PUSR®
www.pusr.com

CANopen Slave Gateway

USR-CAN518

User Manual

V2.0

Your Trustworthy Smart Industrial IoT Partner

Catalog

1. Product Overview

1.1. Product Introduction

The USR-CAN518 is a CANopen slave gateway independently developed by USR-IOT. This product features 1 CAN interface, 1 RS485 interface, and 1 standard 10/100M Ethernet interface. The device supports CANopen to Modbus conversion. It can easily integrate Modbus slave devices from industrial sites into PLCs and other CANopen master devices.

This series of CAN gateways provides a PC configuration tool for flexible function configuration. The product features high speed, low latency, stable performance, ease of use, and high cost-effectiveness. It adopts industrial design standards with comprehensive isolation, achieving up to 3000V isolation on the CAN port. It operates within an industrial wide temperature range of -40℃~85℃ and is powered by a wide voltage DC 9-36V supply, offering robust hardware protection. It supports a wide baud rate range, with an arbitration field baud rate of 5K~1Mbps and a data field baud rate of 100K~5Mbps. After undergoing multiple rigorous environmental tests, it can adapt to various industry scenarios and remains stable and reliable even in harsh environments. The product comes with a DIN rail for quick and convenient installation, suitable for various industrial applications.

Table 1 USR-CAN Series Specification Selection Table

USR-CAN316	CAN(FD) to RS485/RS232, supports CAN(FD) to Modbus RTU, non-isolated
USR-CAN315	CAN(FD) to Ethernet, supports CAN(FD) to Modbus TCP, non-isolated
USR-CAN518	CANopen slave to Modbus RTU/TCP, isolated version
USR-CAN518	Dual CAN ports + 1 Ethernet port + 1 fiber optic port + 1 RS485 port, isolated version CAN(FD) to RS485/Ethernet/Fiber Optic; CAN repeater; Supports CAN(FD) to Modbus RTU/TCP Supports CANopen master to Modbus RTU/TCP*

(Note: Items marked with * are under development)

1.1. Features

- Supports CANopen slave to Modbus RTU/TCP master conversion
- Supports standardCAN2.0 protocol
- Supports 128 groups ofTPDO and 128 groups of RPDO
- Each PDO group can map up to 16 variables
- Supports 5k~1M bps CAN baud rate, customizable CAN baud rate
- SupportsPC parameter configuration
- Supportsnetwork AT command configuration
- Supports PCfirmware upgrade for easier updates
- Reliable communication isolation; power, CAN port, Ethernet port, and serial port are all isolated
- High and low temperature resistance, stable operation from -40℃ to 85℃
- Built-in 120-ohm terminal resistor, accessible via DIP switch

- Supports9-36V wide voltage input with reverse polarity protection
- Reliable hardware protection: ESD, surge, and EFTlevel 3 protection
- Hardware watchdog function, automatic restart after crash, ensuring module stability and reliability

1.2. Technical Parameters

Table 2 Basic Product Parameters

Category	Parameter	Value
Basic Parameters	Operating Voltage	DC 9 ~ 36 V,12V 1A recommended
	Dimensions	110*27*76.1mm
	Mounting Method	DIN Rail Mounting
	Reload Button	Long press to restore factory settings
	Indicator Lights	POWER, WORK, CAN, RS485
Interface Parameters	CAN Port	1 CAN port
	CAN Port Baud Rate	5K~1Mbps, support customization
	Termination Resistor	Built-in two 120Ω CAN bus termination resistors Controlled by DIP switches; switching any one to ON connects a 120Ω resistor in parallel. Switching both to ON connects two 120Ω resistors in parallel.
	Ethernet Port	RJ45, 10/100Mbps, auto MDI/MDIX
Operating Environment	Operating Temperature	-40~85℃
	Storage Temperature	-40~105℃
	Operating Humidity	5% ~ 95% (non-condensing)
	Storage Humidity	5% ~ 95% (non-condensing)
Software Functions	CANopen Features	Device supports CANopen slave
		CANopen slave supports 128 sets of TPDO and 128 sets of RPDO
	Modbus Features	CANopen slave to Modbus TCP/RTU master
		When operating as Modbus RTU master, supports up to 255 Modbus RTU slaves
		When operating as Modbus TCP master (TCP client), supports connection to a single TCP slave

	ESI File	Standard ESI file provided, available for download from the official website
	Configuration Method	PC configuration software provided, easy to use. Can also be configured via AT commands
		Supports one-click parameter reading, CSV file import/export
		Supports firmware upgrade via PC software
Protection Parameters	ESD Protection	Air discharge8kV,contact discharge 6kV
	EFT/Burst Immunity	Power circuit 2kV;Ethernet, CAN circuit1kV
	Surge Immunity Test	Power circuit differential mode 1kV,common mode 2kV;CAN circuit common mode 2kV; Ethernet circuit 1kV
	Communication Isolation	Full isolation for power/serial/Ethernet ports, anti-EMI

2. Hardware Parameters

2.1. Dimension Description

Overall dimensions (including terminals and DIN rail kit): 110*27*76.1mm

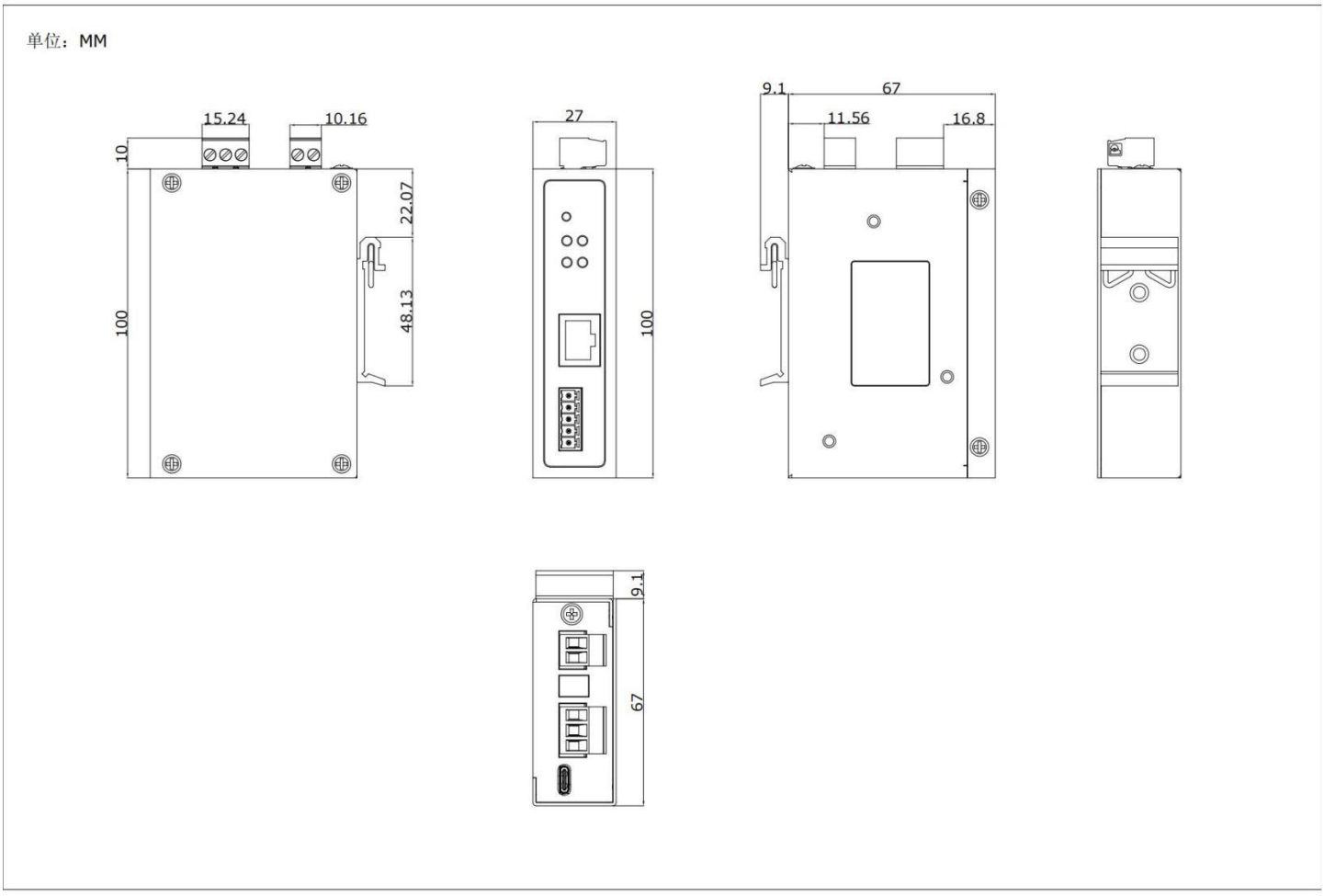


Figure1. USR-CAN518 Standard Product Dimension Drawing

2.2. InterfaceDescription

USR-The CAN518 interface description is as follows.

