The Modbus gateway can be configured

Embedded device networking solutions

ZL DUI 20190714.1.0

Version information

The following changes have been made to the document:

Modification record

date	edition	Document number	Modify content
2019-7-14	Rev.1	ZL DUI 20190714.1.0	Release version

Ownership information

CATALOGUE

1. Overview

Configurable Modbus gateway Addresses, function codes, and register lists of devices at the Modbus RTU can be pre-configured and stored in the gateway. The gateway automatically collects data from these configured RTU devices and maps it to successive Modbus TCP register addresses, allowing the Modbus TCP side to read data from different registers of all devices at once. Its address mapping function is a supplement to the ordinary Modbus RTU to Modbus TCP Modbus gateway.

ZLAN5143, ZLAN7144, ZLAN7146, and ZLAN8303 all support configurable Modbus gateway function. ZLAN5143 requires firmware version 1.579 and above, and ZLVircom5.15 and above to use. 7144 requires firmware version 1.491 or later. Zlan Configurable Modbus gateway is referred to as ZLMB Gateway.

The configurable Modbus gateway has the following advantages:

1. Instruction merging: different station addresses and registers can be mapped to the same piece of register. In this way, the Modbus TCP side can use one instruction to collect all data, rather than separate several instructions to collect, which realizes the transmission efficiency of the Modbus TCP side.
2. Pre-collection: Although the previous "storage Modbus" can automatically collect some data, the automatic collection starts with the query of the first instruction on the Modbus TCP side, so the returned data of the first instruction itself still needs to wait for the return of information from the serial port. The configurable Modbus gateway automatic acquisition starts when the gateway is powered on, and the first command can still respond quickly.
3. Multi-host support: Gateway data can be read from multiple Modbus TCP clients.
4. Support write instructions: For the remapped register address, the Modbus TCP side can not only read data using the remapped address, but also write data to the remapped address using 05 and 06 instructions.
5. Support arbitrary length reading: ZLVircom configuration tool will display the mapping relationship between the register address of RTU and the register address of TCP. The Modbus TCP terminal can not only read the whole chip register and

single read register, but also can read the register content of legal address and length at any length.

6. Offline notification is supported: After an RTU device is offline from the RS485 bus, all the data queried from Modbus TCP turns to 0, indicating that the device is offline.
7. Support for loss retransmission: Because the RS485 bus is prone to interference, if the first query command does not answer, the gateway will send another command, and wait for the timeout period, if the second time is not answered before the device is considered offline. ZLVircom can set the time of the first transmission and the time of the second timeout response.
8. Support a large number of devices and registers: Currently supports 100 RTU query instructions and 500 registers, while being accessed.

2. Simple examples

2.1. RTU collection requirements

Here, ZLAN5143 is taken as an example. First, the firmware version is 1.579 and above, and ZLVircom is 5.15 and above.

Assume that two Modbus RTU devices are connected to the RS485 port of the ZLAN5143. The following information needs to be collected:

Table 1. RTU data to be collected

Record number	Slave address	Function code	Register address	Length
1	1	3	2	4
2	2	1	5	9
3	3	3	2	4

Now ZLVircom needs to be configured so that Modbus gateway can be configured.

2.2. ZLVircom Disposition

Run ZLVircom and enter equipment management→Click to search deviceSelect

a device that you want to configure→Click to edit device→Click the "Firmware and Configuration" button below, and the following dialog box will pop up:

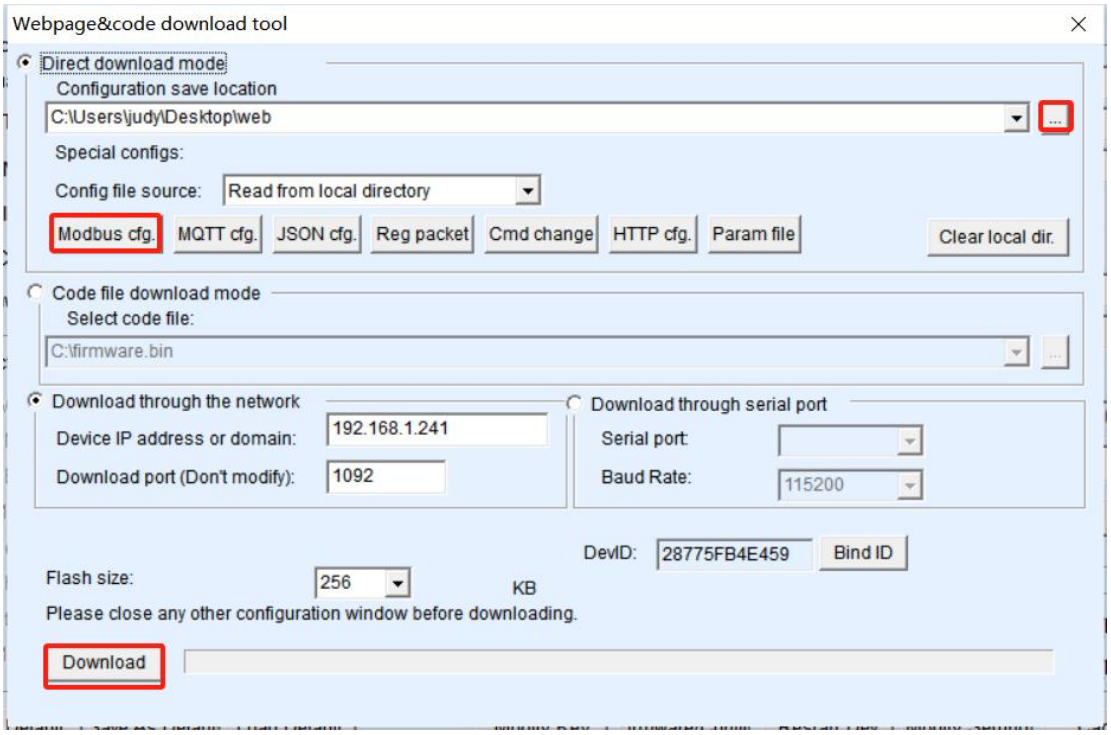


Figure 1 enters the ZLMB configuration method

Here select "Web Directory Download" and click "..." , select an empty directory created earlier to store the configuration information. Then click the "ZLMB Gateway" button, and the following dialog box pops up:

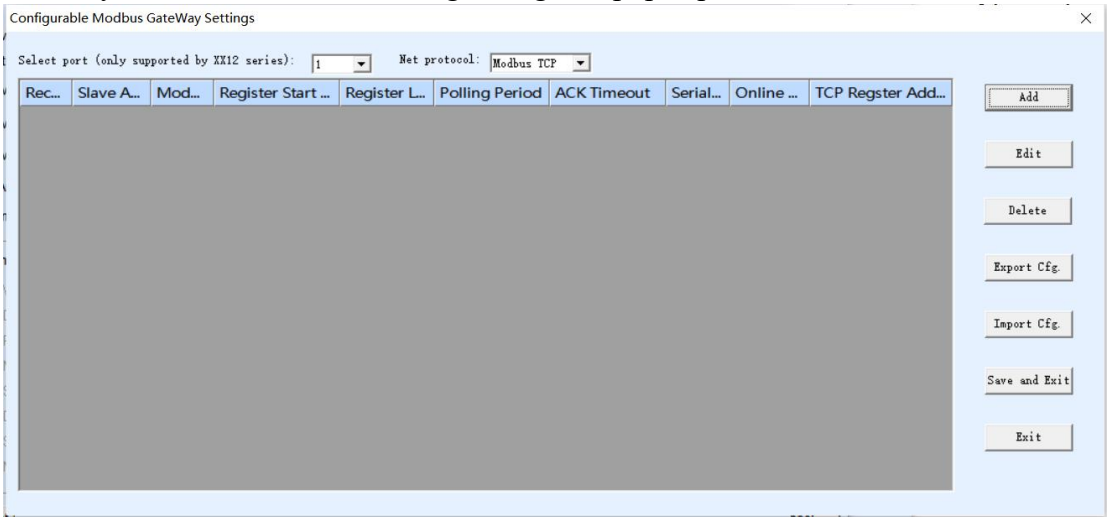


Figure 2 List of configurable Modbus gateway Settings

Click the "Add" button to add the RTU instructions in Table 1 one by one. Note

that the gateway automatically increments the register addresses on this side of Modbus TCP from 0. Therefore, in order to make the instructions of the same function code can be assigned to the same area at one time, the instructions of the same function code should be added consecutively. That is, record numbers 1 and 3 of Table 1 are placed once, and number 2 is placed last. The 3 graphs added are as follows:

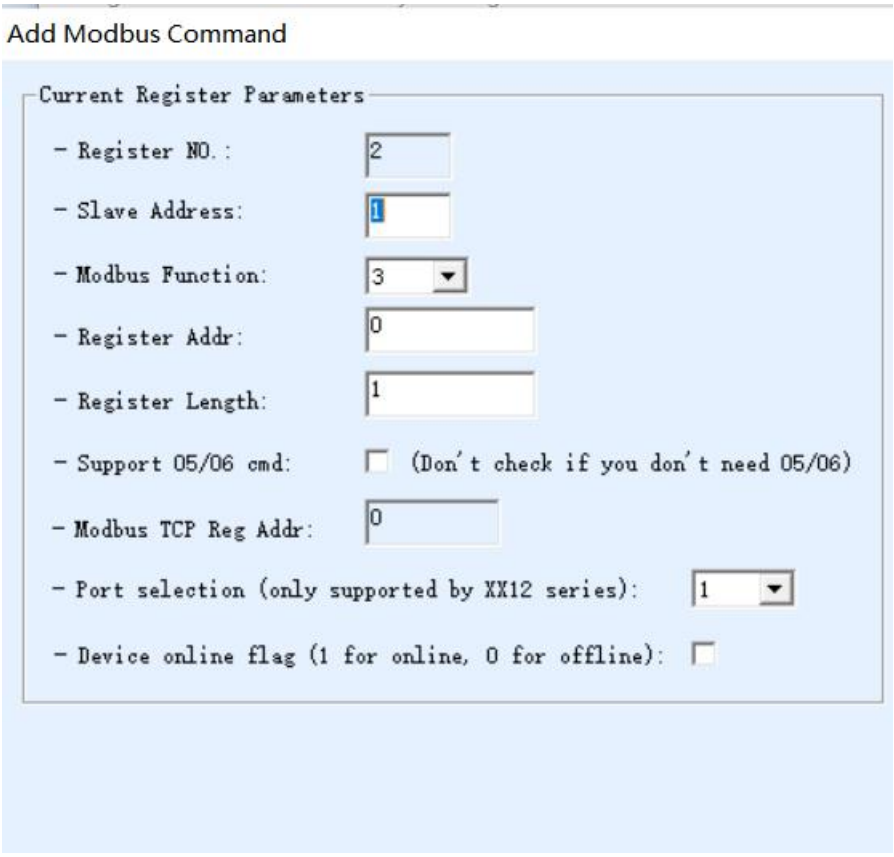


Figure 3 Table 1 Number 1 RTU instruction configuration

The configuration dialog for RTU directive number 1 is explained as follows:

1. Record number: This is the system automatically adds the number according to the order of addition, and cannot be modified. Not necessarily consistent with Table 1.
2. Secondary station address: indicates the secondary station address of the RTU end of the RTU command.
3. Modbus function code: Function code of the RTU command.
4. Register start address: Register start address of RTU instruction.

5. Register length: register query length of RTU instruction.
6. Serial port polling interval: This interval is the time for the gateway to wait for the response to return after sending the query command normally. It's also the speed of polling. This parameter is the same for all RTU instructions, if not the same take the biggest.
7. Timeout response time: This time is the time to wait for the response return when the second query is immediately sent when the normal query instruction is not received. This parameter is the same for all RTU instructions, if not the same take the biggest.
8. Support 06 or 05 write command: Check this option to support the rewrite of RTU device data on the Modbus TCP end. If you do not need to rewrite, do not select it, because selecting it will increase the memory and compute overhead.
9. Modbus TCP register address: This address is automatically calculated by the system in order, and the Modbus TCP end can get the data obtained by this RTU instruction as long as it reads from this starting address. This parameter is the key information for address mapping.

