User Manual

# PXB-82xx

Profinet Protocol Converter Gateway

UM01010101 0.01 Date:2026/8/10

| category | content |
| --- | --- |
| keyword | PXB-82XX protocol converter, user manual, product manual |
| summary | This document provides customers with instructions on the PXB-82XX series protocol converter, including hardware interface introduction, software |

ZLG 致远电子

**Revision History**

| version | date | reason |
|---|---|---|
| V0.90 | 2024/5/19 | Document release |
| V1.00 | 2026/3/19 | officially released |
| | | |

# Contents

# 1. Product Introduction

## 1.1 Product Overview

The PXB-82 series products are various industrial fieldbus protocol converters such as Profinet to Modbus, CAN/CAN FD, DeviceNet, etc. launched by Guangzhou Zhiyuan Electronics Co., Ltd.This series of products has 2 Profinet slave network port interfaces, 1 CAN FD or 1 RS485 interface, and 1 USB Type-C interface. It is equipped with a high-performance Cortex-R52 processor to process the conversion of Profinet to various industrial fieldbus data messages.The PXB-82 series protocol converter provides a configuration tool for the upper computer, which can flexibly configure relevant functions and easily achieve seamless conversion of various industrial bus protocol data to Profinet data.


Figure 1.1 Product Series Group Photo

## 1.2 Product Series Description

### 1.2.1 Naming Convention

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

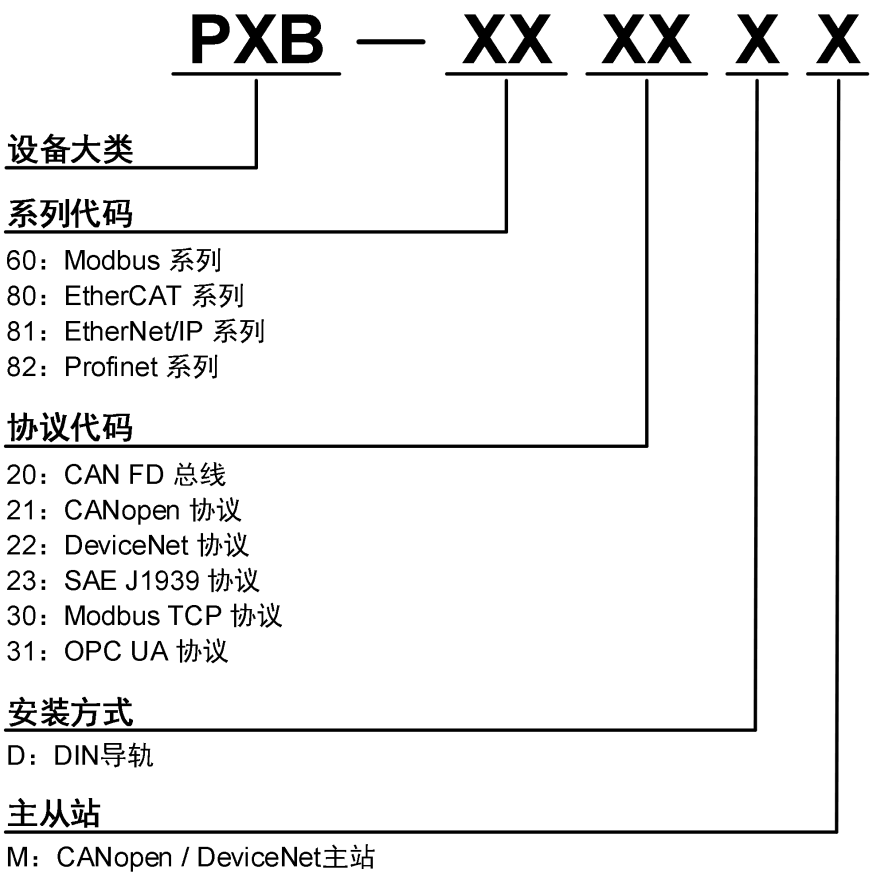



Figure 1.2 Naming Rules for Series Products

### 1.2.2 Order Information

Table 1.1 Ordering Information for Series Products

| Product Model | Protocol conversion type | Installation method |
|---|---|---|
| PXB-8210 | Profinet <=> Modbus | 35mm DIN rail |
| PXB-8220 | Profinet <=> CAN/CAN FD | 35mm DIN rail |
| PXB-8221M | Profinet<=>CANopen (master station) | 35mm DIN rail |
| PXB-8222M | Profinet<=>DeviceNet (master station) | 35mm DIN rail |
| | | |