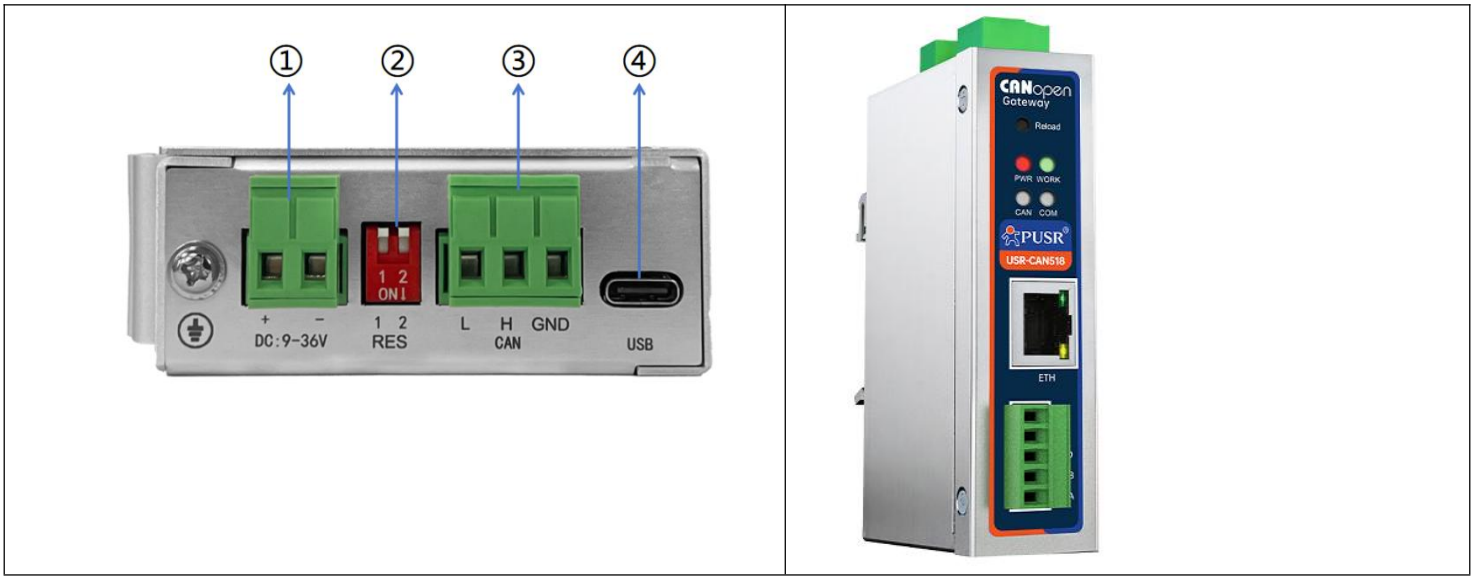


Figure1. Interface Description

Table 3 Terminal WiringDefinition

No.	Interface Name	FunctionDescription
1	DC 9-36V +	Power supply interface,DC 9-36V power positive pole
	DC 9-36V -	Power supply interface,DC 9-36V power negative pole
2	RES 1	Terminal resistor 1,120Ω. Switch down to ON to connect the resistor in parallel to the CAN bus. Default OFF
	RES 2	Terminal resistor 2,120Ω. Switch down to ON to connect the resistor in parallel to the CAN bus. Default OFF
3	CAN L	CAN interface, CAN_L signal line connection terminal
	CAN H	CAN interface, CAN_H signal line connection terminal
	CAN GND	CAN interface, CAN ground signal line connection terminal
4	USB	Standard Type-C interface, firmware upgrade can be performed via this interface
5	485A	Serial port interface, RS485 A
	485B	Serial port interface, RS485 B
	NC	Empty interface
	NC	Empty interface
	NC	Empty interface

<Note>

When connecting USR-CAN518to the CAN bus, connect CAN_H to CAN_H and CAN_L to CAN_L.

RES is for terminal resistor selection.Switch any dip switch to ONto connect the internal 120Ω resistor of the module in parallel to the CAN bus; otherwise, the 120Ω resistor is not connected to the bus.

According toISO 11898specifications, to enhance the reliability ofCAN-bus communication,CAN-bus terminal matching resistors (120Ω) are usually added to both ends of the bus network, as shown in the figure below. The value of the terminal matching resistor is determined by the characteristic impedance of the transmission cable. For example, if the characteristic impedance of the twisted pair is120Ω, then the two ends of the bus should also integrate120Ωterminal resistors.

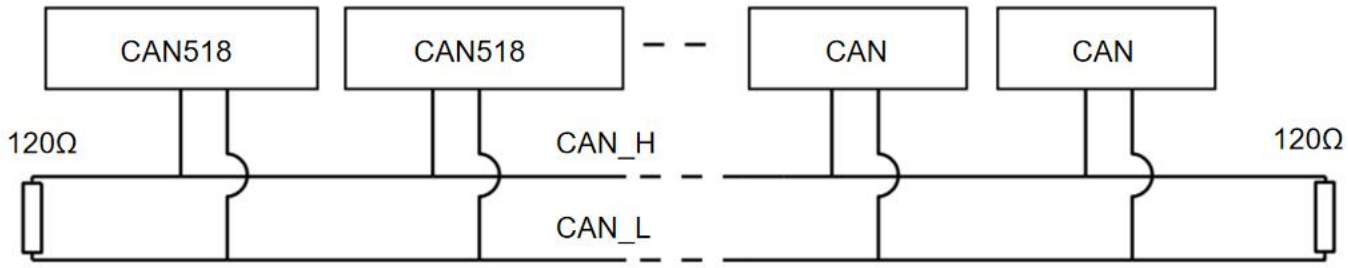


Figure2. CAN Bus Connection

2.3. LED Indicator Description

USR-CAN518has 4 LED indicators:POWER, WORK, CAN, RS485. Users can easily observe the device status through the indicators. The indicatordefinitions are as follows.

Table 4 Indicator Rules

Indicator	Color	FunctionDescription
POWER	Red	Steady on when powered, off when power is lost
WORK	Green	Blinking: Device running normally, frequency 1s; Rapid blinking: Entered CAN bus passive error state; Steady on: CAN bus operation abnormal
CAN	Green/Red	Green blinking: Indicates CAN port is receiving data Red blinking: Indicates CAN port is sending data
RS485	Green/Red	Green blinking: Indicates serial port is receiving data Red blinking: Indicates serial port is sending

		data

3. Product Features

3.1. Function Configuration Description

CAN518 supports parameter configuration via host computer software.

The host computer configuration is simple and easy to use.The following introduces the host computer parameter configuration method; please read this description carefully.

（1）Download the host computer software from the official website and open it. You can select the device connection method as needed. Click the connection method and choose serial port configuration or network configuration.

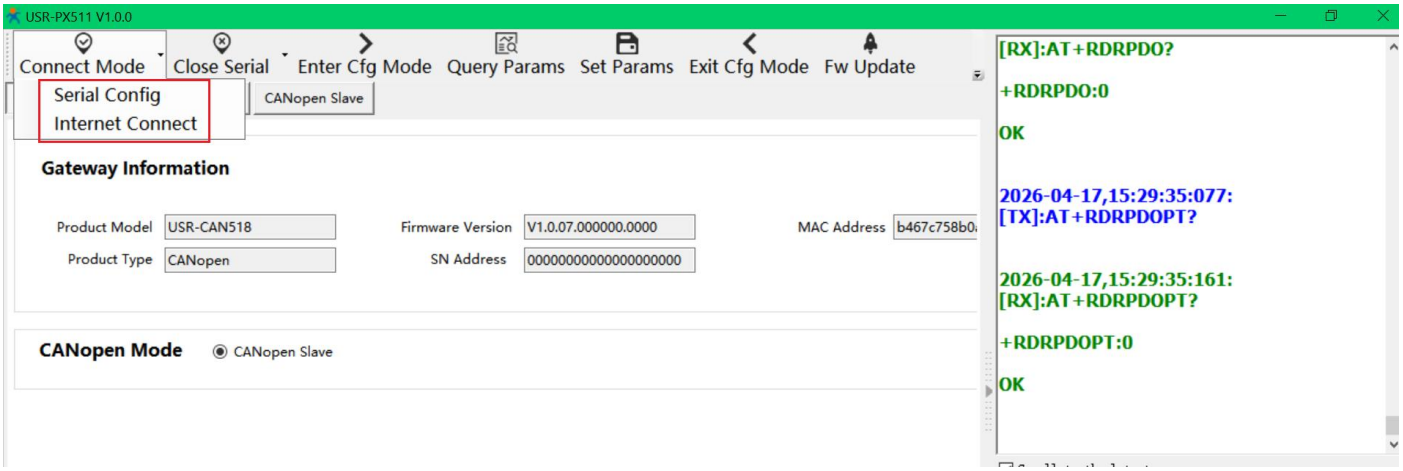


Figure3. Select Configuration Method

（2）If network configuration is selected:

- First, perform a network search to find devices.
- Select the device and click Read Parameters.
- After all current parameters have been read, proceed with parameter configuration.
- After configuration is complete, click Set Parameters to apply all configured parameters.

