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

Revision History

Version	Date	Reson
V1.00	2025/12/15	Initial Release

Contents

1. Product Introduction

1.1 Product Overview

The PXB-81 series products are industrial fieldbus protocol converters launched by Guangzhou ZHIYUAN Electronics Co., Ltd., enabling conversion between EtherNet/IP and various industrial fieldbus protocols such as Modbus, CAN/CAN FD, and DeviceNet. This series features two Ethernet interfaces that support the EtherNet/IP slave protocol, one CAN FD interface or one RS485 interface, and one USB Type-C interface. It is equipped with a high-real-time processor designed to handle the conversion of data packets from EtherNet/IP to various industrial fieldbus protocols. The PXB-81 series protocol converters come with a host computer configuration tool that allows for flexible configuration of related functions, making it easy to seamlessly convert data from various industrial bus protocols into EtherNet/IP data.

Figure 1.1 Product series group shot

1.2 Product Series Description

1.2.1 Naming Rules

The naming rules for the PXB protocol converter series is shown in Figure 1.2.

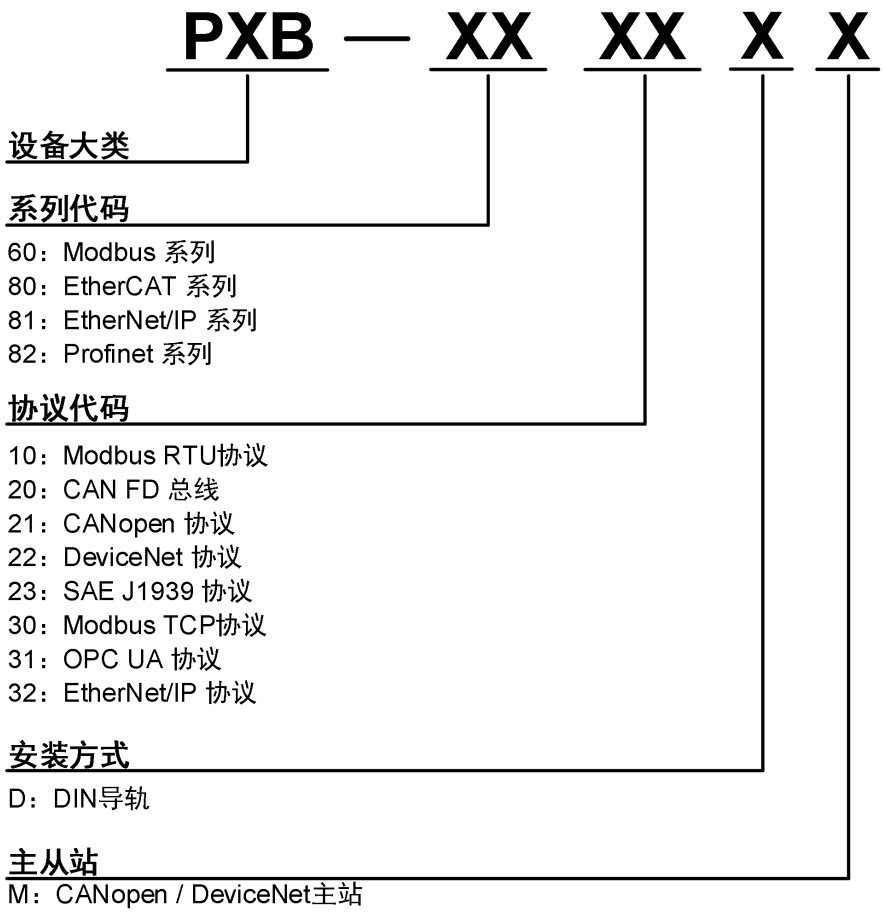


图 1.2 Naming Rules for Series Products

1.2.2 Ordering Information

Table 1.1 Ordering Information for Series Products

Product model	Protocol conversion type	Installation method
PXB-8110	EtherNet/IP<=> Modbus RTU	35mm DIN rail
PXB-8120	EtherNet/IP<=> CAN/CAN FD	35mm DIN rail
PXB-8121M	EtherNet/IP<=>CANopen（Master）	35mm DIN rail
PXB-8122M	EtherNet/IP<=> DeviceNet（Master）	35mm DIN rail

1.3 Hardware Features

Table 1.2 Product Hardware Features

Input voltage	9～36VDC，80mA @ 24VDC
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:common:1.0kV,differential:0.5kV;Signal:0.5kV IEC/EN 61000-4-6 CS(150 kHz 至 80 MHz)：Power supply:3 V/m；Signal:3 V/m

1.4 Software Features

Table 1.3 Software Features

EtherNet/IP Features	Supports standard EtherNet/IP protocol
	Maximum 500 bytes input and 500 bytes output
	Supports dynamic IP assignment via DHCP
	Supports Device Level Ring (DLR) topology
(PXB-8110) Modbus Features	Supports user-defined serial baud rates ranging from 9600 to 2,000,000 bps, with a maximum ultra-high speed of 2 Mbps
	Data bits, stop bits, and parity bits can all be configured
	IP address, slave ID, target IP, and port are all configurable
	Supports Modbus TCP and RTU operating modes (Master/Slave)
	RTU mode supports up to 255 slaves, while TCP master supports up to 8 slaves
	Supports slave status monitoring in master mode
	Supports dynamic slave enablement
	Supports Modbus exception handling
(PXB-8120) CAN /CAN FD Features	Supports multiple transmission modes including single shot, periodic, event-triggered, frame ID-triggered, cyclic group, and conditional sending
	Supports sending and receiving CAN, CAN FD, and CAN FD BRS frames
	Supports a maximum ultra-high CAN FD baud rate of 5 Mbps
	Supports standard, extended, and remote CAN frames
	Supports up to 128 transmit messages and 128 receive whitelist messages
	Supports configuration of CAN type, CAN FD standard, and CAN FD BRS parameters
	Supports frame counter
(PXB-8121M) CANopen Features	Supports a maximum of 126 slave nodes
	Supports baud rate settings from 10K to 1M
	Supports up to 128 RPDO messages and 128 TPDO messages
	Supports SDO configuration to flexibly configure the slave node dictionary
	Supports configuration of PDO attribute parameters
	Supports slave node status monitoring
(PXB-8122M) DeviceNet Features	Standard DeviceNet master, supporting up to 63 standard DeviceNet slaves
	Supports DeviceNet communication rates of 125K, 250K, and 500K
	Supports four I/O connection types: Polling, Bit Strobe, State Change, and Cyclic
	Supports up to 64 bytes of I/O buffer for each I/O connection type
	Supports configurable data update cycle for each DeviceNet slave
	Supports slave node status monitoring
Other Features	Comes with AWPX Tools configuration software, which is simple and user-friendly
	Supports software-configurable CAN and RS485 termination resistors
	Supports one-click factory reset for multiple safety guarantees
	Supports one-click remote firmware upgrade
	Supports edge computing

2. Hardware Description

2.1 Product Appearance

Figure 2.1 Product appearance

The top panel of the PXB-81 series features one power connector, one USB-C port, and one recessed factory reset button.The front panel is equipped with four LED indicators, two EtherNet/IP network ports, and one electrically isolated CAN/CAN FD or RS485 interface.

Note: Please refer to Figure 2.2 (LED Indicator Description) and Figure 2.3 (Interface Definition) for detailed LED and interface definitions and usage.

2.2 Indicator Light Description

The front panel of the PXB-81 series features four LEDs. Depending on the specific product model, these LEDs serve different functions and are labeled with standard industry abbreviations: PWR, RUN, PN, CAN, 485, MS, and NS. Detailed descriptions are provided in the table below:

Table 2.1 PXB-8110 LED Status Description

Identification	Definition	State	Description
PWR	Equipment power indicator	Off	The equipment is not powered on or has abnormal power supply
		Red	The equipment power supply normally
MS	EtherNet/IP Device Status Indicators	Green	The equipment is working normally
		Green flashing	Device requires debugging due to lost, incomplete, or incorrect configuration.
		Red flashing	Recoverable fault
		Red	Unrecoverable fault, replacement required
		Flashing Red and Green	Device is self-testing
NS	EtherNet/IP Network Status Indicators	Off	Device has no IP address or is not powered on
		Green flashing	Device has not established a connection, but has obtained an IP address
		Green	Device is online and has one or more established connections
		Red	IP address is already in use
		Red flashing	Connection(s) to this device as a target have timed out
485	RS-485 Communication Status Indicator	Off	Device is not sending or receiving data
		Green	Device is sending and receiving data normally

Table 2.2 PXB-8120 LED Status Description

Identification	Definition	State	Description
PWR	Equipment power indicator	Off	The equipment is not powered on or has abnormal power supply
		Red	The equipment power supply normally
MS	EtherNet/IP Device Status Indicators	Green	The equipment is working normally
		Green flashing	Device requires debugging due to lost, incomplete, or incorrect configuration.
		Red flashing	Recoverable fault
		Red	Unrecoverable fault, replacement required
		Flashing Red and Green	Device is self-testing
NS	EtherNet/IP Network Status Indicators	Off	Device has no IP address or is not powered on
		Green flashing	Device has not established a connection, but has obtained an IP address
		Green	Device is online and has one or more established connections
		Red	IP address is already in use
		Red flashing	Connection(s) to this device as a target have timed out
CAN	CAN Communication Status Indicator	Off	No data transmission
		Green flashing	Normal data transmission

Table 2.3 PXB-8121M LED Status Description

Identification	Definition	State	Description
PWR	Equipment power indicator	Off	The equipment is not powered on or has abnormal power supply
		Red	The equipment power supply normally
MS	EtherNet/IP Device Status Indicators	Green	The equipment is working normally
		Green flashing	Device requires debugging due to lost, incomplete, or incorrect configuration.
		Red flashing	Recoverable fault
		Red	Unrecoverable fault, replacement required
		Flashing Red and Green	Device is self-testing
NS	EtherNet/IP Network Status Indicators	Off	Device has no IP address or is not powered on
		Green flashing	Device has not established a connection, but has obtained an IP address
		Green	Device is online and has one or more established connections
		Red	IP address is already in use
		Red flashing	Connection(s) to this device as a target have timed out
CAN	CAN Communication Status Indicator	Red	No data transmission
		Green flashing	Communication is normal, but one or more sub-nodes are offline
		Green	Communication is normal, and all sub-nodes are online

表 2.4　PXB-8122M 指示灯状态说明

Identification	Definition	State	Description
PWR	Equipment power indicator	Off	The equipment is not powered on or has abnormal power supply
		Red	The equipment power supply normally
MS	EtherNet/IP Device Status Indicators	Green	The equipment is working normally
		Green flashing	Device requires debugging due to lost, incomplete, or incorrect configuration.
		Red flashing	Recoverable fault
		Red	Unrecoverable fault, replacement required
		Flashing Red and Green	Device is self-testing
NS	EtherNet/IP Network Status Indicators	Off	Device has no IP address or is not powered on
		Green flashing	Device has not established a connection, but has obtained an IP address
		Green	Device is online and has one or more established connections
		Red	IP address is already in use
		Red flashing	Connection(s) to this device as a target have timed out
CAN	DeviceNet Run Indicator	Green	Normal operation
		Red	Device initialization failed; not in operating mode

2.3 Interface Definition

The top of the PXB-81 series provides one OPEN3 power connector, one recessed factory reset button, and one USB Type-C port, as shown in Figure 2.2.

Figure 2.1 Top interface definition diagram

2.3.1 Power Interface

The power supply supports a wide voltage input of 9~36V. Customers can select a power supply within this voltage range based on the site environment. The physical form of the power connector is an OPEN3 locking socket with a 3.81mm pitch, and the housing is silk-screen printed with "9V ~ 36V". The power supply requirements are shown in Table 2.5.

表 2.5 电源接口输入电源规格

Parameter	Min	Typ	Max	Unit
Operating voltage	9.0	24.0	36.0	V
Operating current	—	80	—	mA
Power consumption	—	1.92	—	W

2.3.2 USB Interface

The PXB-81 series provides one USB Type-C port. The physical form of the connector is a standard USB-C receptacle, and the housing is silk-screen printed with "USB". This port can be used as a power supply interface for the entire device 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 indicators, the front panel features two 10/100M Ethernet ports and one isolated RS485 port or isolated CAN/CAN FD port, as shown in Figure 2.2.

Figure 2.2 Front interface definition diagram

2.3.4 EtherNet/IP Interface

As the EtherNet/IP side functions as a slave network interface, the device features two RJ45 ports driven by an internal TSN switch, allowing users to connect via either port. Supporting standard 10/100M Ethernet communication, these ports serve as both EtherNet/IP communication interfaces and device configuration ports. Users can connect to a switch or PC to perform parameter tuning using the accompanying configuration software.

2.3.5 CAN/CAN FD Interface

For the PXB-8120, PXB-8021M, and PXB-8122M models, the bottom interface is a CAN/CAN FD port. The interface markings are shown in Table 2.6.

Table 2.6 CAN FD Interface Labeling Description

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

2.3.6 RS485 Interface

For the PXB-8110 model, the bottom interface is an RS485 port. The interface markings are shown in Table 2.7.

Table 2.7 RS485Interface Labeling Description

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

2.4 Terminal Resistance

The PXB-81 series protocol converters are equipped with built-in 120 Ω termination resistors for the RS485/CAN communication ports. By default, these resistors are disconnected from the bus, but users can configure them using the upper computer software, as shown in Figure 2.2. Users can choose whether to connect or disconnect the termination resistors, eliminating the need to manually add or remove external resistors.

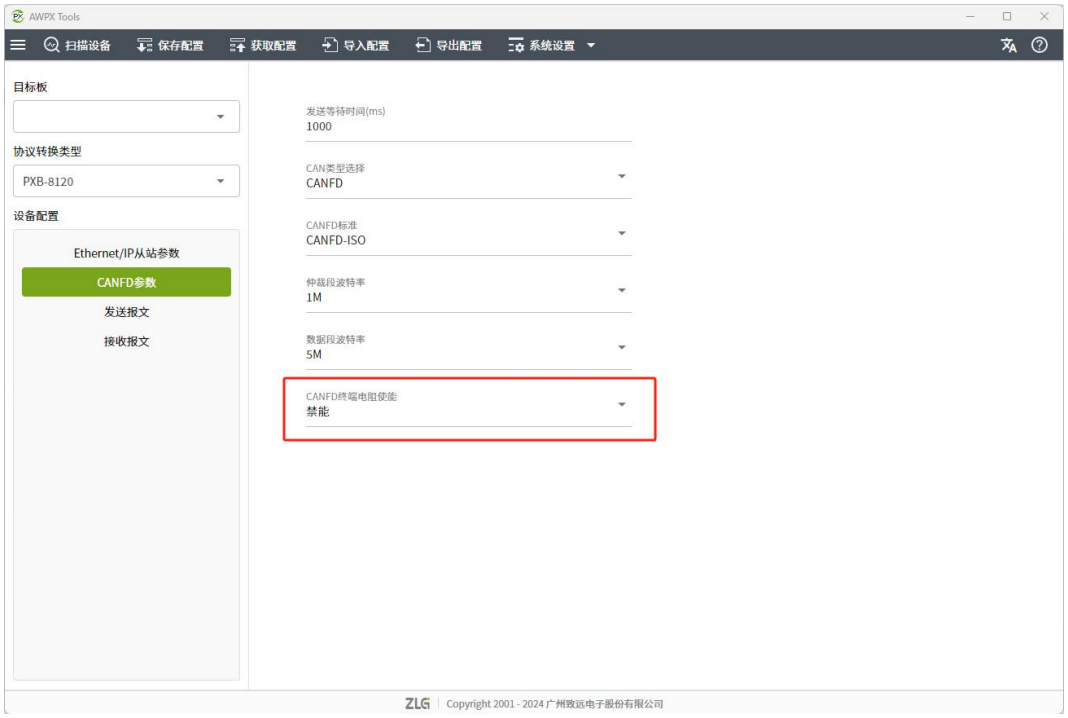


Figure 2.2 Terminal resistance configuration interface

3. Instructions For Use

The operation methods for the PXB-81 series protocol converters are largely similar. You can configure and use the devices by following the steps below:

1、Install the AWPX configuration software and TwinCAT3 software on the PC.