## 1.3 Hardware Features

Table 1.2 Product Hardware Characteristics

| | |
|---|---|
| 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，Support CAN FD acceleration |
| terminating resistor | Built in 120 ohms (can be configured using upper computer software) |
| Shell material | metal + plastic |
| size | 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<br>IEC/EN 61000-4-4 EFT: Power supply: 1.0 kV; Signal: 0.5 kV<br>IEC/EN 61000-4-5 Surge: Power Supply:common:1.0kV,differential:0.5kV;Signal:0.5kV<br>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

| | |
|---|---|
| Profinet features | Supports standard Profinet IO protocol |
| | Supports up to 16 slots |
| | Supports up to 1440 bytes of input and 1440 bytes of output |
| | Supported module types:<br>Input 1/2/4/8/16/32/64/128/256/512 bytes<br>Output 1/2/4/8/16/32/64/128/256/512 bytes |
| (PXB-8210) Modbus Characteristics | Supports three IO modes: polling, status change, and application triggering |
| | Supports 4 working modes: PROFINET slave to Modbus RTU master/slave |
| | When the Modbus side is set as the master, a maximum of 32 Modbus slave devices are supported |
| | Supports 01H, 02H, 03H, 04H, 05H, 06H, 15H, 16H instructions |
| (PXB-8220) CAN/CAN FD Characteristics | Support setting parameters such as CAN type, CAN FD standard, and CAN FD acceleration |
| | CAN message transmission supports multiple triggering methods such as cycle, status change, and single transmission |
| | Support sending and receiving messages through bit, byte, and other operations |
| | Supports standard frames, extended frames, and remote frames |
| (PXB-8221M) CANopen feature | Supports baud rates of 10k~1M |
| | Supports managing up to 127 CANOpen slave devices |
| | Support SDO mechanism, initialize configuration of slave dictionary |
| | Supports 128 sets of TPDO and 128 sets of RPDO |
| | Support synchronous signal mechanism |
| | Support heartbeat management mechanism |
| (PXB-8222M) DeviceNet Features | As a standard DevicetNet master station, this device supports up to 63 standard DevicetNet slave stations for access |
| | DeviceNet communication speed supports 125K, 250K, and 500K |
| | Supports four types of I/O connections: polling, bit gating, state change, and periodic |
| | Each I/O connection type supports up to 64 bytes of input/output cache |
| | Support configuring explicit messages for parameter configuration and information exchange of DeviceNet slave stations |
| | Support configuring the data update cycle for each DevicetNet slave station |
| | Support mutual conversion of Profinet data and DeviceNet data in the form of bits, bytes, etc |
| 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 Physical Image

The PXB-82 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 4 LED indicator lights, 2 Profinet network interfaces, and 1 CAN/CAN FD interface or RS485 interface with electrical isolation.

Note: For specific LED and interface definitions and usage, please refer to 2.2.2 Indicator Light Instructions and 2.3 Interface Definitions.

## 2.2 Indicator Light Instructions

The PXB-82 series products have four LED lights on the front, which have different functional indication instructions in different product models. They are abbreviated according to relevant professional vocabulary to form corresponding labels, namely PWR, RUN, PN, CAN, 485, MS, and NS. The specific relevant instructions are shown in the table below:

Table 2.1 PXB-8220 Indicator Light Status Description

| identification | definition | status | explanation |
|---|---|---|---|
| PWR | Equipment power indicator light | not bright | The device is not powered on or has abnormal power supply |
| | | The red light is | Equipment power supply is normal |

| | | | |
|---|---|---|---|
| | | constantly on. | |
| RUN | Equipment operation indicator light | Green light always on | The equipment is working normally |
| | | The red light is constantly on. | Device initialization failed, did not enter working mode |
| PN | Profinet communication status indicator light | The red light is constantly on. | Profinet initialization failed |
| | | Green light always on | Profinet is ready |
| | | Green light flashing | Profinet has established a connection |
| 485 | 485 communication status indicator light | not bright | The device has not sent or received data |
| | | Green light always on | The device is transmitting and receiving data normally |

Table 2.2 PXB-8220 Indicator Light Status Description

| identification | definition | status | explanation |
|---|---|---|---|
| PWR | Equipment power indicator light | not bright | The device is not powered on or has abnormal power supply |
| | | The red light is constantly on. | Equipment power supply is normal |
| RUN | Equipment operation indicator light | Green light always on | The equipment is working normally |
| | | The red light is constantly on. | Device initialization failed, did not enter working mode |
| PN | Profinet communication status indicator light | The red light is constantly on. | Profinet initialization failed |
| | | Green light always on | Profinet is ready |
| | | Green light flashing | Profinet has established a connection |
| CAN | CAN communication | not bright | The device has not sent or received data |

| | | | |
|---|---|---|---|
| | status indicator light | Green light always on | The device is transmitting and receiving data normally |

Table 2.3 PXB-8221M Indicator Light Status Description

| identification | definition | status | explanation |
|---|---|---|---|
| PWR | Equipment power indicator light | not bright | The device is not powered on or has abnormal power supply |
| | | The red light is constantly on. | Equipment power supply is normal |
| RUN | Equipment operation indicator light | Green light always on | The equipment is working normally |
| | | The red light is constantly on. | Device initialization failed, did not enter working mode |
| PN | Profinet communication status indicator light | The red light is constantly on. | Profinet initialization failed |
| | | Green light always on | Profinet is ready |
| | | Green light flashing | Profinet has established a connection |
| CAN | CANopen communication status indicator light | not bright | CANopen initialization failed |
| | | Green light always on | All child nodes of CANopen network are online |
| | | Green light flashing | CANopen network has child nodes offline or heartbeat timeout |

