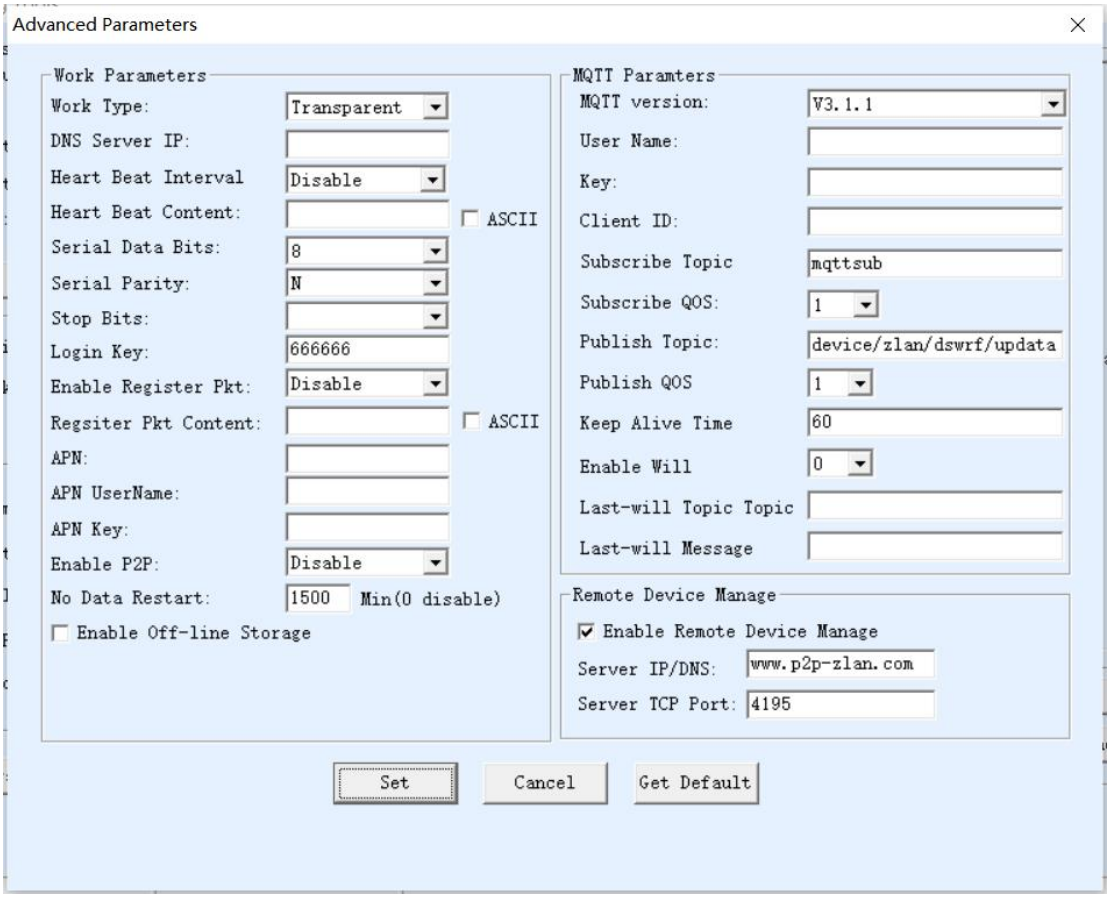


Figure 21 Advanced Parameters

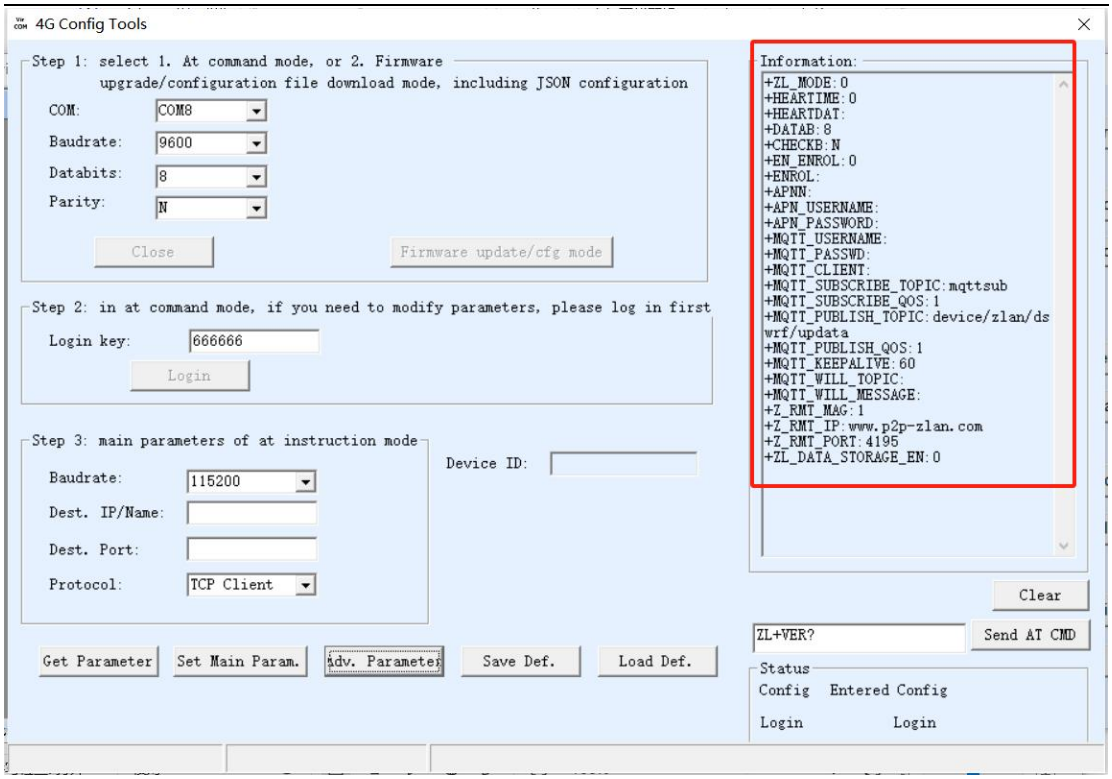


Figure 22 Set advanced parameters and return information

6.2 Firmware/Configuration File Mode

After entering the ConfTool interface, click the Firmware/Configuration File mode button as shown in Figure 23.

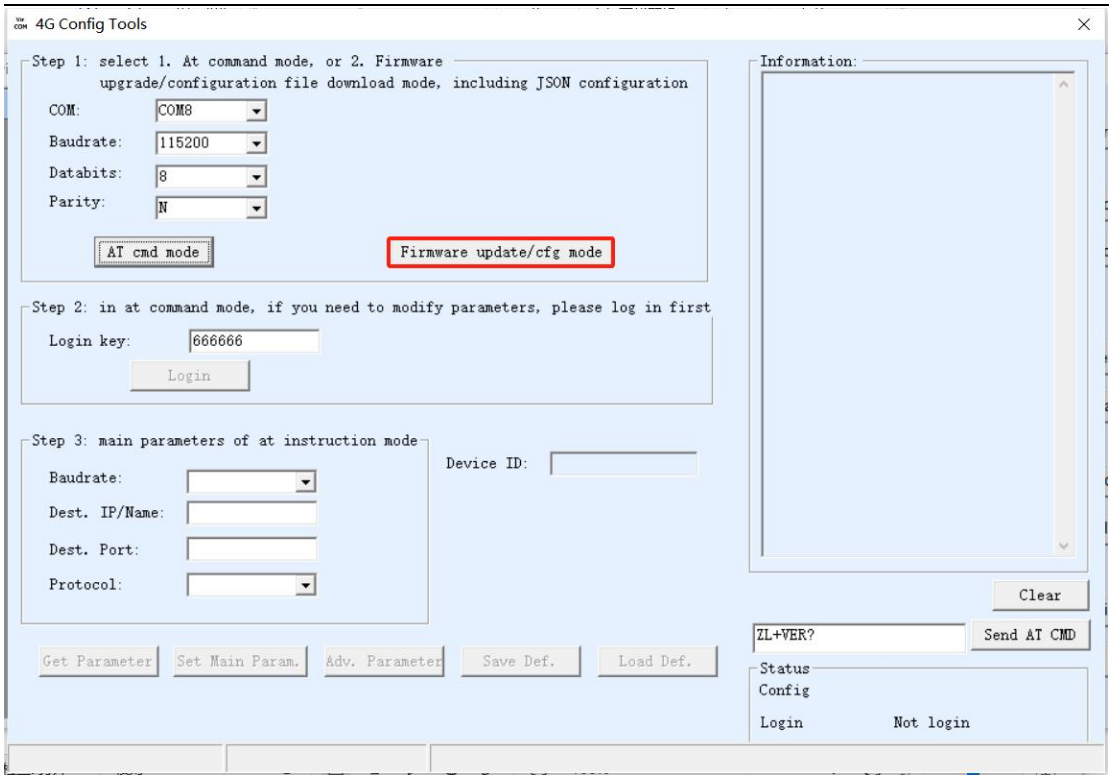


Figure 23 Configuration Interface

Navigate to the firmware/configuration file interface, first create a local configuration webpage root directory to store the configuration files. If MQTT transmission is needed, then click on MQTT configuration to input the information for connecting to the MQTT server. After setting up, click to save the MQTT configuration.

