User Manual

PXB-80xx

EtherCAT Protocol Converter Gateway

UM01010101 1.00 Date:2025/1/14

category	contents
Key word	protocol converter gateway, user manual, product manual
Abstract	PXB-80XX Product Performance Description and User Guide

ZLG 致远电子

Revision History

Version	Date	Reson
V1.00	2025/1/14	Initial Release

Contents

1. Product Introduction

1.1 Product Overview

The PXB-80 series products are industrial fieldbus protocol converters such as EtherCAT to Modbus, CAN/CAN FD, DeviceNet, etc. developed by Guangzhou Zhiyuan Electronics Co., Ltd. This series of products has two EtherCAT Slave network port interfaces, one CAN FD or one RS485 interface, one standard 10/100M Ethernet interface, and one USB Type-C interface. It is equipped with a dedicated ESC chip and a domestically produced high-performance RISC-V processor to process the conversion of EtherCAT to various industrial fieldbus data packets. The PXB-80 series protocol converter provides a configuration tool for the upper computer, which can flexibly conFigure related functions and easily achieve seamless conversion of various industrial bus protocol data to EtherCAT data.

Figure 1.1 Product Series Group Photo

1.2 Product Series Description

1.2.1 Naming Rules

The naming Rules for PXB protocol converter series products is shown in Figure 1.2.

PXB — XX XX X X

Equipment

Series code

60：Modbus series
80：EtherCAT series
81：EtherNet/IP series
82：Profinet series

Protocol code

20：CAN FD bus
21：CANopen protocol
22：DeviceNet protocol
23：SAE J1939 protocol
30：Modbus protocol
31：OPC UA protocol
32：EtherNet/IP protocol

Installation method

D：DIN rail

Master-slave

M：CANopen / DeviceNet Master

Figure 1.2 Naming Rules for Series Products

1.2.2 Ordering Information

Table 1.1 Ordering Information for Series Products

Product model	Protocol conversion type	Installation method
PXB-8020	EtherCAT <=> CAN/CAN FD	35mm DIN rail
PXB-8021M	EtherCAT <=> CANopen(Master)	35mm DIN rail
PXB-8022	EtherCAT <=> DeviceNet	35mm DIN rail
PXB-8022M	EtherCAT <=> DeviceNet(Master)	35mm DIN rail
PXB-8030	EtherCAT <=> Modbus TCP/RTU	35mm DIN rail
PXB-8032	EtherCAT <=> TCP/IP	35mm DIN rail

1.3 Hardware Features

Table 1.2 Product Hardware Features

Input voltage	9～36VDC，150mA @ 12VDC
Power protection	Anti reverse connection protection, short circuit protection
Automatic restart trigger	Built in independent WDT (watchdog timer)
RS485 isolation	Digital isolation, Power isolation
CAN isolation	Digital isolation, Power isolation
RS485 baud rate	Up to 2Mbps
CAN FD baud rate	40k~5Mbps，supports CAN FD acceleration
Terminal resistance	Built in 120 ohms (can be configured using upper computer software)
Shell material	Metal
Dimensions	125.00mm×76.00mm×28.00mm(bare metal)
Installation method	Standard 35mm DIN rail
Operating Temperature	-40 to 85° C(-40 to 185° F)
Storage temperature (including packaging)	-40 to 85° C(-40 to 185° F)
relative humidity	5 to 95% (non condensing)
EMI	EN55032，CLASS A
EMC	IEC/EN 61000-4-2 ESD：Contact：4.0 kV；Air：8.0 kV IEC/EN 61000-4-4 EFT：Power supply：1.0 kV；Signal：0.5 kV IEC/EN 61000-4-5 Surge：Power supply：1.0 kV；Signal：0.5 kV IEC/EN 61000-4-6 CS(150 kHz to 80 MHz)：Power supply：3 V/m；Signal：3 V/m

1.4 Software Features

Table 1.3 Software Features

EtherCAT features	Supports EtherCAT Slave PDO with adjustable size
	Supports EtherCAT Slave with a maximum PDO buffer size of 254 bytes for transmitting and receiving
(PXB-8020) CAN /CAN FD features	Support setting parameters such as CAN type, CAN FD standard, and CAN FD acceleration
	CAN packet transmission supports multiple triggering methods such as cycle, status change, and single transmission
	CAN packet transmission supports multiple data sources
	Support transmitting and receiving packet s in the form of whole frames, bytes, etc
	Supports standard frames, extended frames, and remote frames
	Supports custom transmitting mode, allowing for precise transmitting of CAN or CAN FD packets
(PXB-8021M) CANopen features	Supports 126 Slave nodes
	Support 10K~1M baud rate setting
	Supports 128 sets of RPDO packets and 128 sets of TPDO packets
	Support SDO configuration, flexible configuration of Slave node dictionary
	Support byte and bit operation modes, flexible mapping of data to EtherCAT network
(PXB-8022) DeviceNet features	This device serves as a standard DevicetNet Slave and supports polling based I/O connection types
	DeviceNet supports speeds of 125K, 250K, and 500K
	Supports multi-level input and output bytes, with a maximum of 512 bytes for input and 512 bytes for output
	Support mutual conversion of data between EtherCAT and DeviceNet protocols in the form of bits, bytes, words, etc
	Support DeviceNet I/O scanning
	Support setting the update interval between EtherCAT data and DeviceNet data
	Provide standard EDS files for DeviceNet
(PXB-8022M) DeviceNet features	As a standard DevicetNet Master, this device can access up to 8 standard DevicetNet Slave
	DeviceNet communication speed supports 125K, 250K, and 500K
	Supports four types of I/O connections: polling, bit gating, state change, and cycle
	Each I/O connection type supports a maximum of 64 bytes of input/output buffer
	Support configuring the data update cycle for each DevicetNet Slave
	Support converting EtherCAT data and DeviceNet data to each other in bits, bytes, and other ways
(PXB-8030) Modbus features	IP address, Slave ID, destination IP, and port can all be set
	Supports 4 working modes: EtherCAT Slave to Modbus TCP/RTU Master/Slave
	When the Modbus side is set as the Master, a maximum of 32 Modbus Slave devices are supported

Continued

	Supports 01H, 02H, 03H, 04H, 05H, 06H, 15H, 16H instructions
(PXB-8032) TCP/IP features	Supports data conversion between EtherCAT EoE protocol and TCP Server,TCP Client and UDP
	Supports TCP Server (8 clients), TCP Client, and UDP
	The protocol type, IP address, and port can be set
Other features	Equipped with AWPX Tools configuration software, simple and easy to use
	Support software configuration to enable CAN and RS485 terminal resistors
	Support one click factory reset and multiple guarantees
	Support one click remote firmware upgrade

2. Hardware Description

2.1 Product appearance

Figure 2.1 Product appearance

The PXB-80 series products have one power interface, one USB-C interface, and one hidden button for resetting to factory settings at the top;The front of the product has four LED, two EtherCAT network port interfaces, one standard 10/100M RJ45 Ethernet interface, and one CAN/CAN FD interface or RS485 interface with electrical isolation.

Note：for specific LED, interface definition and usage, please refer to Indicator light Description and Interface Definition。

2.2 Indicator light Description

The PXB-80 series products have four LED on the front, which have different functional instructions for different product models. They are abbreviated according to relevant professional vocabulary to form corresponding labels, namely PWR、RUN、ECT、CAN、485、MS 和 NS。The specific instructions are shown in Table 2.1：

Table 2.1 PXB-8020 Indicator light status description

Ientification	Definition	State	Description
PWR	Equipment power indicator light	Off	The equipment is not powered on or has abnormal power supply
		Red	The equipment power supply normally
RUN	Equipment operation indicator light	Green flashing	The equipment is working normally
		Red flashing	The equipment initialization failed, did not enter working mode
ECT	EtherCAT communication status indicator light	Red	EtherCAT Slave is in the state of INIT、BOOTSTRAP
		Green flashing	EtherCAT Slave is in the state of PRE-OP、SAFE-OP
		Green	EtherCAT Slave is in the state of OP
CAN	CAN communication status indicator light	Off	The equipment has not transmitted or received data
		Green flashing	The equipment is transmitting and receiving data normally
		Red flashing	Abnormal data transmission and reception of equipment

Table 2.2 PXB-8021M Indicator light status description

Ientification	Definition	State	Description
PWR	Equipment power indicator light	Off	The equipment is not powered on or has abnormal power supply
		Red	The equipment power supply normally
RUN	Equipment operation indicator light	Green flashing	The equipment is working normally
		Red flashing	The equipment initialization failed, did not enter working mode
ECT	EtherCAT communication status indicator light	Red	EtherCAT Slave is in the state of INIT、BOOTSTRAP
		Green flashing	EtherCAT Slave is in the state of PRE-OP、SAFE-OP
		Green	EtherCAT Slave is in the state of OP
CAN	CANopen communication status indicator light	Red	The equipment has not transmitted or received data
		Green flashing	The equipment is transmitting and receiving data normally
		Green	Abnormal data transmission and reception

Table 2.3 PXB-8022、PXB-8022M Indicator light status description

Ientification	Definition	State	Description
PWR	Equipment power indicator light	Off	The equipment is not powered on or has abnormal power supply
		Red	The equipment power supply normally
ECT	EtherCAT communication status indicator light	Red	EtherCAT Slave is in the state of INIT、BOOTSTRAP
		Green flashing	EtherCAT Slave is in the state of PRE-OP、SAFE-OP
		Green	EtherCAT Slave is in the state of OP
MS	DeviceNet device status indicator light	Green	The equipment is running normally
		Green flashing	Due to missing, incomplete or incorrect configuration, the device needs to be debugged
		Red flashing	Recoverable faults
		Red	Unrecoverable fault, needs to be replaced
		Red and green flashing	The device is self checking
NS	DeviceNet network status indicator light	Off	The device is not online; The device has not completed duplicate MAC ID detection
		Green flashing	The device is online but not connected and is in an established state
		Green	The device is online and has one or more established connections
		Red flashing	Communication failure: repeated MAC ID detection failed, BUS-OFF
		Red and green flashing	The device has detected a network access error and is in a communication failure state, and has received a point-to-point recognition offline fault request

注：PXB-8022 and PXB-8022M devices will perform LED self-test after power on, and users can determine whether the device is running by observing the status of the LED;

Firstly, MS self-test: display green for 0.25s ->display red for 0.25s ->display green;

Then NS self-test: display green 0.25->display red 0.25 s ->turn off.

Table 2.4 PXB-8030 Indicator light status description

Ientification	Definition	State	Description
PWR	Equipment power indicator light	Off	The equipment is not powered on or has abnormal power supply
		Red	The equipment power supply normally
RUN	Equipment operation indicator light	Green flashing	The equipment is working normally
		Red flashing	The equipment initialization failed, did not enter working mode
ECT	EtherCAT communication status indicator light	Red	EtherCAT Slave is in the state of INIT、BOOTSTRAP
		Green flashing	EtherCAT Slave is in the state of PRE-OP、SAFE-OP
		Green	EtherCAT Slave is in the state of OP
485	Modbus communication status indicator light	Red flashing slowly	Modbus network abnormal, device not connected
		Red flashing fastly	The function code in the configuration information has failed to be sent
		Green flashing fastly	All function codes in the configuration information have been successfully sent

Table 2.5 PXB-8032 Indicator light status description

Ientification	Definition	State	Description
PWR	Equipment power indicator light	Off	The equipment is not powered on or has abnormal power supply
		Red	The equipment power supply normally
RUN	Equipment operation indicator light	Green flashing	The equipment is working normally
		Red flashing	The equipment initialization failed, did not enter working mode
ECT	EtherCAT communication status indicator light	Red	EtherCAT Slave is in the state of INIT、BOOTSTRAP
		Green flashing	EtherCAT Slave is in the state of PRE-OP、SAFE-OP
		Green	EtherCAT Slave is in the state of OP
ETH	TCP/IP communication status indicator light	Red flashing slowly	TCP/IP connection exception
		Red flashing fastly	Data transmission failed due to existence
		Green flashing fastly	All data has been successfully transmitted

2.3 Interface Definition

The PXB-80 series products provide one OPEN3 power interface, one factory reset implicit button, and one USB Type-C interface at the top, as shown in Figure 2.2.

Figure 2.2 Top interface definition diagram

2.3.1 Power interface

The power supply supports a wide voltage input of 9-36V, and customers can choose a power supply within the voltage range to supply power to the equipment according to the on-site environment. The physical form of the power interface connector is OPEN3 3.81mm spacing lockable socket, and the shell silk screen label is "9V~36V". The power supply requirements for the product are shown in Table 2.6.

Table 2.6 Power Interface Input Power Specification

Parameter	Min	Typ	Max	Unit
Operating voltage	9.0	12.0	36.0	V
Operating current	—	150	—	mA
Power consumption	—	1.80	—	W

2.3.2 USB interface

The PXB-80 series products offer one Type-C USB interface. The physical form of the connector is a standard USB-C socket, and the silk screen label on the housing is "USB". This interface can be used as a power supply interface for the equipment during the debugging phase.

2.3.3 Implicit buttons

Considering that customers may have parameter configuration errors during use, resulting in abnormal product operation, an implicit button is reserved at the top of the product to restore factory settings.

During the power on process, pressing and holding the factory reset button will automatically restore the device to its factory settings, but it will not restart; If the product is running and long pressed for 5 seconds, it will automatically reset to factory settings and restart.

After restoring the factory settings, the original configuration parameters inside the device will be cleared.

In addition to four LED indicator lights on the front of the product, there are also two EtherCAT data interfaces: IN and OUT, one standard 10/100M Ethernet port and one isolated RS485 interface or isolated CAN/CAN FD interface, as shown in Figure 2.3.

Figure 2.3 Front interface definition diagram

2.3.4 EtherCAT interface

The EtherCAT side of this product is a Slave network interface. The EtherCAT interface provides two RJ45 network ports (IN and OUT). The IN network port can directly connect to the EtherCAT Master equipment or cascade the OUT network port of the EtherCAT Slave equipment; Similarly, the OUT network port can cascade the IN network port of EtherCAT Slave devices.

2.3.5 CAN/CAN FD interface

When the product is PXB-8020,PXB-8022,PXB-8021M and PXB-8022M, the bottom interface of the product is CAN/CAN FD interface, and the interface identification is shown in Table 2.7.

Table 2.7 CAN FD Interface Identification Description

Identification	Description
CANH	Isolate CAN_H signal line
CANL	Isolate CAN_L signal line
CGND	Isolate CAN ground

2.3.6 RS485interface

When the product is PXB-8030, the bottom interface of the product is RS485 interface, and the interface identification is shown in Table 2.8.

Table 2.8 RS485Interface Identification Description

Identification	Description
485A	Isolate RS485_A signal line
485B	Isolate RS485_B signal line
RGND	Isolate RS485 ground

2.3.7 Standard Ethernet port

The standard Ethernet port is labeled as NET, which supports standard 10/100M Ethernet communication. Users can connect to switches or PCs through this port and configure the corresponding parameters such as working mode, data baud rate, and data format of this product using the matching upper computer software; If the product model is PXB-8030, this network port can not only be used as a device configuration network port, but also as a Modbus TCP communication interface. For other product models, this network port is only used as a device configuration network port.

2.4 Terminal Resistance

The PXB-80 series protocol converter has reserved a terminal resistor of 120 ohms for the RS485/CAN communication port, which is not connected to the bus by default. Users can configure it using the upper computer software, as shown in Figure 2.4. Users can choose to connect or disconnect the terminal resistor to the bus, eliminating the need for external series connection or disconnection of resistors.