Table 2.4 PXB-8222M Indicator Light Status Description

| identification | definition | status | explanation |
|---|---|---|---|
| PWR | Equipment power indicator light | not bright | The device is not powered on or has abnormal power supply |
| | | The red light is constantly on. | Equipment power supply is normal |
| PN | Profinet communication status indicator light | The red light is constantly on. | Profinet initialization failed |

| | | | |
|---|---|---|---|
| | | Green light always on | Profinet is ready |
| | | Green light flashing | Profinet has established a connection |
| MS | DeviceNet device status indicator light | Green light always on | The equipment is running normally |
| | | Green light flashing | Due to missing, incomplete or incorrect configuration, the device needs to be debugged |
| | | Red light flashing | Recoverable faults |
| | | The red light is constantly on. | Unrecoverable fault, needs to be replaced |
| | | Red and green flashing | The device is self checking |
| NS | DeviceNet network status indicator light | not bright | The device is offline.<br>The device has not completed duplicate MAC ID detection |
| NS | DeviceNet network status indicator light | Green light flashing | The device is online but not connected and is in an established state |
| | | Green light always on | The device is online and has one or more established connections |
| | | Red light flashing | Communication failure: repeated MAC ID detection failed, bus BUS-OFF |
| | | Red and green flashing | The device has detected a network access error and is in a communication failure state,<br>And receive point-to-point recognition offline fault request information |

Note:

The PXB-8222M device 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.

## 2.3 Interface Definition

The PXB-82 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 Schematic diagram of top interface definition

### 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 2.5.

Table 2.5 Power Interface Input Power Specification

| parameter | minimum | typical | maximum | unit |
|---|---|---|---|---|
| working voltage | 9.0 | 24.0 | 36.0 | V |
| operating current | — | 80 | — | mA |
| Product power consumption | — | 1.92 | — | W |

### 2.3.2 USB port

The PXB-82 series features one Type-C USB port.The connector is a standard USB-C receptacle, silk-screened with the "USB" label.During the debugging phase,this port can be used to power the entire unit and for configuration via the AWPX software.

### 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.


Figure 2.3 Schematic diagram of front interface definition

In addition to 4 LED indicator lights on the front of the product, there are also 2 10/100M Ethernet ports and 1 isolated RS485 interface or isolated CAN/CAN FD interface.

### 2.3.4 Profinet Interface

Due to the fact that this product has a Profinet side as a slave network interface, the Profinet interface provides two RJ45 network ports and an internal TSN switch, allowing users to connect using either network port.

### 2.3.5 CAN/CAN FD interface

When the product is PXB-8220, PXB-8221M, and PXB-8222M, the bottom interface of the product is CAN/CAN FD interface, and the interface identification is shown in the following table:

Table 2.6 CAN FD Interface Identification Description

| identification | explanation |
|---|---|
| CANH | Isolate CAN_Signal Line |
| CANL | Isolate CAN_L signal line |
| CGND | Isolate CAN ground |

### 2.3.6 RS485 Interface

When the product is PXB-8210, the bottom interface of the product is RS485 interface, and the interface identification is shown in the following table:

Table 2.7 RS485 Interface Identification Description

| identification | explanation |
|---|---|
| 485A | Isolate RS485 signal line |
| 485B | Isolate RS485 signal line |
| RGND | Isolate RS485 ground |

### 2.3.7 Terminating Resistor

The PXB-82 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.Users can choose to connect or disconnect the terminal resistor to the bus, eliminating the need for external series connection or disconnection of resistors.

# 3. Instructions For Use

The PXB-82 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 the AWPX upper computer configuration software on the PC end;

2、After the product is correctly powered on, connect the PC and PXB-82XX device using a USB cable;

3、Open the AWPX upper computer configuration tool, configure according to the requirements in the configuration tool (please refer to 3.3), and issue the configuration to the PXB-82XX protocol converter device;

4、Connect the main station equipment to the PXB-82XX protocol converter using Ethernet cables;

5、Perform data exchange and communication.

## 3.1 Siemens S7-1500 PROFINET Master Station Settings

### 3.1.1 New Project

Open TIA Portal V18 software, click "Create New Project", fill in the project name, select the project path, click "Create", and create a new project, as shown in Figure 3.1.