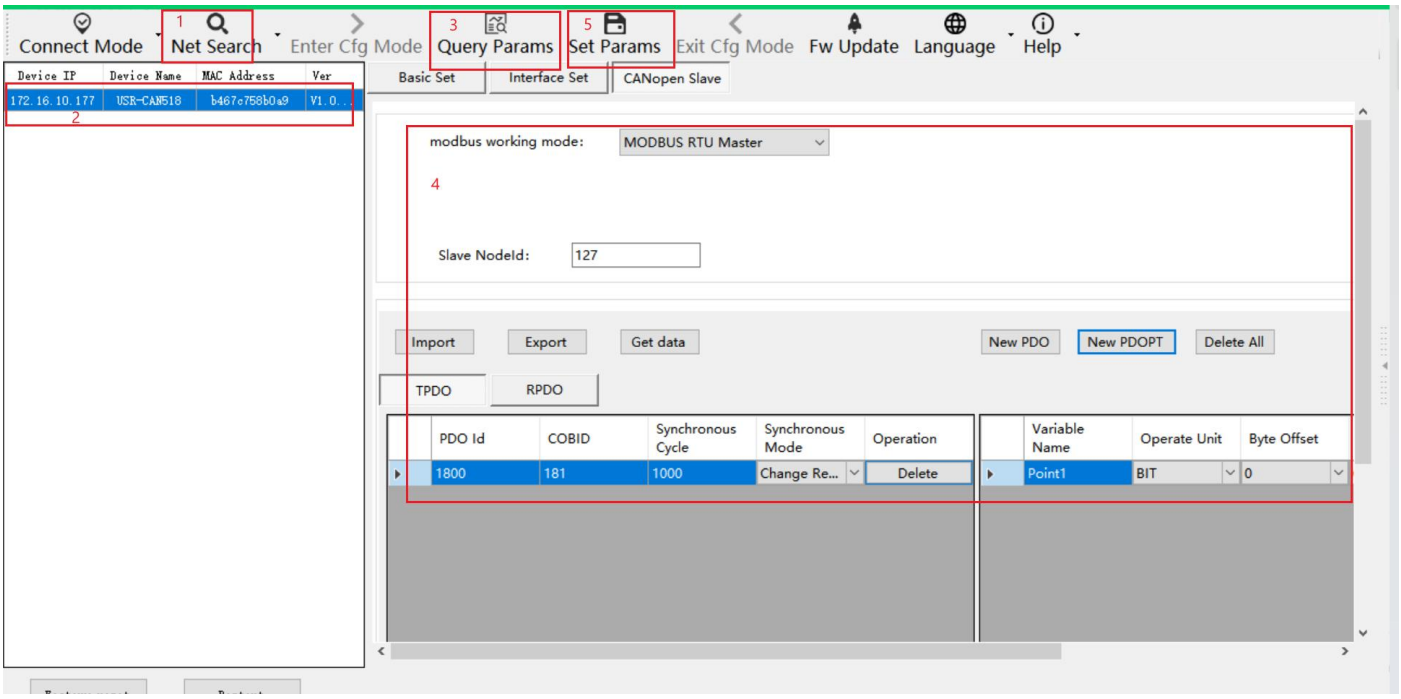


Figure4. Network Configuration Connection

（3）If serial port configuration is selected:

- Open the serial port. At this time, the message interface on the right displays:[Info]:COM95Serial port opened
- Click to enter configuration mode
- Click Read Parameters
- After all current parameters have been read, proceed with parameter configuration
- After configuration is complete, click Set Parameters to apply the parameters

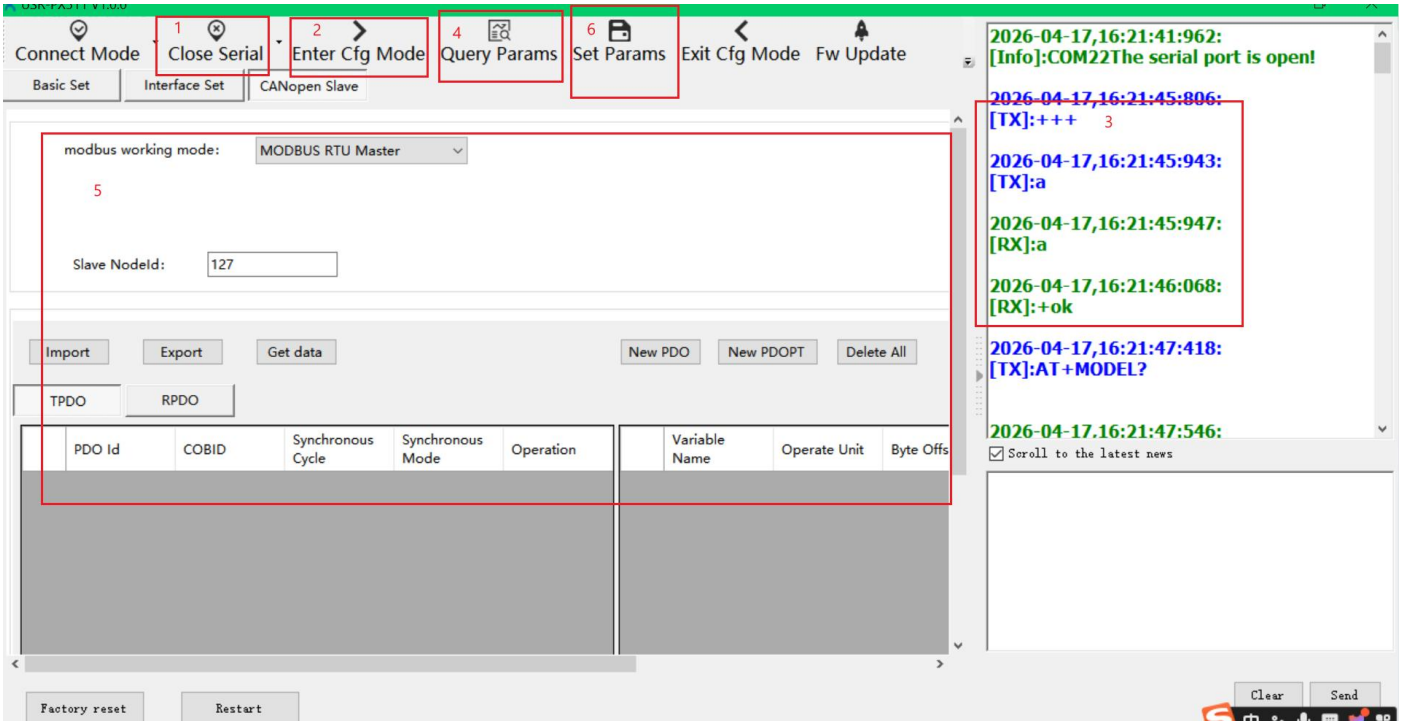


Figure5. Serial Port Parameter Configuration

3.2. CAN Parameters

Protocol: Supports CAN mode.

CAN Baud Rate: Range 5K~1000K, default 100kbps. Common baud rates can be selected directly: 5K, 10K, 20K, 50K, 100K, 120K, 125K, 150K, 200K, 250K, 400K, 500K, 600K, 750K, 1000K. Custom baud rates are supported. Baud rate calculation formula: Baud Rate = 60M / [(1 + BS1 + BS2) * BRP]

BS1: Phase Buffer Segment 1, range 1~16

BS2: Phase Buffer Segment 2, range 1~8

BRP: Prescaler value, range 1~1024

BS1\BS2\BRP are configurable, and the device will automatically calculate the current baud rate based on these three values.

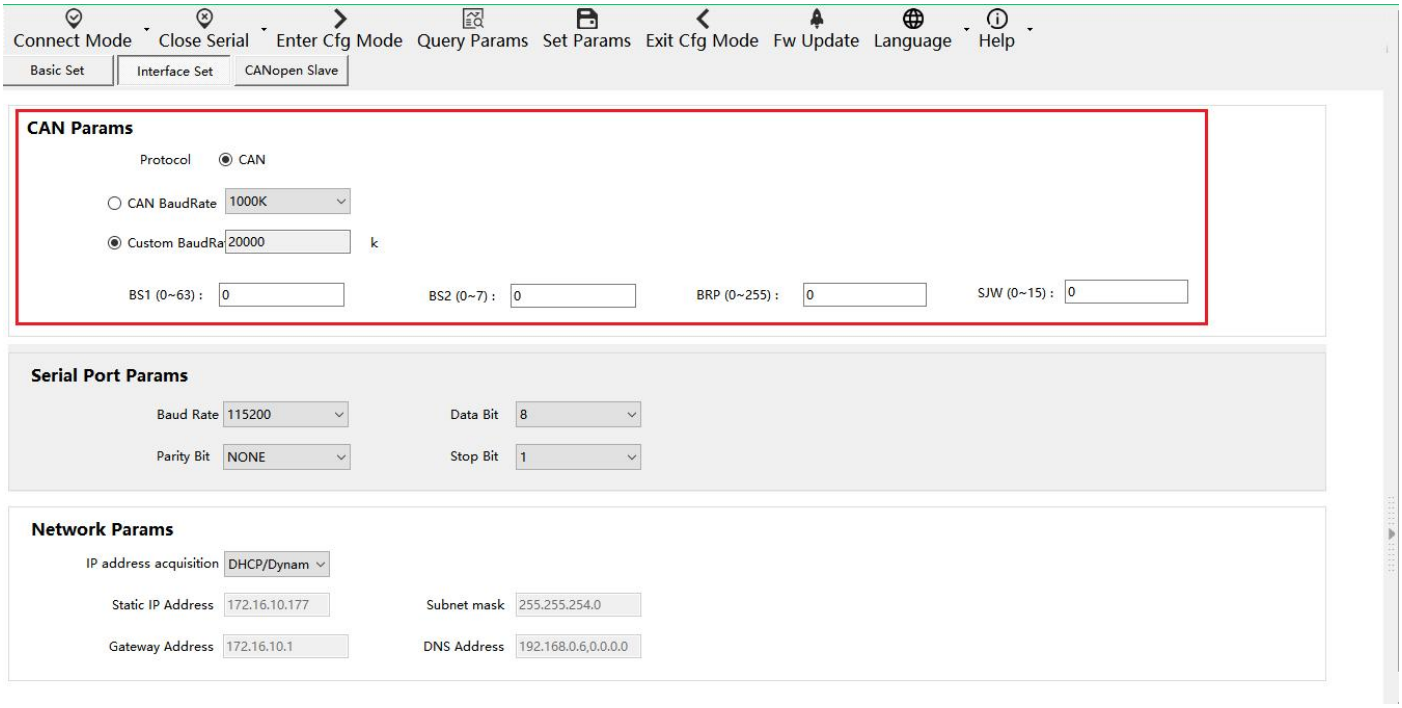


Figure6. CAN Parameter Configuration

3.3. Ethernet Port Parameters

(1) IP Address Acquisition:

The IP address represents the module's identity in the LAN and must be unique within the LAN, so it cannot duplicate other devices in the same LAN.CAN315IP addresses can be obtained via Static IP or DHCP.