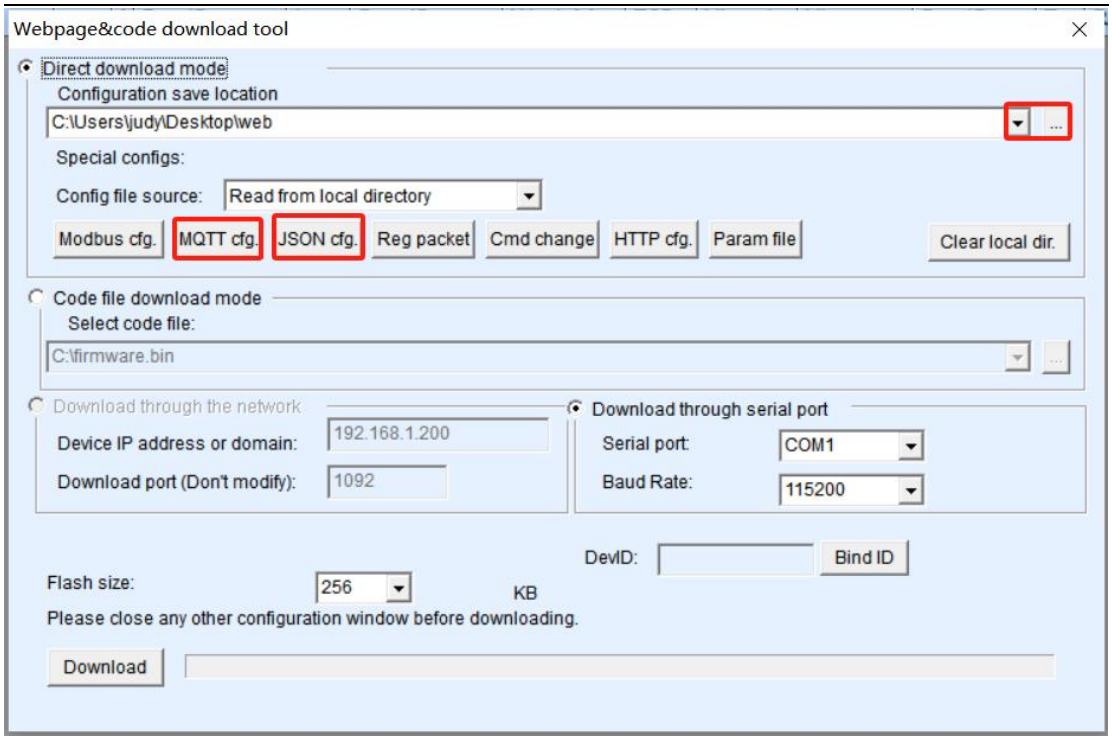


Figure 24 Web Configuration Page

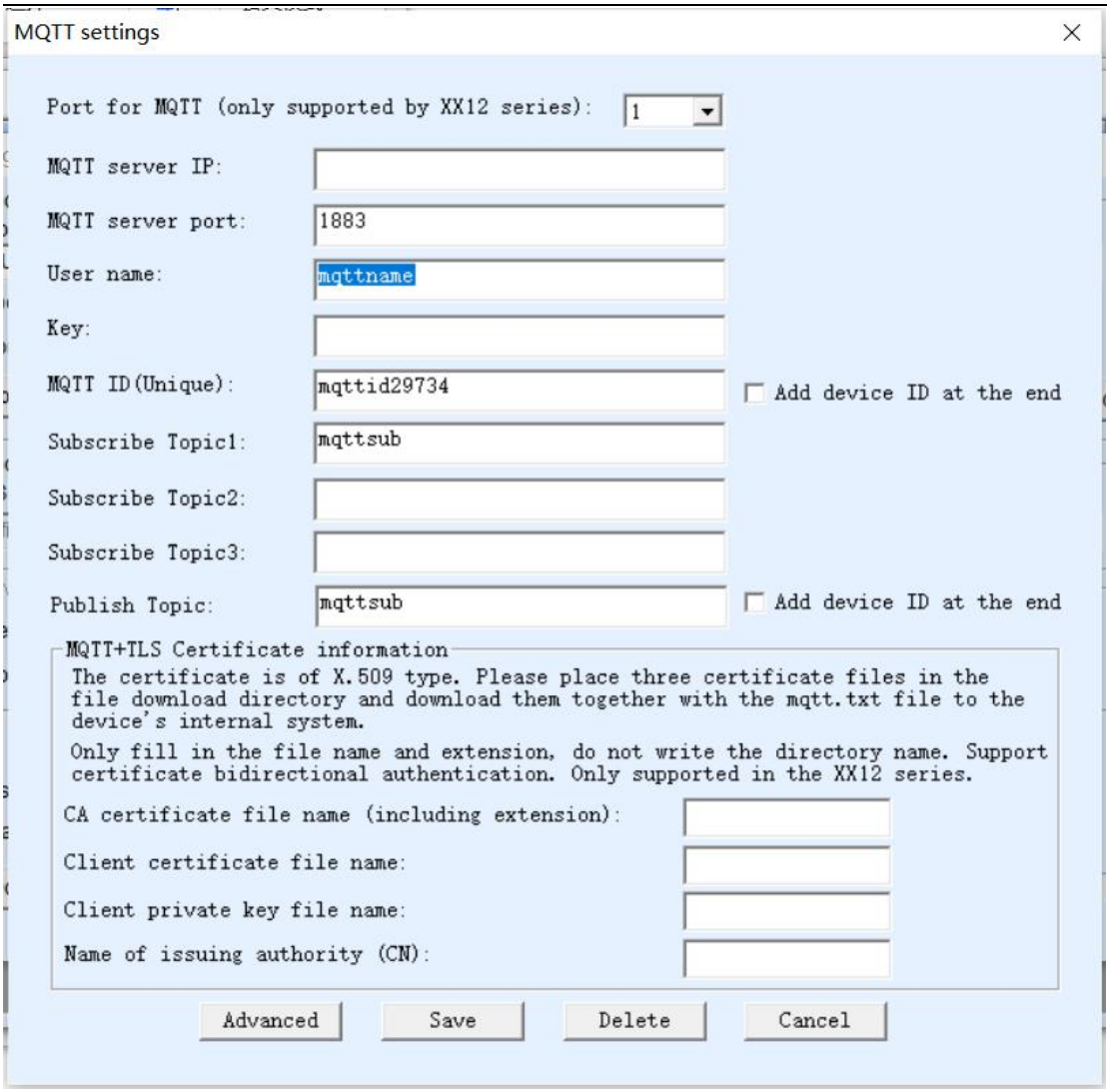


Figure 25 MQTT Configuration Interface

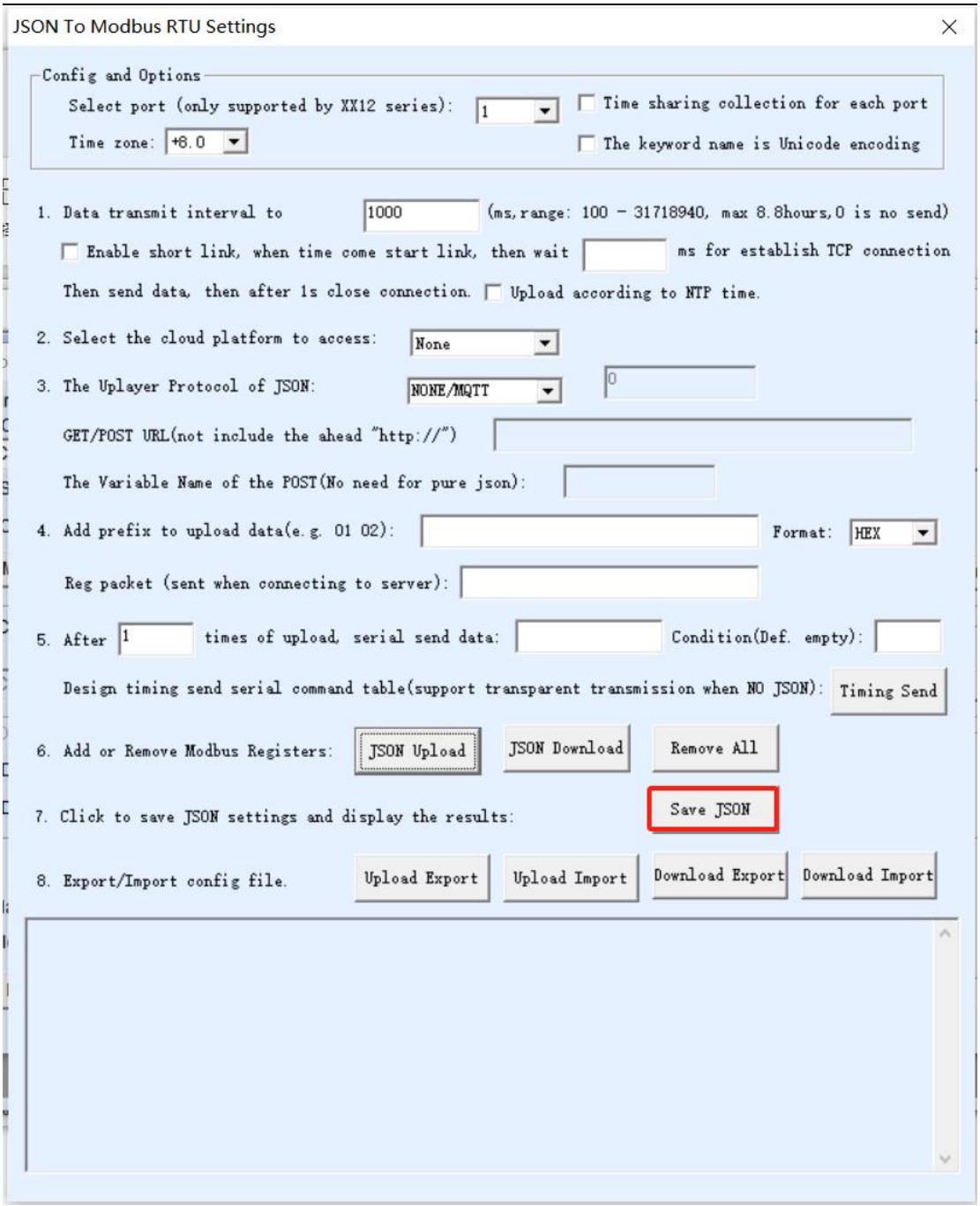


Figure 26 JSON Configuration Interface

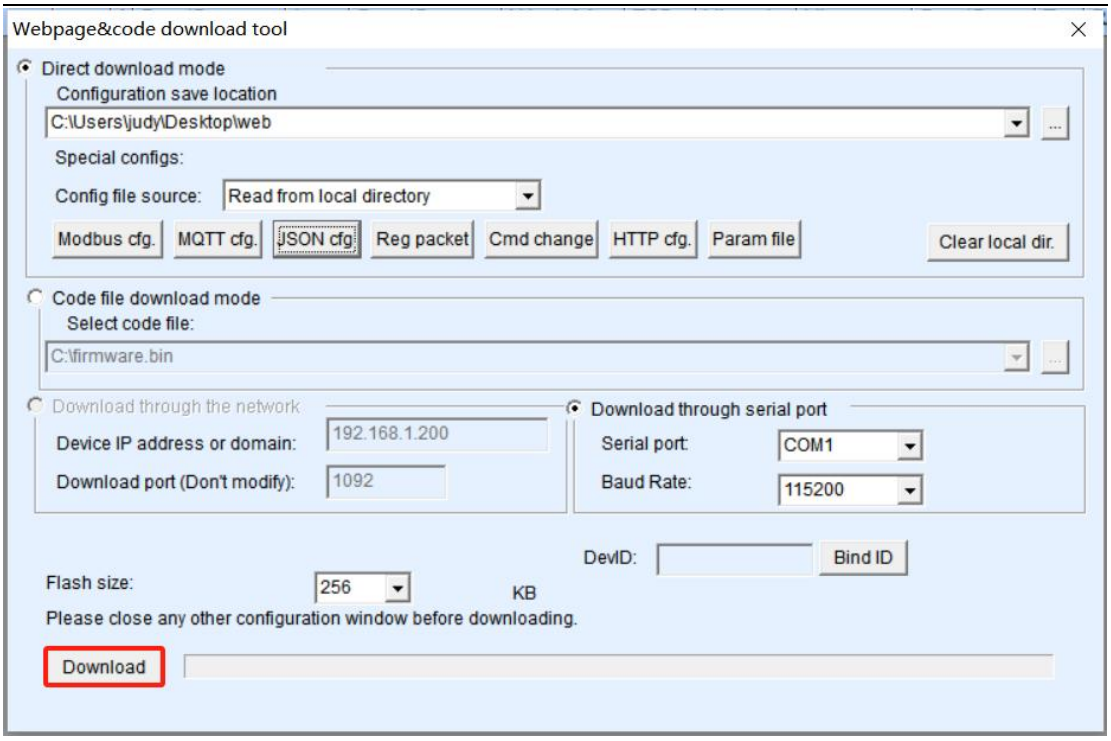


Figure 27 Download Interface

7. Configuration and usage instructions

7.1. Basic parameter settings

Refer to section 6.1 serial AT commands, after entering AI command mode, you can configure parameters such as the gateway's baud rate, IP address, and port. After configuration, click to set the main parameters.