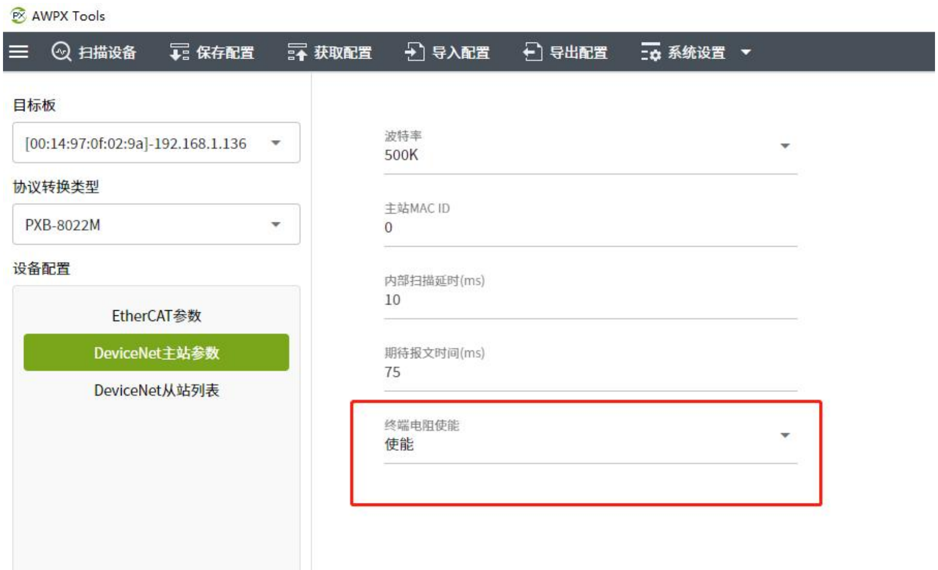

Figure 2.4 Terminal resistance configuration interface

3. Instructions for use

The PXB-80 series protocol converter products have similar usage methods, which can be configured and used according to the following steps. The operation steps can be divided into the following steps:

1、Install AWPX upper computer configuration software and TwinCAT3 software on the PC end;

2、After the product is powered on correctly, insert a network cable into the configuration port marked with "NET" and connect the PXB-80 protocol converter to the PC;

3、Open the AWPX upper computer configuration tool, configure according to the requirements in the configuration tool (please refer to 3.2), and issue the configuration to the PXB-80 protocol converter device;

4、Connect the Master equipment to the network port marked with "IN" on the PXB-80 protocol converter using Ethernet cables;

5、Perform data exchange and communication.

Note：If your Master equipment is not on the TwinCAT3 or x86 platform, you can import an XML file, connect the PXB-80 protocol converter device and configure it correctly according to the different Master devices, or use this device.

3.1 PLC parameter settings

Taking TwinCAT3 as an example, we will explore how to add a PXB-80 protocol converter;

3.1.1 Importing XML Files

Copy and paste the PXB-80 protocol converter XML file into the TwinCAT3 installation directory, for example:

C:\TwinCAT\3.1\Config\Io\EtherCAT；

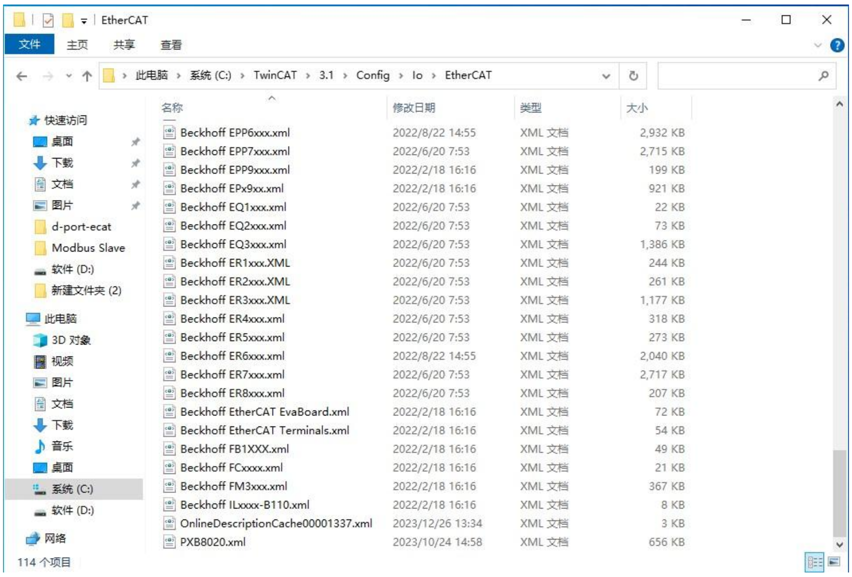

Figure 3.1　XML file placement directory

3.1.2 Configuring Master

1. New project

Open the installed TwinCAT3 software and enter the development environment:

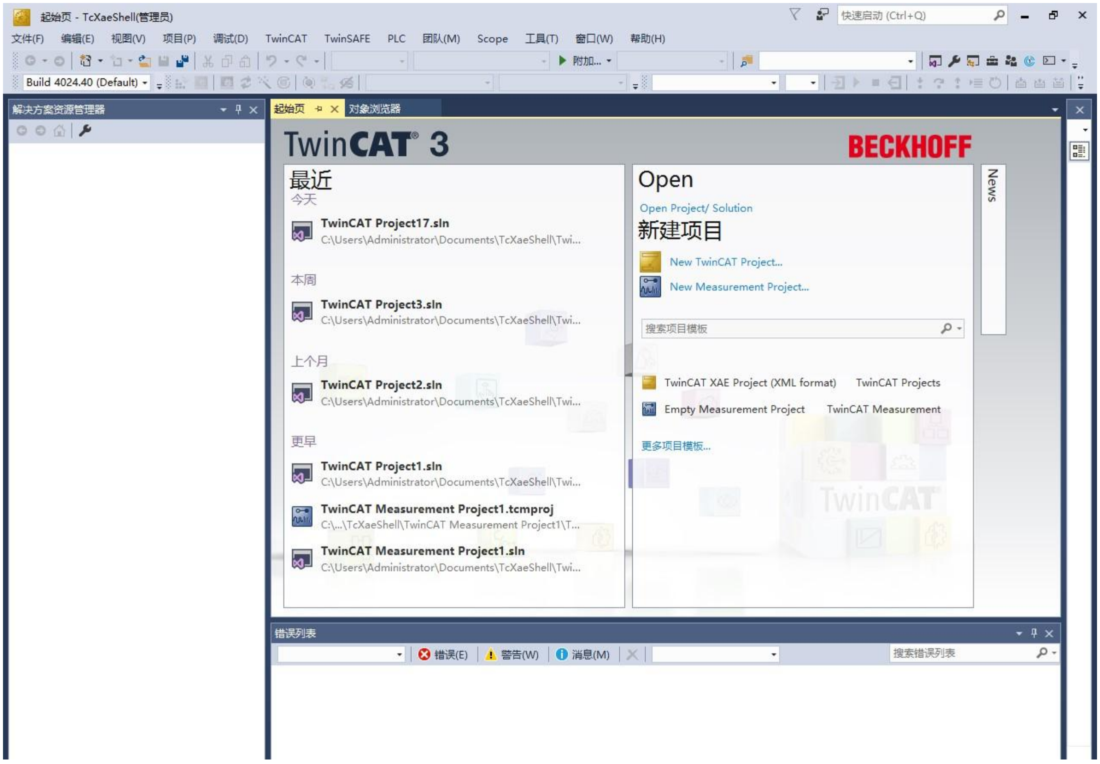

Figure 3.2 TwinCAT3 development interface

Select "File" - "New (N)" - "Project (P)..." from the menu bar in the upper left corner;;

Figure 3.3 New Project Interface

Select TwinCAT Projects, modify the project name, choose a suitable location to save the project, and click "OK":

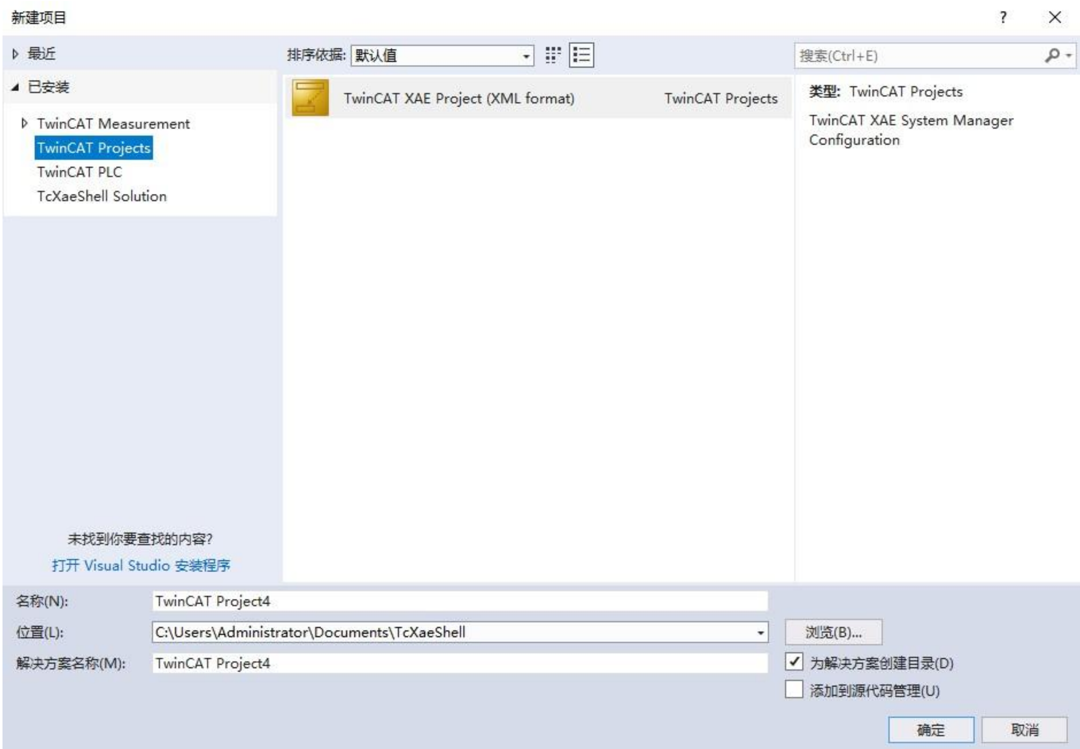

Figure 3.4 Engineering Information Modification Interface

After creating the project normally, the display interface is as follows:

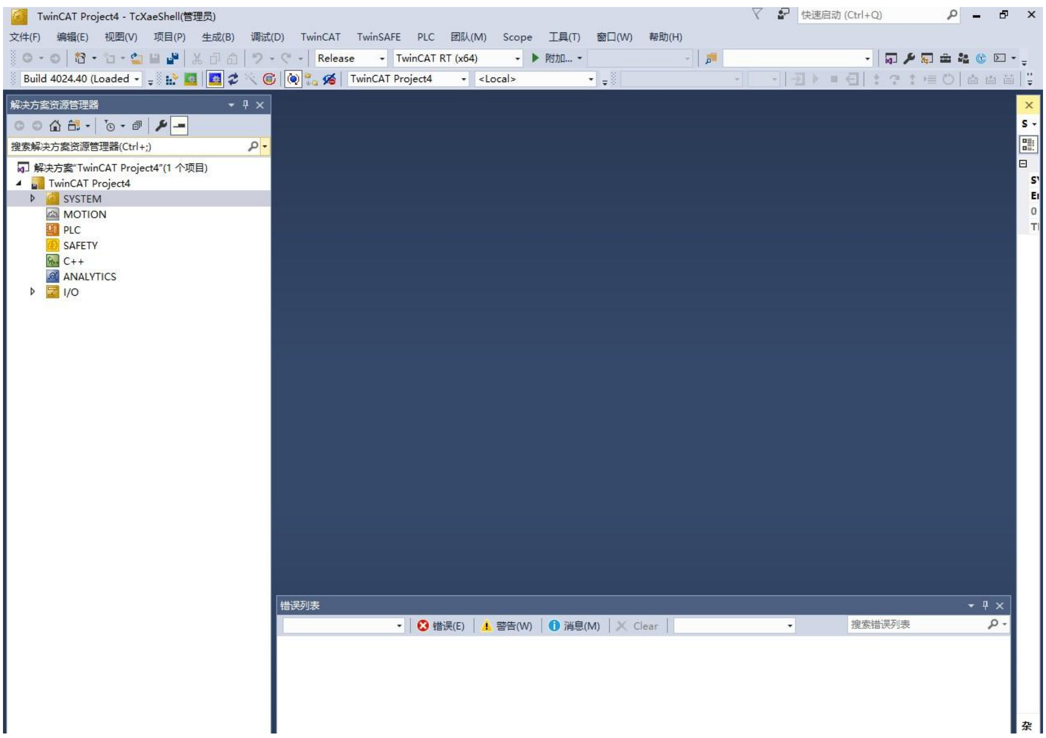

Figure 3.5 Project Creation Success Interface

Install the EtherCAT Master network card driver in TwinCAT3, click on "Show Realtime Ethernet Compatible Devices..." under "TwinCAT" in the main menu, select the local network card, and click "Install";

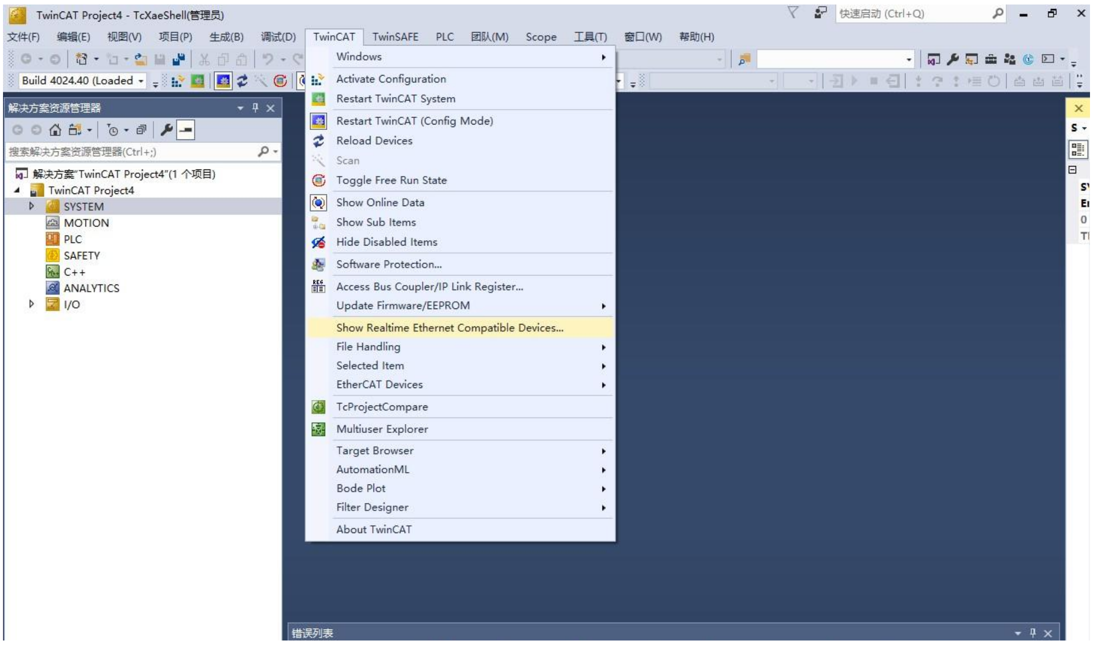

Figure 3.6　Network card installation interface

The normal installation interface of the network card is shown in the following figure:

Note：When using real-time protocols such as EtherCAT, the network port requires the TwinCAT RT driver. Therefore, the EtherCAT Master has requirements for network card compatibility. For details, please refer to the description of the "Supported Network Controllers" section in the TwinCAT3 overview on the official website of Beifu.