2、After powering on the device properly, connect the PXB-81 protocol converter to the PC using an Ethernet cable plugged into any available port.

3、Open the AWPX configuration tool, configure the parameters according to your requirements (see Section 3.2), and download the configuration to the PXB-81 protocol converter.

4、Connect the master device to the PXB-81 protocol converter using an Ethernet cable.

5、Establish data communication and exchange.

Note: If your master device does not run on TwinCAT3 or an x86 platform, you can still use this device. Simply import the EDS file based on your specific master device to connect and properly configure the PXB-81 protocol converter.

3.1 PLC Parameter Configuration

The following section will take TwinCAT3 as an example to demonstrate how to add the PXB-81 protocol converter.

3.1.1 Import the EDS File

Copy and paste the EDS file provided with the PXB-81 protocol converter to the TwinCAT3 installation directory,for example:

C:\TwinCAT\3.1\Config\Io\EtherNetIP；

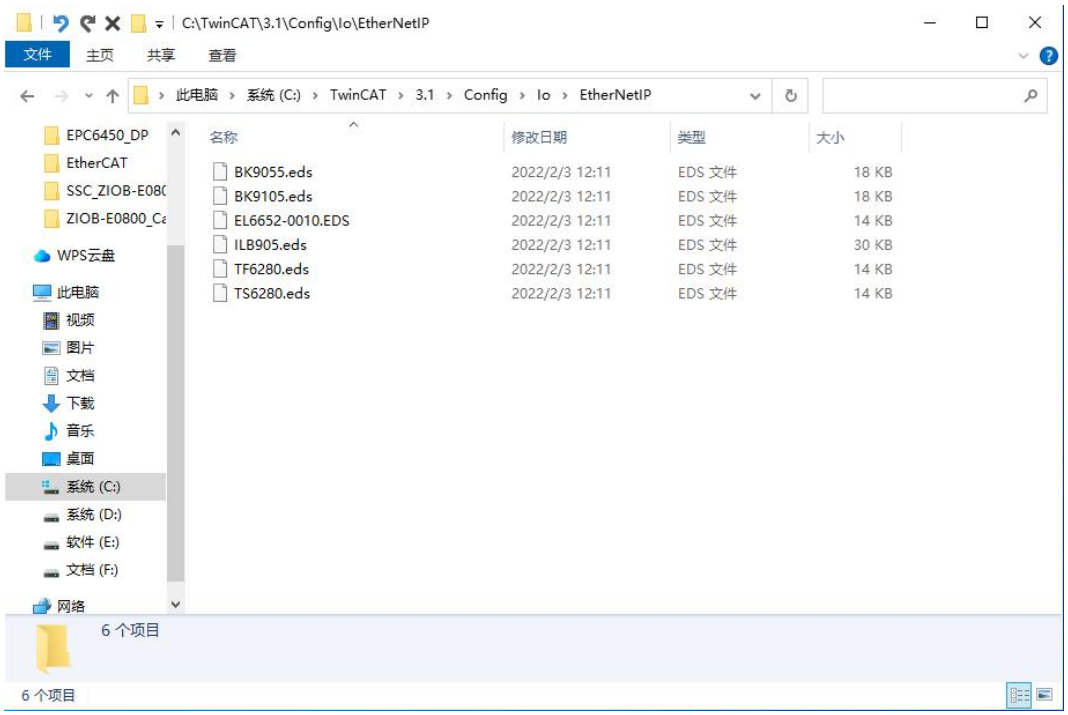


Figure 3.1 edsfile placement directory

3.1.2 Configuring the PXB Protocol Converter

1. Creating a New Project

Open the installed TwinCAT3 software to enter the development environment：

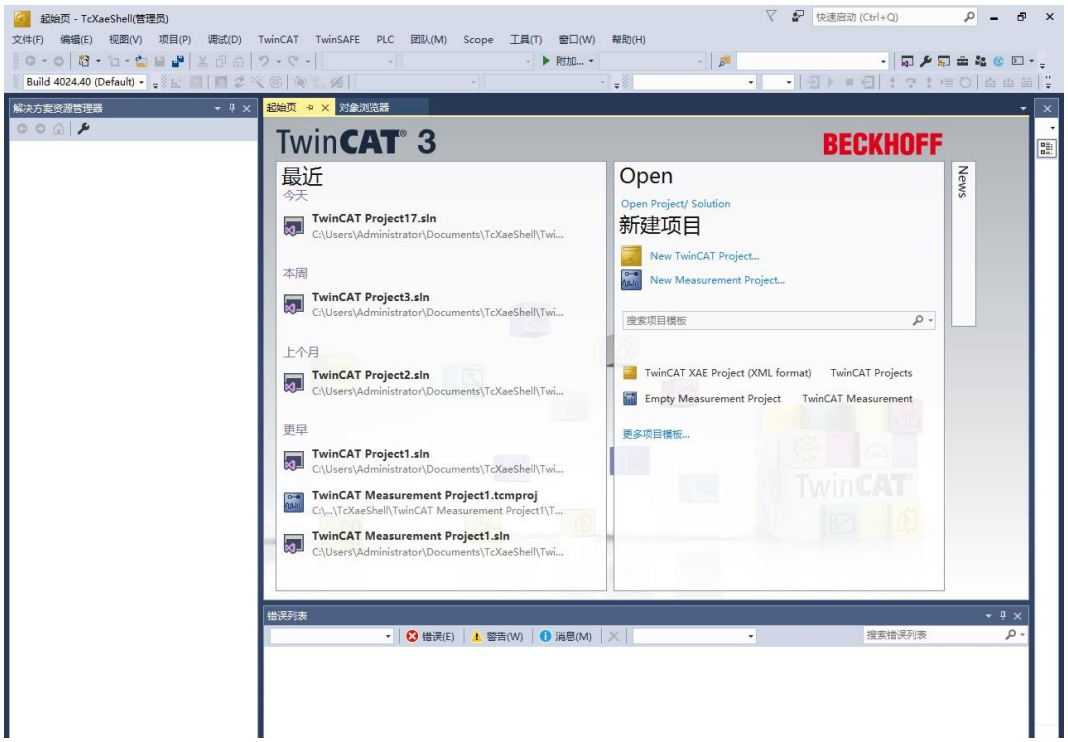


Figure 3.2 TwinCAT3 development interface

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"：

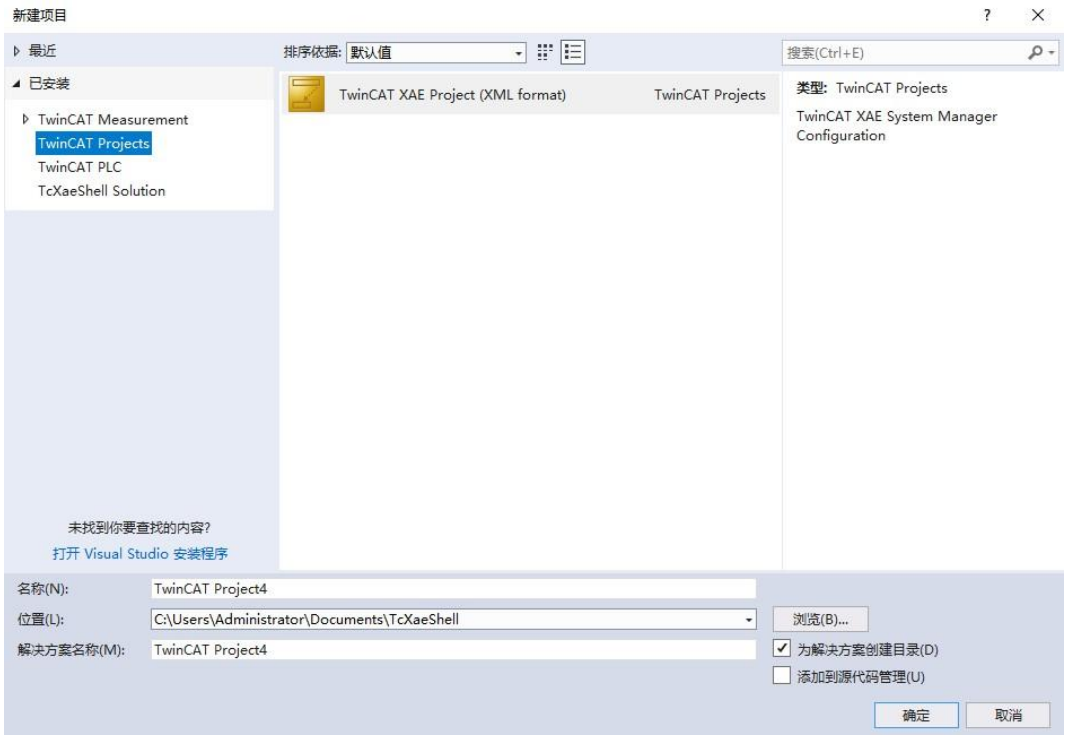


Figure 3.4　Engineering Information Modification Interface

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

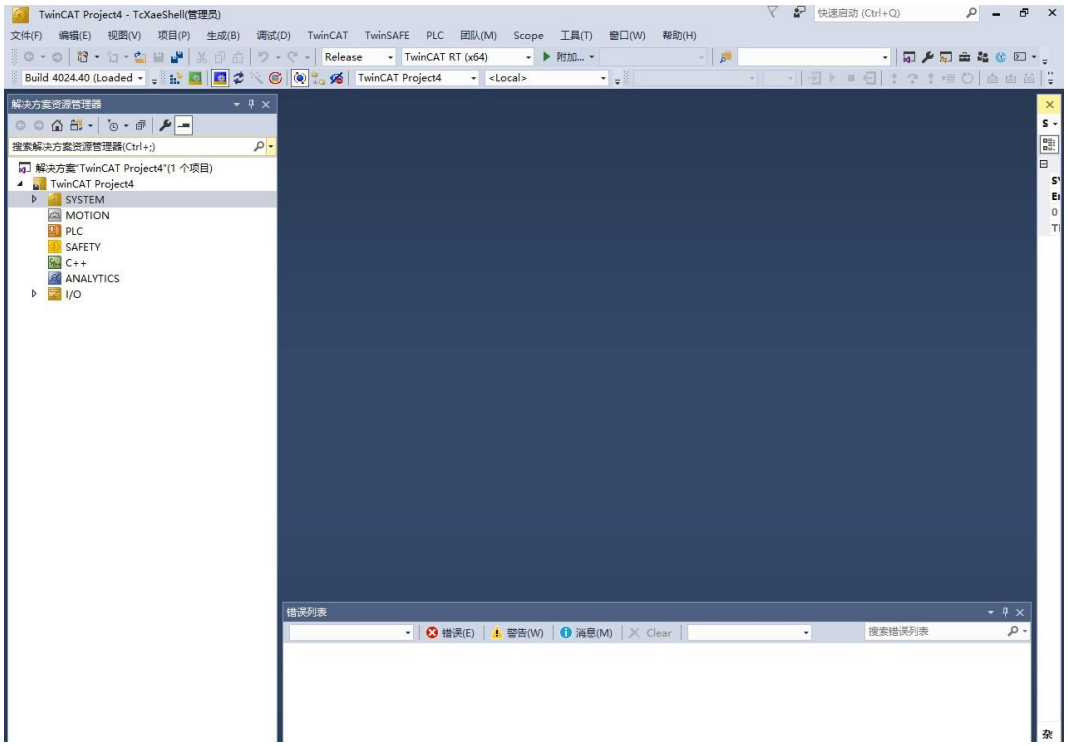


Figure 3.5　Project Creation Success Interface

“Install” 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"；

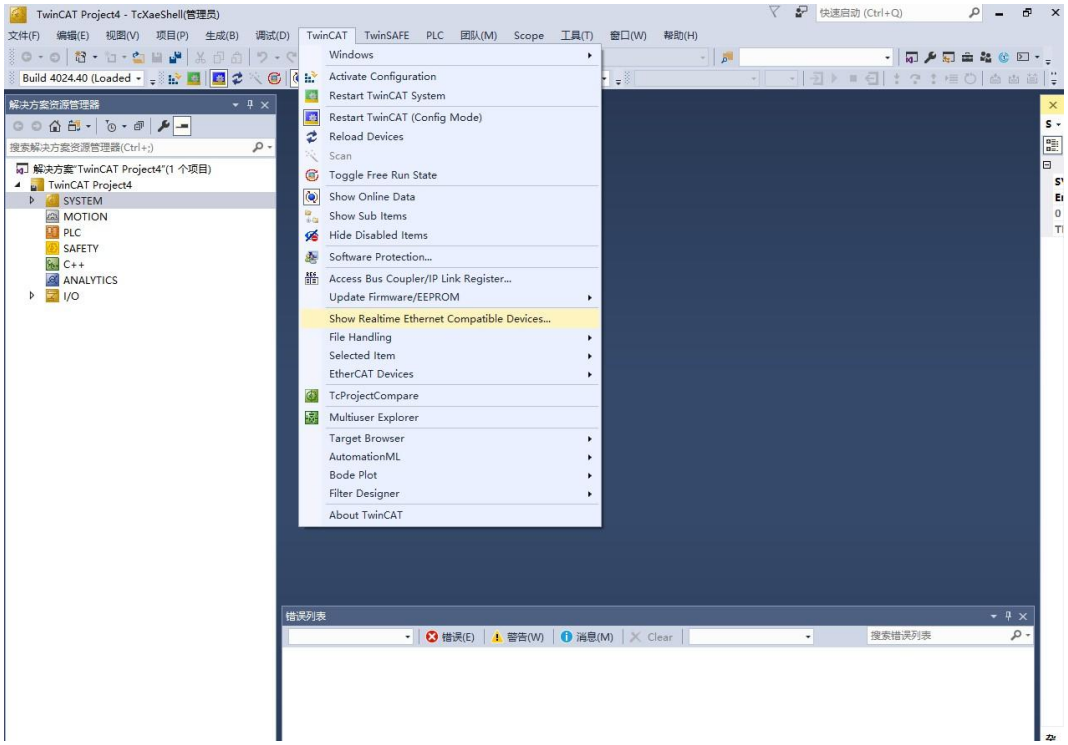


Figure 3.6　Network card installation interface

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.