After confirmation, click "Add" again to add the instructions in Table 1 No. 3:

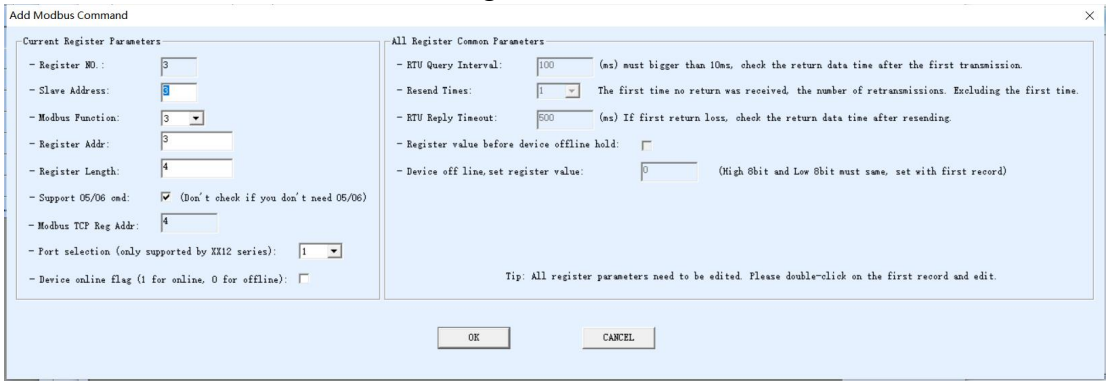


Figure 4 Configuration of the No. 3 RTU instruction in Table 1

Here you see that the Modbus TCP register address is automatically increased to 4. The registers in which the two instructions are combined range from 0 to 7. The whole register range can be read at one time.

Add instructions in Table 1, number 2:

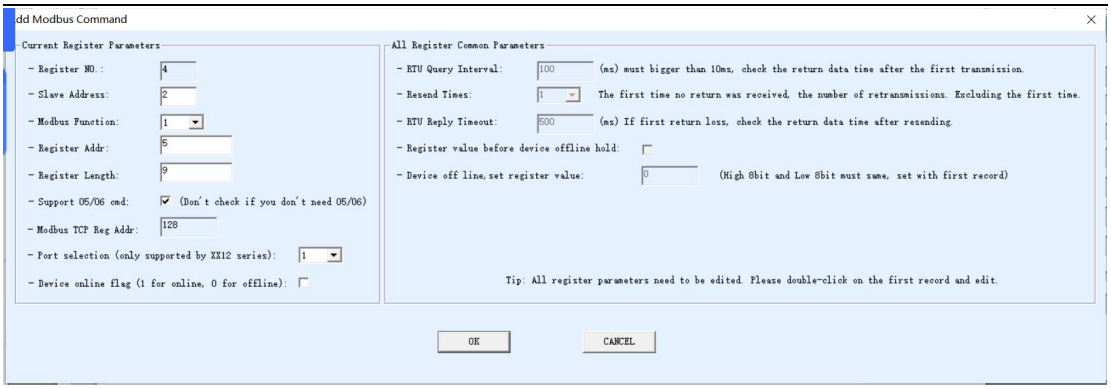
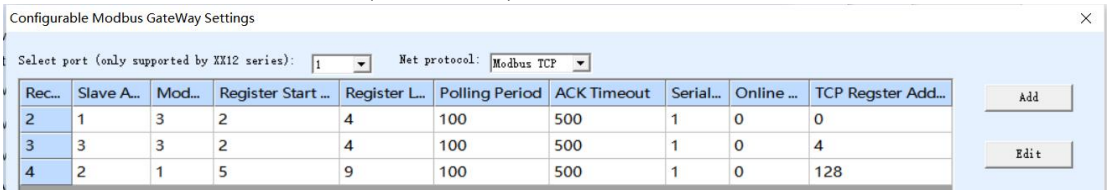


Figure 5 Configuration of the No. 2 RTU instruction in Table 1

The Modbus TCP address automatically assigned by the system is the 128 address of the 01 command (in decimal). The tables listed at the end are:



Rec...	Slave A...	Mod...	Register Start ...	Register L...	Polling Period	ACK Timeout	Serial...	Online ...	TCP Regster Add...
2	1	3	2	4	100	500	1	0	0
3	3	3	2	4	100	500	1	0	4
4	2	1	5	9	100	500	1	0	128

Figure 6 Mapping between RTU and Modbus TCP registers

This table lists RTU and Modbus TCP register mappings. It also clearly shows that if the user needs to obtain the data of a device at the RTU end, what function code and what register address are needed to read the data.

For example, data that needs to read the slave station address 2 function code 1, register address 5, length 9 can be read using any slave station address, function code 1, register address 128, length 9 of the Modbus TCP instruction.