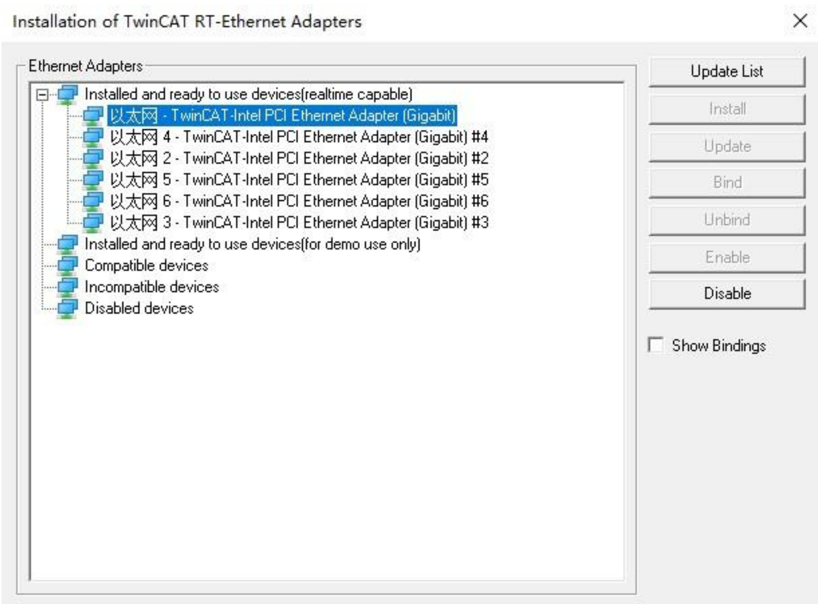

Figure 3.7　Network card driver normal interface

2. Scanning devices

In the engineering directory, select "I/O" to expand, choose "Devices", right-click and select "Scan" to scan the connected Slave devices. Before this operation, it is necessary to ensure that the PXB-80 protocol converter is powered on normally and connected to the EtherCAT Master device using a network cable;

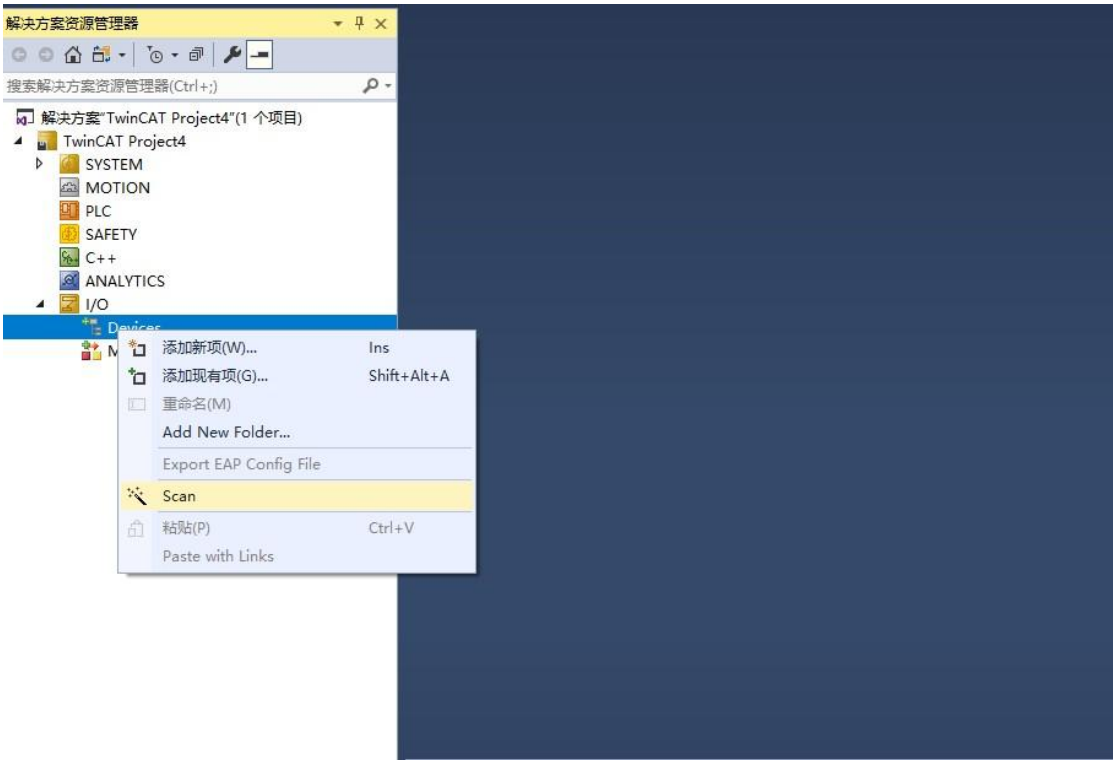

Figure 3.8 Scanning Slave device interface

Pop up the following dialog box, click the "OK" button;

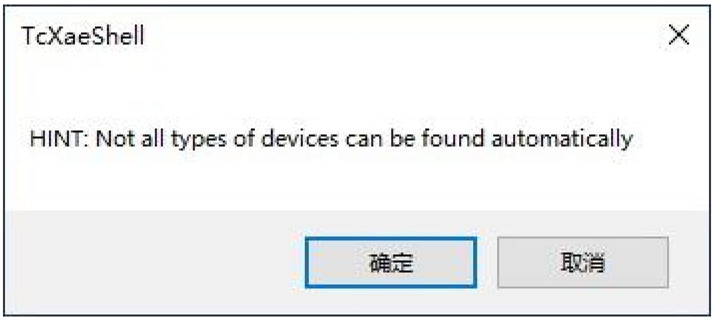

Figure 3.9 Prompt dialog box

Pop up the "6 new I/O devices found" dialog box, select the required Ethernet interface, and click "OK"; As shown in the following figure:

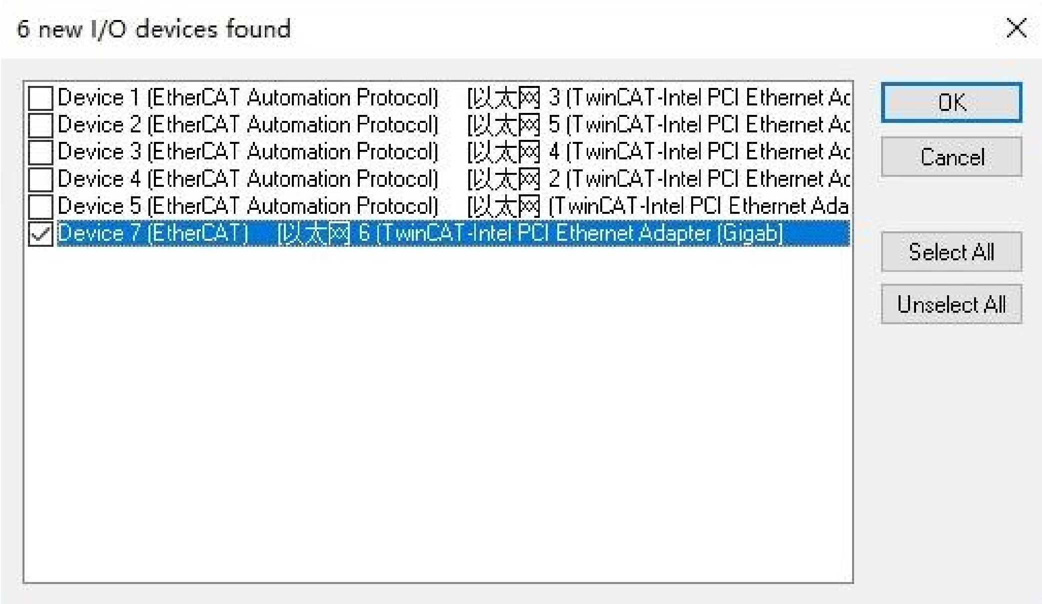

Figure 3.10 Selecting the network card connected to the Slave device

Click "Yes (Y)" to scan for Slave devices;

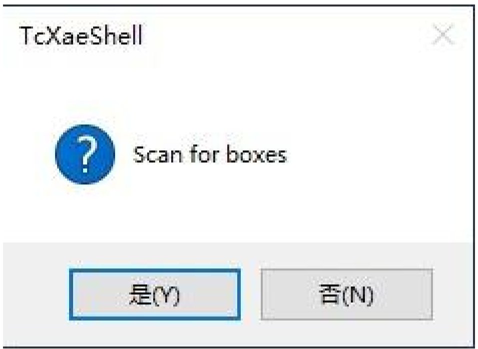

Figure 3.11 Selecting scan Slave devices

Select whether to enter "Activate Free Run" and click "No";

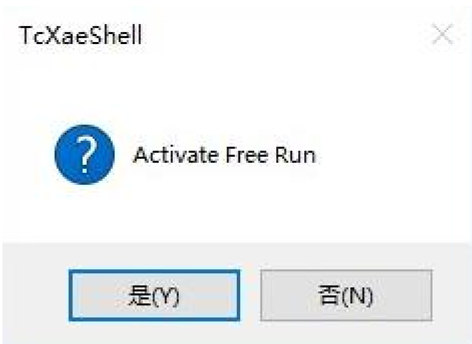

Figure 3.12 Prompt dialog box

The following devices were scanned normally:

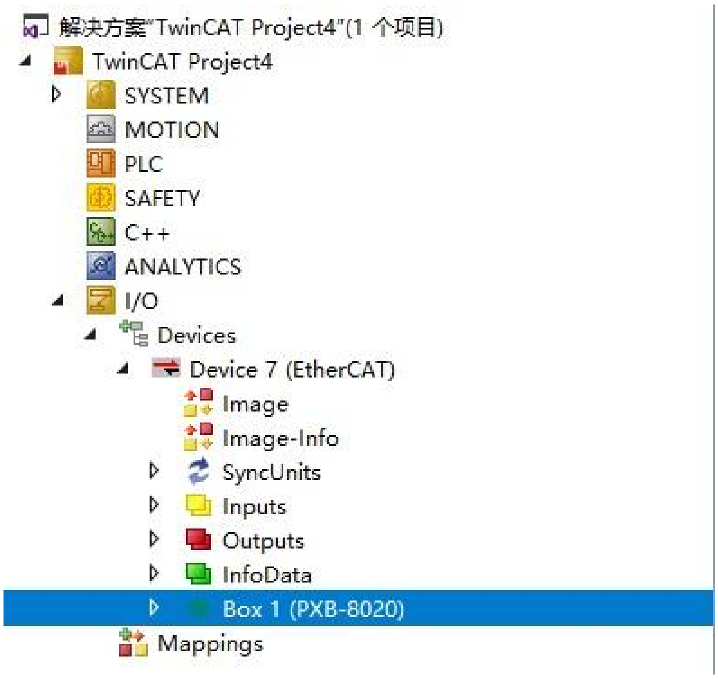

Figure 3.13 Device scanning success interface

It can be seen that "PXB-8020" has been scanned, indicating that the PXB-8020 protocol converter has been successfully recognized by the Master;

3. Check the connection

After establishing communication between the Master controller and the PXB-80 protocol converter, it can be seen that the Slave equipment has entered the "OP" state; The EtherCAT communication link has been successfully established;

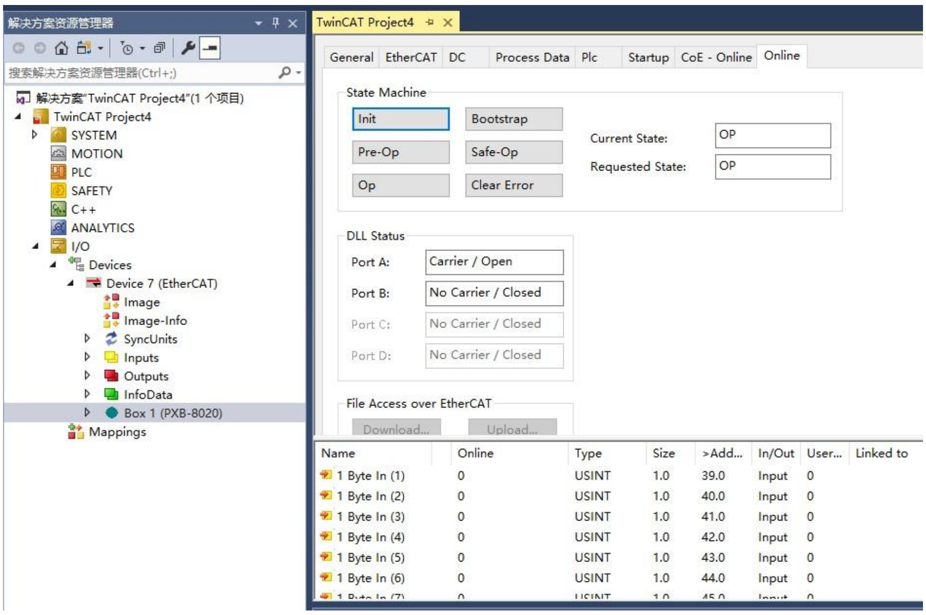

Figure 3.14 Operating status interface of Slave device

In the "Process Data" option page, it will be clear that TwinCAT3 has already allocated I/O data to the PXB-80 protocol converter; The size of MbxOut and MbxIn does not need to be modified; 'Size' represents the byte length of the command used to access data;

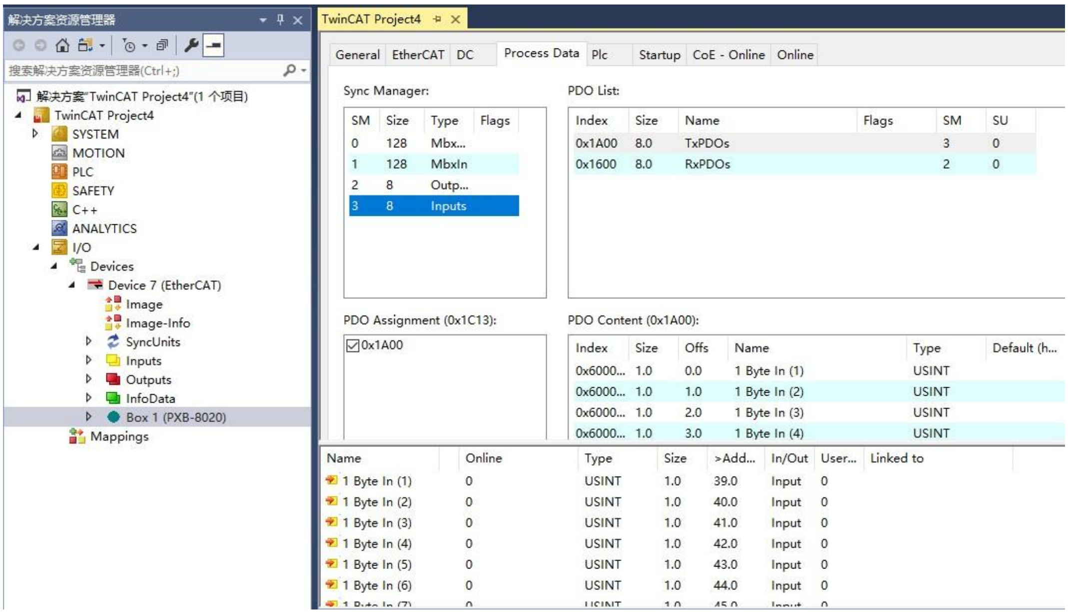

Figure 3.15 Process data interface

3.2 AWPX Configuration Software

3.2.1 Configuration software acquisition and installation

The PXB series protocol converters are configured through AWPX Tools software (hereinafter referred to as AWPX). The AWPX Tools configuration software can be downloaded by searching for "AWPX" on our official website (www.zlg.cn). After downloading, double-click the installation package of AWPX to start installing AWPX. After double clicking, the installation start interface is shown in Figure 3.16,

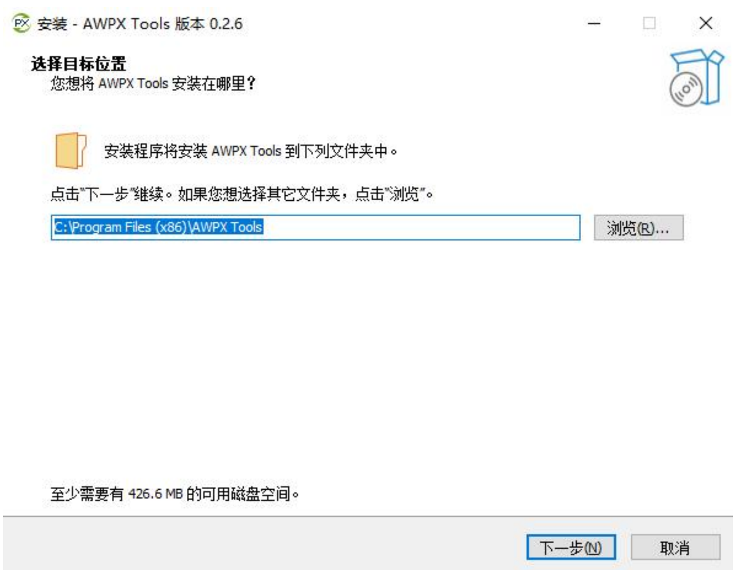

Figure 3.16 Start nstalling AWPX

After clicking on several 'Next' buttons, the installation interface is shown in Figure 3.17.