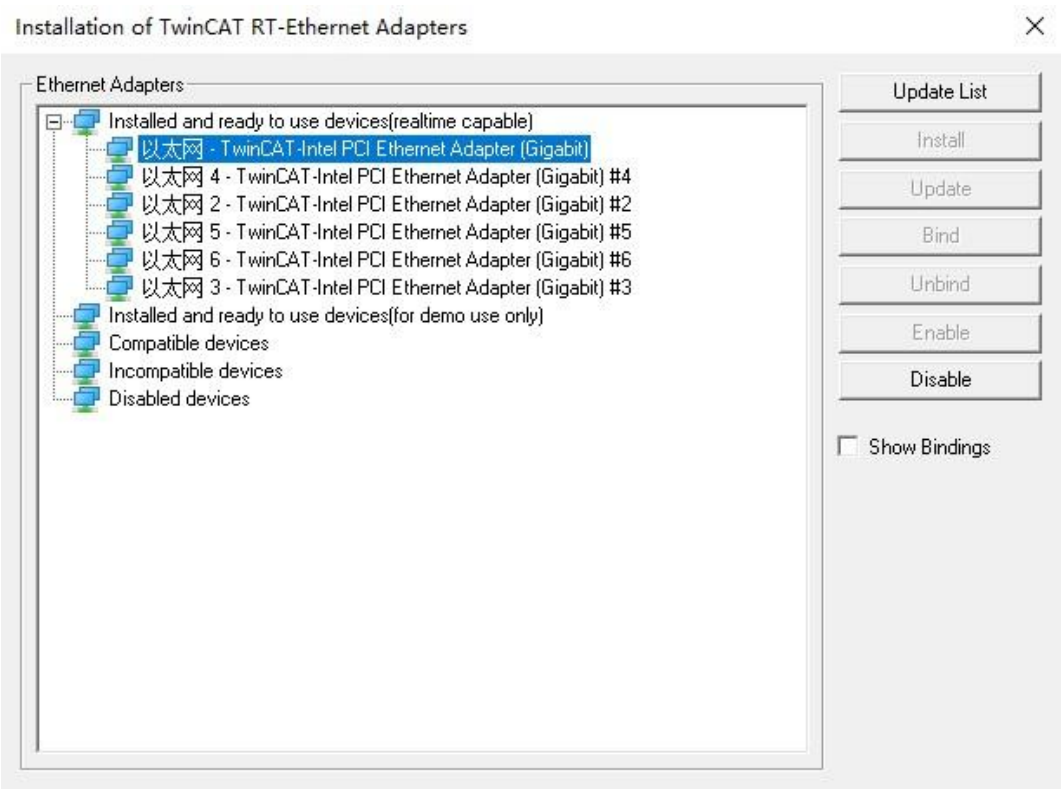


Figure 3.7 Network card driver normal interface

2. Scanning devices

In the project tree, expand “I/O”. Right-click on “Device” and select “Add New Item”。

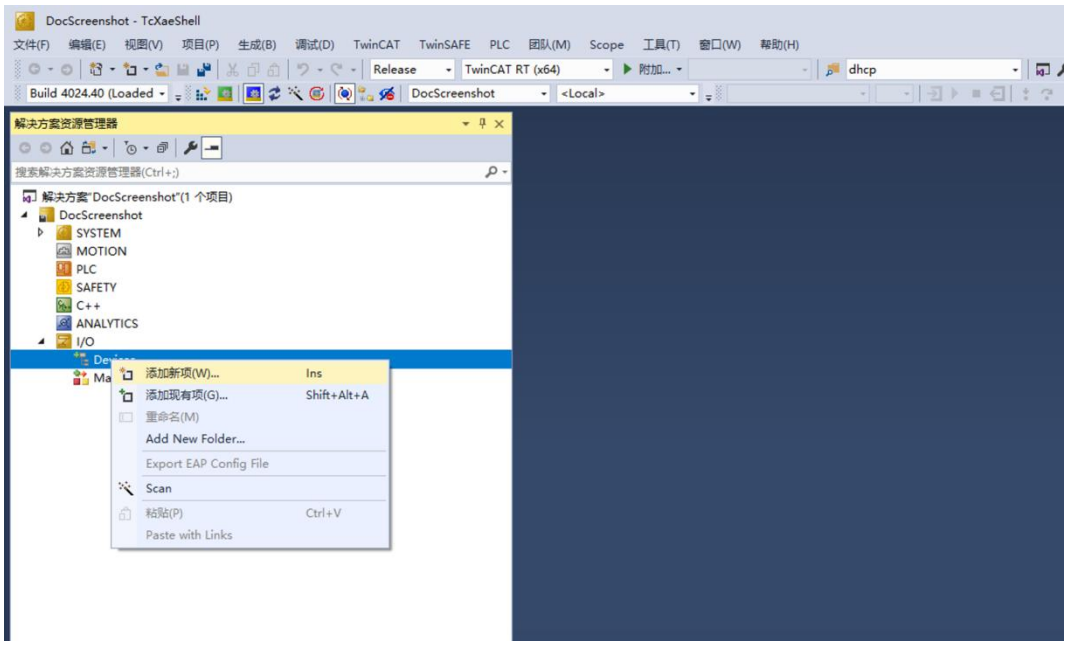


Figure 3.8 Add New Item Interface

Select EtherNet/IP Scanner to add the EIP master.

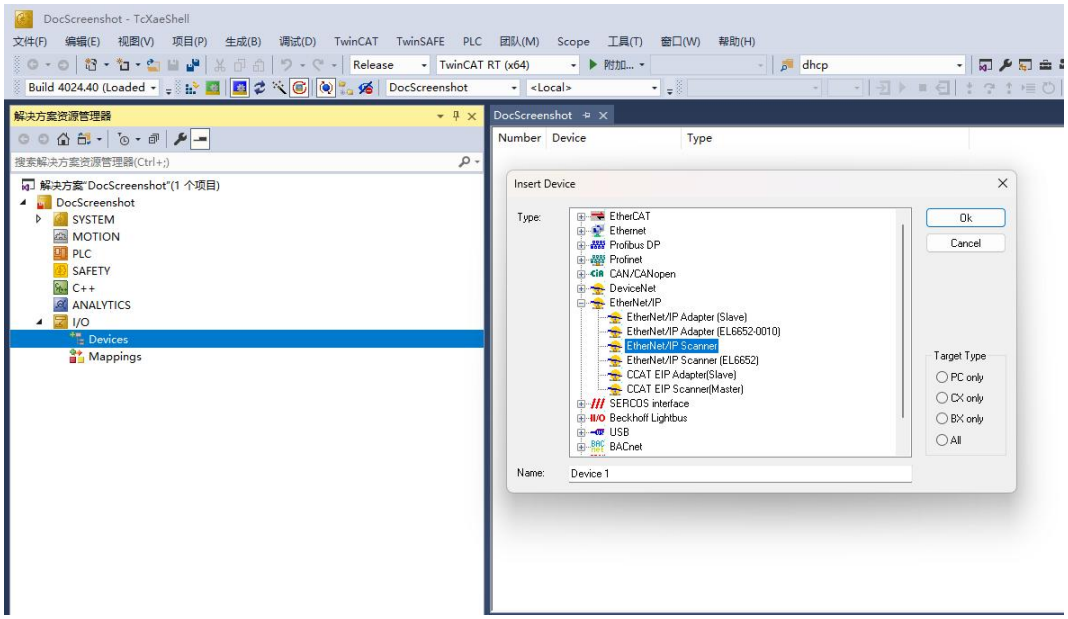


Figure 3.9 Add Master Dialog

Double-click Device 1 (TC3 EIP Scanner) and select the Settings tab. Open the F800:0 option to configure the master IP information (F800:21/22/23). After the configuration is complete, click Reload Device, and then check the corresponding F900 options to verify the settings.

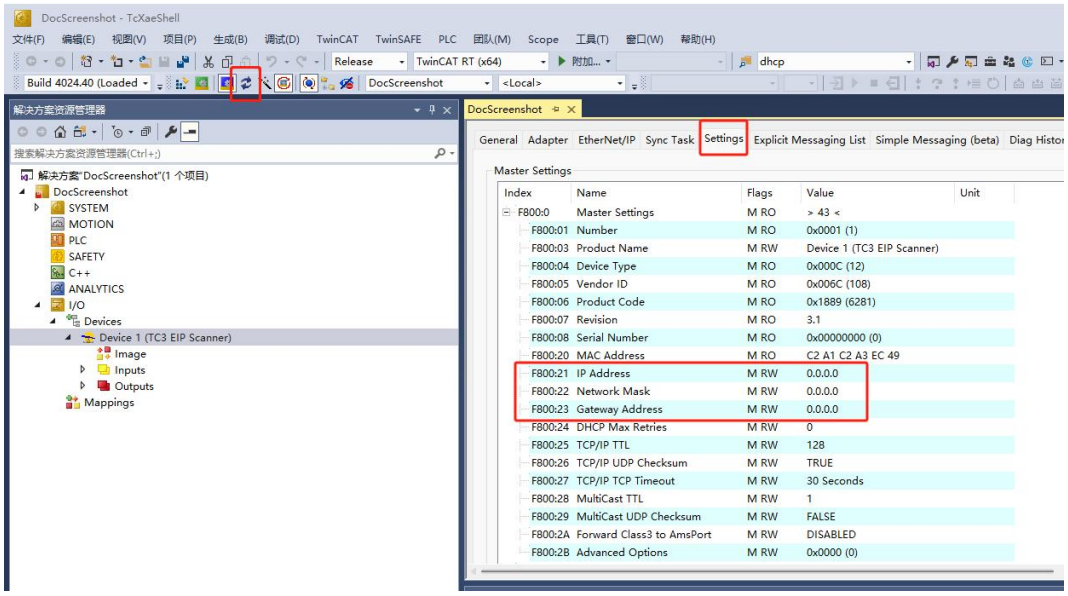


Figure 3.10 Configuring the Master Network

Note: The IP address must be in the same subnet as the slave device, but it does not need to match the physical network adapter. The gateway address must be set to the IP address of the physical network adapter.

Right-click Device 1 (TC3 EIP Scanner) and select Scan to scan for slave devices. Note: The PXB-81xx device must be connected to the corresponding PC network port in advance.

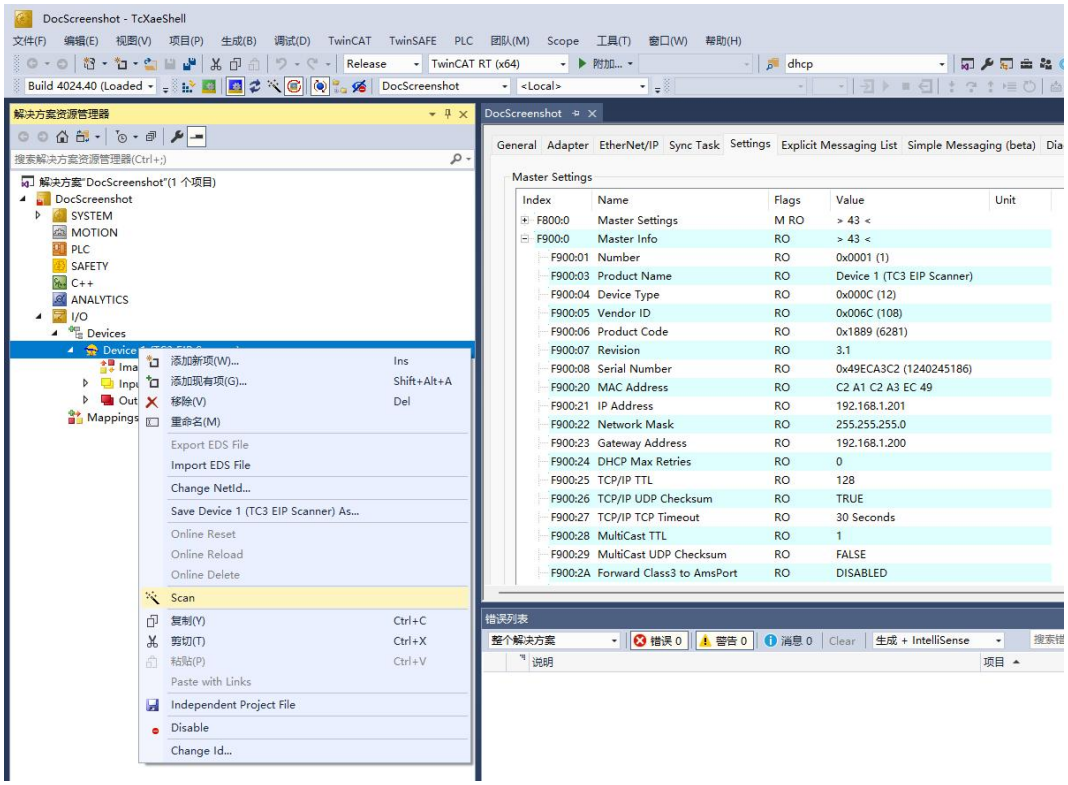


Figure 3.11 Scan for Slaves

After the device is successfully scanned, the following dialog box will appear. If you have not modified the default configuration, the scanned device IP should be 192.168.1.136. Select the device and click OK.

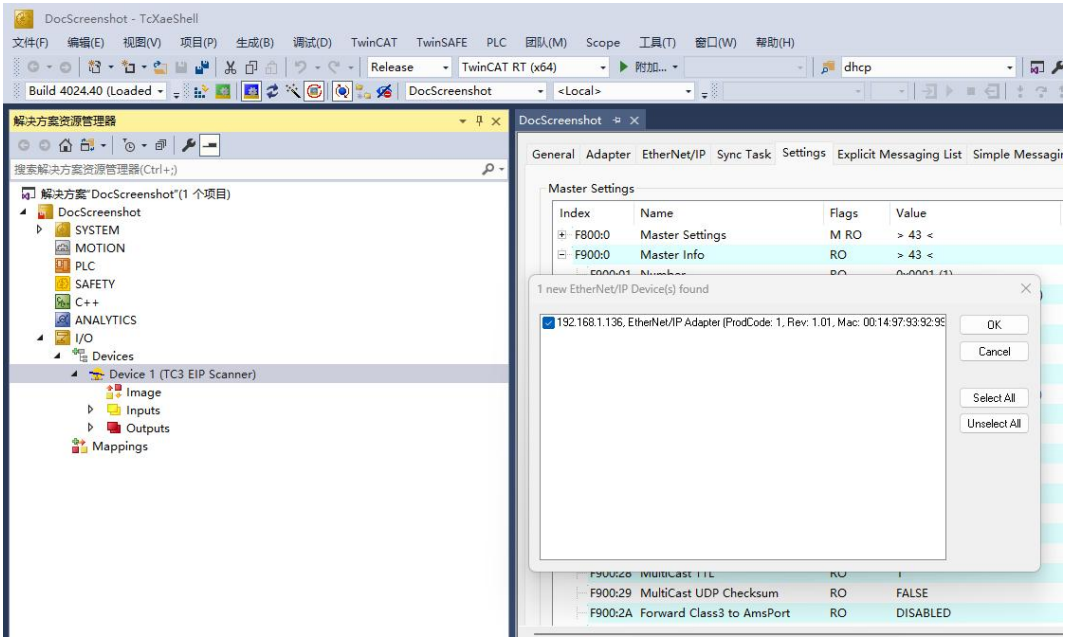


Figure 3.12 EIP Slave Device Discovered

After the device is successfully added, a new EtherNet/IP Adapter will appear under the

Device 1 (Profinet Controller) option. Right-click it and select Load from EDS File.