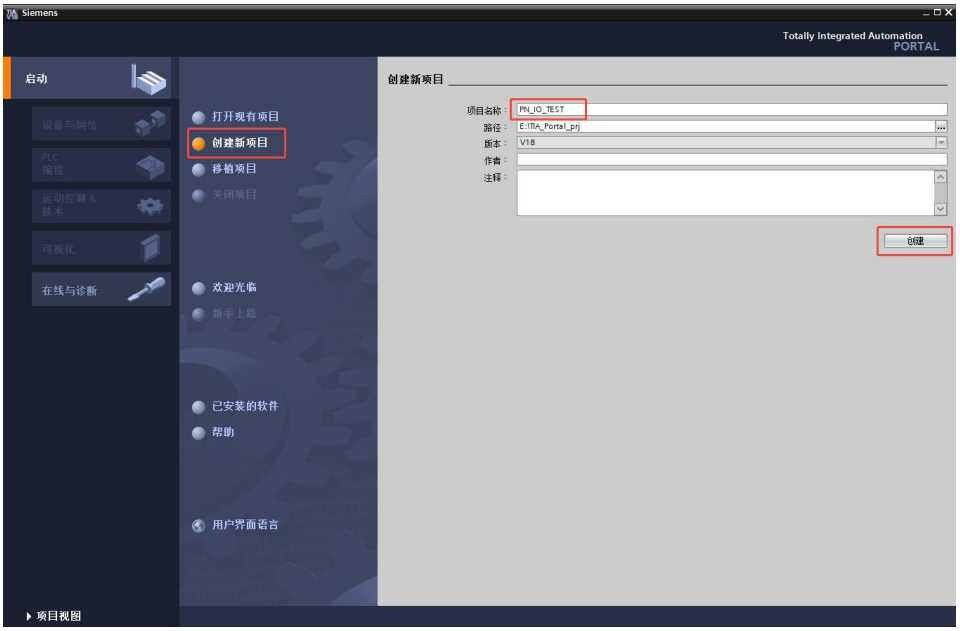



Figure 3.1 Creating a Project

After creation, open the project view as shown in Figure 3.2.


Figure 3.2 Open Project View

### 3.1.2 Add PLC

Click on "Online Access", "Update Accessible Devices", and "Online and Diagnostic" in sequence to view the specific information of the PLC, as shown in Figure 3.3.

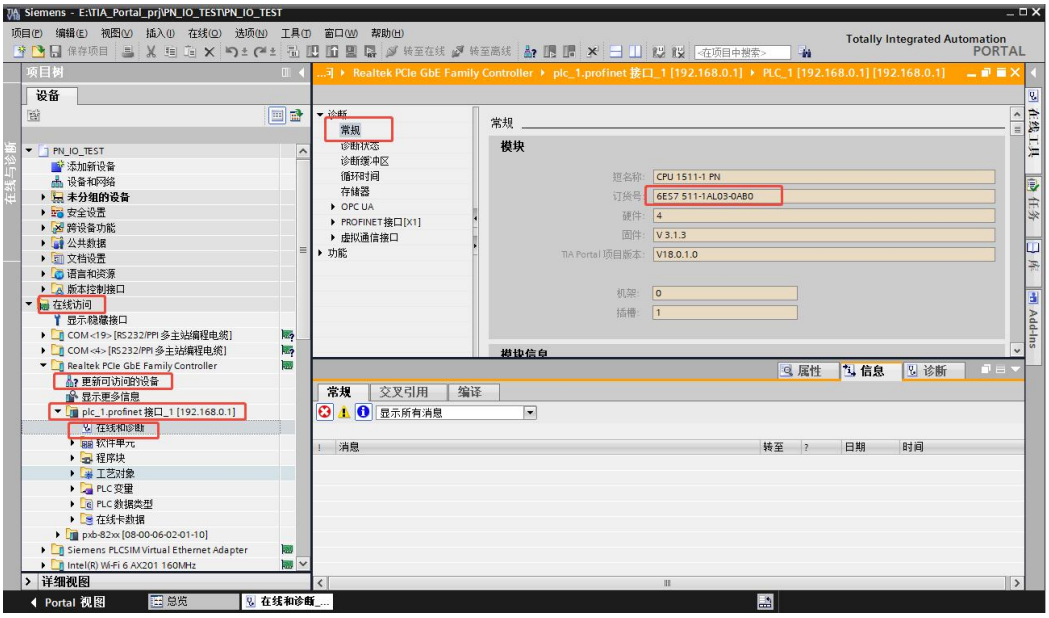



Figure 3.3 Online Viewing of PLC Information

Click on "Devices and Networks", expand the hardware directory, select the PLC with the same model and order number as the "Online Access", double-click to add the PLC to the network, as shown in Figure 3.4.