- Static IP

Static IP requires manual user setup. During setup, ensure that the IP, subnet mask, and gateway are written simultaneously. Static IP is suitable for scenarios where IP and devices need to be statistically tracked and correspond one-to-one.

Advantages: Devices that cannot be assignedIP addressescan befoundvia full-segment broadcast search

Disadvantages: Different LAN subnetsdiffer, leading to inability to perform normalTCP/UDP communication

- DHCP

The main function of DHCP is to dynamically obtain IP addresses, Gateway addresses, DNS server addresses, etc., from the gateway host, thereby eliminating the tedious steps of setting IP addresses.It is suitable for scenarios where there are no specific IP requirements and no strong need for a one-to-one correspondence between IP and module.

Advantages: Devices connected to routers etc.with DHCP Servercan communicate directly,reducingthe hassle of settingIP addresses,gatewayaddressesand subnet masks;

Disadvantages: When connected tonetworks withoutDHCP Server, such asdirect connection to a computer,CAN518willnot work properly.

（2）Subnet Mask:

Subnet mask is a 32-bit address used to mask part of an IP address to distinguish between the network identifier and the host identifier, indicating whether the IP address is on a local area network or a remote network..A subnet mask cannot exist alone; it must be used in conjunction with an IP address..We commonly use Class C subnet mask: 255.255.255.0, the number of IP addresses within the subnet is 2 to the power of 8 minus 2, i.e., 256-2=254,generally host addresses all being 0 or 1 (binary) have special purposes.

（3）Gateway Address:

Gateway address refers to the network number of the network where the module's current IP address is located.If

connecting to external networks via devices like routers, the gateway is the router's IP address; if set incorrectly, external network access will fail. If no such devices are connected, no setting is required, and the default can be used.

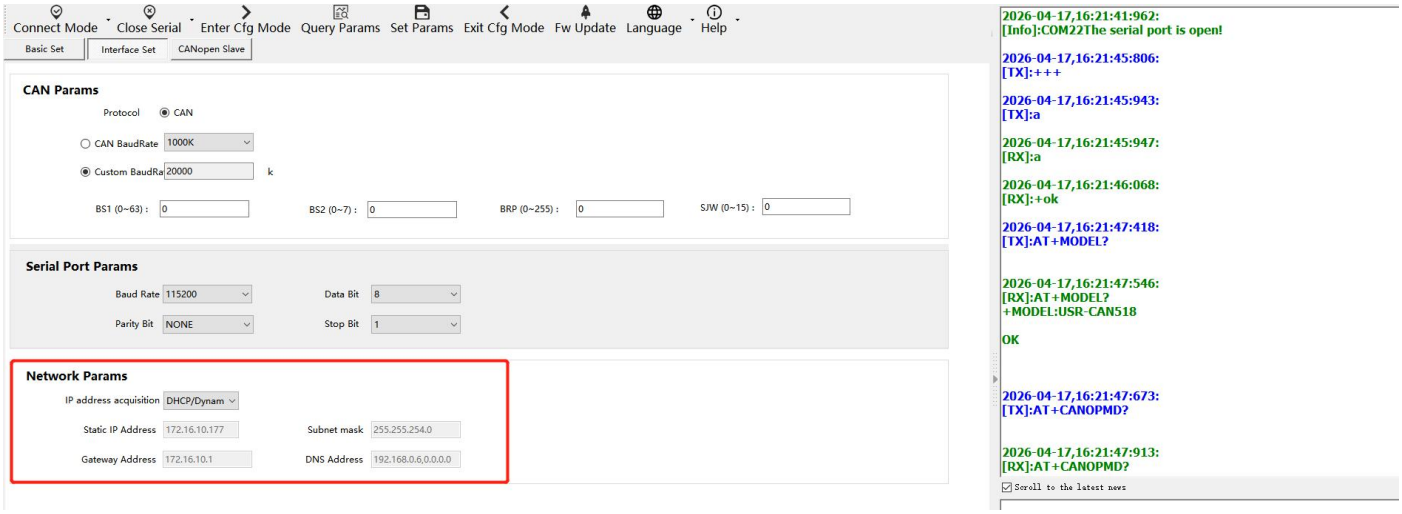


Figure7. Port Parameter Configuration

3.4. Serial Port Parameters

- **Baud Rate:**600~230400bps, default 115200bps
- **Data Bits:**8
- **Parity Bit:**NONE, EVEN, ODD,default is None
- **Stop Bits:**1, 2, default 1
- **Mode Switching:**Optional RS485/RS232 mode, cannot be used simultaneously

3.5. CANopen Gateway

This device supports the standard CANopen slave protocol. The following is an introduction to its features.

3.5.1. CANopen Basic Functions

NMT Node Boot-up Message: After any CANopen slave comes online, to notify the master that it has joined the network (facilitating hot-swapping) or to avoid Node-ID conflicts with other slaves, this slave must send a node boot-up message. The ID of the node boot-up message is 700h+Node-ID, with data being 1 byte of 0. The producer is the CANopen slave.

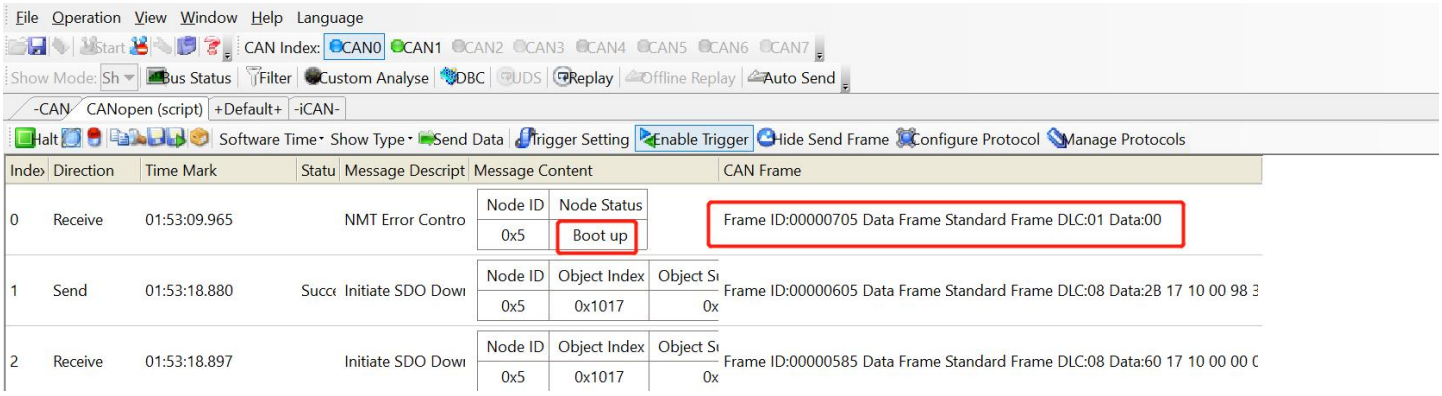


Figure8. Node Boot-up Message

NMT Node Status and Heartbeat Message:

To monitor whether CANopen nodes are online and their current status, CANopen applications typically require online slaves to periodically send status messages (heartbeat messages) so that the master can confirm if a slave has failed or disconnected from the network. The CANopen master can determine the current status of this device through its slave

heartbeat information.

The CANID is the same as the node boot-up message, 700h+Node-ID, with data being 1 byte representing the node's current status.

Heartbeat reports device status: 0x7F Pre-operational state, 0x05 Operational state, 0x04 Stopped state.

CANopen slaves send heartbeat messages according to the heartbeat producer time (ms) filled in object dictionary entry 1017h, while the CANopen master (NMT master) checks according to the heartbeat consumer time filled in 1016h. If no heartbeat message is received from the slave after several heartbeat consumer times, the slave is considered offline or damaged.

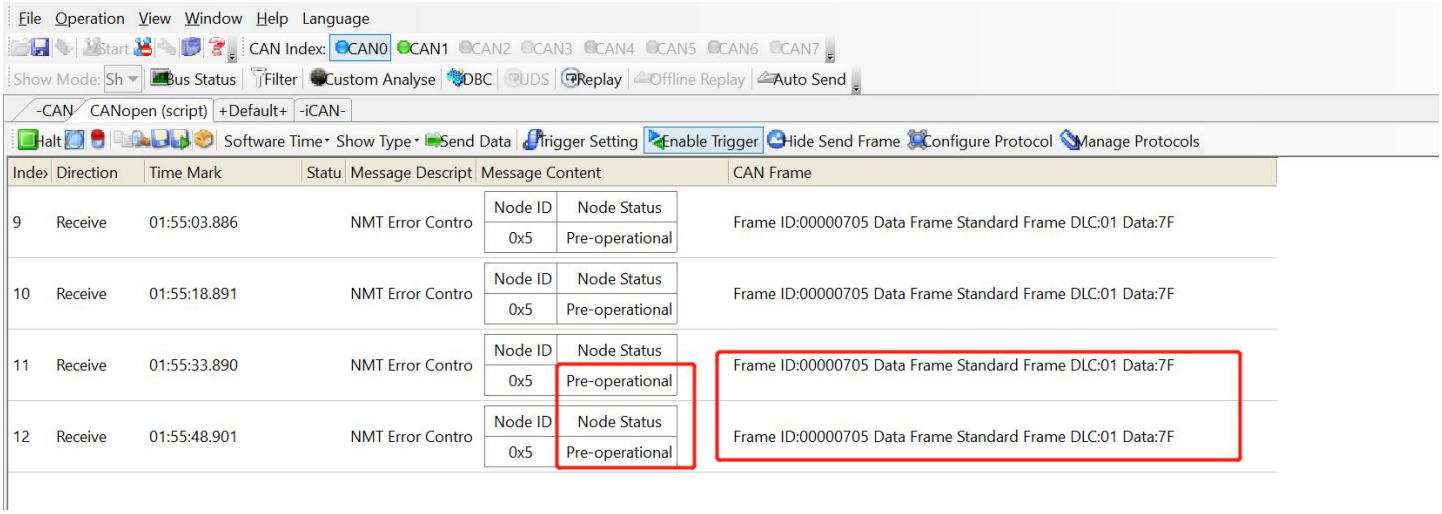


Figure9. Node Heartbeat Message

NMT Node State Change Command:

In the NMT network management of the CANopen protocol, the core element is the NMT node state change command, which is the "command" message issued by the NMT master for network management. Users must memorize these commands.