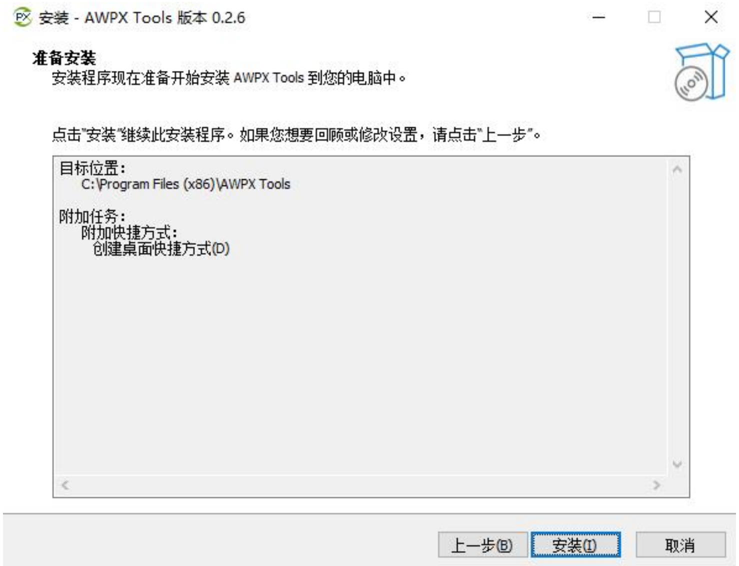

Figure 3.17 AWPX installation interface

Finally, click on 'Install' to officially start the installation of AWPX. Please be patient and wait for the installation to complete.

3.2.2 Introduction to Software Configuration Functions

Run the AWPX software, and the interface is shown in Figure 3.18. After the product is powered on, it is configured by connecting the "NET" marked network port on the PXB-80xx product to the PC host running AWPX software through an Ethernet cable.

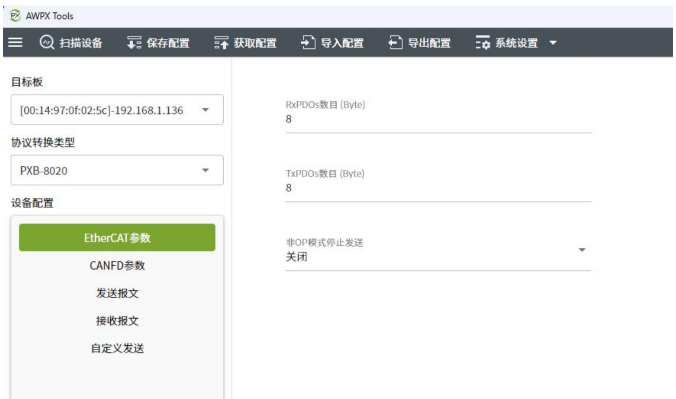

Figure 3.18 AWPX interface

On the left side of the AWPX interface is the device information of the PXB series products, including "Target Board", "Protocol Conversion Type", and "Device Configuration". The specific information of configuration options is located on the right side of the "Device Configuration" column.

The top of the interface is the menu bar button, which includes buttons such as 'Scan Device', 'Save Configuration', 'Get Configuration', etc. As shown in Figure 3.19.

Figure 3.19 Menu bar buttons

1. Scanning equipment

Click the 'Scan Devices' button, and the AWPX software will search for all PXB series devices in the current local area network, and display the IP address and firmware version of the target board in the 'Target Board' dropdown menu. After selecting the correct device in the 'Target Board' drop-down box, you can configure the PXB series devices.

2. Save configuration

After modifying any parameter, click the 'Save Configuration' button to transmit the modified configuration to the PXB series protocol converter, making the modified configuration effective. Saving the configuration will restart the device, wait for the restart prompt at the bottom of the software to disappear.

3. Obtain configuration

After selecting the device, click the 'Get Configuration' button to obtain and display the current configuration of the PXB series protocol converter running.

4. Import configuration

After selecting the device, click the 'Import Configuration' button to import the configuration file with the suffix .awp into the AWPX software. After importing the correct configuration, the imported configuration can be modified or directly saved to the device.

Special note: Do not import configurations across versions for use, for example: please export configurations from devices with firmware version 1.1.7.

5. Export configuration

After selecting the device, click the 'Export Configuration' button to export the current configuration parameters as a configuration file with the. awp suffix. So that the next time you use AWPX, you can quickly import and configure the locally saved configuration file by clicking the 'Import Configuration' button.

Special note: The exported configuration only supports devices with the same firmware version, such as those with firmware version 1.1.7.The configuration exported by the device only supports devices with firmware version 1.1.7.

6. System settings

The 'System Settings' button includes three sub options: 'Network Settings', 'Firmware Upgrade', and 'About'. About it is to explain the version information related to AWPX configuration software. Next, we will focus on describing two functions: network settings and firmware upgrade.

Network settings: The default IP address for PXB-80 series product devices is "192.168.1.136". If you need to change network parameters such as IP address, you can click the 'System Settings' button at the top of the software interface, and then click the 'Network Settings' button in the pop-up menu to perform network settings, as shown in Figure 3.20.

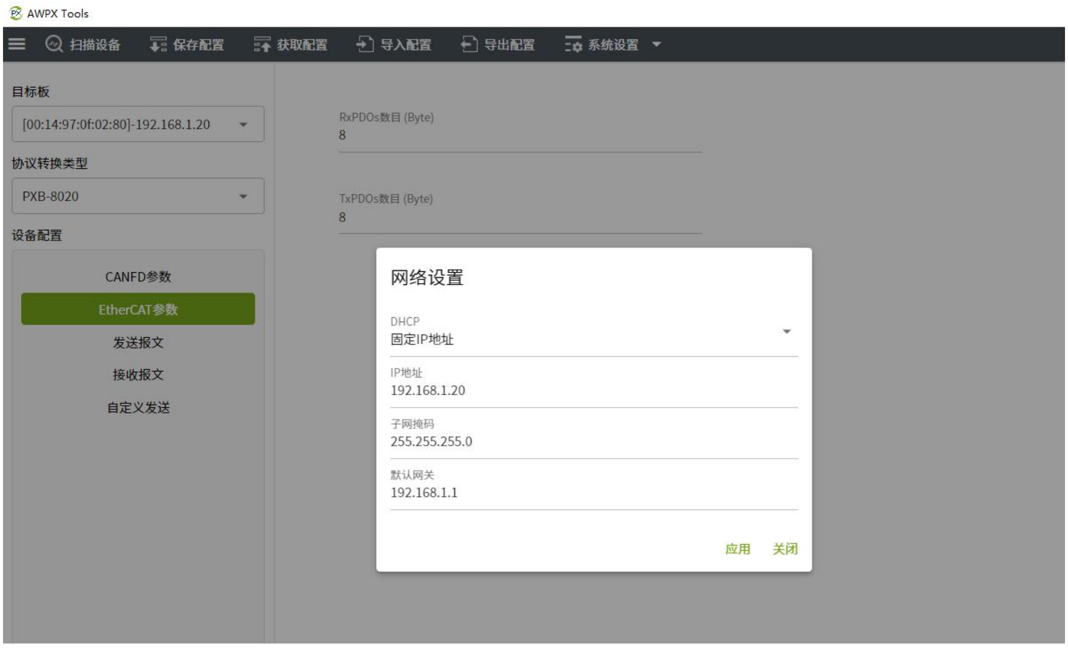

Figure 3.20 Network settings

Firmware Upgrade: Click on 'System Settings' and select 'Firmware Upgrade' from the pop-up menu. AWPX will display the upgrade interface, as shown in Figure 3.21.

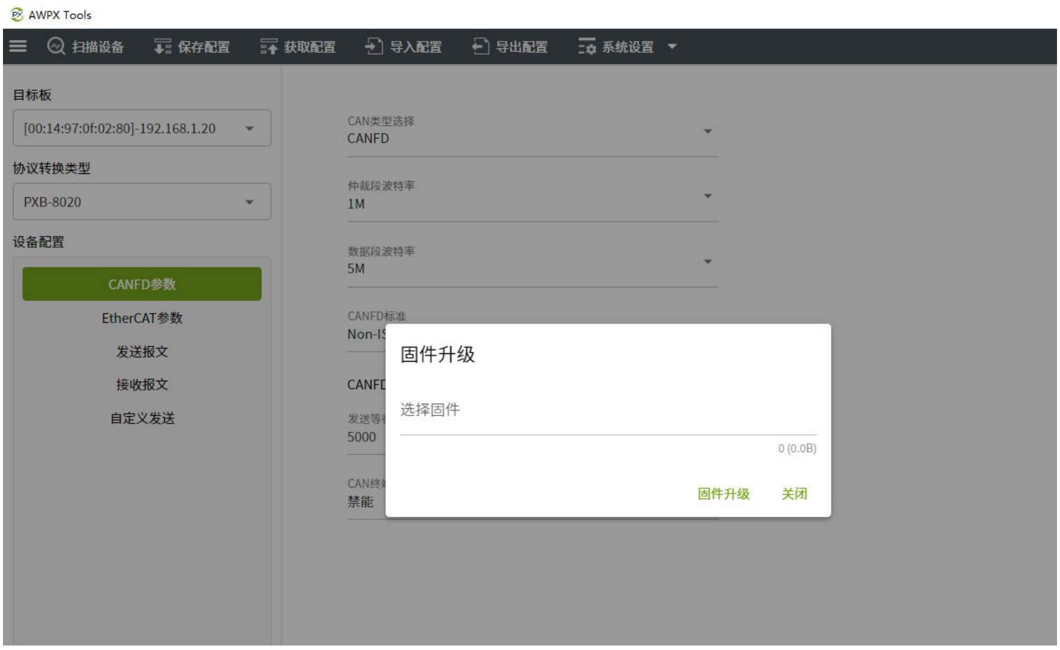

Figure 3.21 Firmware upgrade

Click on 'Select Firmware' in the selection box, choose the firmware (bin file) that needs to be upgraded, and click on 'Firmware Upgrade', AWPX will prompt that firmware is being downloaded to the device, and the entire upgrade process will take about 2 minutes.

After about 1 minute, AWPX will prompt that the device firmware download is successful and the device will automatically restart. Please wait patiently for about 1 minute. During this restart process, do not disconnect the device power. After upgrading the firmware, you need to click the 'Scan Device' button again to rescan and select the device for configuration.

3.3 Equipment Configuration

The main steps for configuring devices using AWPX configuration software are:

Step one, click on 'Scan Devices' and select the correct device. If the device is not scanned and selected, it will not be possible to Configure the PXB series protocol converter accordingly;

Step two, configure parameters. You can configure the parameters in the "Device Configuration" column and network parameters according to your needs;

Step 3: After completing the parameter configuration, click 'Save Configuration' to save the configured parameters to the device and wait for the device to restart just complete it.

3.3.1 EtherCAT Parameter Configuration

Click on 'EtherCAT Parameters' in the 'Device Configuration' column of the AWPX software to configure EtherCAT parameters, including RxPDOs size, TxPDOs size, and stop transmitting in non OP mode. The range of RxPDOs and TxPDOs parameter settings is 1 to 254. The configuration interface is shown in the figure. After successful configuration, the effect scanned by EtherCAT Master software TwinCAT3 is shown in Figure 3.22.

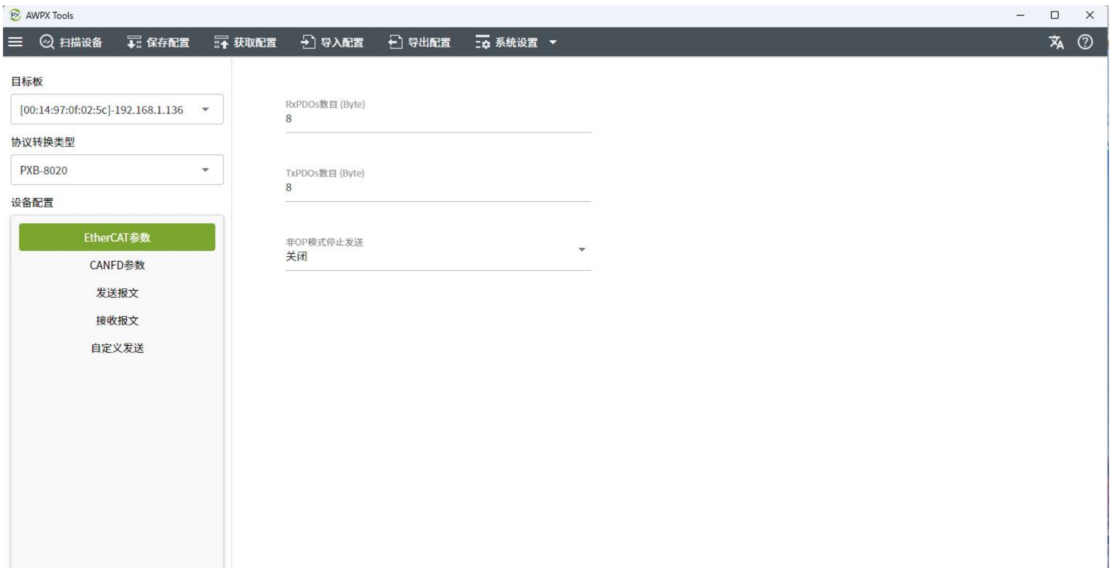

Figure 3.22 EtherCAT parameter settings

Note：If TwinCAT3 and PXB-80XX devices are in data exchange and need to use AWPX to configure the EtherCAT parameters of PXB-80XX devices, first operate on TwinCAT3 to push the Slave device out of the OP state, and then use AWPX to configure and issue.