7.2. MODBUS RTU Convert to JSON configuration

7.2.1. Software configuration

Refer to section 6.2 firmware/configuration file mode, after entering the ConfTool interface, click the firmware/configuration file mode button, then click JSON configuration to enter the configuration page.

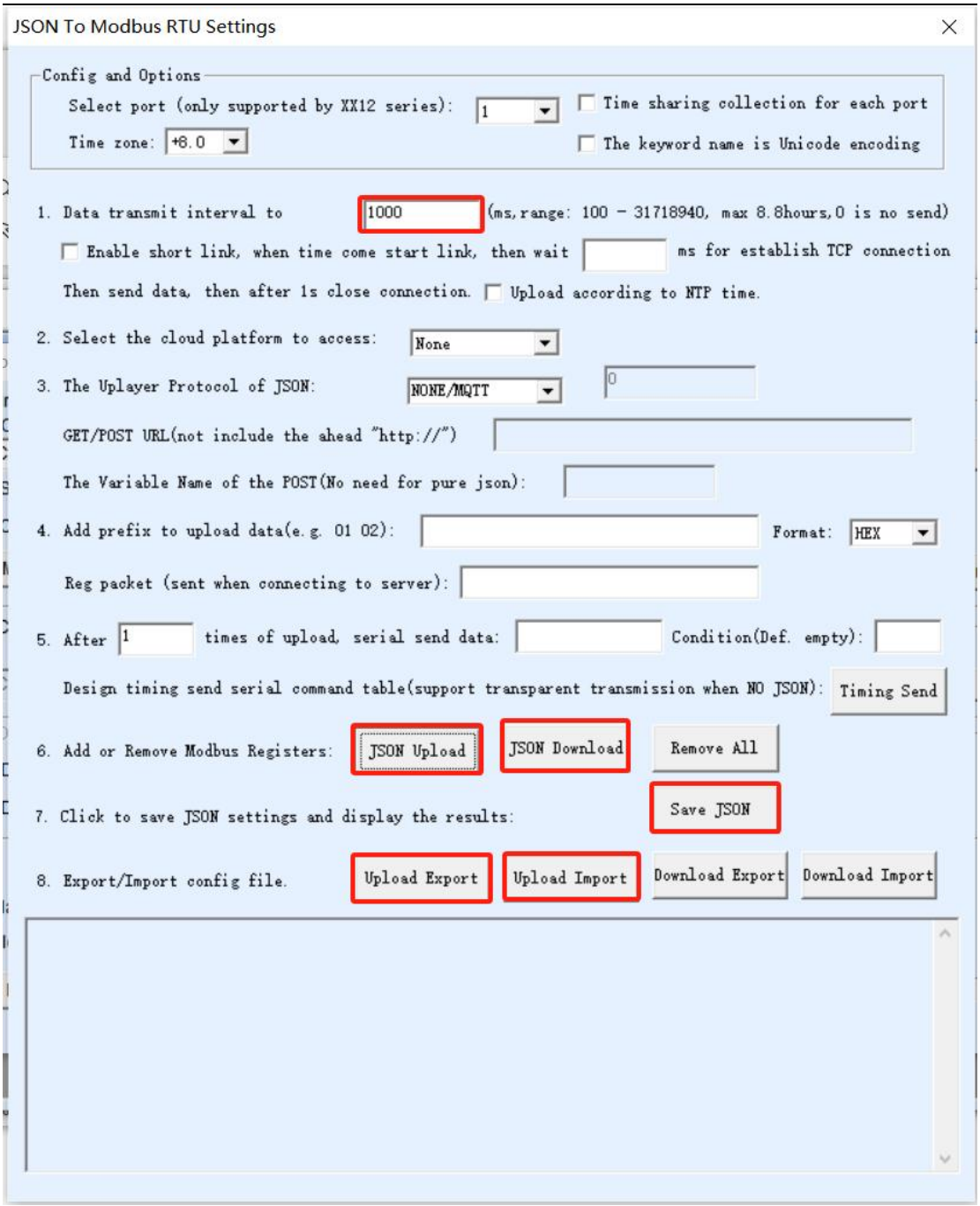


Figure 28 JSON Configuration

As shown in the above figure, several important parameters are introduced as follows:

Server time for sending data: The default interval at which JSON data is sent to the server, measured in milliseconds.

Add or delete Modbus registers: Click on JSON upload/download to configure the JSON upload/download parameters, click on delete all to remove all configured JSON parameters.

Save JSON settings: Click to save the JSON settings and display the results.

Upload import/export: Export the JSON upload configuration to a CSV file or import a CSV file.

Clicking on JSON upload allows you to enter the JSON configuration page.

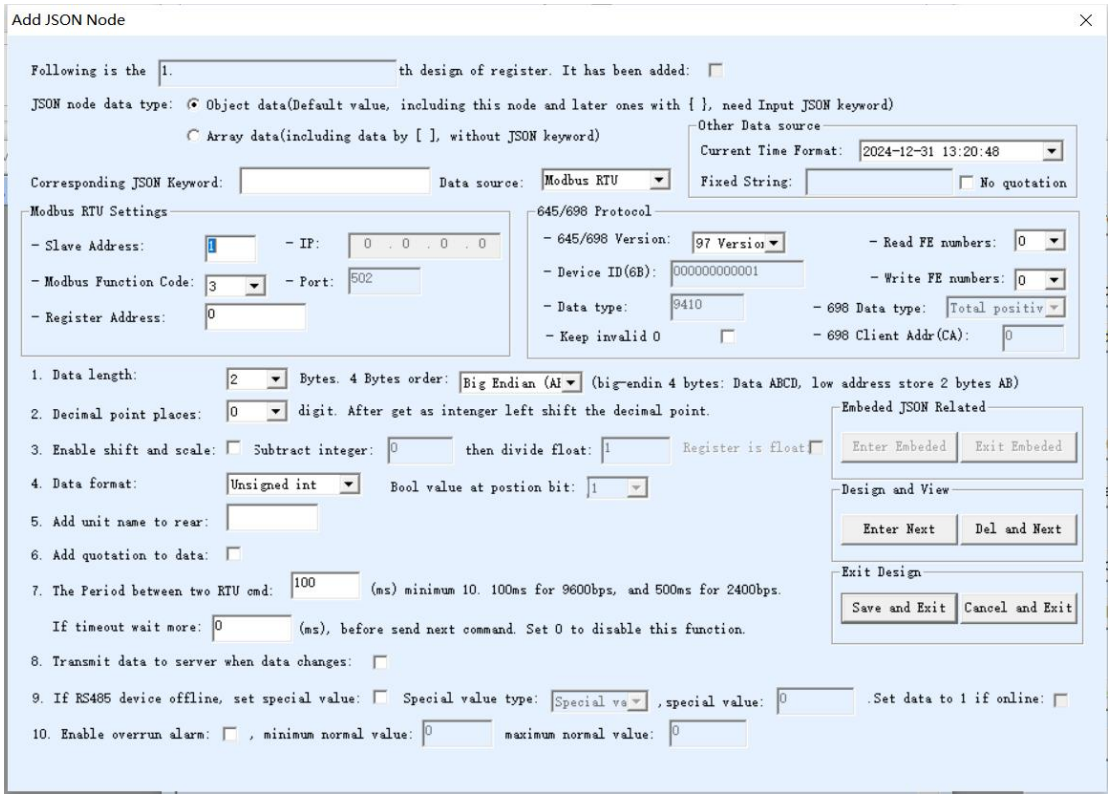


Figure 29 JSON upload

As shown in the above figure, several important parameters are introduced as follows:

1. The following image is the 3.2.4th JSON keyword: Here, "3.2.4" indicates the order of the current design interface's JSON keyword.
2. Already added: If checked, it means that the item has been added. When viewing the configured information, a check mark will appear, indicating that it is in the editing state. If unchecked, it is in the adding state.
3. Corresponding JSON keyword: The name of this JSON node.

4. Data source: Select the source of the JSON data.
   a) Modbus RTU: For example, in the form of addrv:123.45, it indicates that the data comes from a certain Modbus RTU table and is collected through a serial port. The left half of the diagram is all about designing parameters related to Modbus RTU.
   b) Fixed String: For example, in the form of DevName:"MyDev", enter MyDev in the fixed string on the right, and the JSON name is DevName, so that a JSON node for the fixed string can be generated.
   c) Device ID: If the JSON node name is DevID, then the string to be uploaded is DevID:"285301020304", where "285301020304" is the device's MAC address or unique identifier.
   d) Current Time: If the JSON node name is ColletTime, then the string to be uploaded is ColletTime:"2019-05-13 22:23:31". The time is obtained by the system through the NTP protocol.
   e) Nested JSON: If the node name is Alarm, then the format it uploads has the form of Alarm:{temp1:"25.1",temp2:"26.2"}, that is, the content of Alarm is still a JSON collection.
5. Modbus Related settings
   a) From station address: Modbus table address.
   b) Modbus function code: Currently supports function codes 01, 02, 03, and 04.
   c) Register address: Corresponds to 0 here.
   d) Data length: Corresponds to 2 bytes here.
   e) Start translation and scaling: You can perform subtraction and division operations on the acquired value.
   f) Data format: Corresponds to floating point (because translation and scaling are enabled, the 2 bytes must also be set to floating point).
   g) Decimal places to retain: Retain 2 here.
   h) Serial polling time: Set to 200ms here. This refers to the interval between polling this register and the next, not the polling interval for this instruction.
   i) Offline setting of special values: When the serial port does not return data, set the data to a special value such as -9999.Modbus

6. Fixed string: When the source selection is set to a fixed string, you can input the string content.
7. Button
   a) Nested JSON: When the current node source is selected as "Nested JSON" type, you must click this button to enter the design of nested JSON. If the current node is "2.", it will enter the design of node "2.1".
   b) Go back one level: If the current node is at the Nth level of nesting, clicking this button will return to the design of the N-1th level node, and stay on a newly added node at the N-1th level.
   c) Design the next one: After clicking, it will enter the next JSON node locally. If there is no next node designed before, the "already added" check will be cleared, indicating that it is in a new node.
   d) Save design: Complete the design, and when on the last design node interface, click "Save design". After that, return to the main interface, and then click "Save JSON configuration" to save.
   e) Cancel design: Cancel all current designs. If you are viewing design content, you can click this button to exit.

7.2.2. Excel-style editing

JSON parameters can also be configured in a CSV table. Below is an example of a CSV table. When there is already a CSV template, you only need to edit the parameters that need to be changed, such as the Modbus register column.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
1	NO.	JSON KeyWo	Data Sourc	String	Modbus Sla	Modbus Fur	Modbus Reg	645 ID	645 Type	Data Len	4 Bytes Or	Data Forma	Position c	Decimal Pc	Unit	With Quat	Query
2	1		Modbus RTU		1	3	0	ID:0000000	00009410	2	0	Unsigned i	0	0		0	
3	2		Modbus RTU		2	3	1	ID:0000000	00009410	2	0	Unsigned i	0	0		0	
4	3		Modbus RTU		3	3	2	ID:0000000	00009410	2	0	Unsigned i	0	0		0	
5	4		Modbus RTU		3	3	3	ID:0000000	00009410	2	0	Unsigned i	0	0		0	
6																	
7																	

Figure 30 CSV Table

After editing, click on the "Import Configuration" button above, select the edited CSV file, then click "Import". After the import is complete, click "Save JSON Settings". It is generally recommended to modify based on an existing template. Alternatively, configure a set of parameters in the vircom configuration software first, then export them. Make copies and modifications based on the exported CSV table.