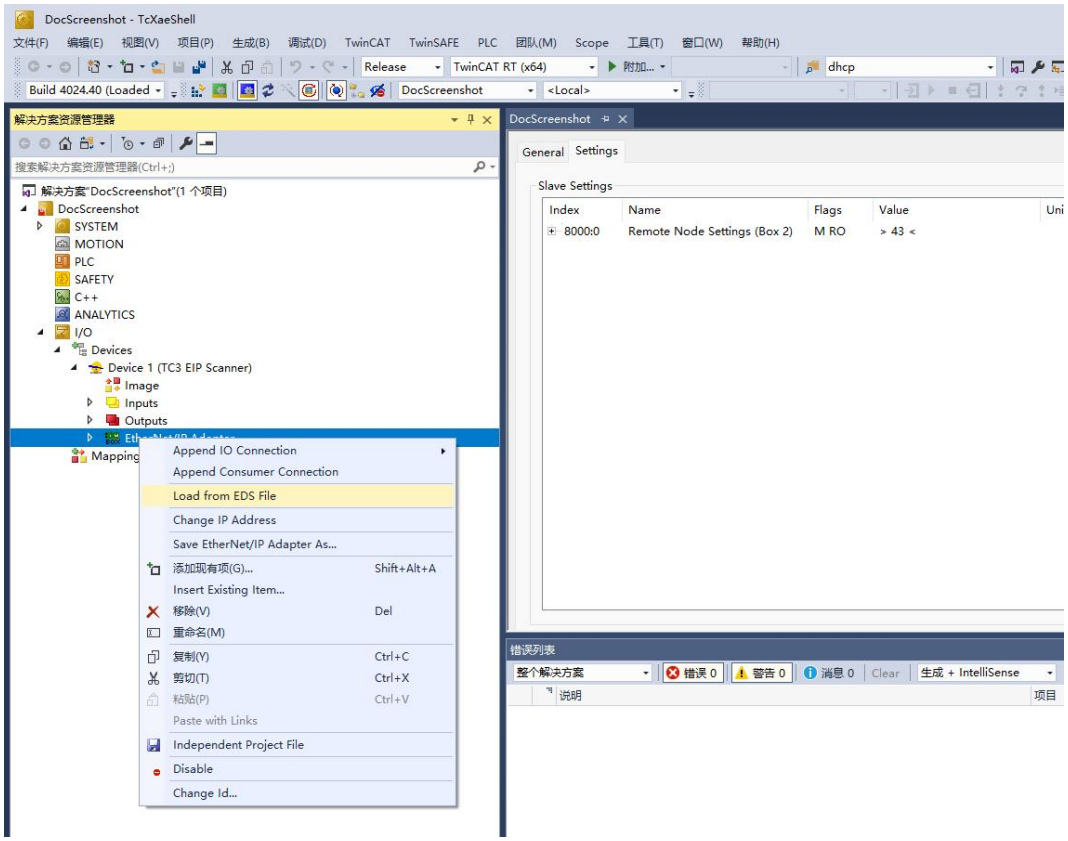


Figure 3.13 Load from EDS file

After selection, the following window will pop up. Select the EDS file for the corresponding model under the ZHIYUAN option.

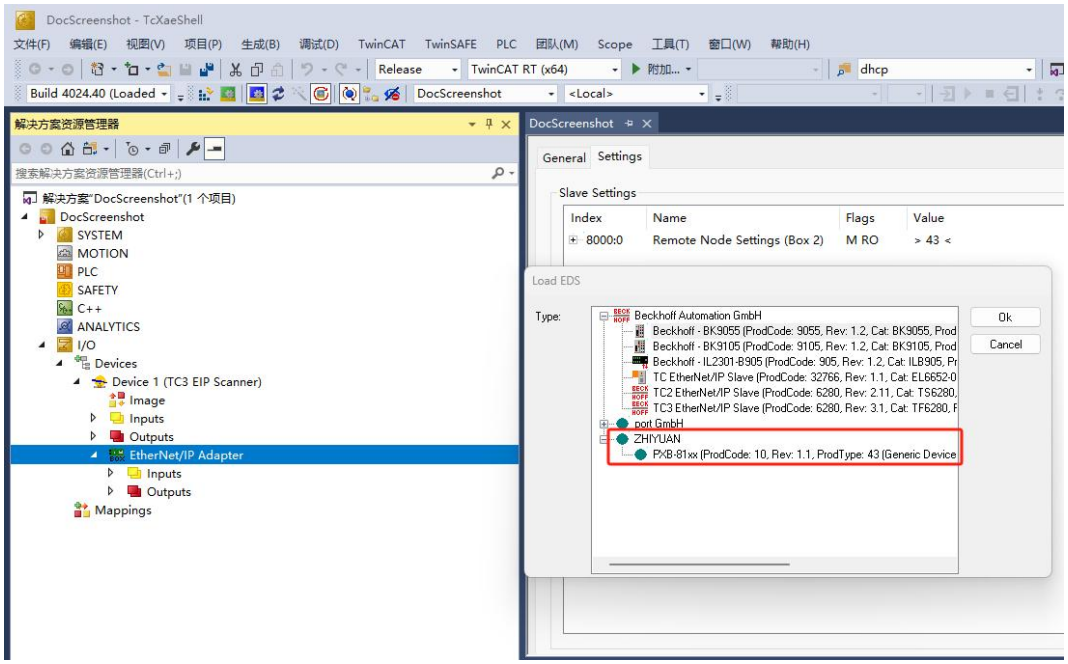


Figure 3.14 Select EDS file

After it is successfully added, right-click the EtherNet/IP Adapter again, select Append IO Connection, and then select Exclusive Owner.

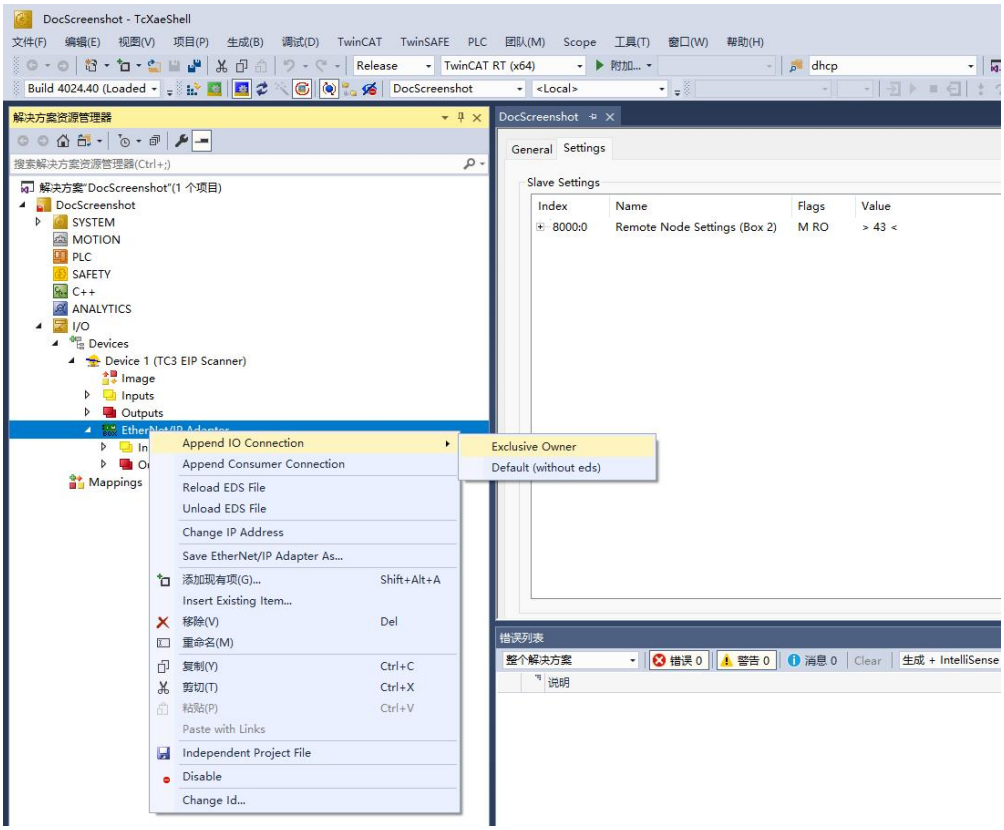


Figure 3.15 Add Connection

Upon completion of the previous step, the Connection 1 (Input/Output) option will be added. Click it to set the Input/Output Data Length. Ensure that this value matches the settings in AWPX, or communication errors will occur.

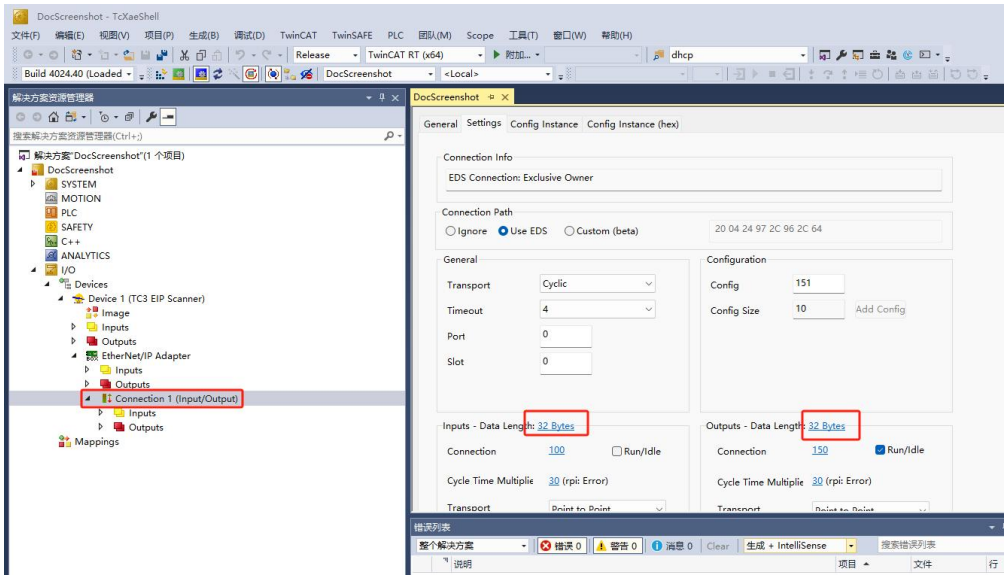


Figure 3.16 Set IO Data Size

After the configuration is complete, click Reload Device. The device will establish a connection and start communication.

3.2 AWPX Configuration Tool

3.2.1 Configuration Software Acquisition and Installation

All PXB series protocol converters are configured using the AWPX Tools software (hereinafter referred to as AWPX). You can download AWPX by visiting our official website (http://www.zlg.cn) and searching for "AWPX". After downloading, double-click the installation package to start the installation. The initial installation interface is shown in Figure 3.17.

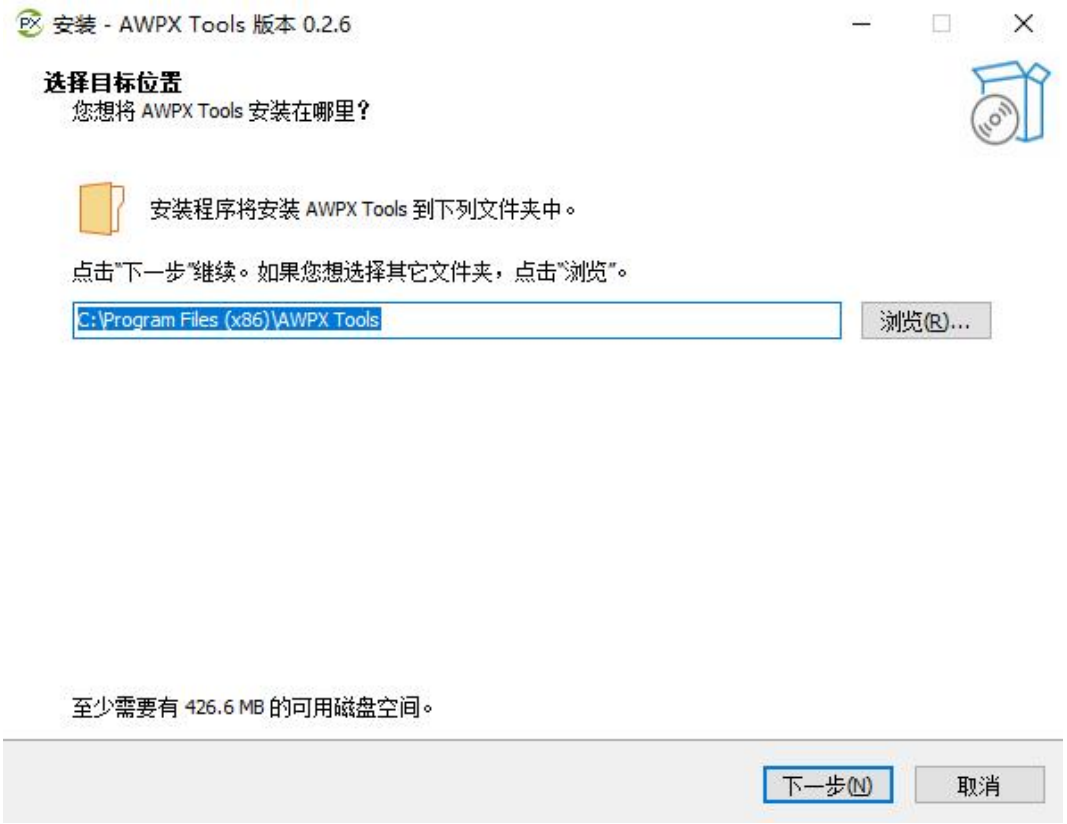


Figure 3.17 Install AWPX

After clicking Next several times, the installation interface is shown in Figure 3.18.

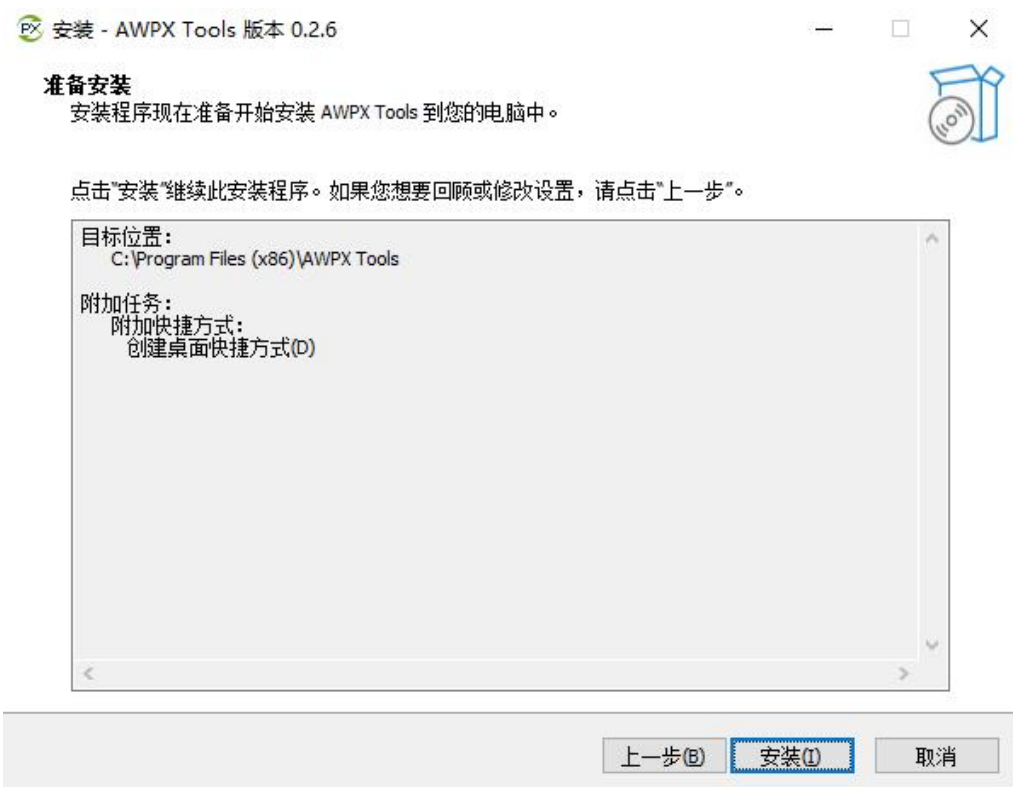


Figure 3.18 AWPX installation interface

Finally, click Install. The AWPX software will begin installing. Please wait patiently until the installation is complete.