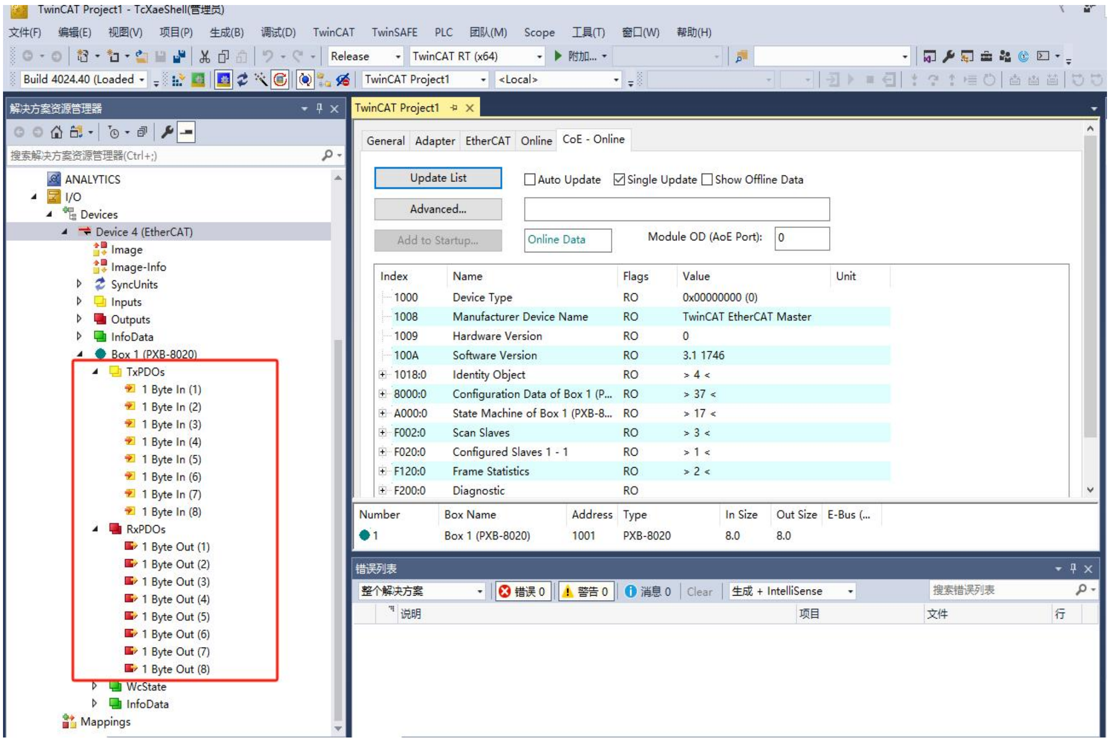

Figure 3.23 TwinCAT3 scanning rendering

3.3.2 PXB-8020 Parameter Configuration

1. CAN FD parameter configuration

PXB-8020 can transmit CAN or CAN FD packets in any working mode. Click on 'CAN FD Parameters' in the 'Device Configuration' column of AWPX software to configure CAN FD parameters, as shown in Figure 3.24.

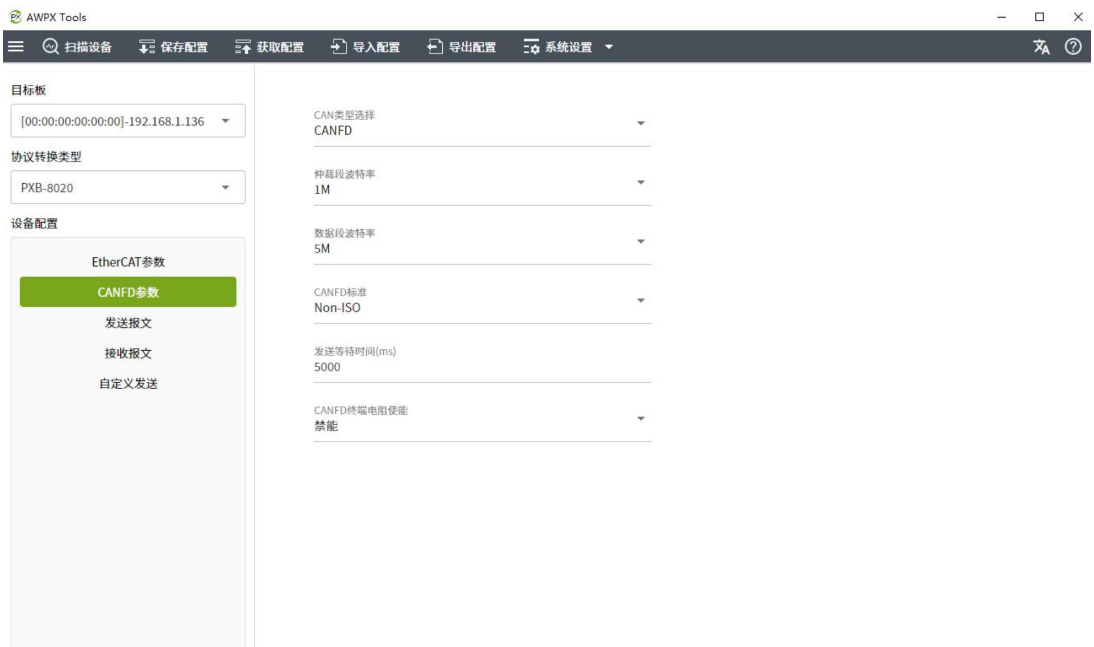

Figure 3.24 CAN FD parameter interface

The CAN FD parameters are described in Table 3.1.

Table 3.1 Description of CAN FD Parameters

Parameter	Description
Type	The type selection can be CAN or CAN FD
Arbitration Rate	Set the baud rate of the arbitration segment for CAN or CAN FD packets
Data Rate	The type is valid for CAN FD, set the data segment baud rate for CAN FD packets
Standard	The type is CAN FD, which is valid. Set the standard used for CAN FD packets
Transmit Wait Time	Used to set how long to wait for PXB-8020 to start transmitting packets after power on initialization is completed
Terminal resistance	Set CAN terminal resistance to enable or disable

2. Transmitting packet configuration

PXB-8020 can transmit CAN, CAN FD, and CANFD acceleration packets in any working mode, depending on the CAN type option selected in the 'Transmit Packets' section.

Click on 'Transmit Packet' in the 'Device Configuration' column of AWPX software to configure the parameters for transmitting packets. The packet transmitting interface is shown in Figure 3.25.

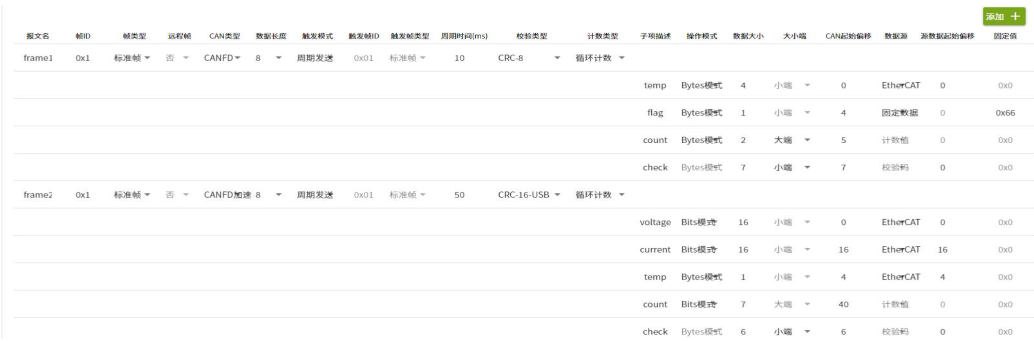

Figure 3.25 Packet transmitting interface

Add up to 128 sent packets by clicking the 'Add+' button in the upper right corner of the interface.

Click the '+Add Data' button to add variables for mapping data, with a maximum of 64 variables added per packet. Can be accessed through the interface

The 'Delete' button on the right deletes the corresponding packet and variable. The parameter description of the packet transmitting interface is shown in Table 3.2.

Table 3.2 Parameter description of packet transmitting interface

Parameter	Description
Packet Name	The name of the packet
Frame ID	Set the frame ID to be sent, which can be decimal or hexadecimal (starting with 0x)
Frame type	Set as standard frame or extended frame
Remote frame	Whether the sent frame is a remote frame, this option is invalid when the CAN type is selected as CAN FD

Continued

CAN type	The types of frames sent are CAN, CANFD, and CANFD acceleration options
Data length	The length of the transmission frame data segment is up to 8 bytes for CAN frames, and up to 64 bytes for CAN FD and CANFD acceleration frames
Trigger mode	Trigger PXB-8020 to transmit CAN packet mode
Trigger ID	The CAN packet ID that triggers PXB-8020 to transmit CAN packets is valid when the trigger mode is selected as 'Frame ID trigger'. Can be decimal or hexadecimal (starting with 0x)
Frame type	The CAN packet type that triggers PXB-8020 to transmit CAN packets is valid when the trigger mode is selected as 'Frame ID trigger'
Cycle time	The time interval for transmitting CAN packets periodically. When the triggering mode is' periodic transmitting ', this parameter is the cycle time; When the triggering method is 'change transmitting' , this time is the cycle for checking data changes; When the triggering mode is 'Single transmit', this time is the waiting time for that single transmit
Check type	Select the verification method required for transmitting the frame
Counting type	Select the counting method required for transmitting the frame
Description	Description of sub item
Operation mode	Choose whether the minimum unit for sub item operations is bits or bytes
Data size	When the data source is EtherCAT, fixed value, count value, the sub item operation size of the data size table, and when the data source is checksum code, the range of the data size table verification.
Byte order	Select data transfer order mode
CAN offset	Select which bit or byte of the CAN packet data segment to start from, and map the data content corresponding to the data source to the selected CAN packet offset.
Data source	Source of CAN packet data segment
Src offset	Select which bit or byte of the source data segment to start from, and map the data content corresponding to the starting offset of the source data to the CAN packet. When the data source bit checksum is used, the starting offset of the source data represents the starting offset of the checksum.
Fixed value	Customize the numerical value of the CAN packet data segment, which can be decimal or hexadecimal (starting with 0x), and the data source is fixed data that takes effect

The data segments for transmitting CAN packets can include custom values, EtherCAT offset values, checksum values, and count values.

Custom Value: After selecting 'Fixed Data' as the 'Data Source' option, set the 'Data Value' option.

EtherCAT Offset value: After selecting 'EtherCAT' as the 'Data source' option, set the 'Src offset'.

Verification code value: When the 'Check Type' is not "None", the 'Data Source' option can be selected as 'Check Code'. In the check code field, the 'CAN offset' can be set to determine the final position of the check value in the CAN packet, the 'Src offset' can be set to determine the starting position of the check CAN frame, and the 'Data Size' can be set to determine the check range (Src offset+data size). In each frame of the transmit packet, only one check field can be added, and the supported check types are shown in Table 3.3:

Table 3.3 Description of check Type Parameters

Check type	Description
CheckSum8	XOR 0xff (starting offset byte+...+ending offset byte)
CheckSum16	XOR 0xffff (starting offset byte+...+ending offset byte)
CheckSum32	XOR 0xffffff (starting offset byte+...+ending offset byte)
CRC-8	The main parameter values are: polynomital: 0x07, initial value: 0x00, XOR value of result: 0x00, input inversion: false, output inversion: false.
CRC-8-ITU	The main parameter values are: polynomital: 0x07, initial value: 0x00, XOR value of result: 0x00, input inversion: false, output inversion: false.
CRC-8-ROHC	The main parameter values are: polynomital: 0x07, initial value: 0xff, XOR value of result: 0x00, input inversion: true, output inversion: true.
CRC-8-MAXIM	The main parameter values are: polynomial: 0x31, initial value: 0x00, XOR value of result: 0x00, input inversion: true, output inversion: true;
CRC-16-IBM	The main parameter values are: polynomial: 0x8005, initial value: 0x0000, XOR value of result: 0x0000, input inversion: true, output inversion: true;
CRC-16-MAXIM	The main parameter values are: polynomial: 0x8005, initial value: 0x0000, XOR value of result: 0xffff, input inversion: true, output inversion: true;
CRC-16-USB	The main parameter values are: polynomial: 0x8005, initial value: 0xffff, XOR value of result: 0xffff, input inversion: true, output inversion: true;
CRC-16-MODBUS	The main parameter values are: polynomial: 0x8005, initial value: 0xffff, XOR value of result: 0x0000, input inversion: true, output inversion: true;
CRC-16-CCITT	The main parameter values are: polynomial: 0x1021, initial value: 0x0000, XOR value of result: 0x0000, input inversion: true, output inversion: true;
CRC-16-CCITT-FALSE	The main parameter values are: polynomial: 0x1021, initial value: 0xffff, XOR value of result: 0x0000, input inversion: false, output inversion: false;
CRC-16-X25	The main parameter values are: polynomial: 0x1021, initial value: 0xffff, XOR value of result: 0xffff, input inversion: true, output inversion: true;
CRC-16-XMODEM	The main parameter values are: polynomial: 0x1021, initial value: 0x0000, XOR value of result: 0x0000, input inversion: false, output inversion: false;
CRC-16-XMODEM2	The main parameter values are: polynomial: 0x8408, initial value: 0x0000, XOR value of result: 0x0000, input inversion: true, output inversion: true;
CRC-16-DNP	The main parameter values are: polynomial: 0x3d65, initial value: 0x0000, XOR value of result: 0xffff, input inversion: true, output inversion: true;
CRC-32	The main parameter values are: polynomial: 0x04c11db7, initial value: 0xffffff, XOR value of result: 0xffffff, input inversion: true, output inversion: true;
CRC-32-C	The main parameter values are: polynomial: 0x1edc6f41, initial value: 0xffffff, XOR value of result: 0xffffff, input inversion: true, output inversion: true;
CRC-32-KOOPMAN	The main parameter values are: polynomial: 0x741b8cd7, initial value: 0xffffff, XOR value of result: 0xffffff, input inversion: true, output inversion: true;

Continued

CRC-32-MPEG2	The main parameter values are: polynomial: 0x04c11db7, initial value: 0xffffff, XOR value of result: 0x0000000, input inversion: false, output inversion: false;

Count value: When the 'Count type' is not none, the 'Data source' option can be selected as 'Count value'. In the count field, the position of the count value can be determined by setting the 'offset'. In each frame of the sent packet, only one count field can be added, and the supported count types are as follows:

Table 3.4 Description of Count Type Parameter

Counting type	Description
Cycle counting	When the operation type is bits, the counting range is 0~1... 1 (B) (data size is 1), and when the operation type is bytes, the counting range is 0~0xF F (2 * data size of F), the first frame starts from 0, the count value increases by one for each frame sent, and the counting cycle repeats

The modes for triggering PXB-8020 to transmit CAN packets include periodic transmitting, variable transmitting, single transmitting, and frame ID triggering.

Periodic transmitting: When this trigger mode is set, PXB-8020 will cyclically transmit CAN packets based on the cycle time.

Change transmitting: When this trigger mode is set, when the EtherCAT mapping address value corresponding to a variable changes, trigger PXB-8020 to transmit a CAN packet.

Single transmit: When setting this trigger mode, only transmit the CAN packet once.

Frame ID trigger: When this trigger mode is set, PXB-8020 will trigger to transmit a CAN packet when it receives a CAN packet that matches the set 'Trigger ID' and 'Trigger Frame Type'.

Example of packet transmitting configuration:

Set 'Frame ID' to 0x01, 'Frame Type' to standard frame, 'Data Length' to 8, 'Trigger Mode' to periodic transmission, 'Cycle Time' to 1000ms, 'Check Type' to None, and 'Count Type' to None. And add a sub item to describe its settings: 'Operation mode' is Byte mode, 'Data size' is 4, 'CAN offset' is 4, the data source is EtherCAT, and the starting offset of the source data is 2.

Then PXB-8020 will read 4 bytes starting from the second RxPDO of RxPDOs. Then shift these 4 bytes one by one to start from the 4th byte of the CAN packet data segment, and then transmit the CAN packet out at a cycle of 1000ms.

If the EtherCAT Slave mapping address data is: 1 Byte Out (3): 0x11, 1 Byte Out (4): 0x22, 1 Byte Out (5): 0x33, 1 Byte Out (6): 0x44, then every 1000ms, PXB-8020 will transmit a CAN standard frame with ID 0x01: 00 00 00 11 22 33 44 (hexadecimal).

3. Receive packet configuration

PXB-8020 can receive CAN or CAN FD packets in any working mode. Whether the received packet is a CAN frame or a CAN FD frame depends on the setting of the CAN type selection in the CAN FD parameters interface.

Click the 'Receive Packet' button in the 'Device Configuration' column of AWPX software to configure the parameters for receiving packets, as shown in Figure 3.26.

Figure 3.26 Receiving packet interface

Add the packets to be received by clicking the 'Add+' button in the upper right corner of the interface, up to a maximum of 128 received packets can be added. Click the '+Add Data' button to add variables for mapping data, with a maximum of 64 variables added per packet. To delete a received packet or variable, you can use the 'Delete' button on the right side of the interface. The parameter description of the packet receiving interface is shown in Table 3.5, and the CAN starting offset when the frame ID is set to -1 is shown in

Table 3.6.