CANIDs are all 000h, having the highest CAN priority. Data consists of 2 bytes:

- \uF075 The 1st byte represents the command type:
  - 01h is the start command (puts the node into operational state). To enable data transmission through the gateway, the slave device must be put into operational state..
  - 02h is the stop command (puts the node into stopped state).
  - 80h is enter pre-operational state (puts the node into pre-operational state).
  - 81h is reset node application layer (resets the node's application to initial state, e.g., train doors return to open state).
  - 82h is reset node communication (re-initializes the node's CAN and CANopen communication, generally used when bus interference causes passive error state or bus-off).
- \uF075 The second byte represents the controlled node's Node-ID. To control all nodes in the entire network simultaneously, this value should be 0.

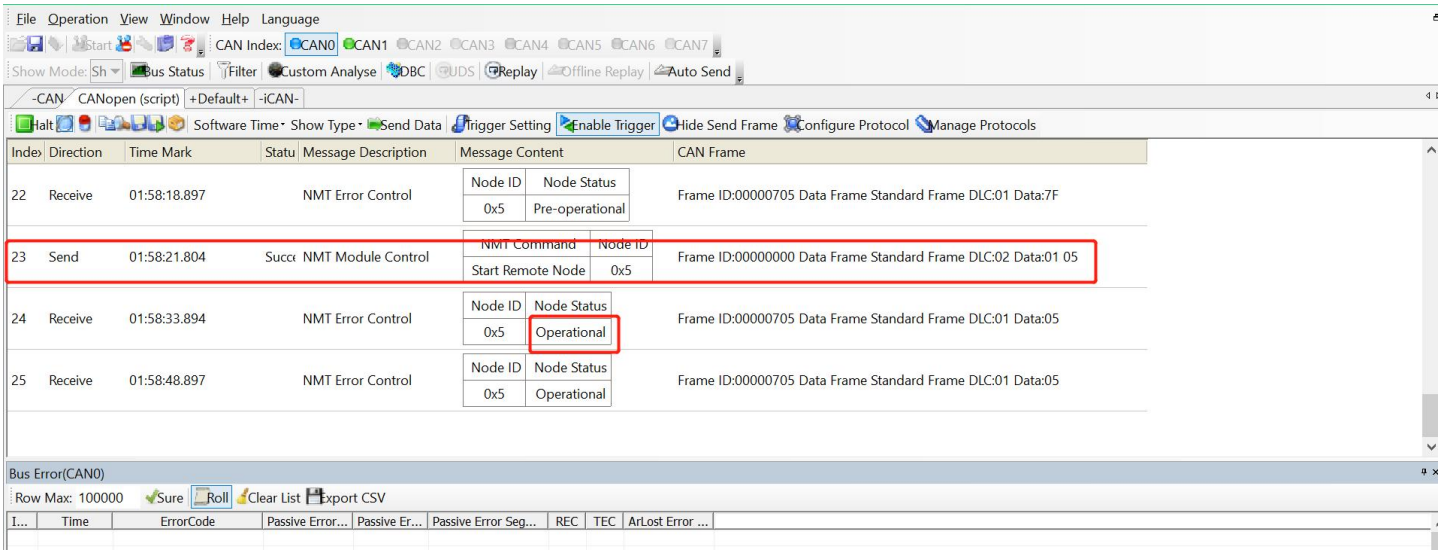


Figure10.Node Status Message

3.5.2. CANopen to Modbus Gateway

This device adopts a simple method to achieve interoperability between CANopen master devices and Modbus slave devices. CAN518 acts as a slave on the CAN side and as a master on the Modbus side. Below is a detailed description of the CANopen to Modbus gateway.

3.5.2.1. Modbus Enable Configuration

(1) Enter parameter configuration state: If connection method is network connection: Network Search --> Select Device --> Read Parameters --> Enter Parameter Configuration State. If connection method is serial port connection: Serial Port Configuration --> Select corresponding serial port, open serial port --> Enter configuration state --> Read parameters.

(2) Configure CAN parameters. According to the upstream CANopen master, configure matching CAN baud rate as needed, supporting custom baud rates.

(3) Select Modbus gateway working mode. Supports device acting as RTU master or TCP master.

When CAN518 works as an RTU master, it supports up to 255 Modbus RTU protocol slaves. Serial port parameters can match slave baud rate, data bits, stop bits, and parity bits.

When CAN518 works as a TCP master, acting as a TCP client. It supports connection to a single Modbus TCP protocol slave.

- Server IP: TCP slave IP address
- Local Port: Device local port number
- Remote Port: TCP slave port number

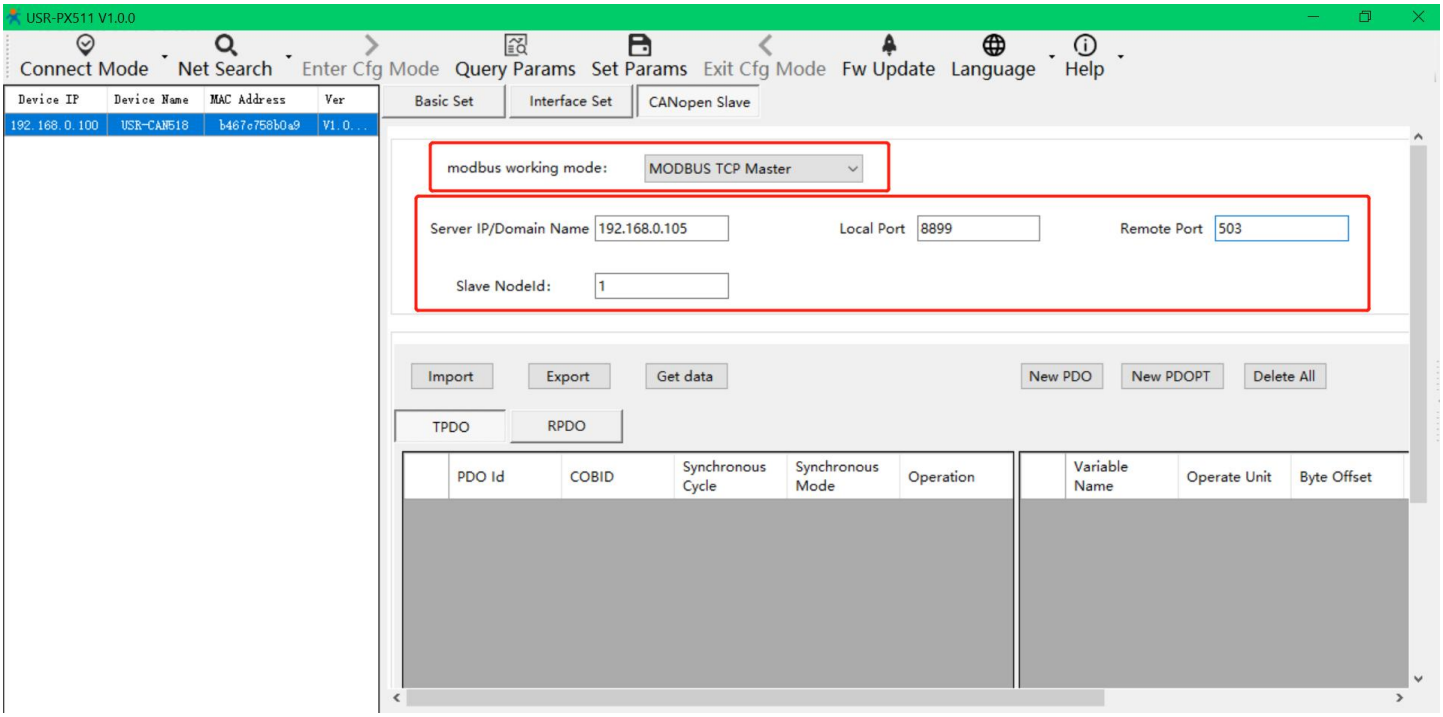


Figure11.Parameter Configuration

（4）CANopen parameter configuration.

Slave Node ID: Set the Node ID for CAN518 as a CANopen slave, between 1 and 127.

TPDO: Configure CAN518 transmit PDO messages, up to 128 entries.

RPDO: Configure CAN518 receive PDO messages, up to 128 entries.

3.5.2.2. TPDO Configuration

By configuring TPDO messages, CAN518 can read data from corresponding address registers of Modbus slaves, map the read Modbus register data to the TPDO data field, and send it to the CANopen master as CAN frames according to set rules. Up to 128 transmit message configurations are supported, with each message capable of adding up to 16 variable data items mapped to corresponding data positions in the CAN frame.