3.2.2 Overview of Configuration Software Functions

Run the AWPX software. The interface is shown in Figure 3.19.

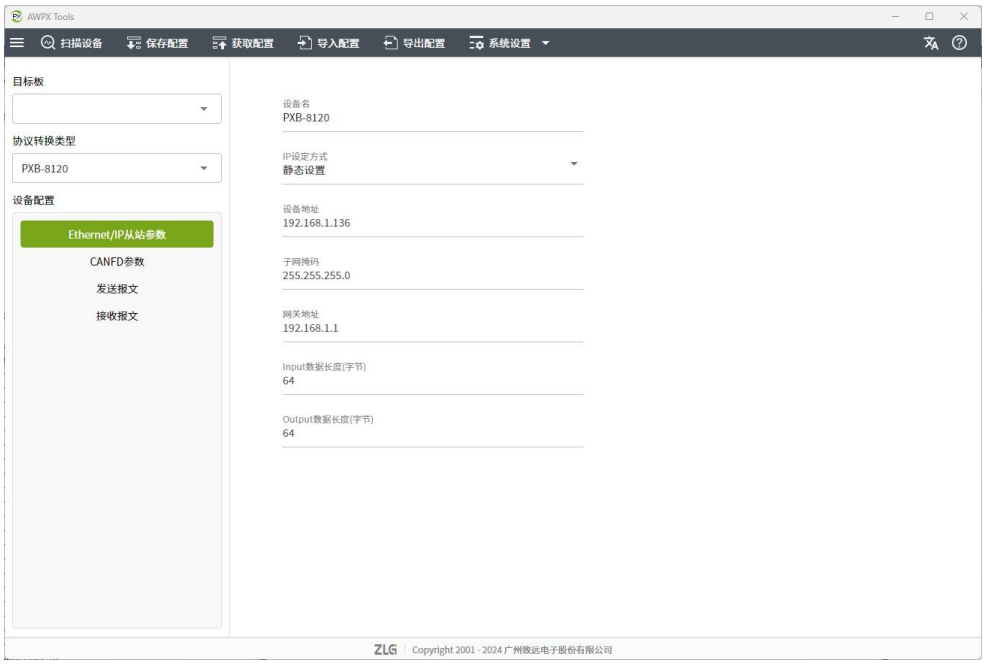


Figure 3.19 AWPX Software Interface

The left side of the AWPX interface displays the device information of PXB series products, including "Target Board", "Protocol Conversion Type", and "Device Configuration". The specific configuration options are displayed on the right side of the "Device Configuration" section.

The top of the interface features a toolbar with buttons such as Scan Device, Save Configuration, and Get Configuration, as shown in Figure 3.20.

Figure 3.20 Toolbar Buttons

1. Scan Device

Click the ‘Scan Device’ button. The AWPX software will search for all PXB series devices on the current local area network (LAN) and display the target board's IP address and firmware version in the ‘Target Board’ drop-down list. After selecting the correct device from the ‘Target Board’ drop-down list, you can configure the PXB series device.。

2. Save Configuration

After modifying any parameters, click the ‘Save Configuration’ button to send the updated configuration to the PXB series protocol converter and apply the changes. Saving the configuration will restart the device. Wait until the restart prompt at the bottom of the software interface disappears.

3. Get Configuration

After selecting a device, click the ‘Get Configuration’ button to retrieve and display the current running configuration of the PXB series protocol converter.

4. Import Configuration

After selecting a device, click the ‘Import Configuration’ button to import a configuration file with the AWPX extension into the AWPX software. Once the correct configuration is imported, you can modify it or save it directly to the device.

5. Export Configuration

After selecting a device, click the ‘Export Configuration’ button to export the current configuration parameters as a configuration file with the AWPX extension. This allows you to quickly import the locally saved configuration file via the ‘Import Configuration’ button during future use of the AWPX software.

6. System Settings

The ‘System Settings’ button includes three sub-options: ‘Network Settings’, ‘Firmware Upgrade’, and ‘About’. The ‘About’ section provides version information related to the AWPX configuration software. The following sections will focus on describing the ‘Network Settings’ and Firmware ‘Upgrade’ functions.

The System Settings button includes three sub-options: Network Settings, Firmware Upgrade, and About. The About section provides version information related to the AWPX configuration software. The following sections will focus on describing the Network Settings and Firmware Upgrade functions

Network Settings: This function allows you to modify the network IP address for the PXB-81xx series products.

Firmware Upgrade: Click System Settings, and then click Firmware Upgrade from the drop-down menu. The AWPX upgrade interface will appear, as shown in Figure 3.21.

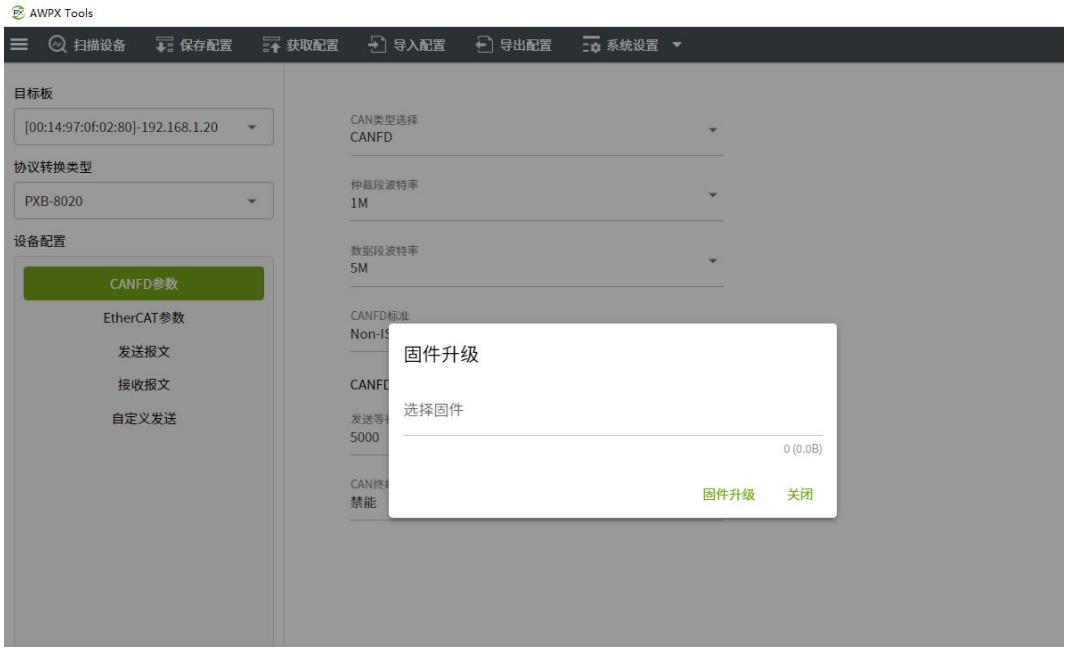


Figure 3.21 Firmware Upgrade

Click the ‘Select Firmware’ button in the selection box, choose the firmware file (.bin) you want to upgrade, and then click ‘Firmware Upgrade’. The AWPX software will prompt that it is downloading the firmware to the device. The entire upgrade process takes about 2 minutes.

After approximately 1 minute, AWPX will indicate that the firmware download is successful, and the device will restart automatically. Please wait patiently for about 1 minute. Do not disconnect the power supply during this restart process. Once the firmware upgrade is complete, click the ‘Scan Device’ button again to rescan and select the device for configuration.。

3.3 Device Configuration

The main steps for configuring a device using the AWPX software are as follows:

Step 1: Click the Scan Device button and select the correct device. If no device is scanned and selected, you will not be able to configure the PXB series protocol converter.

Step 2: Configure the parameters. You can set the parameters in the Device Configuration section and the network parameters according to your requirements.

Step 3: After completing the parameter configuration, click the Save Configuration button to save the parameters to the device. Wait until the device finishes restarting.

3.3.1 Tag Variable Settings

In the Device Configuration section of the AWPX software, click ‘Tag Variable’ to enter the tag variable configuration page. To configure tag variables, you must first create a tag group, and then create tag variables within that group.

1. Tag Group

On the ‘Tag Variable’ page, click the ‘Add’ button in the upper right corner to add a tag group. The group name must be a unique string with a maximum length of 16 bytes, containing only alphanumeric characters (a-z, A-Z, 0-9). When creating a new tag group, the system sets the default name to "group". AWPX supports the creation of multiple tag groups.

2. Tag Variable

After a tag group is created, you can add tag variables within it. There is an Add Data button on the tag group. When adding a tag variable, you need to configure the following fields: Tag Name, Tag Type, Write Permission, and Initial Value. The Tag Name must be a unique string within the tag group, with a maximum length of 16 bytes. Adding a tag variable is shown in Figure 3.24.

Figure 3.22 Add Tag Variable

3.3.2 EtherNet/IP Protocol Settings

1. EtherNet/IP parameters

Click EtherNet/IP Parameters under EtherNet/IP Protocol to configure the device's EtherNet/IP parameters, as shown in Figure 3.23.

Figure 3.23 EtherNet/IP Parameter Settings

The EtherNet/IP parameter descriptions are shown in Table 3.1.

Table 3.1 EtherNet/IP Parameter Description

Parameter	Parameter Description
Device Name	Set the device name for this EtherNet/IP slave

Parameter	Parameter Description
Device Receive Channel Capacity	Set the size of the EtherNet/IP receive channel
Device Send Channel Capacity	Set the size of the EtherNet/IP send channel

2. Input/Output Channel Capacity

To bind the EtherNet/IP receive or send data to tag variables, click ‘Input Channel Capacity’ or ‘Output Channel Capacity’ under the ‘EtherNet/IP Protocol’ section. Please refer to Figure 3.24 for the configuration interface.

Figure 3.24 EtherNet/IP Input/Output Channel

3.3.3 CAN FD Protocol Parameter Configuration

1. CANFD Parameters

Click ‘CAN FD Protocol’ > ‘CAN FD Parameters’ to configure the device's CAN FD parameters, as shown in Figure 3.25.

Figure 3.25 CAN FD Parameter Interface

The description of the CAN FD parameters is shown in Table 3.2.

Table 3.2 CAN FD Parameter Description

Parameter	Parameter Description
CAN Type Selection	Select CAN or CAN FD as the communication type
Arbitration Phase Baud Rate	Set the arbitration phase baud rate for CAN or CAN FD messages
Data Phase Baud Rate	Valid only when the type is set to CAN FD. Sets the data phase baud rate for CAN FD messages
CAN FD Standard	Valid only when the type is set to CAN FD. Specifies the CAN FD standard used for the messages
Transmission Delay Time	Sets the delay time before the device starts transmitting messages after power-on initialization is complete.
Termination Resistor Enable	Enable or disable the termination resistor on the device's CAN FD interface

2. Transmit Message

The available message types depend on the ‘CAN Type Selection’ setting. If ‘CAN Type Selection’ is set to CAN, the device can only transmit CAN messages. If it is set to CAN FD, the device can transmit CAN, CAN FD, and CAN FD BRS (Bit Rate Switch) messages.

Click ‘Send Message’ under ‘CAN FD Protocol’ to configure the message type and other parameters. The interface is shown in Figure 3.26.

Figure 3.26 Transmit Message Interface

Click the [Add +] button in the upper-right corner of the interface to add transmit messages. A maximum of 128 transmit messages can be added. Click the [+ Add Data] button to add mapped data variables, with a maximum of 32 variables allowed per message. The [Clone] button can be used to batch clone tag groups or tags. Click [Delete Selected] to batch delete selected entries, or click [Delete] to remove individual entries. The parameter descriptions for the Transmit Message interface are shown in Table 3.2.

Table 3.3 发送报文界面参数说明

Parameter	Parameter Description
Message Name	The name of the message.
Frame ID	Transmission Frame ID (decimal or hex starting with "0x").
Frame Type	Standard or Extended Frame.
Remote Frame	Specifies whether the transmitted frame is a Remote Frame. This option is invalid when the CAN Type is set to CAN FD.
CAN Type	Selects the type of the transmitted frame. Options include CAN, CAN FD, and CAN FD BRS (Bit Rate Switch).
Data Length	Sets the data length of the transmitted frame. The maximum length is 8 bytes for CAN frames, and up to 64 bytes for CAN FD and CAN FD BRS frames.
Trigger Mode	Sets the trigger mode for the device to transmit CAN messages.
Trigger Frame ID	Sets the Frame ID that triggers the device to transmit a CAN message. This parameter is valid only when the Trigger Mode is set to [Frame ID Trigger]. It can be in decimal or hexadecimal format (hexadecimal values must start with "0x").
Trigger Frame Type	Sets the Frame Type that triggers the device to transmit a CAN message. This parameter is valid only when the Trigger Mode is set to ‘Frame ID Trigger’.
Cycle Time	When Trigger Mode is set to [Periodic Transmission], this is the transmission cycle time. When Trigger Mode is set to [Change Transmission], this is the polling cycle for checking tag value changes. When Trigger Mode is set to [Single Transmission], this is the delay time before the single transmission occurs. When Trigger Mode is set to [Cycle Group Transmission], this is the transmission interval between two CAN messages within a cycle group. When Trigger Mode is set to [Conditional Transmission], this is the periodic transmission cycle time after the corresponding tag value of the Condition Tag Name becomes True.
Cycle Group ID	The ID of the cycle group. A cycle group can be understood as a transmission list. A Cycle Group ID represents one transmission list, and the messages within the list will be transmitted cyclically.
Cycle Group Index	The sequence number of the transmitted message within the cycle group. Messages with a smaller index will be transmitted first.

Parameter	Parameter Description
Condition Tag Name	The tag name used to check if its value is True. This parameter is valid only when the Trigger Mode is set to [Conditional Transmission], and the corresponding tag type must be set to Bool. When the tag value becomes True, the device will periodically transmit the CAN message.
Variable Name	The name of the variable.
Tag Name	The name of the tag. For example, group_1.name1 represents the tag named name1 within the tag group group_1.
Offset	The bit offset of the CAN message, in bits. A value of 1 indicates that the mapping starts from the first bit of the data field. The mapping size depends on the tag type.

Trigger Modes for Transmitting CAN Messages:

Periodic Transmission: When this mode is selected, the device will transmit the CAN message cyclically based on the ‘Cycle Time’.

Change Transmission: When this mode is selected, the device will transmit a CAN message whenever the value of any tag within the transmission entry changes.

Single Transmission: When this mode is selected, the CAN message will be transmitted only once.

Frame ID Trigger: When this mode is selected, the device will transmit a CAN message upon receiving a CAN message that matches both the configured ‘Trigger ID’ and ‘Trigger Frame Type’.

Cycle Group Transmission: When this mode is selected, messages with the same Cycle Group ID will be transmitted cyclically based on their sequence number. Messages with a lower sequence number are transmitted first. The transmission interval between two CAN messages is defined by the ‘Cycle Time’.