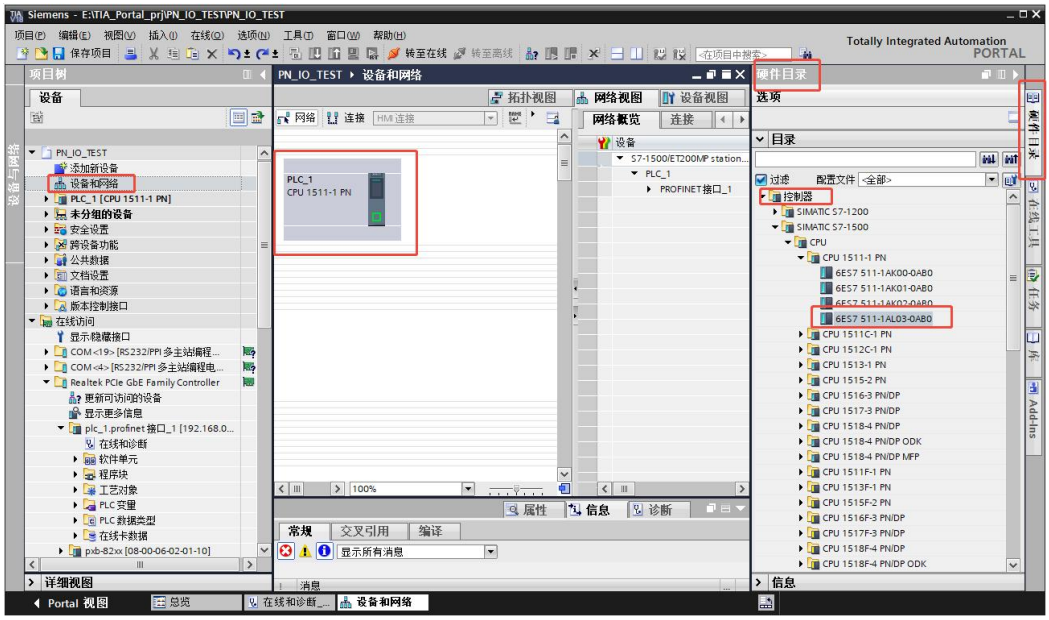


Figure 3.4 Adding PLC

### 3.1.3 Install GSD File

Click on "Options" ->"Manage General Station Description Files (GSD)", as shown in Figure 3.5.

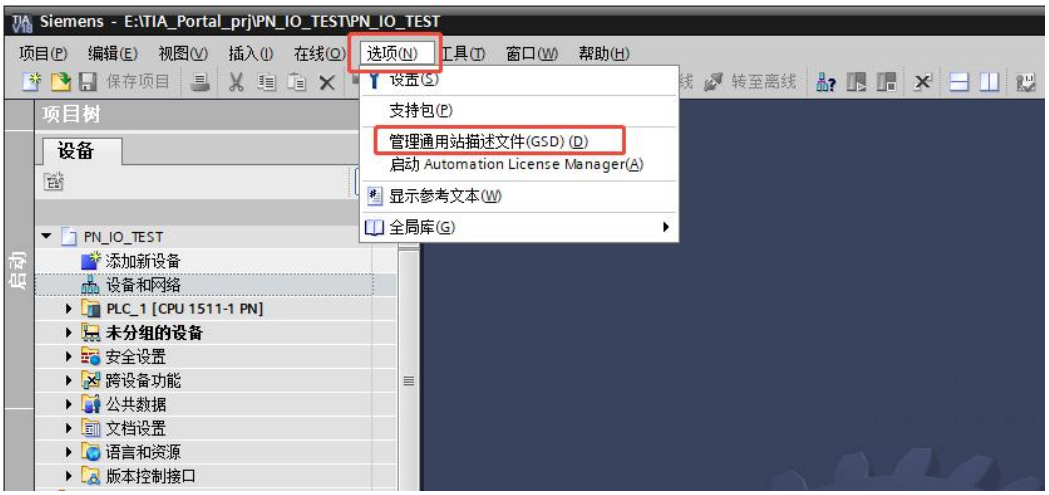



Figure 3.5 GSD Installation

Select the path where the GSD is located on the pop-up interface, click "Install", and install the GSD file for the PXB-8280M device, as shown in Figure 3.6.

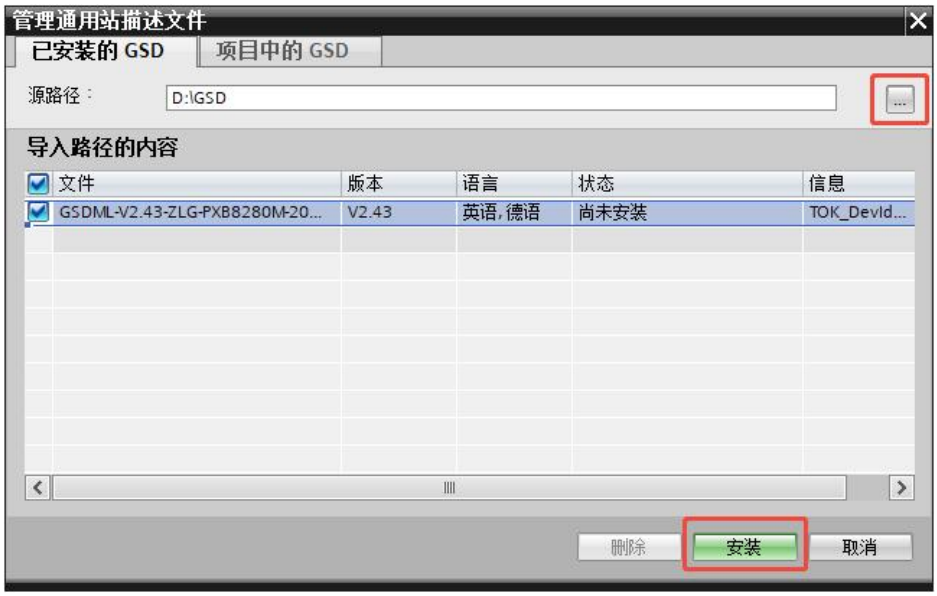



Figure 3.6 Installing GSD

### 3.1.4 Add PROFIENT Device

After installing GSD, expand the hardware directory, select the PXB device name, and double-click to add the device.As shown in Figure 3.7.