Finally click "Save and return" to save the design, if you click "do not save return" to abandon the design. At the same time, you can select a line and click "Edit" to edit, and click "Delete" to delete an RTU command.

Click "Save and return" to return to the "ZLAN Web page/program download tool" interface. Click the "Download button" in this interface. After the download is complete, the "Transfer complete..." is displayed. Button, then click OK, and the device will automatically restart and load the configuration.

2.3. Data testing

Now use the serial port to simulate the RTU device and conduct tests. The simulation parameters of the RTU end are shown in the figure below.

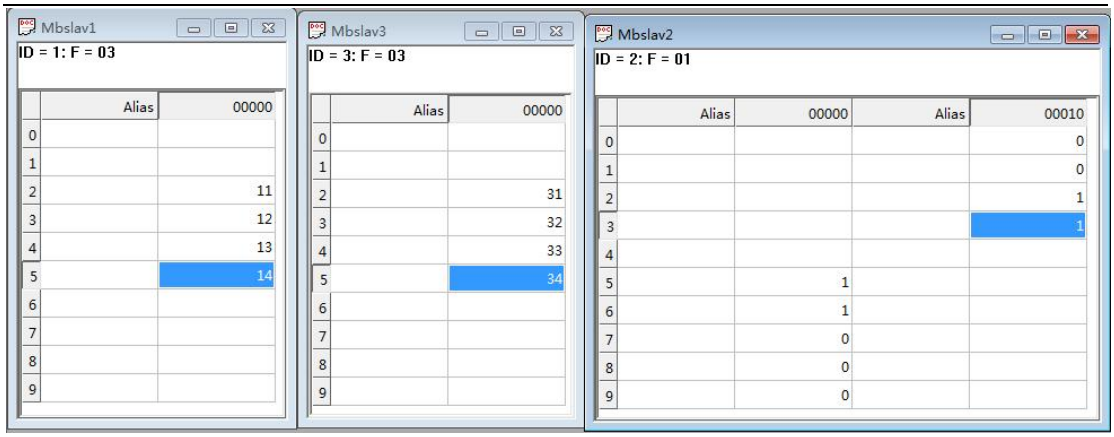


Figure 7 simulates the three RTU devices in Table 1

Then connect the ZLAN5143 to this computer serial port. If the IP address of the device is 192.168.0.200 and port 4196, the Modbus TCP software reads the following data after connecting the IP address and port. According to the mapping table, 8 registers should be read from 0 to 7, and the function code is 03; The 9 bit sex registers 128 to 136 should be read with the function code 01.

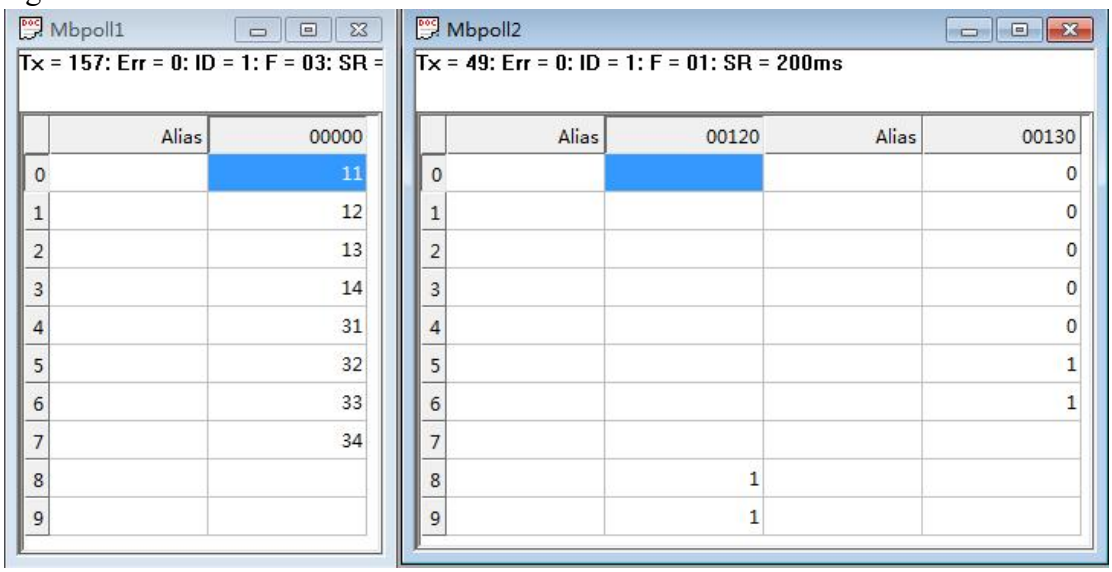


Figure 8 Modbus reads gateway data

As shown in the figure, the configurable Modbus gateway realizes the data merging of station address 1 and station address 3, and combines them into the whole area of registers 0 ~ 7 to realize the function of instruction merging.