Conditional Transmission: When this mode is selected, the device will periodically transmit the CAN message when the value of the tag specified by the ‘Condition Tag Name’ becomes True.

Example of Message Transmission Configuration: The message transmission configuration interface is shown in Figure 3.27.

The device will read the value of the temp tag and map it to the first and second bytes of the CAN message data field. If the value of temp is 0x1122, the device will transmit a standard CAN frame with an ID of 0x01 every 1000 ms. The transmitted frame will be: 00 22 11 00 00 00 00 00 (in hexadecimal).

Transmitted Message Counter:The device supports counting successfully transmitted CAN messages. The counter value is stored in a dedicated transmission counter tag. The configuration steps are as follows: Create a canfdsys tag group -> Create a transmission counter tag using the format tc_ + Frame ID. The Frame ID must be in hexadecimal format, padded to 8 bytes, and the tag type must be set to uint64. For example, to count the transmission of messages with Frame IDs

0x1 and 0x1A000000, the tag names should be set to tc_00000001 and tc_1A000000, as shown in Figure 3.27.

Figure 3.27 Transmitted Message Counter Configuration

3. Received Messages

The types of receivable messages depend on the ‘CAN Type Selection’ setting. If ‘CAN Type Selection’ is set to CAN, the device can only receive CAN messages. If it is set to CAN FD, the device can receive CAN, CAN FD, and CAN FD BRS messages.

Click ‘Received Messages’ under ‘CAN FD Protocol’ to configure parameters such as the frame ID and frame type for receiving messages. The interface is shown in Figure 3.28.

Figure 3.28 Received Messages Interface

Click the [Add +] button in the upper-right corner of the interface to add received messages. A maximum of 128 messages can be added. Click the [+ Add Data] button to add variables for data mapping. Up to 32 variables can be added per message. The [Clone] button allows you to batch clone tag groups or tags. Click [Delete Selected] to batch delete selected items, or click [Delete] to remove individual items. For parameter descriptions of the Received Messages interface, see Table 3.4.

Table 3.4 Received Messages Interface Parameter Description

Parameter	Parameter Description
Message Name	The name of the message.
Frame ID	Sets the frame ID of the received message. It can be in decimal or hexadecimal

Parameter	Parameter Description
	format (hexadecimal values must start with 0x).
Frame Type	Sets whether the received message is a standard frame or an extended frame.
Variable Name	The name of the variable.
Tag Name	The name of the tag. For example, group_1.name1 represents a tag named name1 within the group_1 tag group.
Offset	The offset of the CAN message, measured in bits. Setting it to 1 means the mapping starts from the first bit of the data field of the received CAN message. The mapping size depends on the tag type.

Receiving Message Configuration Example: The interface for configuring receiving messages is shown in Figure 3.29.

When the device receives a standard CAN frame with a frame ID of 0x2 and frame data of 11 22 33 44 55 66 77 88 (in hexadecimal), it will map two bytes of data starting from the second byte of the CAN frame data section to the tag flow. The value of the tag flow will then become 0x4433.

Receive Message Counter: The device supports counting successfully received CAN messages, with the count value stored in a designated receive counter tag. The setup steps are as follows: Create a canfdsys tag group -> Create a receive counter tag formatted as rc_ + Frame ID. The Frame ID must be in hexadecimal and padded to 8 bytes. The tag type must also be set to uint64. For example, to count messages with frame IDs 0x1 and 0x1A000000, the tag names should be set to rc_00000001 and rc_1A000000, as shown in Figure 3.29.

Figure 3.29 Receive Message Counter Configuration

3.3.4 CANopen Master Protocol Configuration

1. CANopen Master Parameters

In the Device Configuration pane of the AWPX software, click [CANopen Master Protocol] and select [CANopen Master Configuration] to configure the relevant CANopen master parameters. The configuration interface is shown in Figure 3.30.

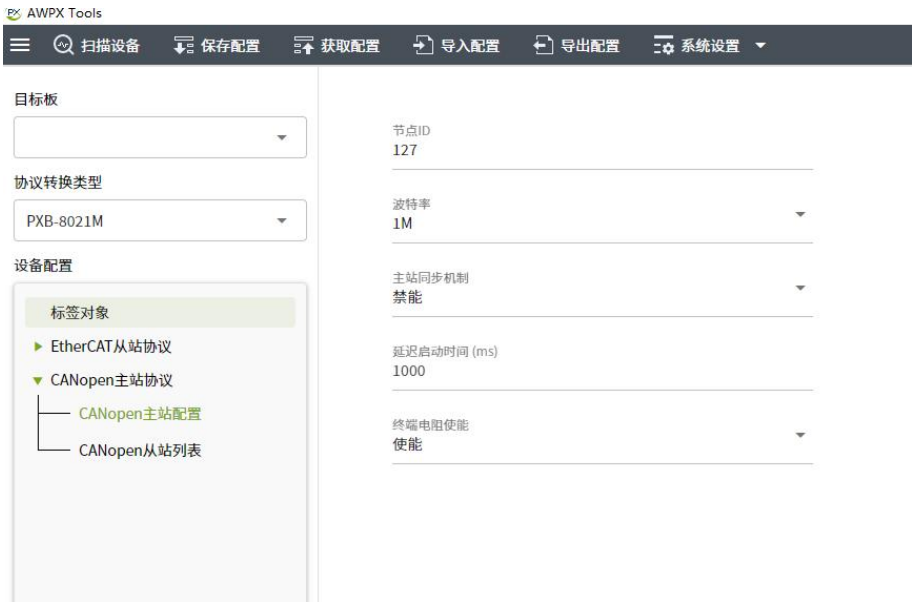


Figure 3.30 CANopen Master Parameter Configuration Interface

The description of the CANopen master parameters is shown in Table 3.5.

Table 3.5 CANopen Master Parameter Description

Parameter	Parameter Description
Node ID	Master Node ID
Baud Rate	Set the CAN baud rate of the master (10K, 20K, 50K, 100K, 125K, 250K, 500K, 800K, 1M).
Master Synchronization Mechanism	Set whether to enable the master synchronization message mechanism.
Delayed Start Time	Set whether to delay the start of the master.
Terminating Resistor Enable	Enable or disable the terminating resistor.

2. CANopen Slave Parameters

After clicking ‘CANopen Master Protocol’ in the Device Configuration pane, click ‘CANopen Slave List’ to view the established CANopen slave nodes. You can also right-click to add or delete CANopen slave nodes. The interface is shown in Figure 3.31 and Figure 3.32.

Figure 3.31 Create Slave Node

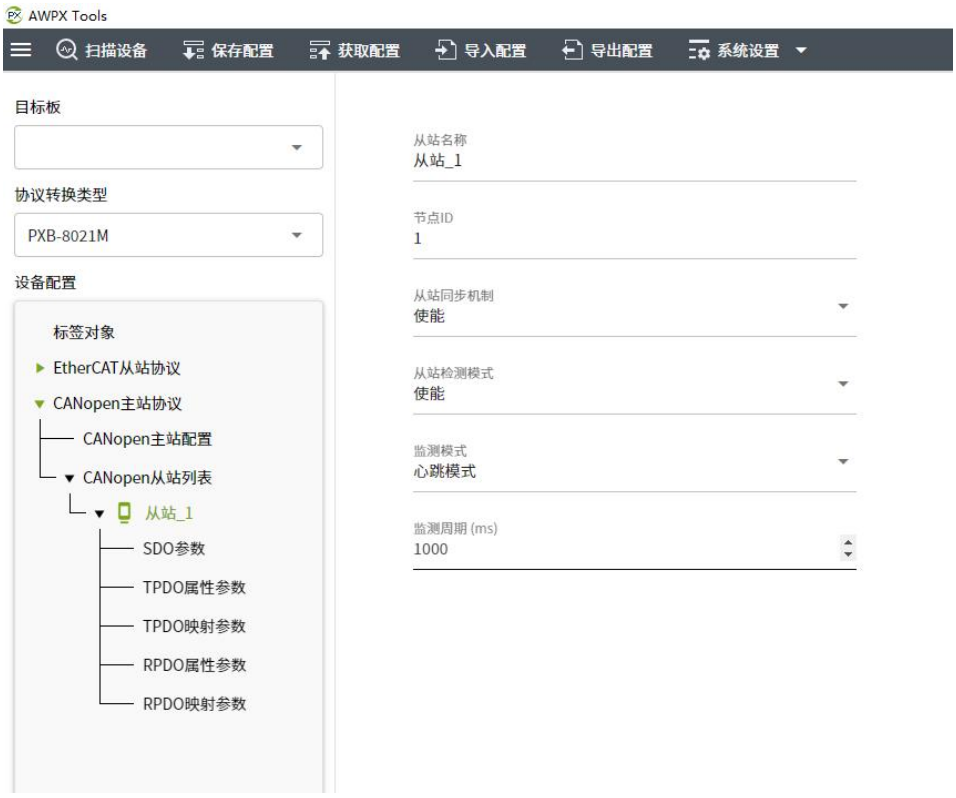


Figure 3.32 Slave Configuration Interface

The description of the CANopen slave parameters is shown in Table 3.6.

Table 3.6 Slave Configuration Interface

Parameter	Parameter Description
Slave Name	The name of the slave node, used as a user mnemonic.
Node ID	The Node ID of the slave.
Slave Synchronization Mechanism	Set whether to configure the slave synchronization message mechanism.
Slave Detection Mode	Set whether to enable the detection mode of the slave.
Detection Mode	When the detection mode is enabled, configure the node guarding mode or the heartbeat mode.
Detection Cycle	The detection time cycle (ms).

1. SDO Parameter Configuration

Click ‘SDO Parameters’ in the Device Configuration pane. The interface is shown in Figure 3.33.

Figure 3.33 SDO Parameter Configuration

The description of the CANopen slave SDO parameters is shown in Table 3.7.

表 3.7 SDO Configuration Parameter Description

Parameter	Parameter Description
Main Index	The main index of the slave dictionary. It can be in decimal or hexadecimal format (must start with 0x).
Sub-Index	The sub-index of the slave dictionary.
Description	User remarks or notes.
SDO Transfer Method	Currently, only the download protocol is supported.
Tag Name	The data payload transmitted by the SDO.

2. TPDO Attribute Parameters

Click [TPDO Attribute Parameters] in the Device Configuration pane. The interface is shown in Figure 3.34.

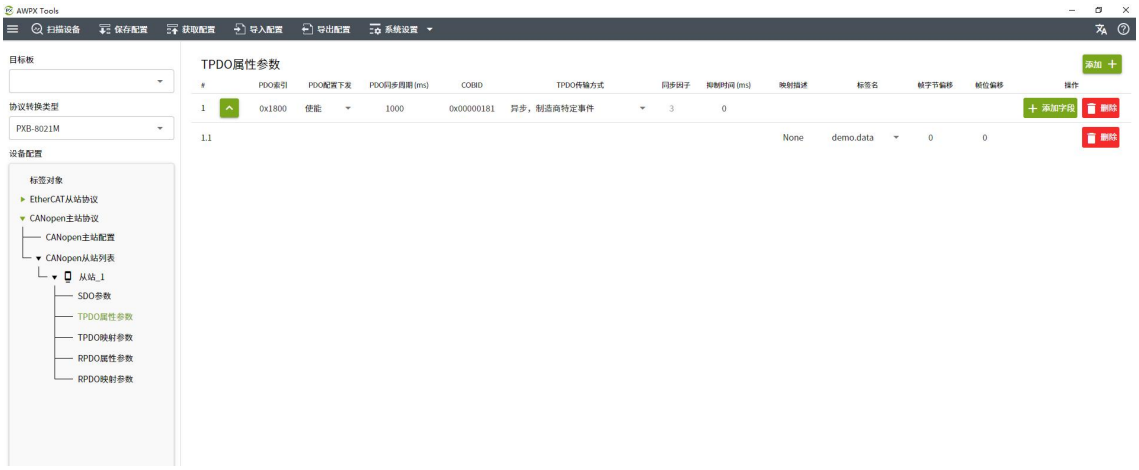


Figure 3.34 TPDO Attribute Parameters

The description of the CANopen slave TPDO attribute parameters is shown in Table 3.8.

Table 3.8 TPDO Attribute Parameters

Parameter	Parameter Description
PDO Index	The main index of the slave dictionary TPDO. It can be in decimal or hexadecimal format (must start with 0x).
Download PDO Config	Determines whether to download the current PDO configuration to the slave.
PDO Sync Cycle	The periodic sending cycle of the slave.

Parameter	Parameter Description
	(0 indicates sending upon data change).
COBID	The COB-ID of the slave dictionary TPDO attribute parameters. It can be in decimal or hexadecimal format (must start with 0x).
TPDO Transmission Type	The transmission type selection for the slave TPDO.
Sync Factor	In synchronous cyclic mode, the slave sends data once after receiving N SYNC messages.
Inhibit Time	The minimum cycle between two consecutive message transmissions from the slave.
Mapping Description	User remarks or notes.
Tag	The data distribution object for the TPDO message.
Frame Byte Offset	The byte offset of the PDO. (Distributes the byte at the offset to the corresponding tag).
Frame Bit Offset	The bit offset of the PDO. (Only applicable when the tag type is a bit).

3. TPDO Mapping Parameters

Click ‘TPDO Mapping Parameters’ in the Device Configuration pane. The interface is shown in Figure 3.35.

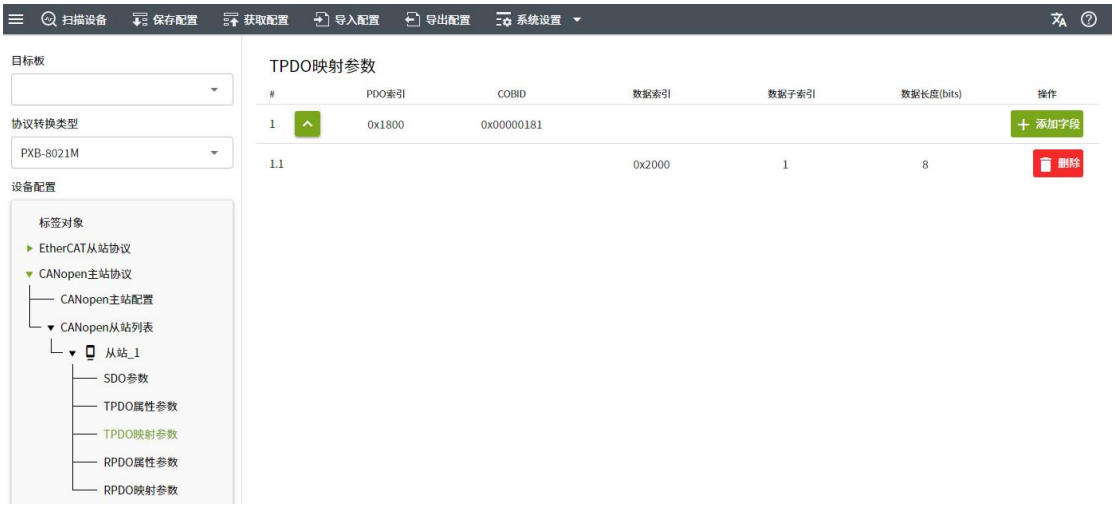


Figure 3.35 TPDO Mapping Parameters

The description of the CANopen slave TPDO mapping parameters is shown in Table 3.9.