This device provides convenient upper computer software for easy TPDO message configuration. Configuration can be done directly in the upper computer, or point tables can be imported/exported via .csv files. This is suitable for scenarios with many collection and mapping points. Upper computer configuration methods are as follows:

Direct Configuration:

(1) Enter parameter configuration state: If connection method is network connection: Network Search --> Select Device --> Read Parameters --> Enter Parameter Configuration State. If connection method is serial port connection: Serial Port Configuration --> Select corresponding serial port, open serial port --> Enter configuration state --> Read parameters. Network Search --> Select Device --> Read Parameters --> Enter Parameter Configuration State.

(2) Click Add PDO to configure the required TPDO transmit message content. Select the message, click Add Variable, and configure mapping point parameters.

(3) If adjustments are needed, individual messages or individual points can be deleted. Alternatively, click Delete All to remove all messages directly.

(4) After point configuration is complete, set and save to take effect.

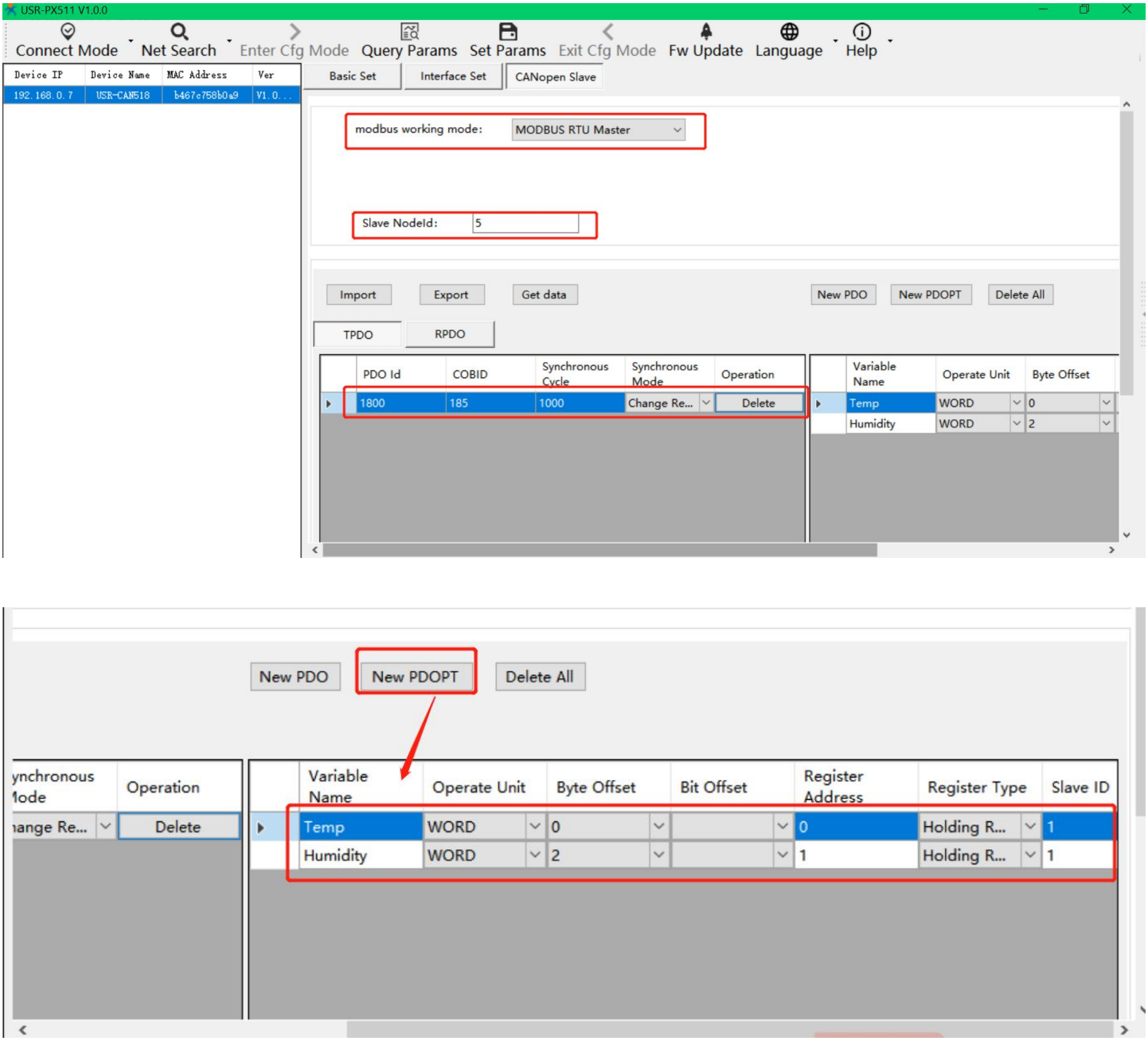


Figure12.TPDO Configuration Diagram

Parameter Information Introduction:

Parameter	Meaning	Range	Corresponds to in .csv document
Message Parameters			
PDO Index	PDO Index is a number used in the CANopen protocol to uniquely identify a Process Data Object (PDO). It is a key parameter for locating and configuring PDOs in the Object Dictionary. Each PDO has a unique index in the Object Dictionary.	1800 ~ 19FF (Hexadecimal)	1800 ~ 19FF
COB-ID	Sets the Communication Object Identifier for the PDO (the message frame ID of this communication object).	181~ 57F (Hexadecimal)	181~ 57F
Synchronization Period	Sets the time interval for updating Modbus slave register data to the TPDO, in milliseconds (ms).	0~65535 ms	0~65535

Synchronization Mode	Triggermode for the device to report CAN messages to the CANopen master. Cyclic Report: Report according to the set period time. Change Report: Report after any data point in this group of messages changes.	Cyclic Report Change Report	Cyclic Report--0 Change Report--1
Variable Parameters			
PDO Index	Consistent with the PDO Index in Message Parameters.	1800 ~ 19FF (Hexadecimal)	1800 ~ 19FF
Variable Name	The name of the variable, which is not mapped to CAN data and is used for mnemonic purposes.	Supports English letters or numbers.	Corresponds to English letters or numbers.
Operation Unit	Describes the size of the data range mapped from a CAN message frame to Modbus. BIT: Bit BYTE: 1 byte WORD: 2 bytes DWORD: 4 bytes QWORD: 8 bytes	BIT BYTE WORD DWORD QWORD	BIT--0 BYTE--1 WORD--2 DWORD--4 QWORD--8
Byte Offset	Selects which byte in the CAN message data segment to start mapping Modbus register data to the CAN message data segment sequentially.	0~7	0~7
Bit Offset	Only effective when the Operation Unit is BIT. When the Operation Unit is BIT: Both byte offset and bit offset are supported. The byte offset determines which byte in the CAN message the current data is in, and the bit offset determines which bit position in the byte the data is in.	0~7	0~7
Register Type	Modbus Register Type When the Operation Unit is BYTE, WORD, DWORD, or QWORD, function codes 03 and 04 are supported; when the Operation Unit is BIT, function codes 01 and 02 are supported.	01 (Coil Register) 02 (Input Register) 03 (Holding Register) 04 (Input Register)	01 02 03 04
Register Address	The starting address of the register in the device or Modbus slave for the sent message data.	0~65534	0~65534
Slave ID	Modbus Slave Address	1~255	1~255
Byte Order	Modbus Data Storage Method	Little Endian Big Endian	Little Endian--0 Big Endian--1

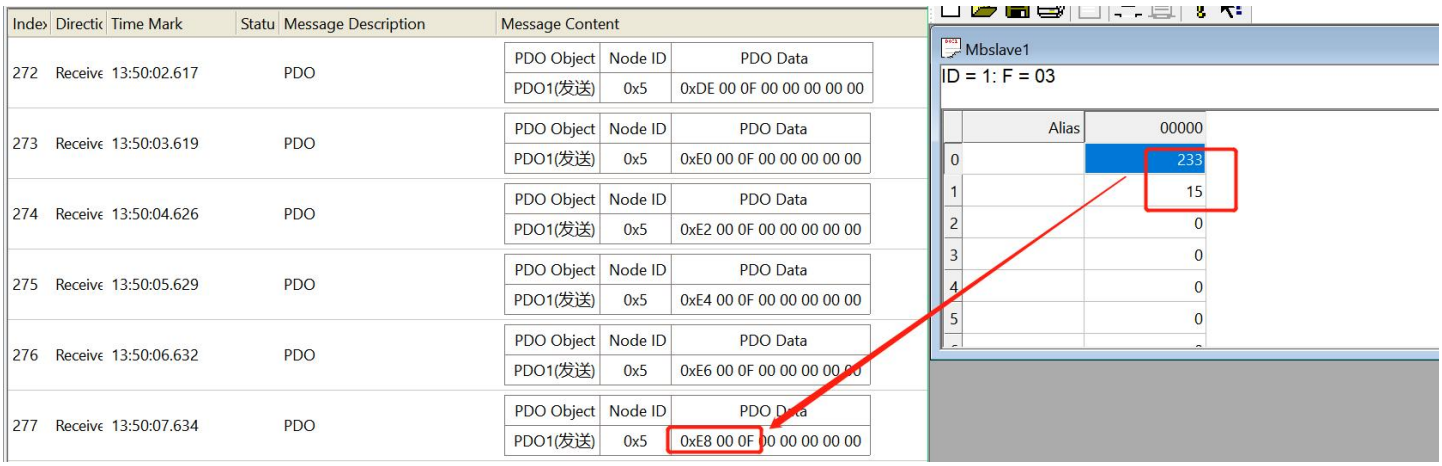


Figure13.TPDO Sending Simulation Example

3.5.2.3. RPDO Configuration

By configuring RPDO messages, the CAN518 can read data from the corresponding address registers of the Modbus slave, then map the read Modbus register data to the TPDO data field and send it to the CANopen master, transmitting it as CAN frames according to set rules. It supports up to 128 sending message configurations, with each message capable of adding up to 16 variable data items, mapped to the corresponding data positions in the CAN frame.