Table 3.5 Parameters Description of Receiving Packets Interface

Parameter	Description
Packet Name	The name of the packet
Frame ID	Set the frame ID for receiving packets, which can be decimal or hexadecimal (starting with 0x); When the frame ID is set to -1, it means that all frames that meet the frame type can be received and mapped
Frame type	Set whether the received packet is a standard frame or an extended frame
Variable Name	The name of the variable
Operation mode	Choose whether the minimum unit for sub item operations is bits or bytes
Data size	Fill in the sub item operation size
CAN offset	Select which bit or byte of the CAN packet data segment to start from, and map the corresponding data content of the CAN packet to the EtheCAT starting offset; When the frame ID is set to -1, the CAN starting offset setting can represent the CAN frame ID, frame information, and frame length.
ECAT offset	Select which bit or byte of the RxPDOS data segment of EtheCAT to start from, and map the data content corresponding to the CAN packet to the selected EtheCAT starting offset.

Table 3.6 Description of CAN Initial Offset When Frame ID is Set to -1

CAN offset	Map address PDO	Description
-1	4	Received CAN frame ID
-2	1	Received CAN frame information Bit0: 0：CAN，1：CANFD Bit1: 0: CANFD acceleration turned off, 1: CANFD acceleration turned on Bit2: 0: Data frame, 1: Remote frame bit Bit3: 0: Standard frame, 1: Extended frame bits Bit4~Bit7: Reserved, default is 0
-3	1	Received CAN frame data length
>=0	n	Received CAN frame data content

PXB-8020 will write the content of the received CAN packet data segment into the TxPDOs mapping address of EtherCAT.

Example of receiving packet configuration:

Set 'Frame ID' to 0x02, 'Frame Type' to standard frame, 'Operation Mode' to bytes mode, 'Data Size' to 6, 'CAN Offset' to 2, and 'ECAT Offset' to 0.

When PXB-8020 receives a CAN standard frame with frame ID 0x02 and frame data of 11 22 33 44 55 66 77 88 (hexadecimal), PXB-8020 will start from the second byte of the CAN frame data segment and sequentially write the subsequent 6 bytes of the CAN packet data segment into the 0th TxPDO of EtherCAT. The value of 1 Byte In (1) in the TxPDOs of EtherCAT Slave is 0x33, and the value of 1 Byte In (2) is 0x44, corresponding to the value of 1 Byte In (6) being 0x88.

4. Customize transmitting configuration

Click on 'Custom transmit' in the 'Device Configuration' column of AWPX software to configure the parameters for custom transmitting packets, including CAN, CANFD, and CANFD acceleration packets. The custom transmitting interface is shown in Figure 3.27

Figure 3.27 Custom transmitting interface

The 'EtherCAT Offset' option is used to configure the data source for custom packet transmitting. The data type for custom packet transmitting comes from the CANFD parameter value, and its data source is the RxPDOs starting offset of EtherCAT.

Click the 'Custom transmit' button to enable custom transmitting,Then, on the EtherCAT Master,sequentially fill in the data format shown in the Table 3.7 with the starting offset of RxPDO in the PXB-8020 device.

Table 3.7 CAN/CAN FD Data area format

field	Map address PDO	Description
Transaction Number	1	A value greater than 0 indicates that the following data area is valid, and this sequence number needs to be incremented every time an update is sent. When this value is at 255, it can return to 1
CANID	4	The frame ID calculates the 4 bytes after the transaction sequence number using the Byte1<<24 | Byte2<<16 | Byte3<<8 | Byte4 method.
Frame information	1	Bit0: 0：CAN，1：CANFD Bit1: 0: CANFD acceleration turned off, 1: CANFD acceleration turned on Bit2: 0: Data frame, 1: Remote frame bit Bit3: 0: Standard frame, 1: Extended frame bits Bit4~Bit7: Reserved, default is 0
Frame length	1	The length options range from 0 to 8, 12, 16, 20, 24, 32, 48, 64 (0 to 0x40)
CAN/CAN FD data	0~64	Fill in the value of the corresponding address based on the frame length (note: when exceeding the maximum mapping address, 0 is added by default)

Custom transmitting example:

If custom transmitting is enabled in AWPX software, the CAN type in the CANFD parameter is set to CANFD, and the 'EtherCAT offset'for custom transmitting is set to 29.

Then, in the EtherCAT Master, sequentially fill in 0x01, 0x00, 0x00, 0x01, 0x23, 0x03, 0x08, 0x11, 0x22, 0x33, 0x44, 0x55, 0x66, 0x77, and 0x88 in the 1 Byte Out (30)~1 Byte Out (44) of the RxPDOs in the PXB-8020 device. When the value of 1 Byte Out (30) increases significantly, PXB-8020 transmits a frame with the ID of 0x0123, frame length 8, frame type standard frame, frame content 11 22 33 44 55 66 77 88 (hexadecimal) CANFD accelerate frames.

3.3.3 PXB-8021M Parameter Configuration

1. CANopen Master parameter configuration

Click on 'CANopen Master setting' in the'Device Configuration'column of the AWPX software to configure the relevant parameters of the CANopen Master . The configuration interface is shown in Figure 3.28.

Figure 3.28 CANopen Master setting interface

The CANopen Master parameter configuration instructions are shown inTable 3.8.

Table 3.8 CANopen Master parameter configuration description

Parameter	Description
Node ID	Set the Master node ID (1~127), default is 127
Baud rate	Set the baud rate of the CAN bus from 10Kbps to 1Mbps, with a default of 1Mbps
synchronization mechanism	Whether to activate the Master synchronization mechanism, disabled by default
Synchronize packet ID	Default is 0x00000080 (hexadecimal needs to be prefixed with 0x, otherwise it is decimal)
Synchronization period	Default is 1000 (0~65535)ms
Delay start time	Is the Master delayed in starting(0~2147483647)ms
Terminal resistance	Enable terminal resistor, default enabled

2. CANopenSlave Configuration

In the 'Device Configuration' column of AWPX software, click on 'CANopen Slave List' and 'Add Slave+' in order to configure CANopen Slave parameters. The configuration interface is shown in Figure 3.29.

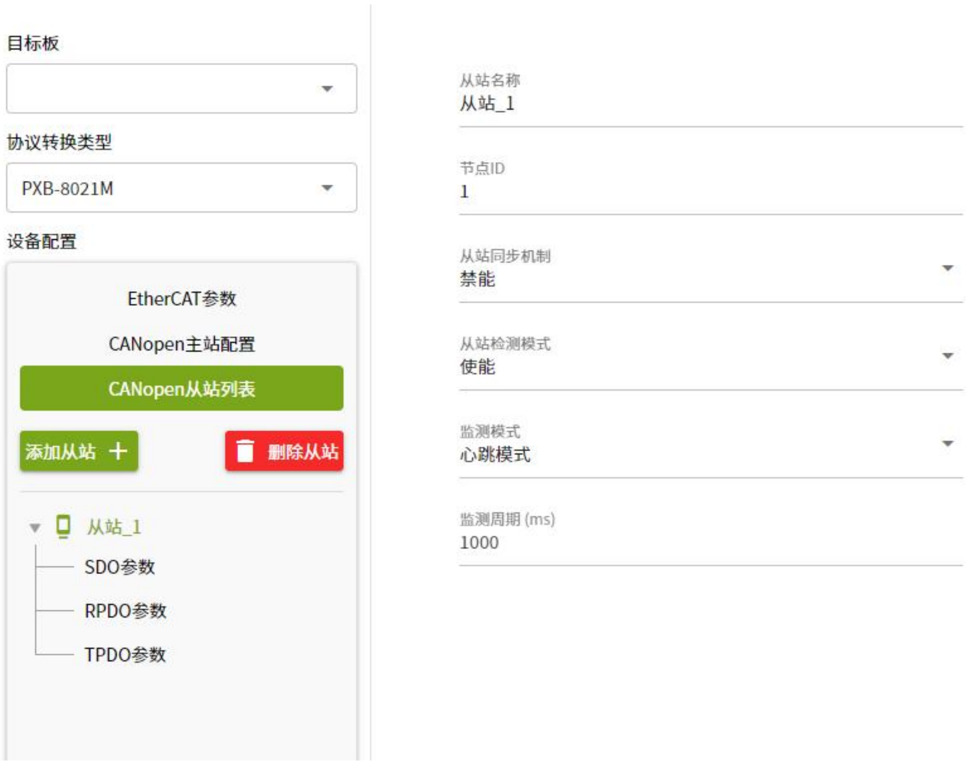

Figure 3.29 CANopen Slave list

The CANopen Slave parameter configuration instructions are shown in Table 3.9.

Table 3.9 CANopen Slave parameter configuration description

Parameter	Description
Slave Name	Slave name, user-defined
Node ID	Slave node ID (1~127)
Slave synchronization mechanism	Whether to synchronize the synchronization packet information of the Master, default not to synchronize
Slave monitoring mode	Whether to issue monitoring configuration to the current Slave
Monitoring mode	Slave node online monitoring mode, default to heartbeat protocol
Monitoring cycle	Default 1000 (0~65535)ms
Monitoring factors	The number of retransmissions is only valid when the detection mode is node guarding

3. CANopen Slave packet configuration

Click on 'CANopen Slave List' in the 'Device Configuration' column of AWPX software, select the corresponding Slave, and configure the CANopen Slave packet parameters. There are three types of Slave packet types: SDO, RPDO, and TPDO, and their configuration interfaces correspond to Figure 3.30, Figure 3.31, Figure 3.32 , respectively.

Figure 3.30 SDO Packet Configuration

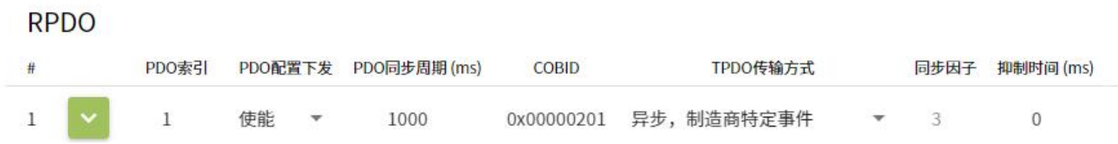

Figure 3.31 RPDO Packet Configuration for Slave

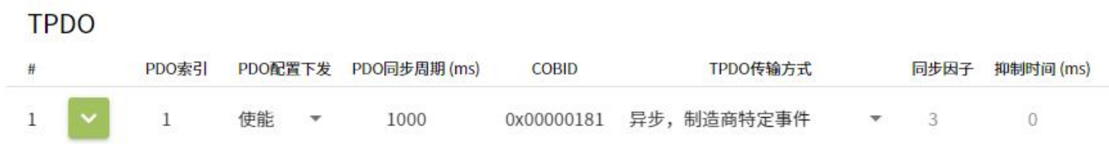

Figure 3.32 TPDO Packet Configuration for Slave

The configuration instructions for CANopen Slave packet parameters correspond to Table 3.10, Table 3.11 respectively.

Table 3.10 SDO Packet Configuration Description

Parameter	Description
Description	User defined description content for mnemonic purposes
Main index	Slave dictionary main index
Sub index	Slave dictionary sub index
PDO fast transfer mode	Upload or download
Data size	The maximum data size for this SDO fast transfer is four bytes
Data value	The data issued to the Slave (hexadecimal needs to be prefixed with 0x, otherwise it is decimal) is transmitted in small endian byte order in the CANopen network

Table 3.11 PDO Configuration Description

Parameter	Description
Pdo configuration distribution	Should the PDO configuration information of the interface be issued to the Slave
PDO index	The communication object of PDO in the Slave dictionary is offset based on the main index of 0x1400 or 0x1800. The interface configuration value will be reduced by 1 during actual transmission. When the RPDO index is configured as 1, the actual modified parameters are: 0x1400 + pdo_index - 1 = 0x1400 When the TPDO index is configured as 1, the actual modified parameters are: 0x1800 + pdo_index - 1 = 0x1800
COBID	COBID of the Slave (hexadecimal needs to be prefixed with 0x, otherwise it is decimal)
PDO synchronization period	PDO data refresh cycle (0~65535) ms, default is 1000ms
Transmission mode	Default is asynchronous transmission
Inhibition time	Minimum interval between transmitting two PDO packets (0-6500) ms
Synchronization factor	Only effective when the transmission method is configured as synchronous cycle (1-240)
Mapping description	Used to describe the mapping function, user-defined
Operation mode	Type: bits operation, byte operation
Operating data size	When operating in bits mode, the size range is 1-64 bits, When operating in byte mode, the size range is 1-8 bytes
ECAT byte offset	ECAT channel byte offset, size range 0~254
ECAT starting bit position	Based on a bit offset of a certain byte, the size range is 0-7
CAN frame byte offset	CAN frame byte offset, size range 0-7
Starting bit position of CAN frame	Based on the bit offset of a certain byte in the packet, the size range is 0-7
Fixed value	Fixed numerical values for mapping(Hexadecimal needs to be prefixed with 0x, otherwise it is decimal), Transmitted in small endian byte order in CANopen network

3.3.4 PXB-8022 Parameter Configuration

1. DeviceNet parameters

Click on 'DeviceNet Parameters' in the 'Device Configuration' column of AWPX software to configure the DeviceNet parameters of the Slave. The configuration interface is shown in Figure 3.33.

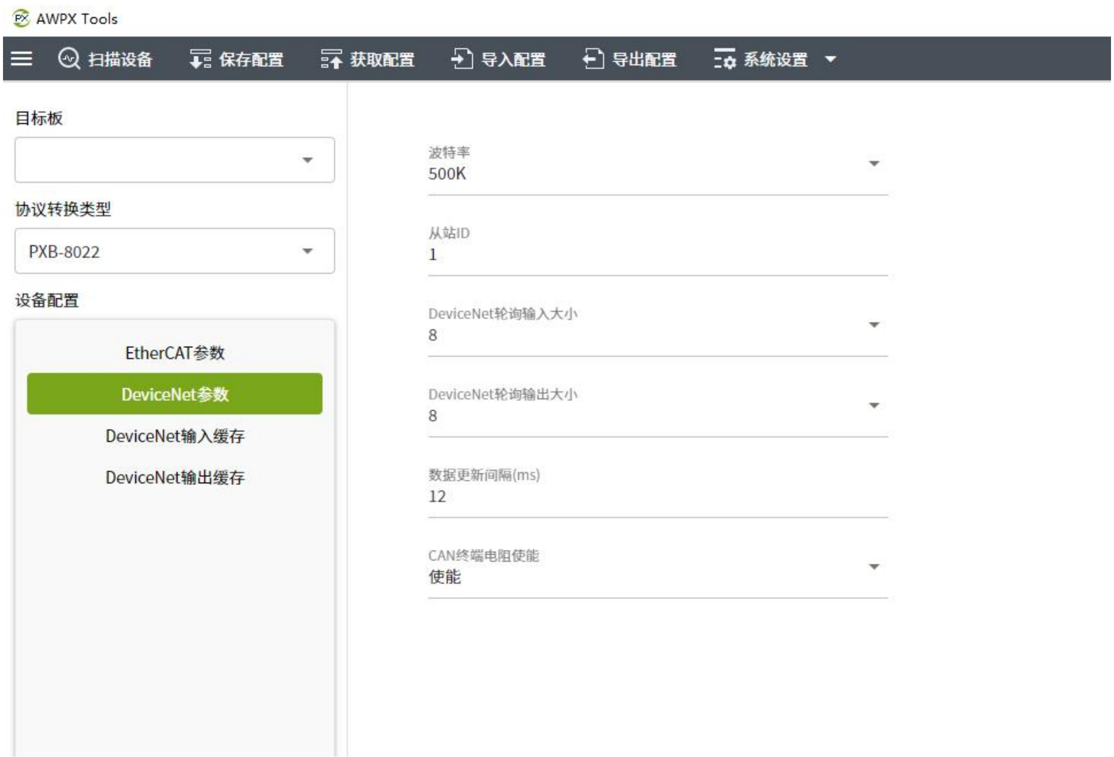

Figure 3.33 DeviceNet parameter interface

The parameter description of DeviceNet is shown in Table 3.12.