Table 3.9 TPDO Mapping Parameter Description

Parameter	Parameter Description
PDO Index	Automatically populated by AWPX and cannot be modified by the user.
COBID	Automatically populated by AWPX and cannot be modified by the user.
Data Index	The main index of the slave dictionary mapping parameters. It can be in decimal or hexadecimal format (must start with 0x).
Data Sub-index	The sub-index of the slave dictionary mapping parameters.
Data Length (bit)	The length of the mapped data (1-64).

4. RPDOAttribute Parameters

点击“设备配置”栏的【RPDO 属性参数】后，其界面如图 3.38 所示。

Figure 3.36 RPDO Attribute Parameters

CANopen slave RPDO attribute parameters are described in Table 3.10.

Table 3.10 RPDO 属性参数说明

Parameter	Parameter Description
PDO Index	The main index of the slave dictionary RPDO. It can be in decimal or hexadecimal format (must start with 0x).
Download PDO Config	Determines whether to download the current PDO configuration to the slave.

Parameter	Parameter Description
PDO Sync Cycle	The periodic sending cycle of the master. (0 indicates sending upon data change).
COBID	The COB-ID of the slave dictionary RPDO attribute parameters. It can be in decimal or hexadecimal format (must start with 0x).
RPDO Transmission Type	The transmission type selection for the slave RPDO.
Sync Factor	Reserved.
Inhibit Time	Reserved.
Mapping Description	User remarks or notes.
Tag	The tag data to be filled into the corresponding PDO message.
Frame Byte Offset	The byte offset of the PDO. (The tag data is filled into the corresponding PDO message).
Frame Bit Offset	The bit offset of the PDO. (Only applicable when the tag type is a bit).

5. RPDO Mapping Parameters

After clicking [RPDO Mapping Parameters] in the Device Configuration section, the interface is shown in Figure 3.37.

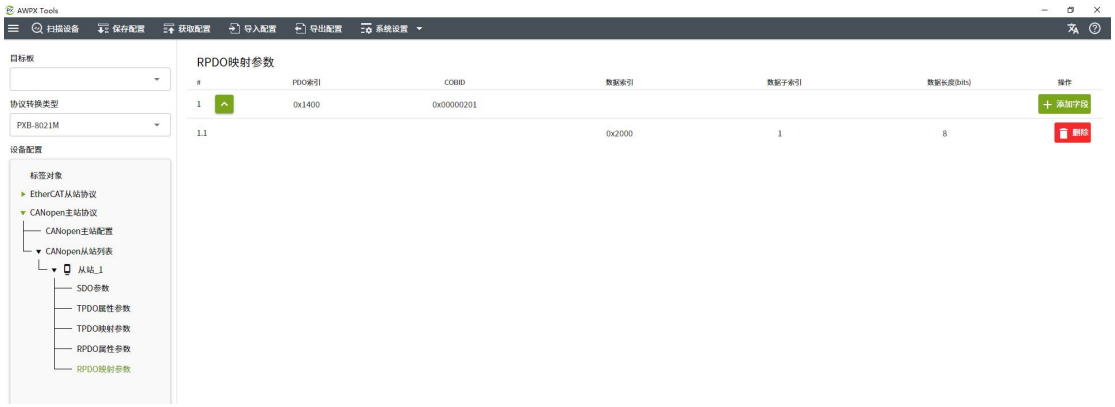


Figure 3.37 RPDO Mapping Parameters

The description of the CANopen slave RPDO mapping parameters is shown in Table 3.11.

Table 3.11 RPDO Mapping Parameter Description

Parameter	Parameter Description
PDO Index	Automatically populated by AWPX. Cannot be modified by the user.
COBID	Automatically populated by AWPX. Cannot be modified by the user.
Data Index	The main index of the slave dictionary mapping parameter. Can be in decimal or hexadecimal format (hexadecimal values must start with 0x).
Data Sub-index	The sub-index of the slave dictionary mapping parameter.
Data Length (bits)	The length of the mapped data (1 – 64 bits).

3.3.5 DeviceNet Master Protocol Parameter Configuration

1. DeviceNet Master Parameters

In the Device Configuration section of the AWPX software, click [DeviceNet Master Protocol] to configure the relevant parameters of the DeviceNet master. The configuration interface is shown in Figure 3.38.

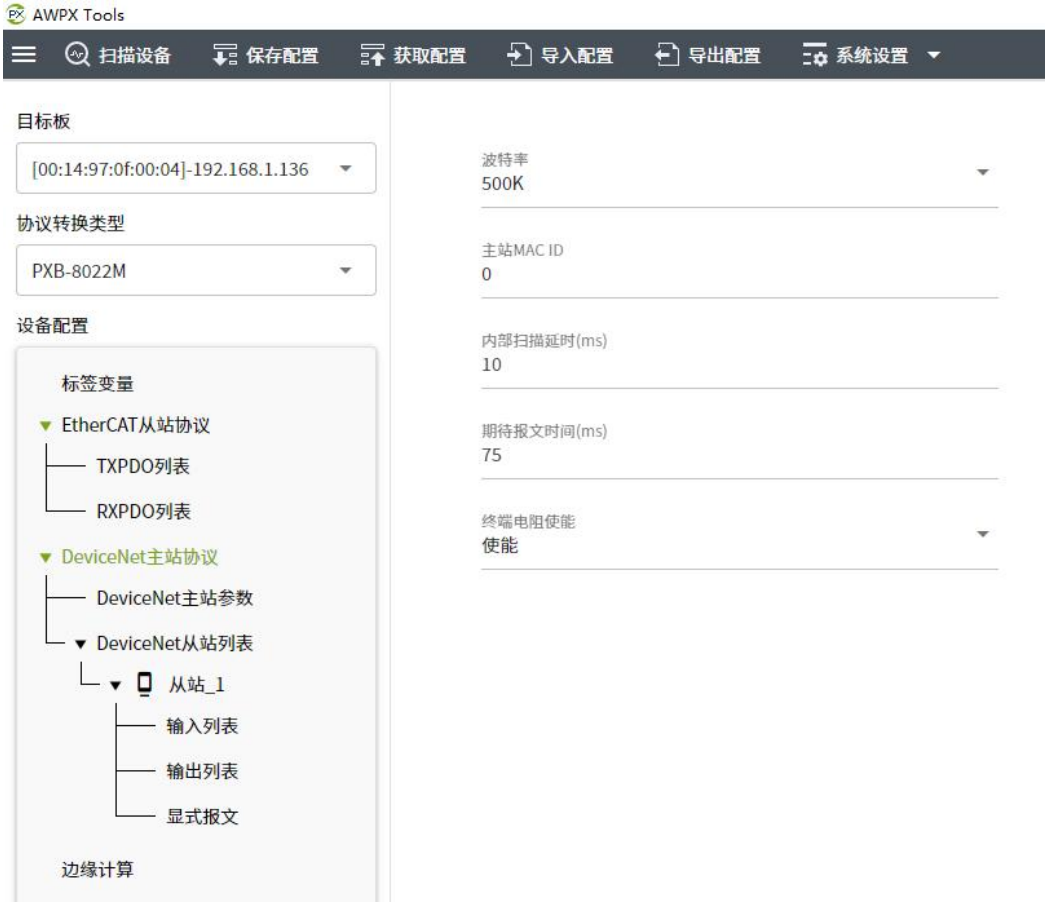


Figure 3.38 DeviceNet Master Parameter Interface

The description of the DeviceNet master parameters is shown in Table 3.12.

Table 3.12 DeviceNet Master Parameter Description

Parameter	Parameter Description
Baud Rate	The CAN baud rate of the PXB-8122M.
Master MAC ID	The device address of the PXB-8122M when acting as a DeviceNet master.
Internal Scan Delay	The minimum time allowed for external devices to access the network after the scanner's continuous I/O scan. A value that is too large will result in a longer network scan, which may affect I/O execution. Conversely, a value that is too small will cause the scanner module to respond slowly to external devices.
Expected Packet Time	The expected packet rate, which determines the message timeout for bit strobes and polls.
Termination Resistor Enable	Enables or disables the termination resistor on the CAN interface.

2. DeviceNet Slave List

Click ‘DeviceNet Master Protocol’ in the Device Configuration section, and then click [DeviceNet Slave List] to view the established DeviceNet slave nodes. You can also right-click to add or remove DeviceNet slave nodes. The interface is shown in Figure 3.39.

Figure 3.39 DeviceNet Slave List Interface

3. DeviceNet Slave Attributes

Click ‘Create Slave’ to open the DeviceNet slave configuration interface, which is used to configure the relevant attribute information of the DeviceNet slave to be connected. The interface is shown in Figure 3.40.

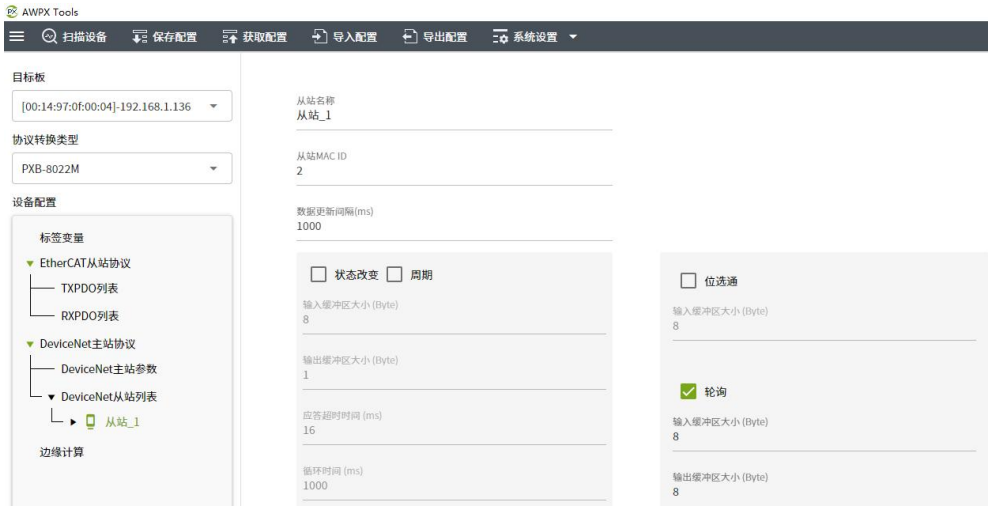


Figure 3.40 DeviceNet Slave Attribute Interface

On the left side of Figure 3.40, click the corresponding slave name to configure the relevant attribute information of the DeviceNet slave. The attribute information description is shown in Table 3.13.

Table 3.13 DeviceNet Slave Attribute Description

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

Below the ‘Data Update Interval’ parameter, you can select the I/O connection type. There are four I/O connection types available: Bit Strobe, Polled, Status Change, and Cyclic. Each DeviceNet slave must have at least one I/O connection type selected, and a maximum of three can be selected. Note that Status Change and Cyclic cannot be selected simultaneously.

1、 Bit Strobe: Requires configuring the input buffer size of the DeviceNet slave. The configurable size ranges from 1 to 64 Bytes;

2、Polled: Requires configuring both the input and output buffer sizes of the DeviceNet slave. The configurable size for both ranges from 1 to 64 Bytes;

3、Status Change: When the Status Change type is selected, the interface is shown in Figure 3.42.

Figure 3.41 Status Change Method

The parameter description for this interface is shown in Table 3.14.

Table 3.14 Status Change Parameters

Parameter	Parameter Description
Input Buffer Size	The maximum size of I/O data that can be input to this DeviceNet slave per cycle. Range: 1-64 Bytes.
Output Buffer Size	The maximum size of I/O data that this DeviceNet slave can output per cycle. Range: 1-64 Bytes.
Response Timeout	The time interval from when the DeviceNet slave sends a message due to a status change to when it receives the response from the DeviceNet master.
Heartbeat Interval	The time interval at which the DeviceNet slave sends heartbeat messages. When using the status change method, heartbeat messages can periodically query the device status to prevent the DeviceNet slave from going offline.
Message Production Time	The message production time of the DeviceNet slave, which must be less than the heartbeat interval.

4、Cyclic: When the cyclic type is selected, the interface is shown in Figure 3.42.

Figure 3.42 Cyclic Mode

The parameter descriptions for this interface are shown in Table 3.15.

Table 3.15 Cyclic Mode Parameter Description

Parameter	Parameter Description
Input Buffer Size	The communication cycle interval of the DeviceNet slave, which can be adjusted to reduce unnecessary network traffic.
Output Buffer Size	The maximum size of I/O data that this DeviceNet slave can output per transaction. Range: 1 – 64 Bytes.
Response Timeout	The time from when the DeviceNet slave sends a message to when it receives a response from the DeviceNet master.
Cycle Time	The communication cycle interval of the DeviceNet slave, which can be adjusted to reduce unnecessary network traffic.

4. Input and Output List

On the ‘DeviceNet Slave List’ interface, click ‘Input List’ or ‘Output List’ below the slave name. The interface is shown in Figure 3.43. Click the ‘▼’ button to the left of the slave name to hide the output list, input list, and explicit messages of that slave. The ‘Output List’ and ‘Input List’ are defined relative to the DeviceNet master. The ‘Output List’ contains the mapping entries for I/O data output by the DeviceNet master, while the ‘Input List’ contains the mapping entries for I/O data input to the DeviceNet master.

Figure 3.43 Output List Interface

Click the [Add +] button in the upper-right corner of the interface to add a mapping entry, and then you can edit the mapping parameters. To delete a mapping entry, click the [Delete] button on the far right of the entry. The descriptions of the output list parameters are shown in Table 3.16.

Table 3.16 Input and Output List Parameter Description

Parameter	Parameter Description
Variable Name	The name of this mapping entry, which can be used as a mnemonic.
IO Type	Selects the checked I/O connection type.
Tag Name	Specifies the tag name mapped to the current I/O data offset. This tag must exist in

	the tag variables. For the output list, data is written from the specified tag to the DeviceNet I/O data; for the input list, data is read from the DeviceNet I/O data to the specified tag.
Offset	Specifies the bit position of the current I/O type input/output data.

5. Explicit Message

On the ‘DeviceNet Slave List’ interface, click ‘Explicit Message’ below the slave name. The interface is shown in Figure 3.46. Click the ‘▼’ button to the left of the slave name to hide the output list, input list, and explicit messages of that slave. After an explicit message connection is established, you can use this interface to configure various parameters and exchange information with the DeviceNet slave.

Figure 3.44 Explicit Message Interface

Click the [Add +] button in the upper-right corner of the interface to add a mapping entry, and then you can edit the mapping parameters. To delete a mapping entry, click the [Delete] button on the far right of the entry. The descriptions of the explicit message parameters are shown in Table 3.17.

Table 3.17 Explicit Message Parameter Description

Parameter	Parameter Description
Message Name	The name of this message, which can be used as a mnemonic.
Service Code	Specifies the service code corresponding to this explicit message.
Class ID	Specifies the class ID for configuring or retrieving attributes via the explicit message.
Instance ID	Specifies the instance ID for configuring or retrieving attributes via the explicit message.
Request Data Length	Specifies the length of the service data sent in this explicit message, in bytes.
Request Timeout	Specifies the maximum time (in ms) allowed for the DeviceNet slave to respond correctly after the explicit message is sent. Failure to respond within this time results in a request timeout.
Request Type	Specifies the method for sending this explicit message. Options include ‘Timed Request’ and ‘Single Request’. ‘Timed Request’: After the system starts, this explicit message is sent cyclically at the interval specified by ‘Request Period’. ‘Single Request’ means that after the system starts, this explicit message is sent only once following the delay specified by ‘Request Period’.
Request Period	Specifies the time interval (in ms) for sending ‘Timed Request’ and ‘Single

	Request' messages.