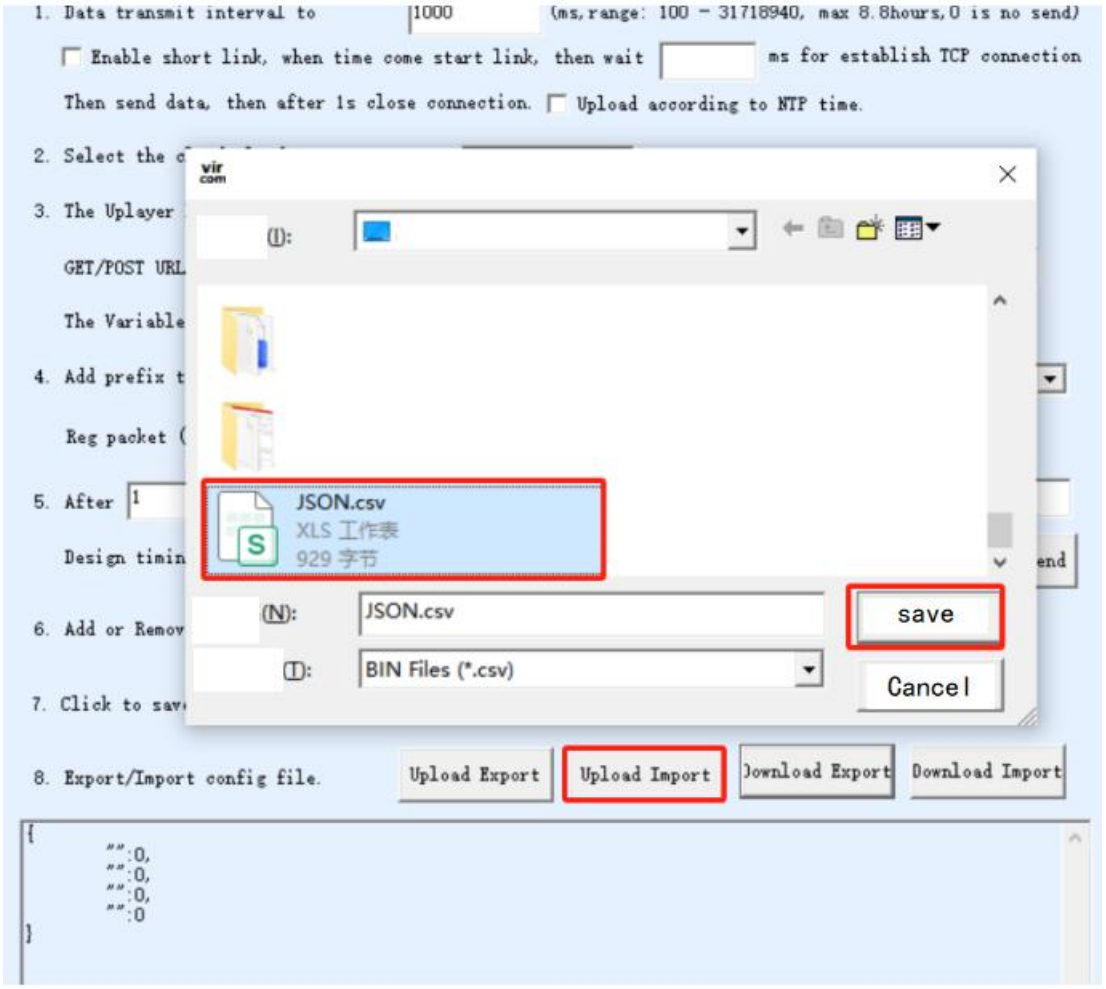


Figure 31 CSV Import

7.3. Server Transparent Transmission Test

Assuming the following network structure as shown in the diagram, 6408 is configured to connect to the server at IP address ***.***.***.*** on port ***. Please configure using the methods described in the "Configuration Method" section. After configuration, upon rebooting, it will take 20 to 40 seconds to connect to the server.

Figure 32 Networking Structure Diagram

We are running the TCP tool SocketDlgTest on our server.（http://www.zlmcu.com/document/tcp_debug_tools.html）。

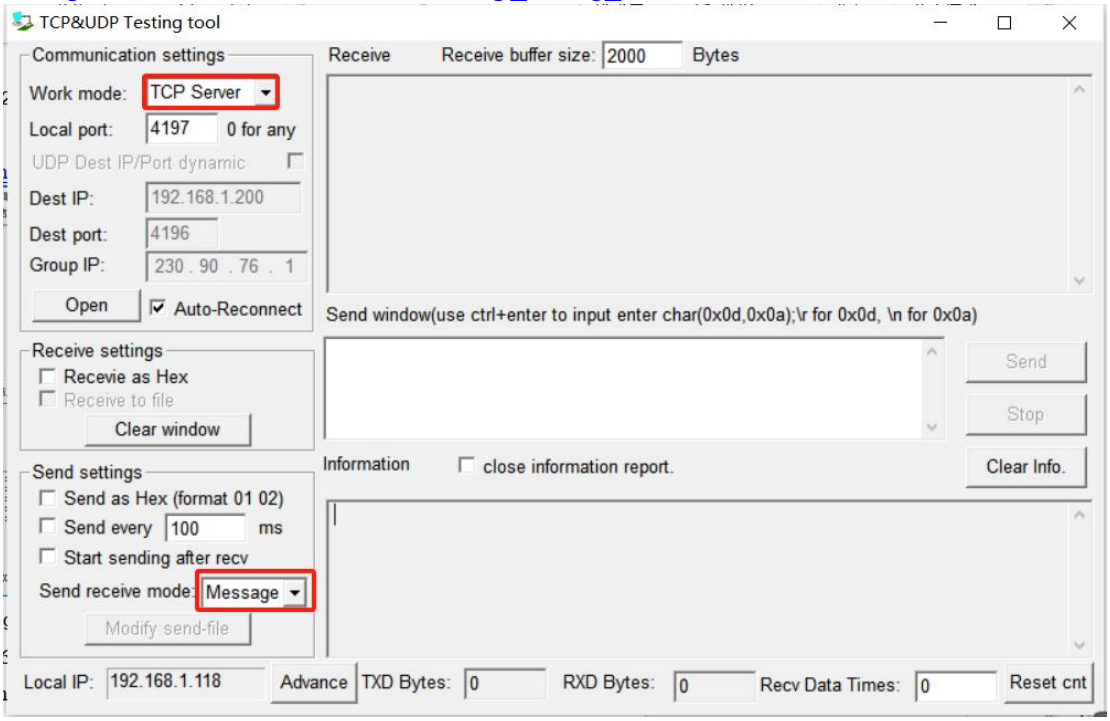


Figure 33 Server-side tools

Select the local port as 4196 (note that if you are running the ZLVircom tool, you will need to switch to a different port), then click the "Open" button. Once the 6408 device connects to the server, it will display the message "The NO... is accepted!".

Now connect the -4G,+4G serial port of the 6408 device to a USB to 485 serial port cable, and open the serial port debugging tool (http://www.zlmcu.com/document/com_debug_tools.html), and open the correct

COM port.

When the serial port sends data, the server will reply with corresponding data, and similarly, the device will receive the message from the server through the serial port and output it. This demonstrates the two-way communication from serial to 4G network, as shown in the figure 34 below:

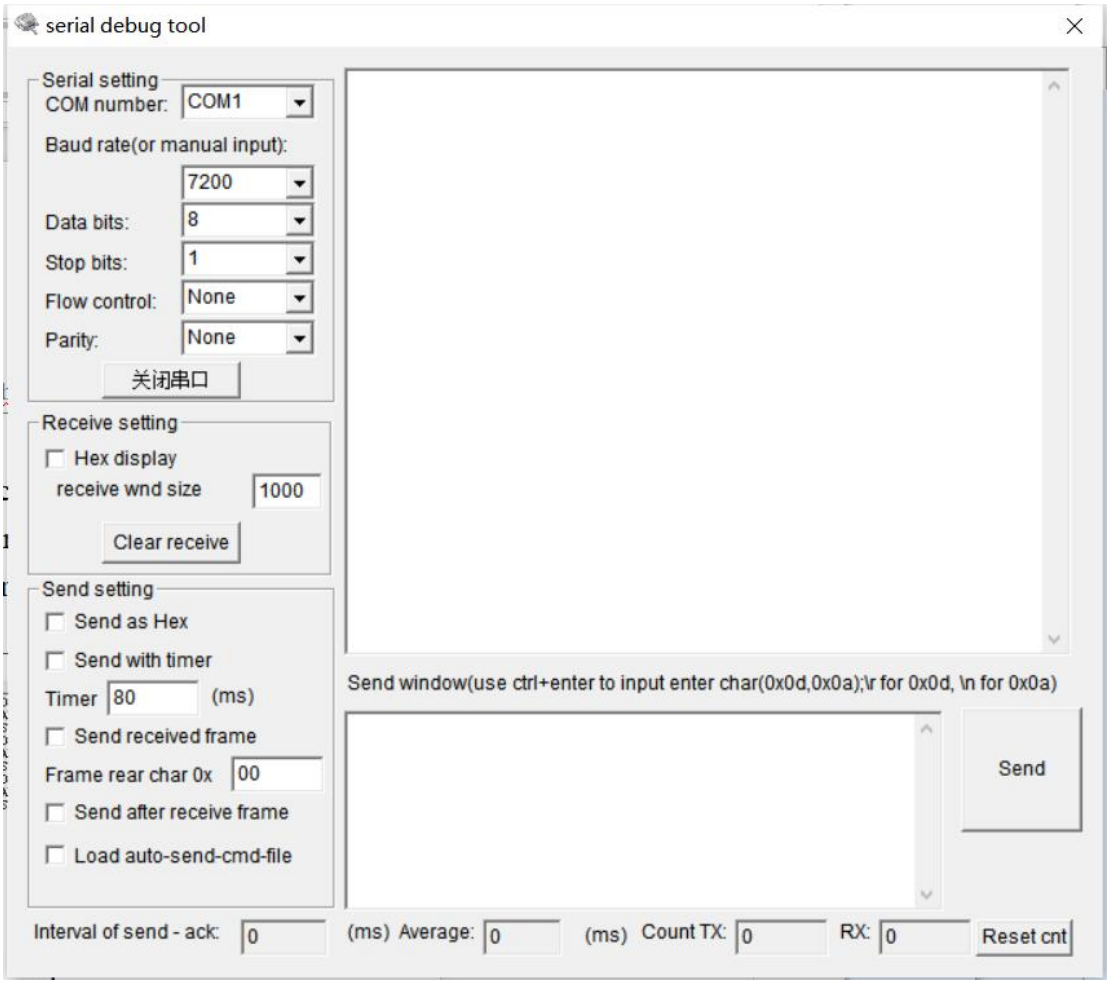


Figure 34 Serial Port Debugging Tool for Devices

7.4. Modbus Protocol conversion test

The configuration parameters are basically the same as the transparent transmission test without protocol, just change the conversion protocol to MODBUS protocol.

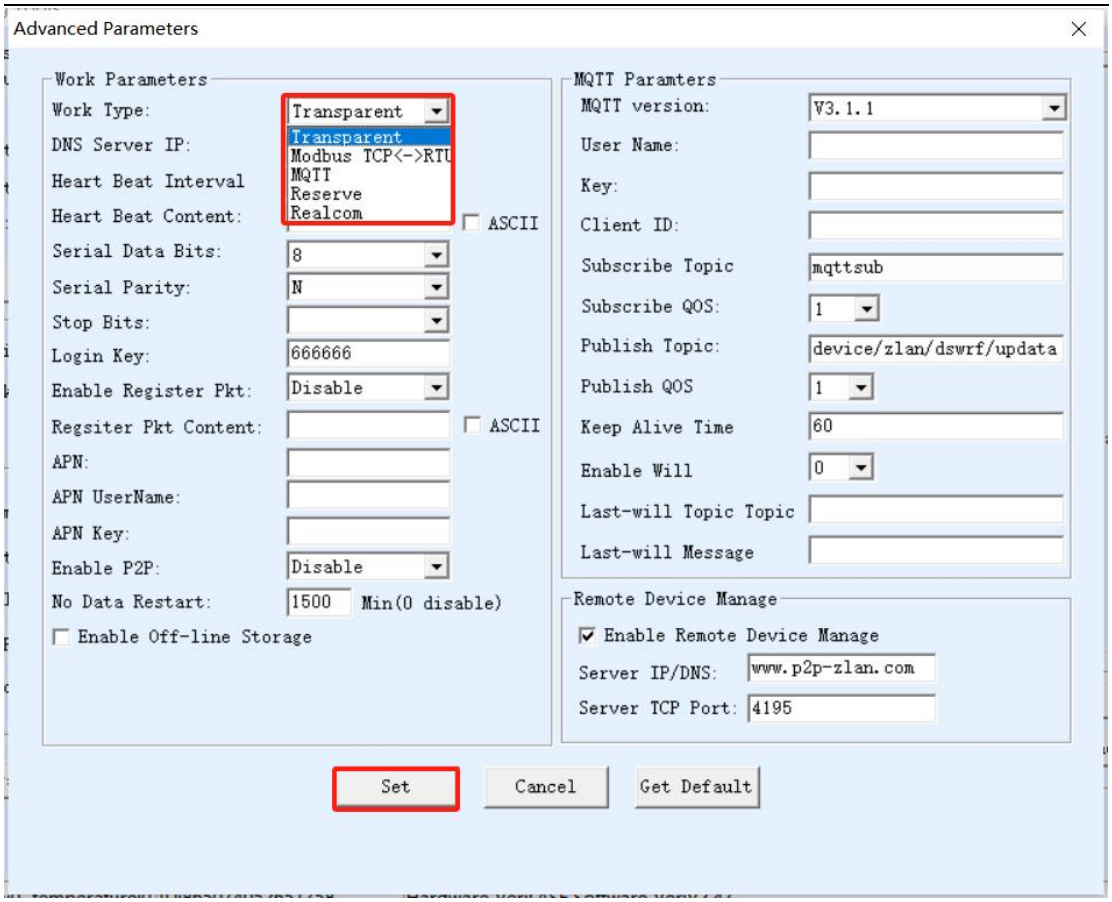


Figure 35 Modbus Protocol Conversion Settings

It can achieve the conversion of serial MODBUS RTU protocol to network MODBUS TCP protocol, and convert network MODBUS TCP protocol to serial MODBUS RTU protocol.

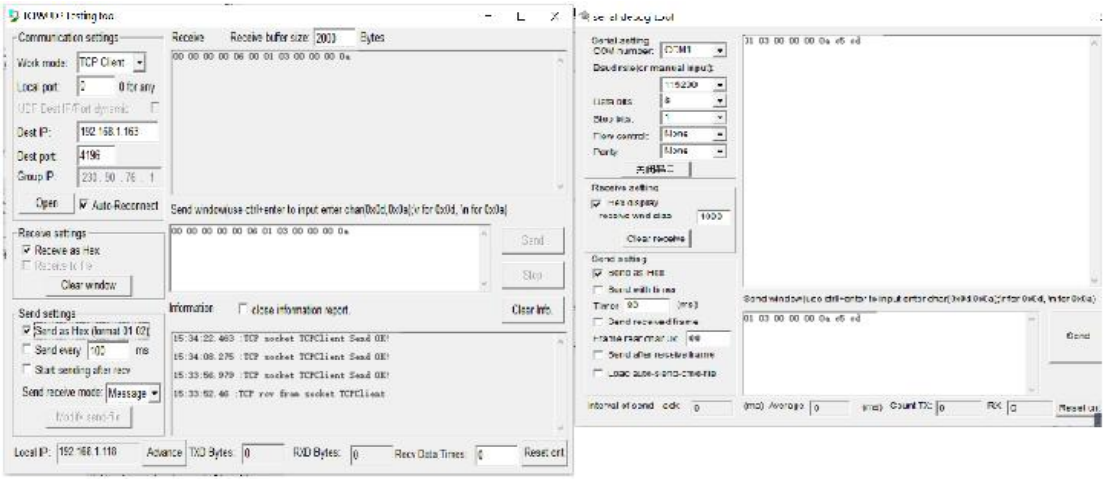


Figure 36 Modbus Protocol Conversion Test

7.5. MQTT Protocol testing

This test is for connecting to Alibaba Cloud. Create a new subscription topic named zlan_test and a publication topic named zlan_1 on Alibaba Cloud, as shown in Figure 37. First, enter the IP and port configuration of the MQTT server, save the parameters as shown in Figure 38. Then, on the advanced parameters page, fill in the MQTT ID, username, password, including the subscription and publication topics, and the keep-alive time, as shown in Figure 39. Make sure to select the working mode as MQTT mode.