Table 3.12 DeviceNet Parameters Description

Parameter	Description
Baud rate	Set the CAN baud rate of PXB-8022
Slave ID	Set the ID of PXB-8022 as a DeviceNet Slave
DeviceNet poll input size	Set the maximum polling data size that PXB-8022 can receive at a time, in bytes
DeviceNet poll output size	Set the maximum polling data size that PXB-8022 can transmit at a time, in bytes
Data update interval	The synchronization interval between EtherCAT data and DeviceNet data. The smaller the value, the better the real-time performance.The unit is ms.
CAN terminating resistor	Enable or disable the terminal resistance of CAN interface

2. DeviceNet input buffer

The DeviceNet input buffer is the polling input data transmitted from the DeviceNet Master to PXB-8022. PXB-8022 then maps the DeviceNet input buffer to TxPDOs, and finally, PXB-8022 transmits TxPDOs to the EtherCAT Master . The DeviceNet input buffer configuration interface is shown in Figure 3.34.

Figure 3.34 DeviceNet input buffer interface

Add a mapping entry by clicking the 'Add+' button in the upper right corner of the interface, and then edit the mapping parameters. On the far right side of the entry, click the 'Delete' button to delete the mapping entry. The maximum number of mapping entries that can be added is 128. The parameter description for DeviceNet input buffer configuration is shown in Table 3.13.

Table 3.13 Description of DeviceNet input buffer parameters

Parameter	Description
Var Name	The name of this mapping entry can be used for mnemonic purposes
Operate Unit	The size of the mapped data. BYTE:1 byte, WORD:2 bytes, DWORD: 4 bytes, QWORD: 8 bytes
DeviceNet byte offset	Specify which byte of polling input data to map TxPDOs to
DeviceNet byte bit offset	The bit offset after byte offset is valid when the operation unit is BIT. Specify the byte and bit of the polling input data to be mapped to TxPDOs
EtherCAT byte offset	Specify the byte at which polling input data is mapped to TxPDOs
EtherCAT byte bit offset	The bit offset after byte offset is valid when the operation unit is BIT. Specify which byte and bit of TxPDOs to map polling input data to

3. DeviceNet output buffer

The output buffer of DeviceNet is PXB-8022. After receiving the RxPDOs sent by the EtherCAT Master, PXB-8022 maps the RxPDOs to the output buffer. Finally, PXB-8022 transmits the output buffer to the DeviceNet Master. The DeviceNet output buffer configuration interface is shown in Figure3.35.

Figure3.35　DeviceNet output buffer Interface

Add a mapping entry by clicking the 'Add+' button in the upper right corner of the interface, and then edit the mapping parameters. On the far right side of the entry, click the 'Delete' button to delete the mapping entry. The maximum number of mapping entries that can be added is 128. The parameter specifications for DeviceNet output buffer configuration are shown in Table 3.14.

Table 3.14　Description of DeviceNet output buffer parameters

Parameter	Description
Var Name	The name of this mapping entry can be used for mnemonic purposes
Operate Unit	The size of the mapped data. BYTE: 1 byte, WORD: 2 bytes,, DWORD: 4 bytes, QWORD: 8 bytes
DeviceNet byte offset	Specify which byte of polling output data to map RxPDOs to
DeviceNet byte bit offset	The bit offset after byte offset is valid when the operation unit is BIT. Specify the byte and bit of the polling output data to be mapped to RxPDOs
EtherCAT byte offset	Specify which byte of RxPDOs to map to polling output data
EtherCAT byte bit offset	The bit offset after byte offset is valid when the operation unit is BIT. Specify which byte and bit of RxPDOs to map polling output data to

3.3.5 PXB-8022M Parameter Configuration

1. DeviceNet Master parameters

Click on 'DeviceNet Master Parameters' in the 'Device Configuration'column of the AWPX software to configure the relevant parameters of the DeviceNet Master. The configuration interface is shown in Figure 3.36.

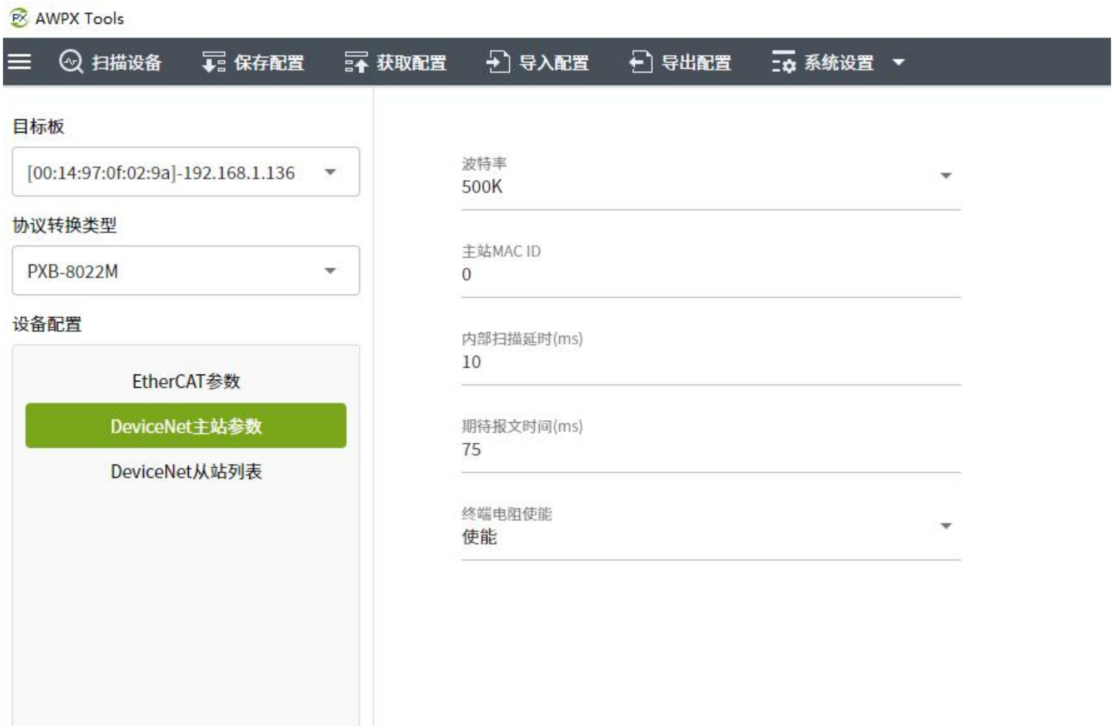

Figure 3.36 DeviceNet Master Parameter Interface

The parameter description of DeviceNet Master is shown in Table 3.15.

Table 3.15 Description of DeviceNet Master parameters

Parameter	Description
Baud Rate	CAN baud rate of PXB-8022M
Master MAC ID	Set the device ID of PXB-8022M as a DeviceNet Master
Interscan Delay	The internal scanning delay is the minimum time allowed for external devices to access the network after continuous I/O scanning by the scanner. If the value is too high, it will cause a longer network scan, which will affect the execution of input and output. If the value is too small, it will slow down the scanner module's response to external devices
Expected packet rate	Expecting packet rate, determining the timeout time for bit gating and polling packet
Terminating Resitor	Enable or disable the terminal resistance of CAN interface

2. DeviceNet Slave list

Click on the 'DeviceNet Slave List' in the 'Device Configuration' column to view the established DeviceNet Slave nodes and add or remove them. The interface is shown in Figure 3.37.

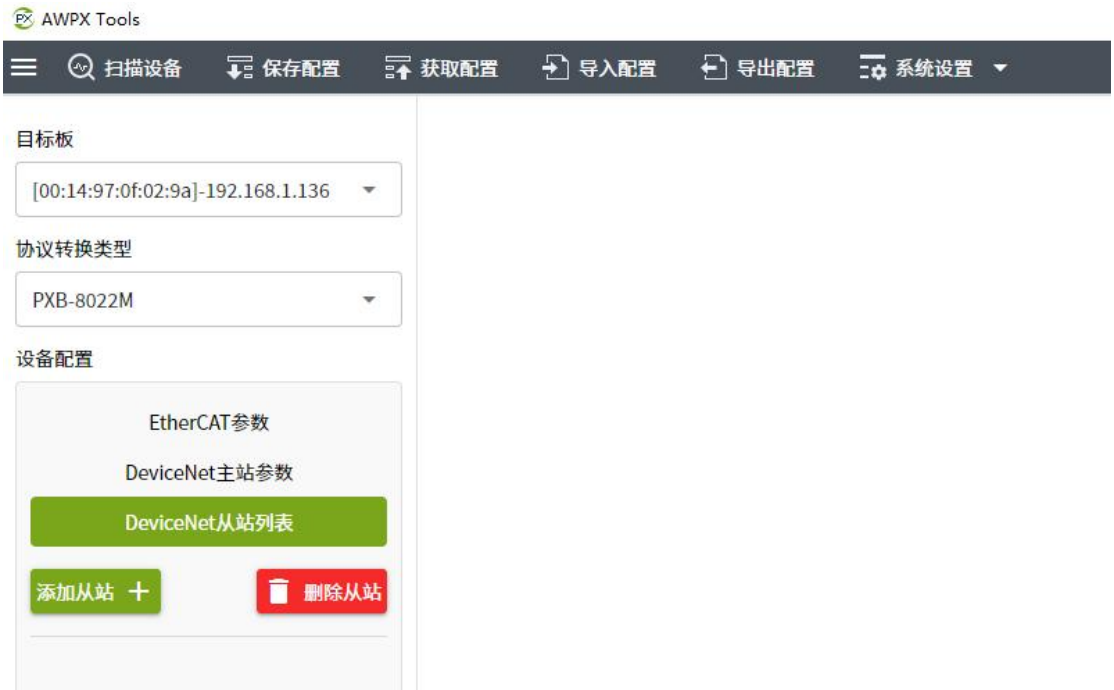

Figure 3.37 DeviceNet Slave list interface

3. DeviceNet Slave parameters

Click on 'Add Slave+' to bring up the DeviceNet Slave Configuration interface, which is used to configure the related attribute information of the DeviceNet Slave that needs to be connected. The interface is shown in Figure 3.38.

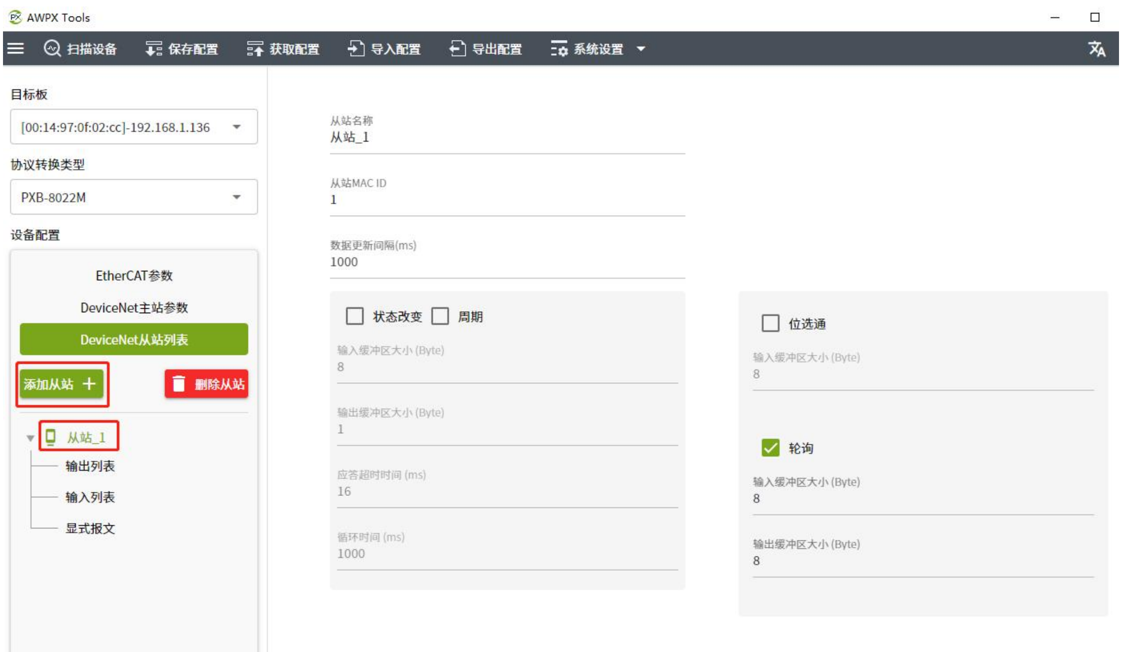

Figure 3.38 DeviceNet Slave parameters interface

On the left side of Figure 3.38 and Figure 3.37, On the left side of Figure 3.38 and Figure 3.37, click on the corresponding Slave name to configure the relevant attribute information of the

DeviceNet Slave.The description of DeviceNet Slave attribute information is shown in Table 3.16.

Table 3.16 Description of DeviceNet Slave parameter

Parameter	Description
Slave Name	The name of this DeviceNet Slave can be changed for mnemonic purposes
Slave MAC ID	The device address of this DeviceNet Slave
Data Update Interval	EtherCAT data and DeviceNet data synchronization interval. The smaller the value, the better the real-time performance

Below the 'Data Update Interval' parameter, you can select the I/O connection type, which includes four types: bit select, polling, state change, and cycle. Each DeviceNet Slave must select at least one I/O connection type and a maximum of three I/O connection types. State change and cycle cannot be selected simultaneously.

1. Bit selection: The input buffer size of DeviceNet Slave needs to be configured, which can be configured to be 1-64Byte;

2. Polling: The input and output buffer sizes of DeviceNet Slave need to be configured, with configurable sizes ranging from 1 to 64 bytes;

3. State change: When the state change type is selected, the interface is shown in Figure 3.39.

Figure 3.39 Change mode of state

The parameter description of this interface is shown in Table 3.17.

Table 3.17 Description of state change mode parameters

Parameter	Description
Input Buffer Size	The maximum size of I/O data that can be input to this DeviceNet Slave at a time, ranging from 1 to 64 bytes
Output Buffer Size	The maximum I/O data size that this DeviceNet Slave can output at a time, ranging from 1 to 64 bytes

Continued

Acknowledgment Timeout	The time from transmitting a packet to receiving a response from the DeviceNet Master when the DeviceNet Slave status changes
Loop Time	The interval between when DeviceNet Slave transmits heartbeat packets. When using the state change mode, the heartbeat packet can query the device status at regular intervals to prevent DeviceNet Slave from disconnecting
Production inhubit time	The time for generating DeviceNet Slave packets should be less than the time for heartbeat packets

4、Cycle: When selecting the cycle type, the interface is shown in Figure 3.40.

Figure 3.40 Cyclic type

The parameter description of this interface is shown in Table 3.18.

Table 3.18 Description of cyclic type parameters

Parameter	Description
Input Buffer Size	The maximum size of I/O data that can be input to this DeviceNet Slave at a time, ranging from 1 to 64 bytes
Output Buffer Size	The maximum I/O data size that this DeviceNet Slave can output at a time, ranging from 1 to 64 bytes
Acknowledgment Timeout	The time from transmitting a packet from the DeviceNet Slave to receiving a response from the DeviceNet Master
Loop Time	The time interval of DeviceNet Slave cyclic communication can reduce unnecessary network traffic

4. Output List

On the 'DeviceNet Slave List' interface, click on the 'Output List' below the Slave name, as shown in Figure 3.41. Click the '▼' button to the left of the Slave name to hide its output, input list, and explicit packets. The output list is a mapping entry list for outputting I/O data to the DeviceNet Master relative to the DeviceNet Master.

Figure 3.41 Output list interface

Add a mapping entry by clicking the 'Add+' button in the upper right corner of the interface, and then edit the mapping parameters. On the far right side of the entry, click the 'Delete' button to delete the mapping entry. The parameter description of the output list is shown in Table 3.19.