By clicking 'Add Data', you can establish a mapping relationship between the request or response data of an explicit message and a tag variable. The Data Type is used to select either request or response data; the Tag Name is used to set the mapped tag variable; and the Offset is used to specify the data position.

6. Slave Status

Retrieve the status of DeviceNet master-managed slaves using tag variables. Create a tag group named dvnmst_sys in the tag configuration interface, and add a uint16 tag variable named slvX_status (where X is the slave's MAC ID; for example, use slv2_status for MAC ID 2). Refer to Table 3.18 for slave status value descriptions.

Table 3.18 DeviceNet Slave Status Value Description

Status Value (Decimal)	Status Value Description
0	No Error
57	Other Undefined Internal Error
61	Slave Not Found
62	Data Transmission Failed
63	No Data
66	Master Not Online
67	Connection Does Not Exist
72	Device Communication Stopped
77	Data Length Mismatch
78	Device in Scan List but No Response
84	Device Not Initialized (Master Status)
86	Device Entered Idle State Automatically

3.3.6 Modbus Protocol Parameter Configuration

1. Modbus Settings

In the Device Configuration pane of the AWPX software, click 'Modbus Protocol' and then click 'Modbus Settings' to configure the Modbus parameters. The configuration interface is shown in Figure 3.45.

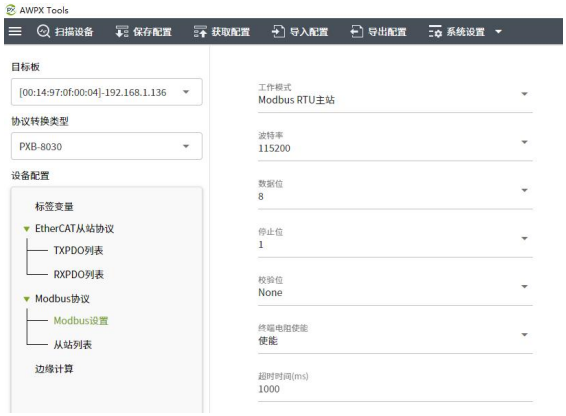


Figure 3.45 Modbus Settings

The Modbus protocol supports four operating modes: Modbus RTU Master, Modbus RTU Slave, Modbus TCP Master, and Modbus TCP Slave. You can select the desired mode from the Operating Mode drop-down list. Each operating mode has its corresponding parameters for configuration.

- Modbus RTU Master

When the operating mode is set to Modbus RTU Master, the Modbus configuration parameters are shown in Figure 3.46. Baud Rate, Data Bits, Stop Bits, and Parity are used to configure the RS485 communication parameters as required. Terminal Resistor Enable determines whether the onboard RS485 terminal resistor is connected to the network. Timeout sets the timeout period for RTU commands, measured in milliseconds (ms). The Modbus RTU Master mode supports communication with multiple RTU slaves.

Figure 3.46 Master Station Parameters

- Modbus RTU Slave

When the operating mode is set to Modbus RTU Slave Mode, the Modbus setting parameters are shown in Figure 3.52. "Baud Rate", "Data Bits", "Stop Bits", and "Parity Bit" are used to configure the RS485 communication parameters as needed. "Local Slave Station ID" sets the Modbus ID for this device when acting as a slave station. "Termination Resistor Enable" controls whether the onboard RS485 termination resistor is connected to the network.

Figure 3.47 RTU Slave Station Parameters

- Modbus TCP Master

When the operating mode is set to Modbus TCP Master Mode, the Modbus setting parameters are shown in Figure 3.53. Modbus TCP Master Mode supports access to a maximum of 8 TCP slave stations. You can click "Add" to add slave station information. The "Slave Station ID" is used to distinguish entries in the slave station list and must be unique.

Figure 3.48 TCP Master Station Parameters

- Modbus TCP Slave

When the operating mode is set to Modbus TCP Slave Mode, the Modbus setting parameters are shown in Figure 3.54. The "Local Slave Station ID" is used to set the ID of this device when acting as a slave station.

Figure 3.49 TCP Slave Station Parameters

2. Slave Station List

Right-click on the "Slave Station List" under "Modbus Protocol". In the pop-up menu, you can select "Create Slave Station" or "Delete Slave Station". When Modbus is in Master Mode, multiple slave stations can be created; however, in Slave Mode, only one slave station is allowed. Please note that the parameter settings and function code configurations for slave stations in Master Mode differ from those in Slave Mode.

- Master Mode

When operating in Master Mode, the slave station parameters are configured as shown in Figure 3.50. "Slave Station ID" sets the connected slave's ID; for RTU, configure as needed, but for TCP, it must match the ID in "Modbus Settings". "Polling Cycle" determines the command sending frequency, which should balance the data volume in the "Function Code List" with real-time application requirements. "Send Mode" controls the writing method with two options: "Trigger Send" (writes when associated data changes) and "Polling Send" (writes based on the polling cycle). "Function Code Retry Count" sets the retry limit upon transmission failure. "Local Data on Read Failure" determines whether associated variables retain their values or clear to zero upon a Modbus read failure. Finally, "Conditional Enable" activates the slave station when the condition variable is "True" and deactivates it when "False".

Figure 3.50 Slave Station Settings in Master Mode

The function code list settings for Master Mode are shown in Figure 3.56. Master Mode supports function codes "01", "02", "03", "04", "05", "06", "15", and "16". Users can configure the "Function Code", "Start Address", and "Register Quantity" as needed. After adding a function code, click the "Add Data" button to bind data variables to the current function code. In this process, "Tag Name" is used to select the variable, and "Offset" is used to set the position of the variable associated with the current function code.

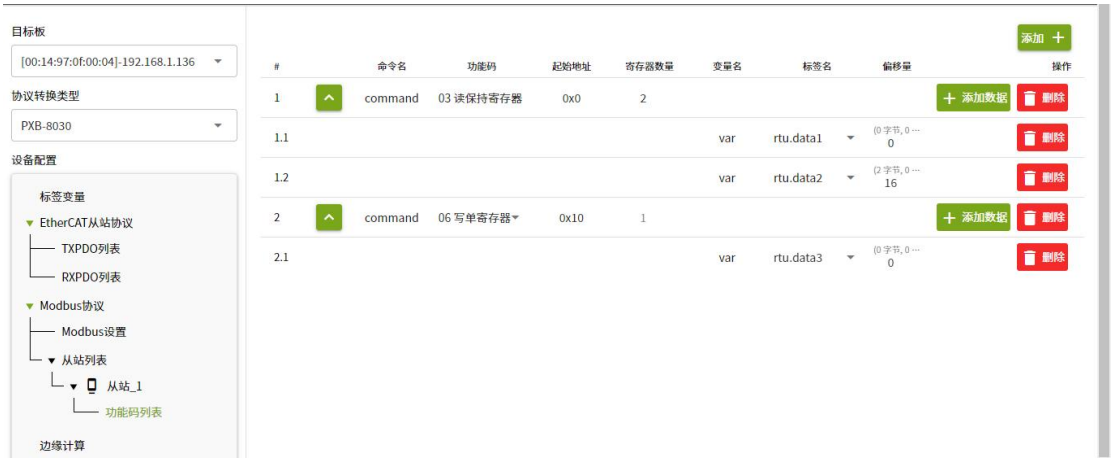


Figure 3.51 Function Code List in Master Mode

- Slave Mode

When the operating mode is set to Slave Mode, the parameter settings for this slave station are shown in Figure 3.52. As can be seen, the configuration is relatively simple, with only the Endianness parameter being configurable.

Figure 3.52 Slave Settings in Slave Mode

The function code list settings for the Slave Mode are shown in Figure 3.58. The available register types include "Coil", "Discrete Input", "Holding Register", and "Input Register". Users can configure the "Register Type", "Start Address", and "Register Quantity" as needed. After adding a function code, click the "Add Data" button to bind data variables to the current function code. In this setting, "Tag Name" is used to select a variable, and "Offset" specifies the position of the variable associated with the current function code.

Figure 3.53 Function Codes in Slave Mode

3.3.7 Edge Computing Parameter Configuration

In the "Device Configuration" section of the AWPX software, click ‘Edge Computing’ to configure the related parameters. The configuration interface is shown in Figure 3.47.

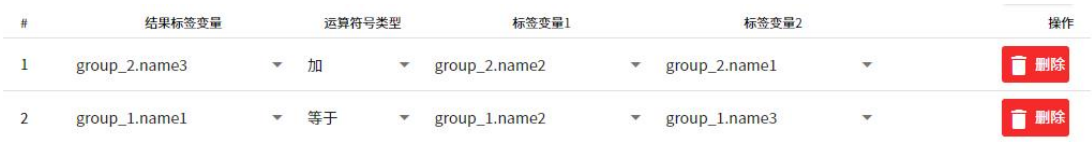


Figure 3.54 Configure Edge Computing

The Edge Computing Configuration Form contains four configuration items: "Result Tag Variable", "Operator Type", "Tag Variable 1", and "Tag Variable 2". The "Result Tag Variable", "Tag Variable 1", and "Tag Variable 2" must be existing tag variables from the tag variable list. For the operator types and operation rules, please refer to Table 3.19.

Table 3.19 Edge Computing Overview

Operator Type	Operation Rule Description
Addition	Result Tag Variable = Tag Variable 1 + Tag Variable 2
Subtraction	Result Tag Variable = Tag Variable 1 - Tag Variable 2
Multiplication	Result Tag Variable = Tag Variable 1 * Tag Variable 2
Integer Division	Result Tag Variable = Tag Variable 1 / Tag Variable 2
Greater Than	Result Tag Variable (Boolean)= Tag Variable 1 > Tag Variable 2
Greater Than or Equal To	Result Tag Variable (Boolean)= Tag Variable 1 >= Tag Variable 2
Equal To	Result Tag Variable (Boolean) = Tag Variable 1 == Tag Variable 2
Not Equal To	Result Tag Variable (Boolean) = Tag Variable 1 != Tag Variable 2
Less Than or Equal To	Result Tag Variable (Boolean) = Tag Variable 1 <= Tag Variable 2
Less Than	Result Tag Variable (Boolean) = Tag Variable 1 < Tag Variable 2
Maximum	Result Tag Variable = Tag Variable 1 > Tag Variable 2 ? Tag Variable 1 : Tag Variable 2
Minimum	Result Tag Variable = Tag Variable 1 > Tag Variable 2 ? Tag Variable 2 : Tag Variable 1
Modulo	Result Tag Variable (Int) = Tag Variable 1 (Int) % Tag Variable 2(Int)
Left Shift	Result Tag Variable (Int) = Tag Variable 1 (Int) << Tag Variable 2(Int)
Right Shift	Result Tag Variable (Int) = Tag Variable 1(Int) >> Tag Variable 2(Int)
AND	Result Tag Variable (Int) = Tag Variable 1(Int) & Tag Variable 2(Int)
OR	Result Tag Variable = Tag Variable 1（Int） | Tag Variable 2（Int）
NOT	Result Tag Variable （Int） = not (Tag Variable 1) （Int）
XOR	Result Tag Variable （Int）= Tag Variable 1（Int）xor Tag Variable 2（Int）
Exponentiation	Result Tag Variable （Int） = Tag Variable 1（Int） ^ Tag Variable 2（Int）
Square Root	Result Tag Variable = √ Tag Variable 1 （Int）
Logarithm	Result Tag Variable = ln(Tag Variable 1)
Conditional Operation	Result Tag Variable = Tag Variable 1（Boolean）? Tag Variable 2：Tag Variable 3

4. Product Installation

4.1 Mechanical Dimensions

PXB-81series product dimensions: 125.00mm×76.00mm×28.00mm (bare metal).

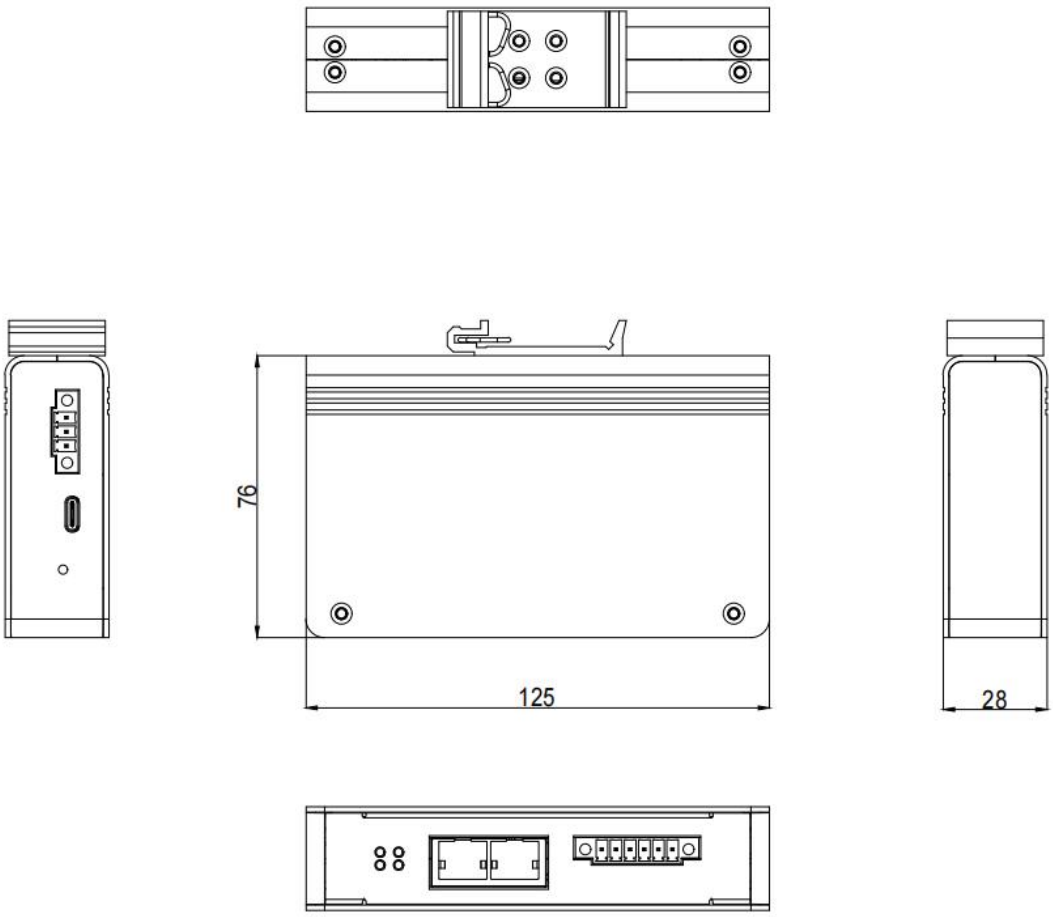


Figure 4.2 Product Dimensional Diagram

4.2 Installation Methods

For product installation methods, please refer to the information on the official website.《PXB Series DIN-Rail Mount Installation Instructions》.

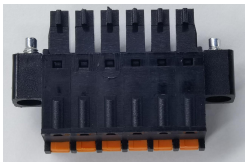
5. Product Maintenance And Precautions

- 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

表 6.1 产品装箱清单

Num	Name	Quantity	Unit	Physical picture
1	PXB-81xx Protocol Converter Gateway	1	PCS	
2	3-Pin connecting terminal	1	PCS	
3	6-Pin connecting terminal	1	PCS	
4	certificate	1	PCS	

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 to lerance 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