Figure 37 Adding a Theme to Alibaba Cloud

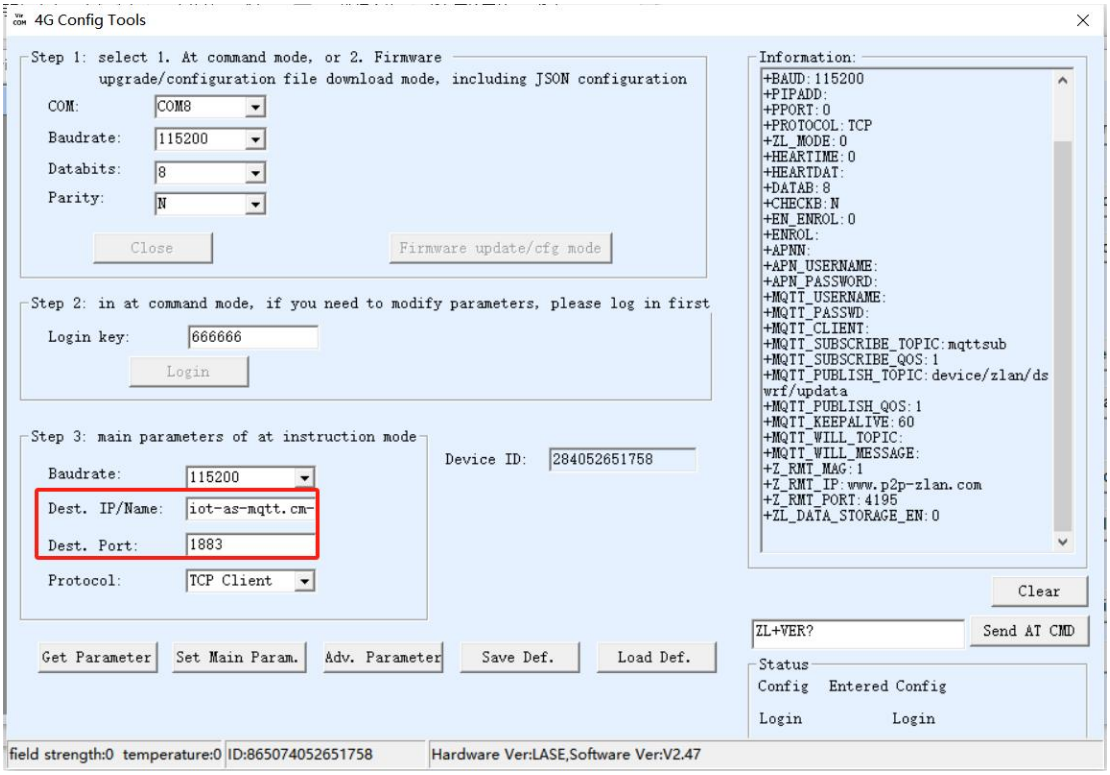


Figure 38 Alibaba Cloud IP and Port

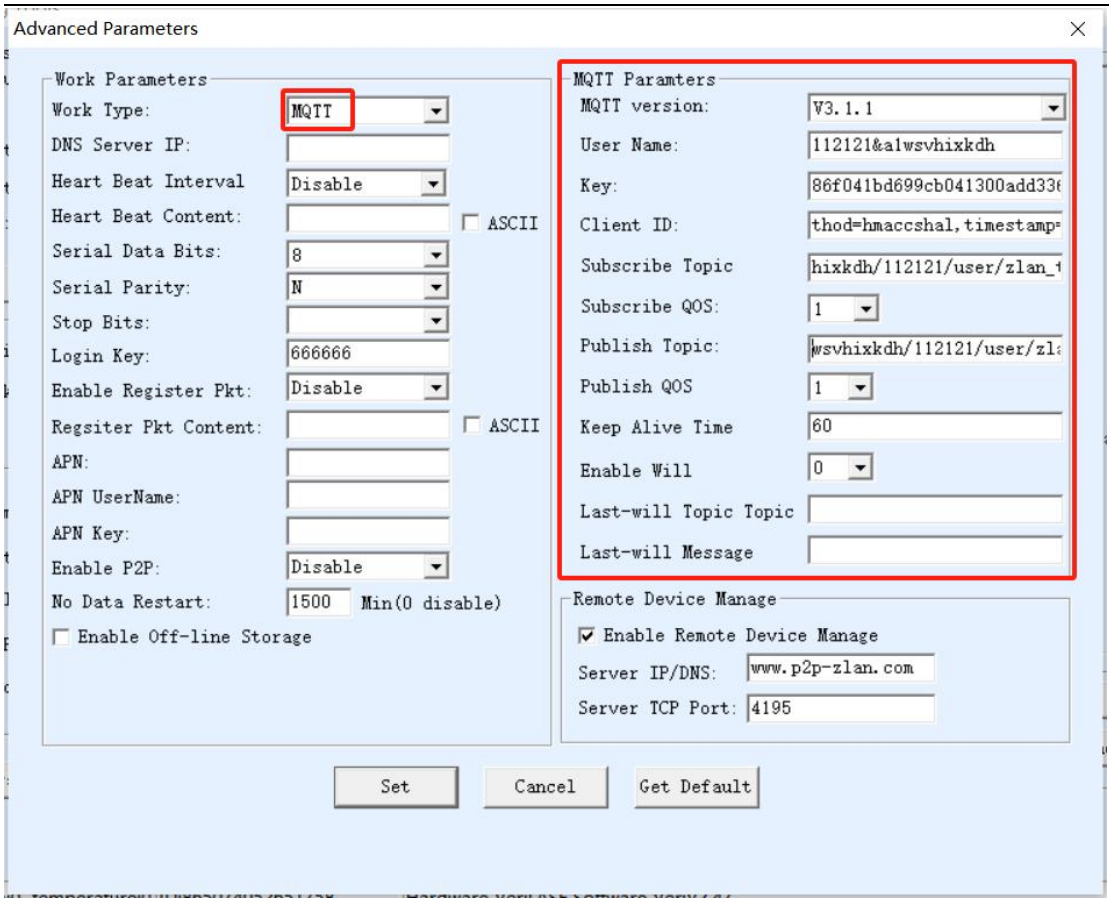


Figure 39: Alibaba Cloud MQTT Configuration

After setting up, open the Alibaba Cloud Device Management interface, go to the Log Service page to check the messages sent by the device, as shown in Figure 40. Send data through the device's serial port, and send messages ("ZLAN8308TEST") through the zlan_1 topic to Alibaba Cloud's MQTT server. Once Alibaba Cloud receives the data, as shown in Figure 41, the Alibaba Cloud server sends messages ("ALI_send") through the zlan_test topic to the device's serial port, as shown in Figure 42. This completes the MQTT send and receive test.