This device provides quick host computer software for easy configuration of TPDO messages. Configuration can be done directly in the host computer software, or point tables can be imported and exported via .csv files. This is suitable for scenarios with many collection points and mapping points. The host computer configuration methods are as follows:

Direct Configuration:

(1) Enter parameter configuration state: If network connection is selected: Network Search --> Select Device --> Read Parameters --> Enter Parameter Configuration State. If serial port connection is selected: Serial Port Configuration --> Select Corresponding Serial Port, Open Serial Port --> Enter Configuration State --> Read Parameters. Network Search --> Select Device --> Read Parameters --> Enter Parameter Configuration State.

(2) Click Add PDO to configure the required TPDO sending message content. Select the message, click Add Variable, and configure the mapping point parameters.

(3) If adjustments are needed, individual messages or individual points can be deleted. You can also click Delete All to directly delete all messages.

(4) After point configuration is complete, set and save to take effect.

By configuring the device's RPDO messages, the CAN518 can receive RPDOs sent by the CANopen master and then write the RPDO data field content to the Modbus slave registers. It supports up to 128 receiving message configurations, with each message capable of adding up to 16 variable data items.

This device provides quick host computer software for easy configuration of RPDO messages. Configuration can be done directly in the host computer software, or point tables can be imported and exported via .csv files. This is suitable for scenarios with many collection points and mapping points. The host computer configuration methods are as follows:

Direct Configuration:

(1) If serial port connection is selected: Serial Port Configuration --> Select Corresponding Serial Port, Open Serial Port --> Enter Configuration State --> Read Parameters. Network Search --> Select Device --> Read Parameters --> Enter Parameter

Configuration State. Enter Parameter Configuration State: Network Search --> Select Device --> Read Parameters --> Enter Parameter Configuration State.

(2) Click Add PDO to configure the required RPDO sending message content. Select the message, click Add Variable, and configure the mapping point parameters.

(3) If adjustments are needed, individual messages or individual points can be deleted. You can also click Delete All to directly delete all messages.

(4) After point configuration is complete, click Configure Data, save and restart to take effect.

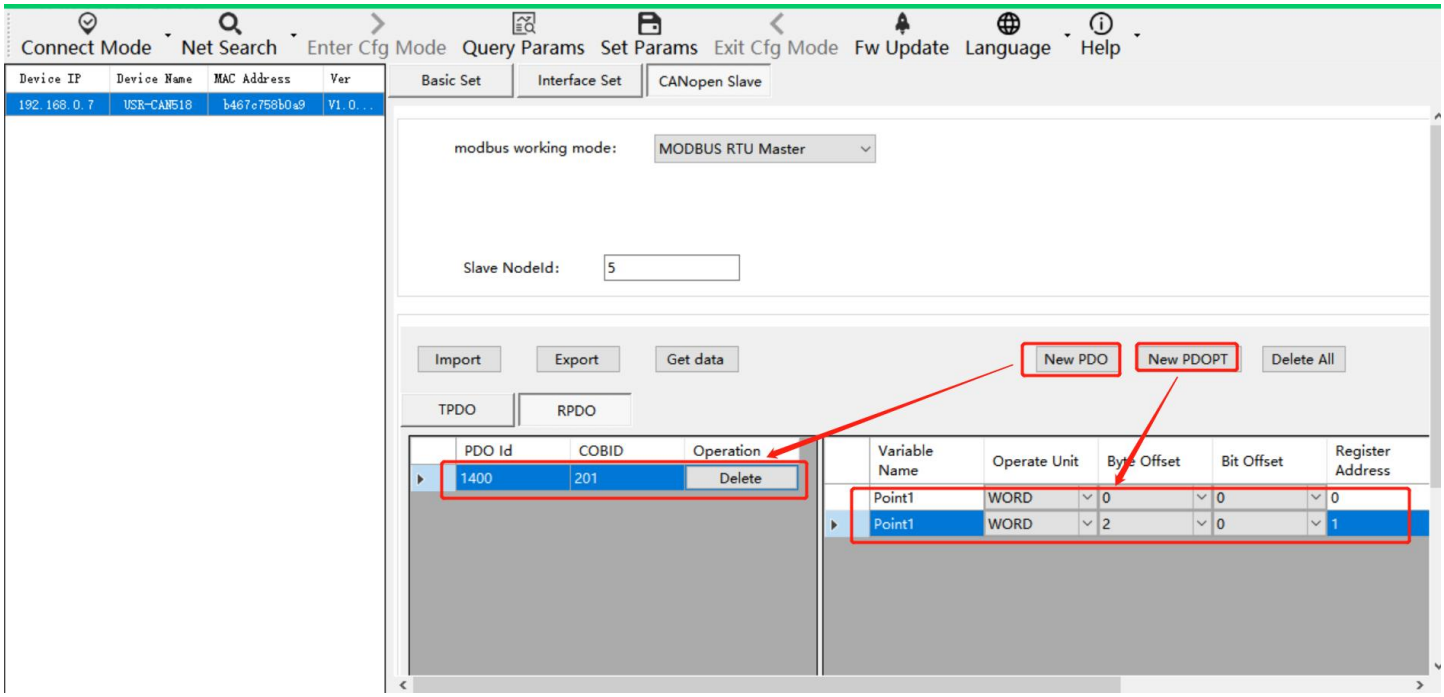
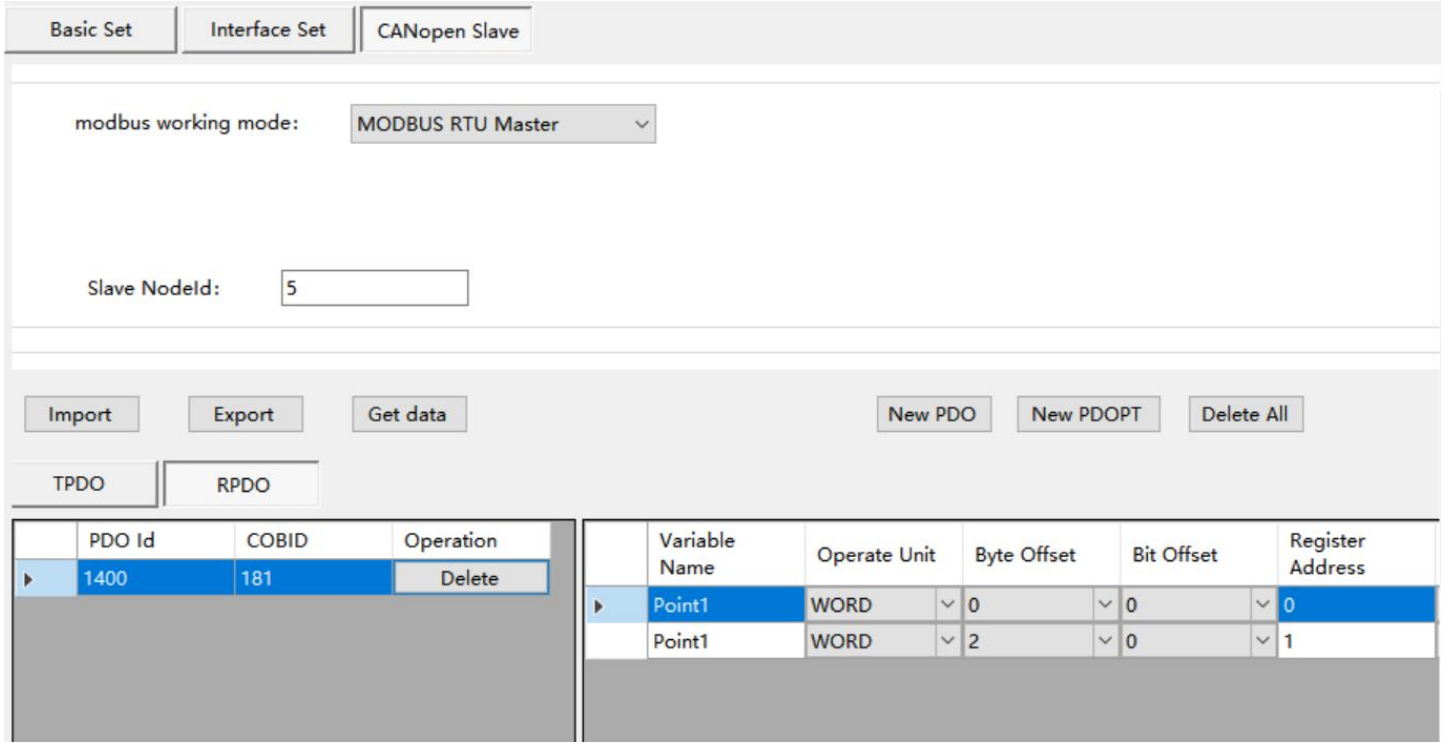


Figure14.RPDO Message Configuration Diagram

Parameter Information Introduction:

Parameter	Meaning	Range	Corresponding in .csv document
Message Parameter			
PDO Index	PDO Index is a number used in the CANopen protocol to uniquely identify a Process Data Object (PDO). It is a key parameter for locating and configuring PDOs in the Object Dictionary. Each PDO has a unique index in the object dictionary.	1400 ~ 15FF (Hexadecimal)	1400 ~ 15FF
COB-ID	Sets the Communication Object Identifier for the PDO (the frame ID of this communication object's message).	181~ 57F (Hexadecimal)	181~ 57F
Variable Parameter			
PDO Index	Consistent with the PDO Index in the Message Parameters.	1400 ~ 15FF (Hexadecimal)	1400 ~ 15FF
Variable Name	The name of the variable, which is not mapped to CAN data and is used for mnemonic purposes.	Supports English letters or numbers	Corresponding English letters or numbers
Operation Unit	Describes the size of the data range mapped from one CAN	BIT	BIT--0

	message frame to Modbus. BIT: Bit BYTE: 1 byte WORD: 2 bytes DWORD: 4 bytes QWORD: 8 bytes	BYTE WORD DWORD QWORD	BYTE--1 WORD--2 DWORD--4 QWORD--8
Byte Offset	Selects which byte in the CAN message data segment to start from, sequentially writing the CAN message data segmentintoModbus register data.	0~7	0~7
Bit Offset	Only effective when the Operation Unit is BIT. When the Operation Unit is BIT: Both byte offset and bit offset are supported. The byte offset determines which byte in the CAN message to start writing to the Modbus register, and the bit offset determines which bit within that byte to start writing the data.	0~7	0~7
Register Type	Modbus Register Type When the Operation Unit is BYTE, WORD, DWORD, or QWORD, Function Code 03 is supported; when the Operation Unit is BIT, Function Code 01 is supported.	01 (Coil Register) 03 (Holding Register)	01 03
Register Address	The starting address of the register in the device or Modbus slave where the sent message data is stored.	0~65534	0~65534
Slave ID	Modbus Slave Address	1~255	1~255
Byte Order	Modbus data storage method	Little Endian Big Endian	Little Endian--0 Big Endian--1



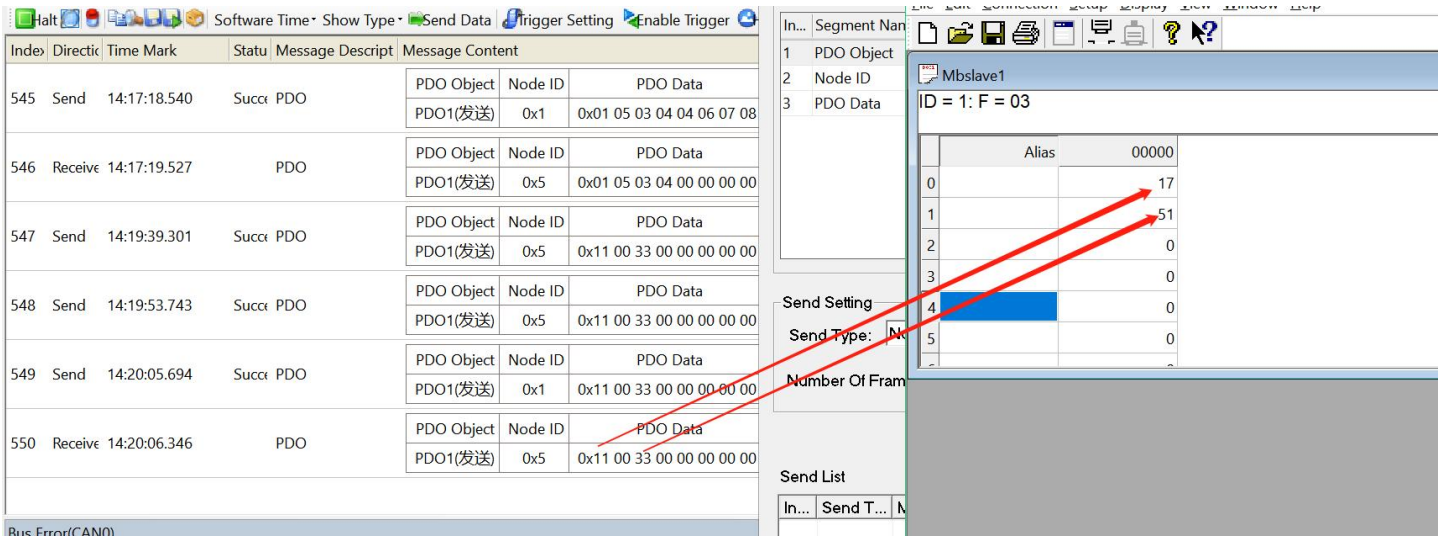


Figure15.TPDO Reception Simulation Example

3.6. Firmware Upgrade

The device supports easy firmware upgrades via PC software. If a firmware upgrade is required, please contact USR-IOT technical support to obtain the latest firmware. Do not attempt unauthorized operations. For detailed firmware upgrade methods, please refer to the "USR-CAN518 Product Firmware Upgrade Manual".

3.7. Factory Reset

Hardware Factory Reset: The module can be reset to factory settings via hardware.After powering on,press and hold the Reload button, keep it pressed, and release it after3-15seconds to perform a hardware factory reset.

Software Factory Reset: Factory settings can be restored via software configuration.

Restore Factory Settings via AT Commands: In AT command mode, send the command AT+CLEAR,followed by Enter.When you receive the correct response+OK, the factory settings are restored.

Software Configuration Settings:

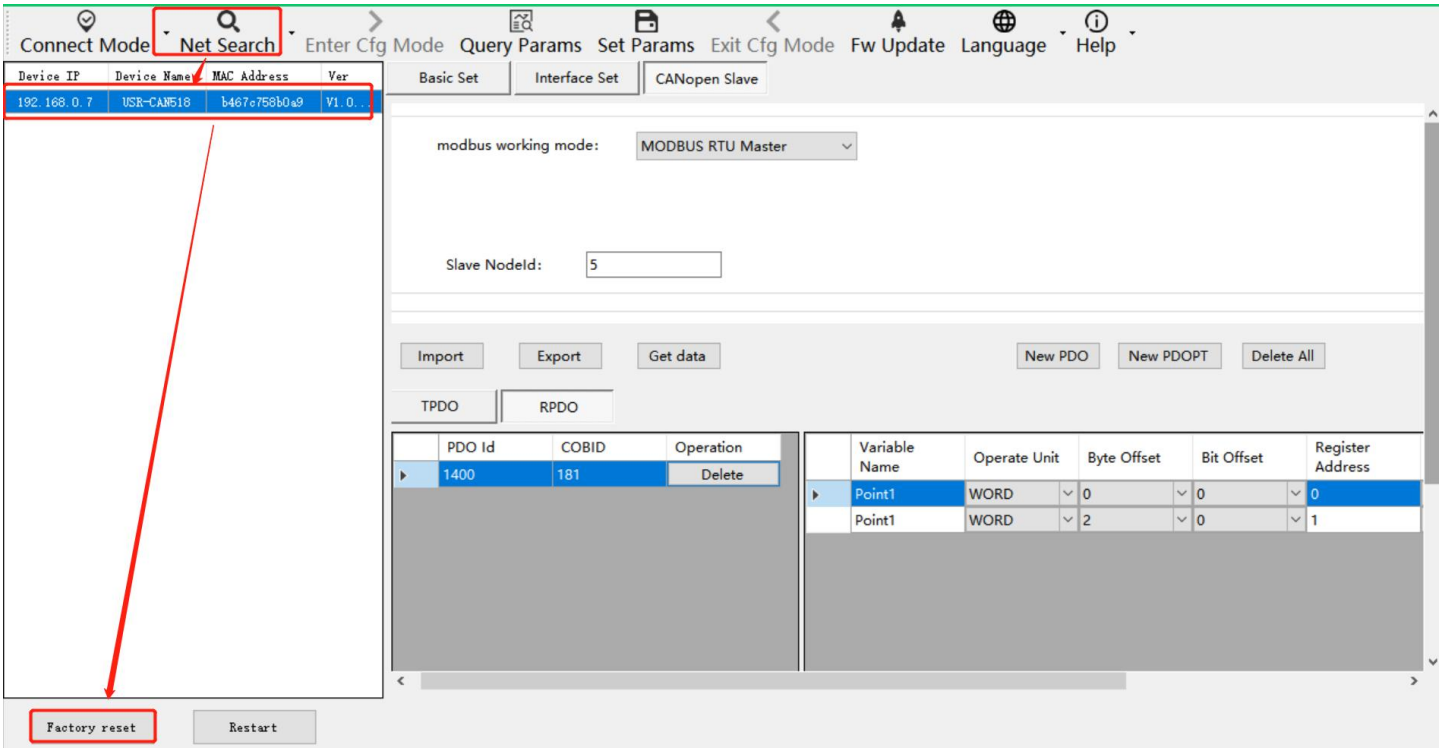


Figure16.Software Configuration Diagram

4. Contact Us

Jinan USR IOT Technology Limited

Address：Floor 12 and 13, CEIBS Alumni Industrial Building, No. 3 Road of Maolingshan, Lixia District, Jinan, Shandong, China

Official website: https://www.pusr.com

Official shop: https://shop.usriot.com

Technical support: http://h.usriot.com/

Email : sales@usriot.com

Tel : +86-531-88826739

Fax : +86-531-88826739-808

5. Disclaimer

6. Revision History

Document Version	Update Content	Update Time
V1.0.0	Initial Version	2026-4-19

Your Trustworthy Smart IOT Partner

PUSR®
www.pusr.com

Official Website: www.pusr.com

Official Shop: shop.usriot.com

Technical Support: h.usriot.com

Inquiry Email: inquiry@usriot.com

Skype & WhatsApp: +86 13405313834

Click to view more: Product Catalog & Facebook & Youtube