Use SockeDlgTest to connect to port 4196 at 192.168.0.200 and send 12 34 00 00 00 06 01 03 00 00 00 00 00 command to return data immediately.

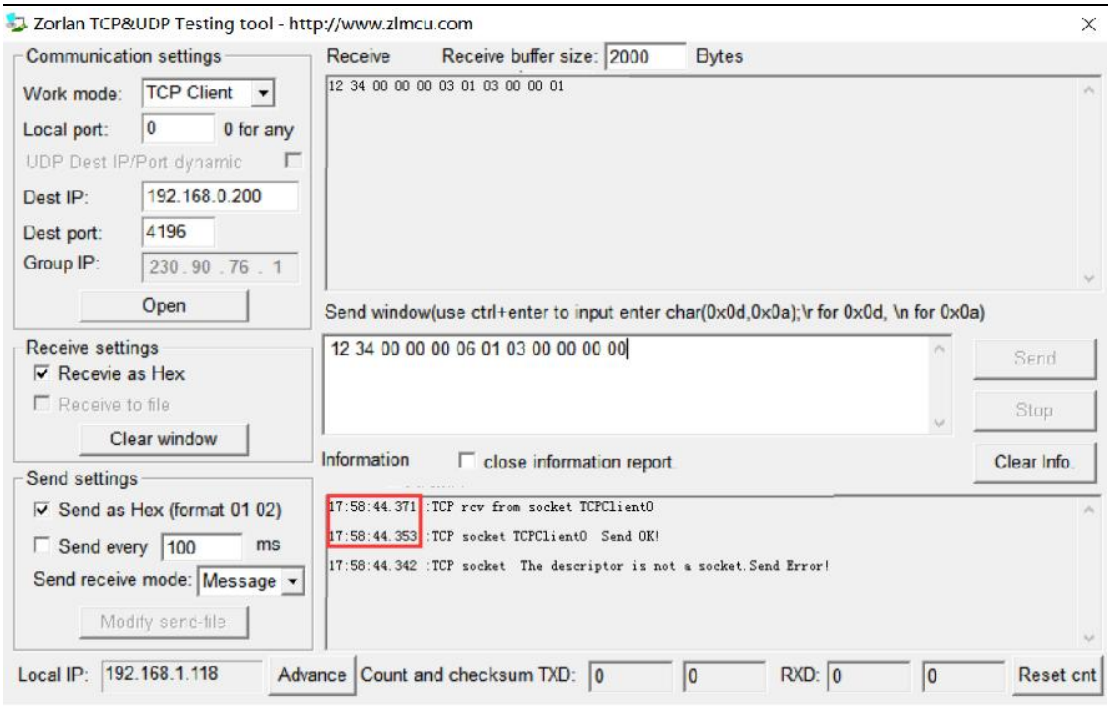


Figure 9 Pre-collection effect

As shown in the figure, the return time is about 18ms. This response time is much faster than the first response time for "storage" Modbus.

Multiple Modbus TCP clients can be opened to collect data on the gateway at the same time. Note The multi-host function is supported.

When the value of a register is modified on the Modbus TCP side, the data on the RTU side is modified, and the data is also fed back to the Modbus TCP side. This shows that sending 05/06 single coil modification and single register modification instruction write is supported.

The Modbus TCP software can also read registers of any length, for example, registers of 06 lengths starting from 01. Data of any length can be read.

Close the Modbus RTU software or disable the serial port. All the data in the Modbus TCP software becomes 0, and it can be seen that the offline or online status of an instrument can be reflected by all the data becomes 0.

Close Modbus RTU software and use ZLComDebug to open COM port to receive data. It can be seen that when an instruction is not answered, it will be sent again. Note The gateway has the data resending function.