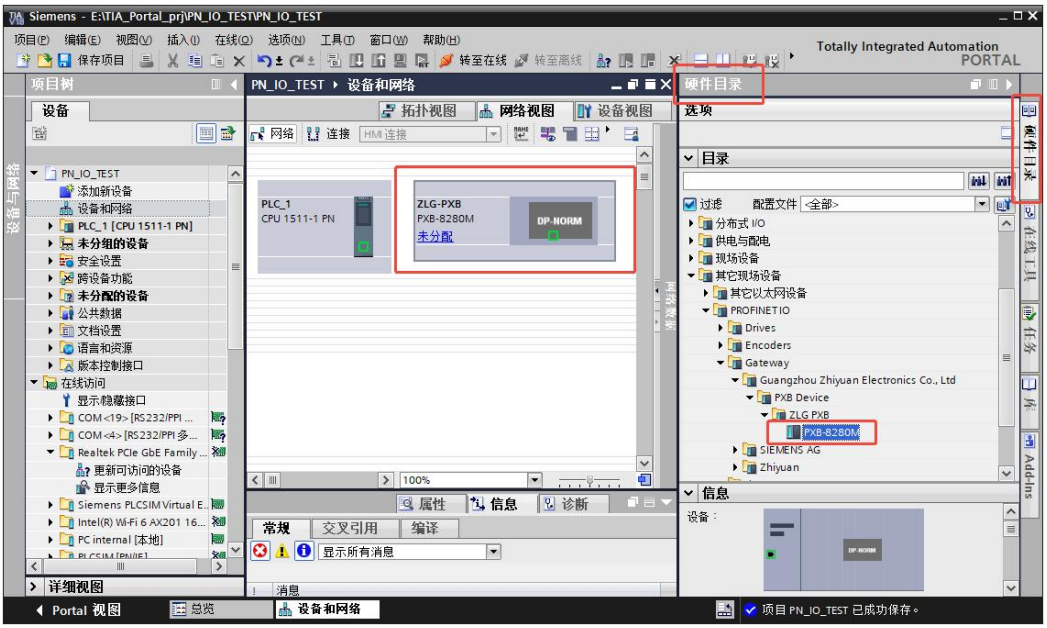



Figure 3.7 Adding PXB Device

Place the mouse arrow above "Unallocated" and right-click to select "Assign to New IO Controller", as shown in Figure 3.8.

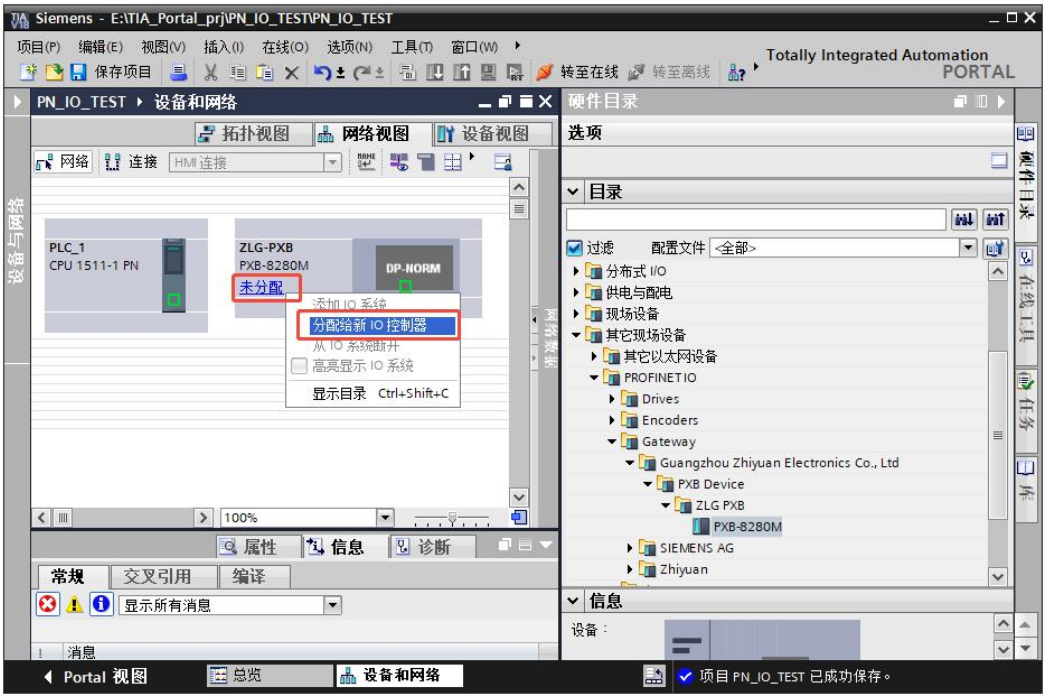



Figure 3.8 Allocation of IO controllers

Check the IO controller and click OK, as shown in Figure 3.9.