Table 3.19 Description of Output List Parameters

Parameter	Description
Var Name	The name of this mapping entry can be used for mnemonic purposes
Operate Unit	The size of the mapped data. BYTE: 1 byte, WORD: 2 bytes, DWORD: 4 bytes, QWORD: 8 bytes
IO Type	Select the selected I/O connection type
DeviceNet byte offset	Specify the byte that maps RxPDOs to I/O output data
DeviceNet byte bit offset	The bit offset after byte offset is valid when the operation unit is BIT. Specify which byte and bit of the I/O output data to map RxPDOs to
EtherCAT byte offset	Specify which byte of RxPDOs to map to I/O output data
EtherCAT byte bit offset	The bit offset after byte offset is valid when the operation unit is BIT. Specify which byte and bit of RxPDOs to map to I/O output data

5. Input list

In the 'DeviceNet Slave List' interface, click on 'Input List' below the Slave name, as shown in Figure 3.42. Click the '▼' button to the left of the Slave name to hide its output, input list, and explicit packets. The input list is a mapping entry list for the DeviceNet Master to receive input I/O data, relative to the DeviceNet Master.

Figure 3.42 Input list interface

Add a mapping entry by clicking the 'Add+' button in the upper right corner of the interface, and then edit the mapping parameters. On the far right side of the entry, click the 'Delete' button to delete the mapping entry. The input list parameters are described in Table 3.20.

Table 3.20 Description of input list parameter

Parameter	Description
Var Name	The name of this mapping entry can be used for mnemonic purposes
Operate Unit	The size of the mapped data. BYTE: 1 byte, WORD: 2 bytes, DWORD: 4 bytes, QWORD: 8 bytes
IO Type	Select the selected I/O connection type
DeviceNet byte offset	Specify which byte of I/O input data to map to TxPDOs
DeviceNet byte bit offset	The bit offset after byte offset is valid when the operation unit is BIT. Specify the byte and bit of the I/O input data to be mapped to TxPDOs
EtherCAT byte offset	Specify the byte at which I/O input data is mapped to TxPDOs
EtherCAT byte bit offset	The bit offset after byte offset is valid when the operation unit is BIT. Specify which byte and bit of TxPDOs to map I/O input data to

6. Msg List

In the 'DeviceNet Slave List' interface, click on 'Msg List' below the Slave name, as shown in Figure 3.43. Click the '▼' button to the left of the Slave name to hide its output, input list, and explicit packets. After establishing an explicit information connection, configure various parameters and exchange information for DeviceNet Slave through this interface by setting explicit packets.

Figure 3.43　Msg List interface

Add a mapping entry by clicking the 'Add+' button in the upper right corner of the interface, and then edit the mapping parameters. On the far right side of the entry, click the 'Delete' button to delete the mapping entry. The input list parameters are described inTable 3.21.

Table 3.21　Description of msg list parameters

Parameter	Description
Packet Name	The name of this packet can be used for mnemonic purposes
Service code	Specify the service code corresponding to the explicit packet
Class ID	Specify the ID of the class corresponding to the attribute obtained through explicit packet configuration
Instance ID	Specify the ID of the instance corresponding to the attribute obtained through explicit packet configuration
Req Data Src	Specify the source of the service data that transmits the explicit packet, which can be either 'Fixed Data' or 'Target Protocol'. If you select 'Fixed Data', the source will be the data filled in from 'Data Fixed'. If you choose 'Target Protocol', the source will be RxPDOs
Req Data Len	The unit is Byte. Specifying the length of the service data that transmits the explicit packet
Req Data offset	The unit is Byte, which specifies the starting byte of RxPDOs and maps them to the service data that transmits explicit packets. When the 'Req Data Src' is selected as the 'Target Protocol', this option is valid
Data Fixed	When the 'Req Data Src' is selected as 'Fixed Data', fill in the service data content for transmitting the explicit packet here
Ack Data Dst	Specify the method for processing DeviceNet Slave response packets after transmitting explicit packets, which can be either 'Drop Data' or 'Target Protocol'. When selecting 'Drop Data', the data of the response packet will not be processed. When selecting 'Target Protocol', map the return data of DeviceNet Slave response packet to TxPDOs
Ack Data Len	The unit is Byte, which actually maps the return data of DeviceNet Slave to the length of TxPDOs. If the actual length of the response data is less than the length filled in here, the mapping will be based on the actual length of the response data. If the actual length of the response data is greater than the length filled in here, then map according to the length of the response data here
Ack Data Offset	The unit is Byte, which specifies the starting byte of TxPDOs to map the return data of DeviceNet Slave response packets to TxPDOs. When the 'Ack Data Dst' is selected as 'Target Protocol', this option is valid

Continued

Req Timeout	The unit is ms, and after transmitting the explicit packet, the DeviceNet Slave must respond correctly within the time set here, otherwise it is a request timeout
Req Type	Specify the method for transmitting the explicit packet, which can be either 'Interval Req' or 'single Req'. The timed request is to transmit the explicit packet in a loop at the 'Interval Req' set by the 'Req Period' after the system is started. 'single Req' refers to transmitting the explicit packet only once after the time set in the 'Req Period' after the system is started
Req Period	Unit is ms, set the time interval for transmitting 'Interval Req' and 'single Req'

3.3.6 PXB-8030 Parameter Configuration

1. Modbus Parameter Setting

2. Click on 'Modbus Settings' in the 'Device Configuration' column of AWPX software to configure Modbus parameters. The configuration interface is shown in Figure 3.44.

Figure 3.44 Modbus Settings

The Modbus of PXB-8030 supports four working modes, which can be selected through the 'Work Mode' drop-down list box. Each workmode has corresponding parameters, and the functional descriptions and corresponding parameter descriptions of these four work modes are shown in Table 3.22.

Table 3.22 Description of Modbus parameter

Work mode	Function	Configuration	Description
Modbus TCP Master	The Device is configured as Modbus TCP Master	Slave IP address	Unique TCP slave IP address
		Slave port	Unique TCP slave port number
		Slave ID	Modbus target slave number
Modbus TCP Slave	The Device is configured as Modbus TCP Slave	Local Port	Local port number
		Local Slave Num	Modbus Slave Number

Continued

Modbus RTU Master	The Device is configured as Modbus RTU Master	Baud Rate	RTU communication parameters
		Data Bits	
		Stop Bits	
		Parity	
		Terminating Resitor	Enable or disable the terminal resistance of RS485 interface
Modbus RTU Slave	The Device is configured as Modbus RTU Slave	Baud Rate	RTU communication parameters
		Data Bits	
		Stop Bits	
		Parity	
		Terminating Resitor	Enable or disable the terminal resistance of RS485 interface
		Local Slave ID	Modbus Slave Number

3. Function code information setting

4. Simply click on 'Function Code Information' in the 'Device Configuration' column of the AWPX software to configure the PXB-8030 function code information. The configuration interface is shown in Figure 3.45.

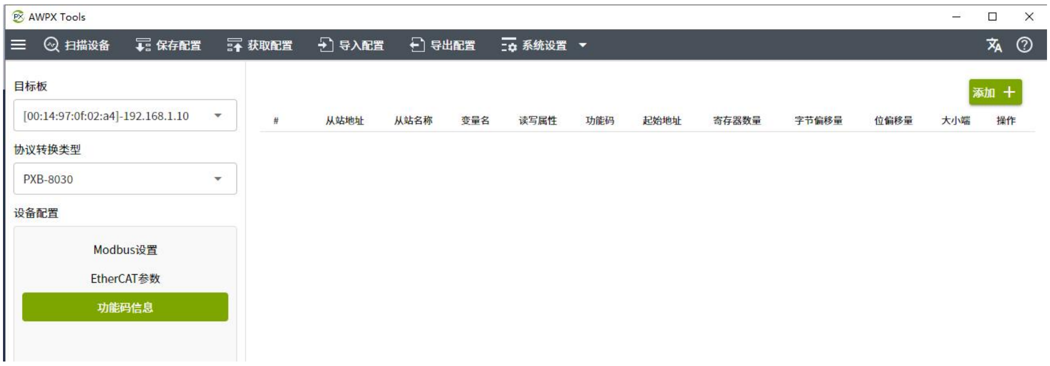

Figure 3.45 Modbus Master function code information

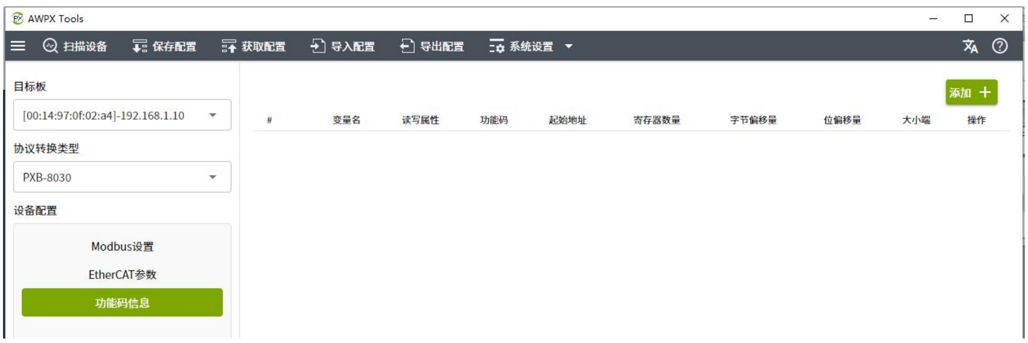

Figure 3.46 Modbus Slave function code information

The parameter description of function code information is shown in Table 3.23.

Table 3.23 Description of function code parameters

Parameter	Description
Slave Address	Set the ID of the device connected to the Modbus bus, limited to 1-255, unique to the master mode
Slave Name	Set slave name, unique to master mode
Var Name	Set the identifier name for the function code
Read/Write	Read Modbus and write Modbus respectively
Function code	支 Support 01H, 02H, 03H, 04H, 05H, 06H, 15H, 16H
Start Address	Starting address of operation register
Quantity	Number of operation registers
Byte Offset	Offset of EtherCAT PDO data in data mapping
Bit Offset	The bit offset of EtherCAT PDO data mapping only supports coil and discrete operations
Byte Order	Do data operations require size conversion. Coil and discrete operations are not supported

3.3.7 PXB-8032 Parameter Configuration

Click on 'Converter Config' in the 'Device Configuration' column of AWPX software to configure the converter parameters, as shown in Figure 3.47.

First, click the 'Add+' button to add a converter group that can be configured with EoE port numbers and protocol types. After adding, click the 'Add Data' button to add the corresponding converter configuration.

Figure 3.47 Converter parameter configuration interface

The configuration instructions for the converter are shown in Table 3.24.

Table 3.24 Description of converter parameters

Parameter	Description
EoE Port	EoE Port number
EoE Protocol Type	EoE protocol type, configurable as TCP Server, TCP Client, and UDP
Converter IP	The IP address of converter
Converter Port	Port number corresponding to the converter protocol
Converter Protocol Type	Converter protocol types, configurable as TCP Server, TCP Client, and UDP
UDP Port	Device UDP port number

4. Product installation

4.1 Mechanical dimensions

PXB-80 series product dimensions: 125.00mm×76.00mm×28.00mm (bare metal).

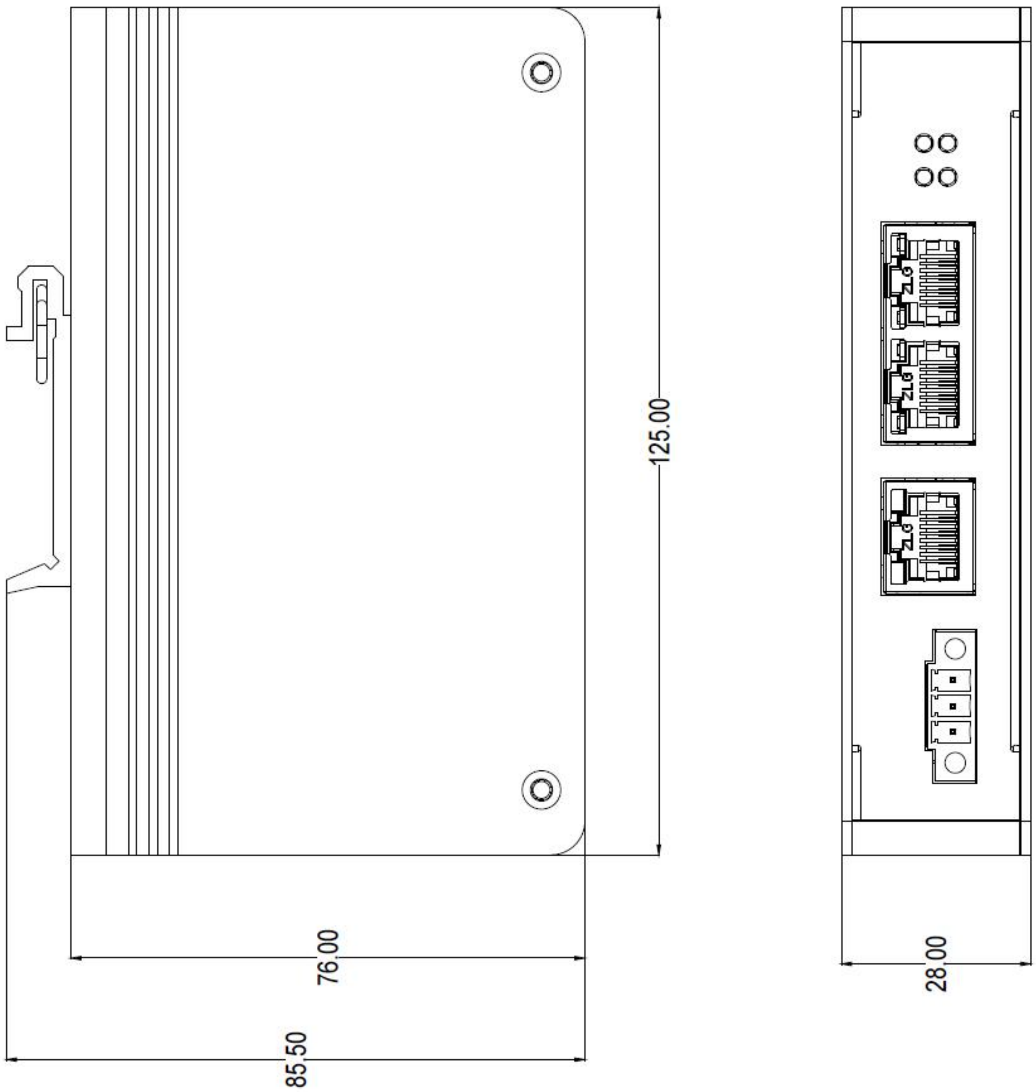

Figure 4.1 Product Dimensional Diagram

5. Product maintenance and precautions

- Before powering on the product, please check if the power input voltage is within the required range, if the product wiring is reasonable, and if there are any short circuits or incorrect signal lines;
- The product does not have IP protection level requirements and needs to be protected from water ingress, which may affect the normal operation of the product;

6. Appendix

6.1 Product Packing List

Table 6.1 Product Packing List

Num	Name	Quantity	Unit	Physical picture
1	PXB-80xx Protocol Converter Gateway	1	Pcs	
2	3-Pin connecting terminal	2	Pcs	
3	certificate	1	Piece	

7. Disclaimer

Based on the principle of providing better service to users, Guangzhou ZHIYUAN Electronics Co., Ltd. (hereinafter referred to as "ZHIYUAN Electronics") will present detailed and accurate product information to users as much as possible in this manual. However, due to the timeliness of the content in this manual, ZHIYUAN Electronics cannot fully guarantee the timeliness and applicability of this document at any time. ZHIYUAN Electronics reserves the right to update the content of this manual without prior notice. In order to obtain the latest version of information, please visit the official website of ZHIYUAN Electronics regularly or contact the staff of ZHIYUAN Electronics. Thank you for your tolerance and support!

Integrity and win-win situation, continuous learning, customer first, professional focus, and focus on being the first

Guangzhou ZHIYUAN

Electronics Co., Ltd

For more details, please visit

www.zlg.cn

Welcome to call the national service hotline

400-888-4005