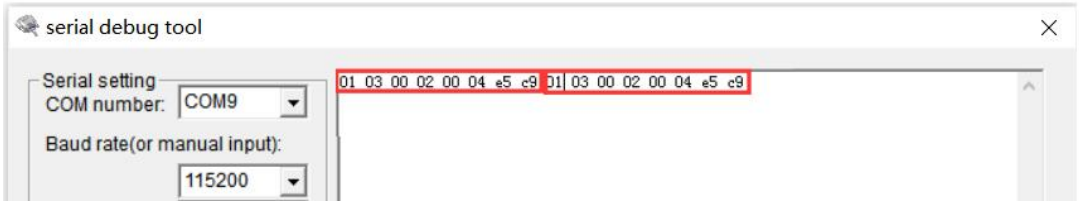


Figure 10 Data retransmission function

2.4. Set the offline default value

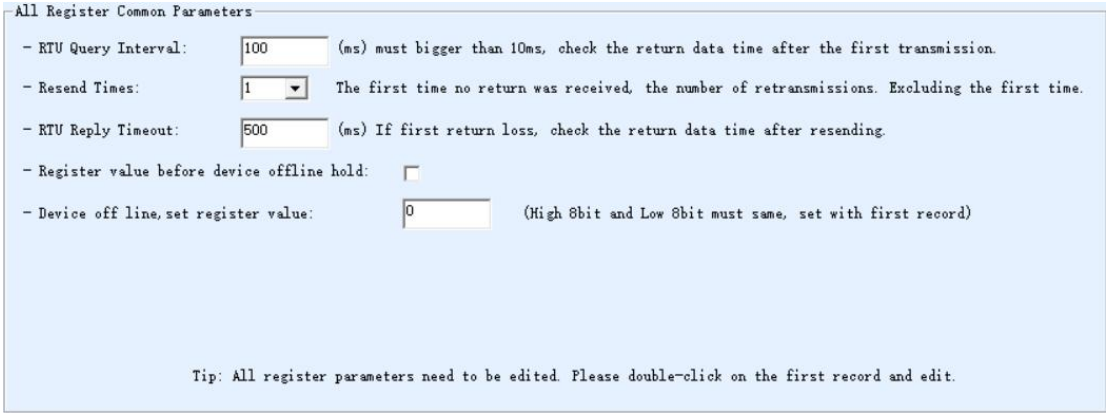


Figure 11 Setting offline defaults

Setting the offline default allows the device to collect a special value when it is offline, such as 65535 (that is, signed -1). Because the previous default value was 0, for some devices 0 is a possible value and it is impossible to distinguish between offline and valid data.

This offline only requires high and low bytes the same, that is, you can set: v*256+v, where v<=255 and v>=0.

Only the setting value of the first record is valid; the other records mimic the parameters of the first record. Here only look at the first record of the parameters also include: serial port polling interval, timeout response time, support for 06 or 05 write command.

2.5. Set the value to keep offline

Sometimes when the device is offline, it does not want the data to become 0 or become a special value, but to keep the last value collected. ZLVircom6.68 and above can set the function of "register value before the device is kept offline".

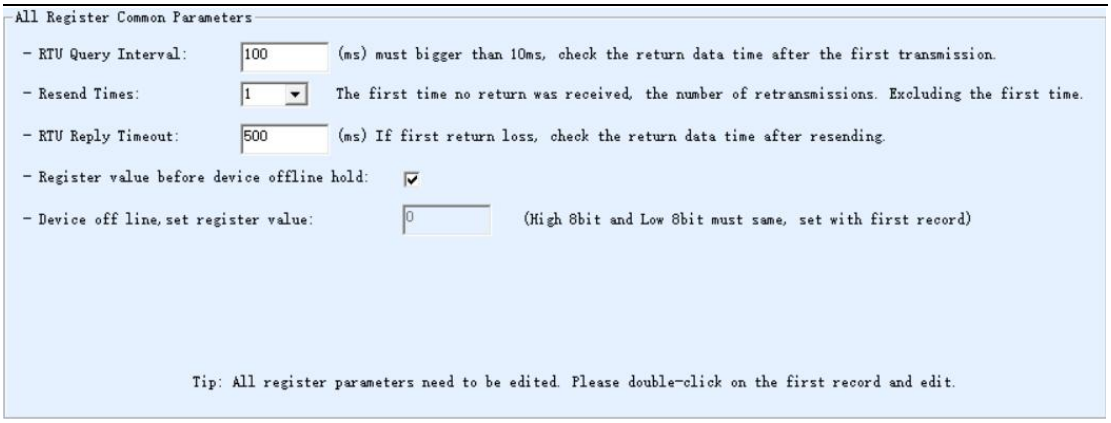


Figure 12 Keeping the original value offline

Note that this feature is global, not specific to a register setting.