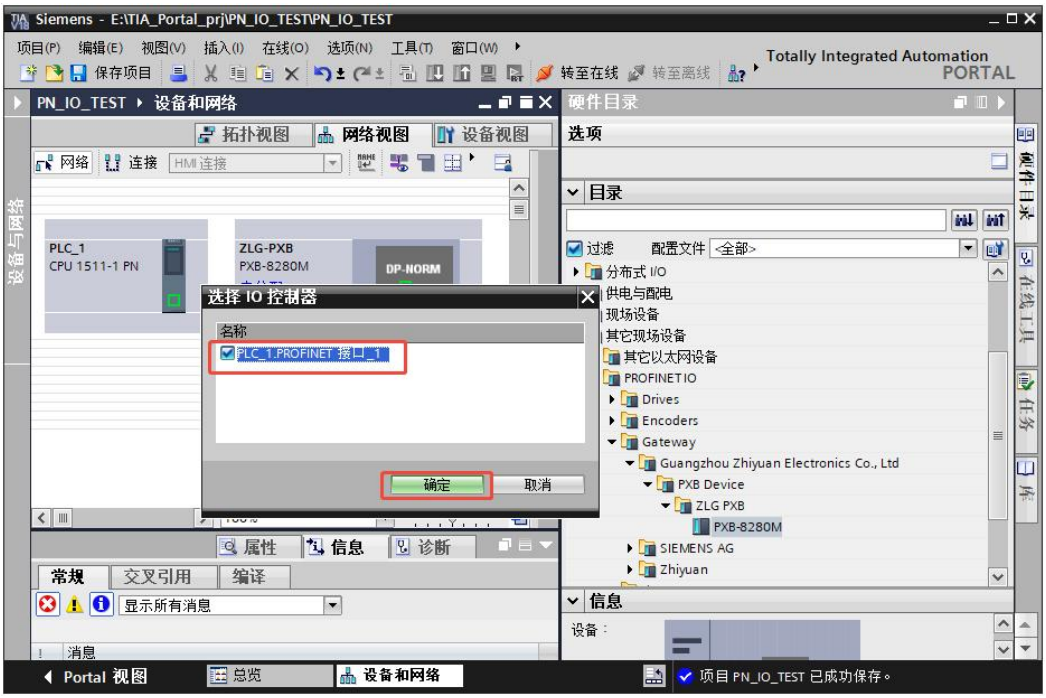



Figure 3.9 Checking IO Controller

After clicking "OK", the PXB device is displayed as connected to the network. Click on PLC to view PLC properties and change parameters such as the PLC's IP address, as shown in Figure 3.10.The default IP address for PLC is 192.168.0.1.

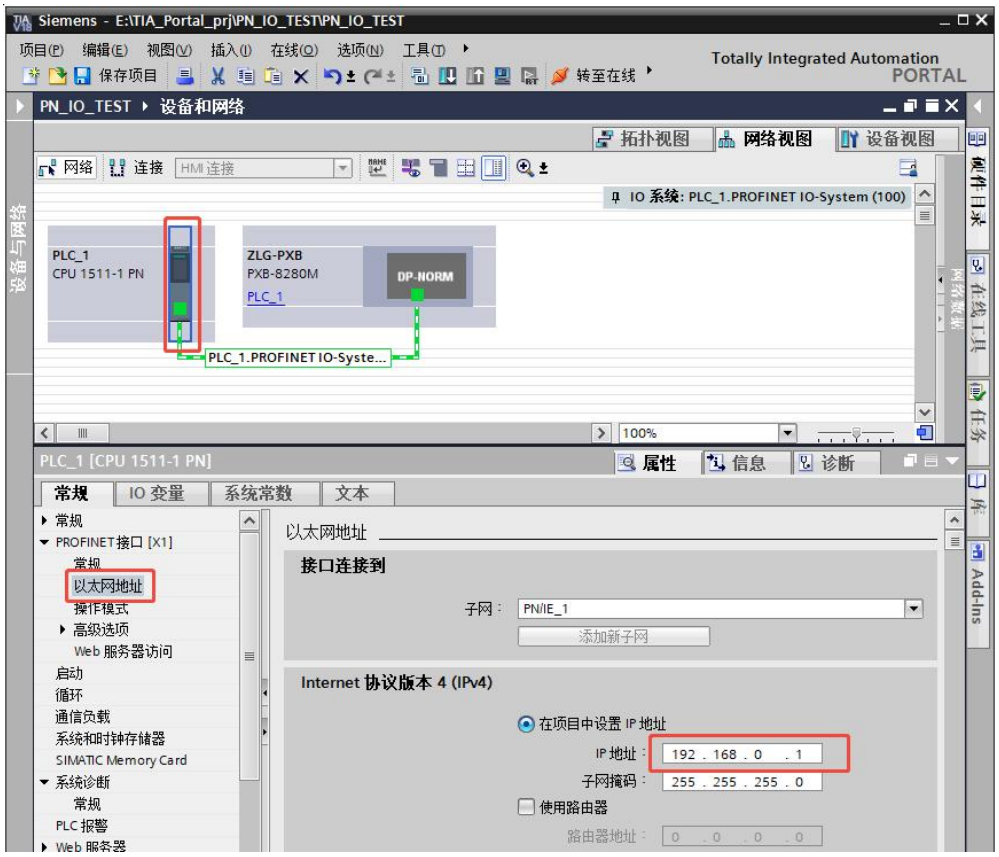



Figure 3.10 PLC Properties

Click on the PXB device, view its properties, set the IP address of PXB to 192.168.0.2, cancel the "Automatically generate PROFIENT device name" option, and fill in the PROFIENT name as "pxb xxxx". After downloading the project, the PLC will connect to the PROFIENT device named "pxb xxxx", while other PROFIENT devices with different names will not be connected by the PLC, as shown in Figure 3.11.