Figure 40 Alibaba Cloud Log Service

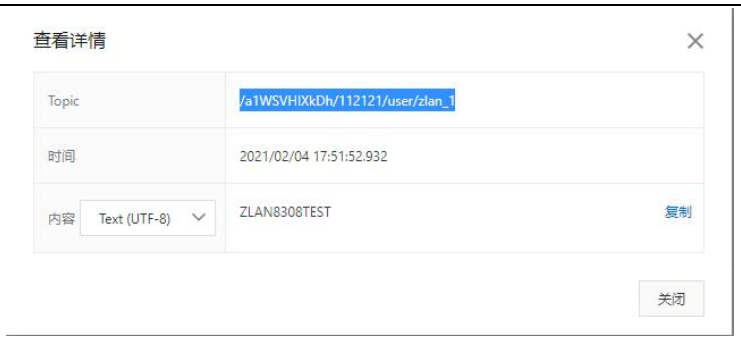


Figure 41: Aliyun receives serial port

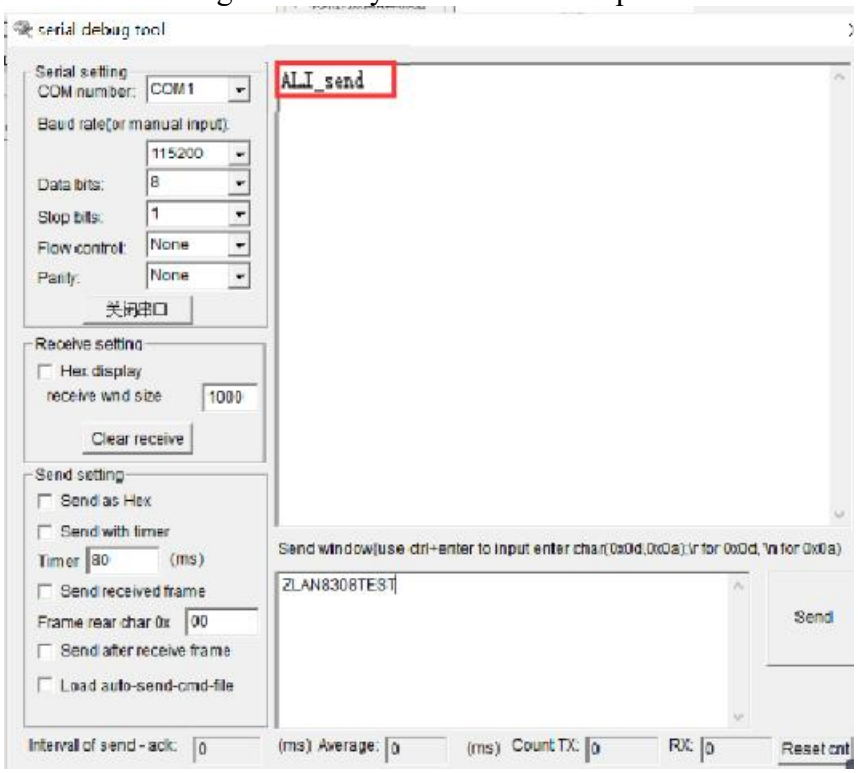


Figure 42 Serial port receives data from Alibaba Cloud

Appendix 1: Parameter Summary

This chapter mainly deals with the technical details of parameter setting and reading. It also helps users configure and modify parameters with their own software. For general applications, this section can be skipped.

The parameters related to reading and setting are listed separately from the register table below.

Table 2. Parameter-related read operations

Function code	Function	Address range	Address range
03	Read the basic parameters.	63~67	63~67
03	Read extended parameters	68~162	68~162
06	Set parameters	63~67	63~67
06	Set the extended parameters	68~162	69~162
16	Set the basic parameters.	63~67	63~67
16	Set the extended parameters	68~162	68~162

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

Table 3. Basic Parameter Registers

Register address	Parameter name	Length (bytes)	Explanation
63(0x3F)	addr/Device address	1	High byte of the register value
63(0x3F)	upLoad/Enable DI active reporting	1	The low byte of the register value, 1 indicates enabled, 2 to 255 indicates periodic transmission.
64(0x40)	dst_addr/DI Report the address.	1	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 for the RS485 interface of the 485-IO.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)	Composite parameter settings	1	The low byte of the register value. Bit1: 32-bit DI count saving, 1 indicates saving Bit2: DI logic inversion, 1 indicates inversion Bit3: DI delay function, after DI becomes 1, it continues to remain 1 for 2 seconds even after the DI input becomes 0, meaning that DI can still be read as 1 within 2 seconds. After selecting this function, the DI adds a 50ms debounce function.
66(0x42)	A1UpLoadH/AIHigh byte of reporting cycle	1	High byte of the register value
66(0x42)	A1UpLoadL/ AIReport cycle low byte	1	Low byte of the register value
67(0x43)	A2UpLoadH/AIHigh byte of reporting cycle	1	Set the high byte of the register value to the same value as A1UpLoadH.

67(0x43)	A2UpLoadL/ AI 上 Report low byte of the period	1	Low byte of the register value, please andA1UpLoadLSet the same value

Table 10. Extended Parameter Registers

Register address	Parameter name	Length (bytes)	Instructions (DO 无效)
68(0x44)	dostate/DO configuration after power-on	1	High byte of the register value, 0xF0 indicates the last 4 paths are engaged.
68(0x44)	checkb/Check digit	1	Low byte of register value. 0: None parity 1: Odd parity 2: Even parity 3: Mark 4: Space
69(0x45)	baud_UART_0_2/Network communication and the baud rate of 485-4G	1	The high byte of the register value, currently read-only, is automatically adapted by the network module and does not require configuration.
69(0x45)	datab/Data bit	1	The low byte of the register value. To be expanded later.
70(0x46)	stopb/Stop bit	1	High byte of the register value, reserved for future expansion.
70(0x46)	TCP_LINK_FLAG/Reserved	1	The low byte of the

			register value. To be expanded later.
71(0x47)	FirmwareType/Device type	1	High byte of the register value. 0：6002/6042 1：6808-1 3：6808-2，6808-3，6808-8，6808-7，6408 4：6802/6842 9：6808-9
71(0x47)	DOMaintain time	1	The low byte of the register value. The duration DO status is maintained.
72(0x48)	DI controls its own DO.	1	The first bit (Bit0) of the high byte of the register value. 1: Enabled 0: Disabled (Invalid)
72(0x48)	Maintain the previous DO state after power-up.	1	The second bit (Bit1) of the high byte of the register value. 1: Enabled; 0: Disabled
72(0x48)	AI proactively reports high-precision numerical values.	1	The third bit (Bit2) of the high byte of the register value. 1: Enabled; 0: Disabled.
72(0x48)	Write DO no return instruction.	1	The fourth bit (Bit3) of the high byte of the

			register value. 1: Enabled; 0: Disabled.
72(0x48)	reserver/Reserved	1	The low byte of the register value. To be expanded later.
73(0x49)	reserver/Reserved	2	To be expanded later.
74~89 (0x4a~0x59)	V1~V8 are the adjustment coefficients for each AI channel.	32	Big-endian format data, refer to the "AI High Precision Usage" chapter for details.
90（0x5a）	AI calibration status	2	1 indicates that the AI is in calibration mode.
91~98 (0x5b~6a)	32-bit counting	32	There are 8 registers in total, 4 DIs, each with 2 registers.
107（0x6b）	Single/Multiple DO Hold	2	Set whether DO1-DO8 single/multiple channels maintain.
108~130 (0x6c~82)	Reserved	46	There are 23 registers.
131~162 (0x83~a2)	DI Combination pair DO logic control	32	There are 16 registers.

Appendix 2: AI Calibration

Steps: For example, calibrating using RS485-IO serial communication method

1. Send 01 06 00 5a 00 01 68 19, set "AI calibration status" to 1, enter calibration mode.
2. Send 01 04 00 00 00 04 f1 c9 to query the data of 4 AI channels. For the received

data 01 04 08 02 81 00 00 00 00 00 00 34 dc, calculate the values for each channel V1~V4. For example, for the first channel.

a) The value is 02 81, converted to decimal is Vin=641. According to the formula in the "AI Usage Instructions" for calculating the input point voltage: Vi = (Vin/1024)*5, where Vin is 641, Vi is the known voltage, for example 3.3V. Thus, the adjustment coefficient V1 = Vi/((Vin/1024)*5) = 3.3 /((641/1024)*5) = 1.0543525.

b) Represent V1 as a float data type and convert it to big-endian HEX format as 0x3F86F506.Write 0x3F86 into the first register 0x4a corresponding to V1, and write 0xF506 into the second register 0x4b corresponding to V1. That is, 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 calibration mode.

Users can calibrate themselves using the "AI Calibration Function" in the "IO Controller" dialog box of the ZLVircom. However, each ZLAN6808 device has been professionally calibrated at the factory and, unless necessary, users do not need to perform calibration. The calibration steps are as follows:：

1. Please select the correct product sub-model in the model: Only by selecting the correct model can you determine the AI type of each channel as 5V, 10V, or 4~20mA. Only then can calibration be performed.

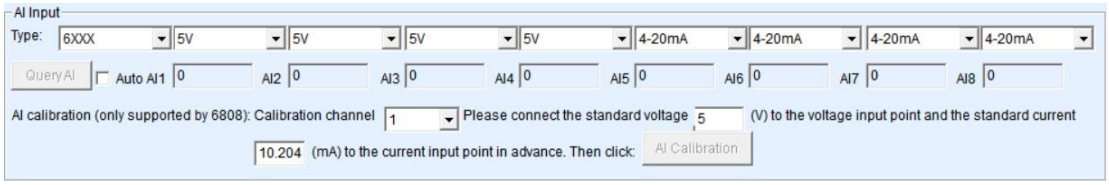


Figure 43 AI Calibration

2. Select which channel to calibrate in the calibration process. Since the user may not be able to connect to all four test points at the same time, it is more convenient to calibrate one channel at a time.
3. Click the "AI Calibration" button, and the system starts the calibration process. After calibration, the AI's values are more accurate. No restart is needed after calibration, as the system automatically saves the calibration parameters.

After-sales service and technical support

Shanghai Zhuolan Information Technology Co., Ltd.

Address: Room 2001, Jin Yuan Center, No. 28 Yuanwen Road, Minhang District,Shanghai

Phone: 021-64325189

Fax: 021-64325200

Website: http://www.zlmcu.com

Email: support@zlmcu.com