2.6. Set the number of resends

Before ZLMB default to provide a retransmission opportunity, now using ZLVircom6.68 and above version can be set to provide multiple retransmission function:

1. Send the Modbus RTU command for the first time and wait for Serial Port Polling Interval.
2. Receive the data. If the data is received, proceed to the next item. If no data is received, the Modbus RTU command is resend.
3. Wait for Reply Timeout to receive the data. If the data is received, proceed to the next message. Otherwise, the retransmission count is reduced by one, and if the retransmission count becomes 0, the next instruction continues. Otherwise skip to step 2.

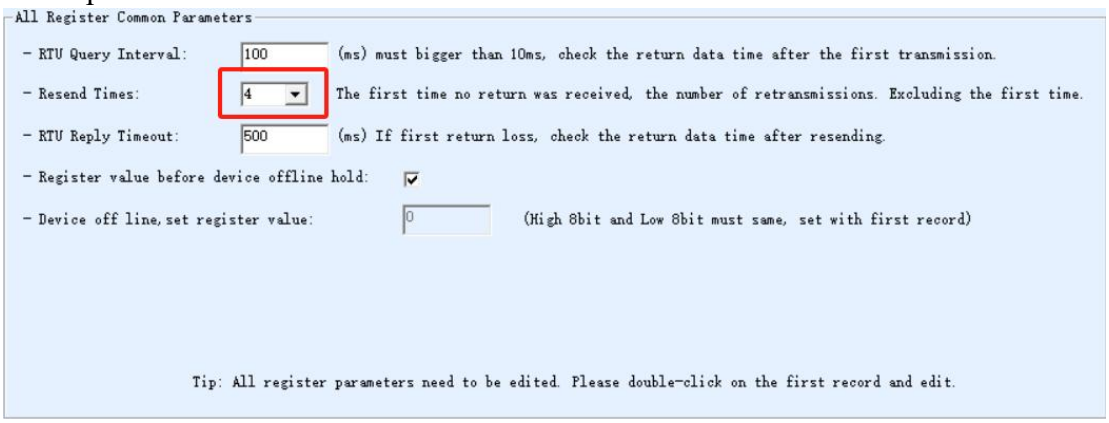


Figure 12 Setting the number of resends

For RS485 communication lines that are prone to packet loss, multiple retransmissions can ensure that valid data is found each time. You can set the number of resending times from 0 to 9. 0 indicates that no resends are sent. Note that this feature is global, not specific to a register setting.

3. Other precautions

1. Some bit data cannot be merged. As shown below, the data length of the first instruction of the 01/02 instruction is 9, not a multiple of 16. So you can't read all the data together by reading 19 addresses starting with 128. Instead, you should read 9 of the 128 addresses and 10 of the 144 addresses given by the software, respectively.

Rec...	Slave A...	Mod...	Register Start ...	Register L...	Polling Period	ACK Timeout	Serial...	Online ...	TCP Register Add...
1	1	3	2	4	100	500	0		
2	3	3	2	4	100	500	4		
3	2	1	5	9	100	500	128		
4	4	2	3	10	100	500	144		

Figure 12 Continuity of Bit data

2. The Modbus TCP station address does not matter, generally 01 can be used.
3. If it is found that the 05/06 instruction cannot be supported, please confirm whether the 05/06 support option has been checked during the design.
4. Note that the register of 02 function cannot be modified.
5. In the Conversion Protocol section of the Modbus gateway, do not select Modbus TCP to RTU. Select None.
6. 15/16 instructions for multi-register writing can now be supported. But do not write across regions. For example, the original address of 2 registers is arranged together, if you try to write more than 2 registers at a time across the region, there will be an error.

4. After-sales service and technical support

Shanghai ZLAN Information Technology Co., LTD

Add: 2001, Jinyuan Center, No.28 Yuanwen Road, Minhang District, Shanghai

Tel:021-64325189

Fax:021-64325200

Website：http://www.zlmcu.com

Email：support@zlmcu.com