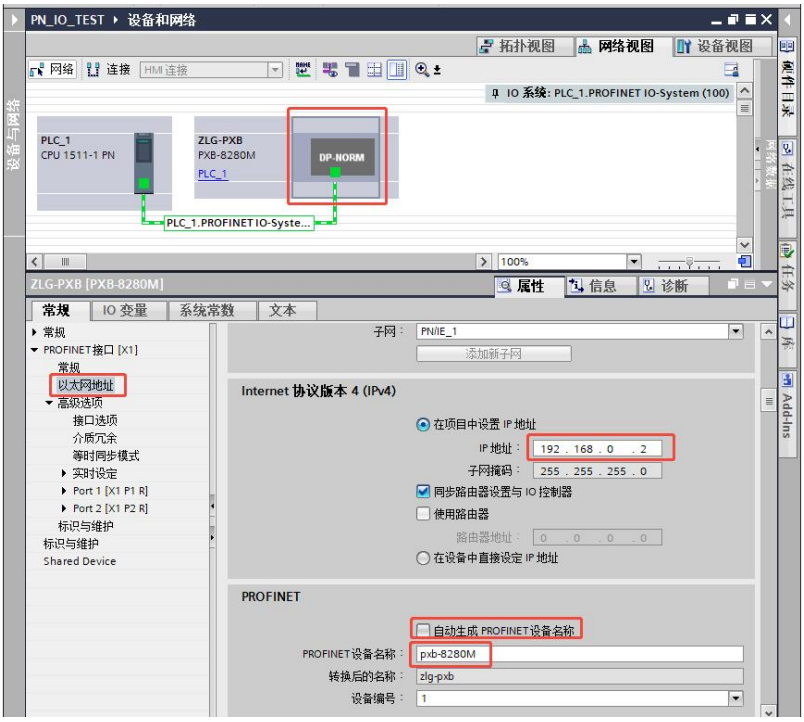



Figure 3.11 PXB Attributes

Double click PXB and click on "Device View". Drag "1 byte I" and "1 byte O" to slot 1 and slot 2, respectively, as shown in Figure 3.12.

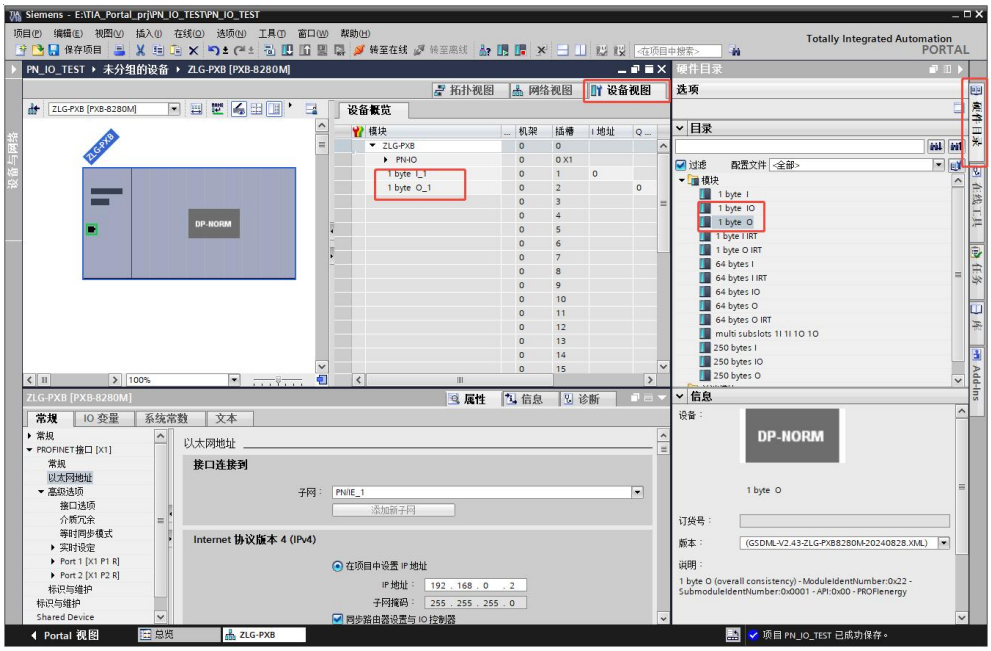



Figure 3.12 Insertion into Slot

### 3.1.5 Assign Device Name

Click on "Network View", right-click on the connection between PLC and PXB, and select "Assign Device Name", as shown in Figure 3.13.

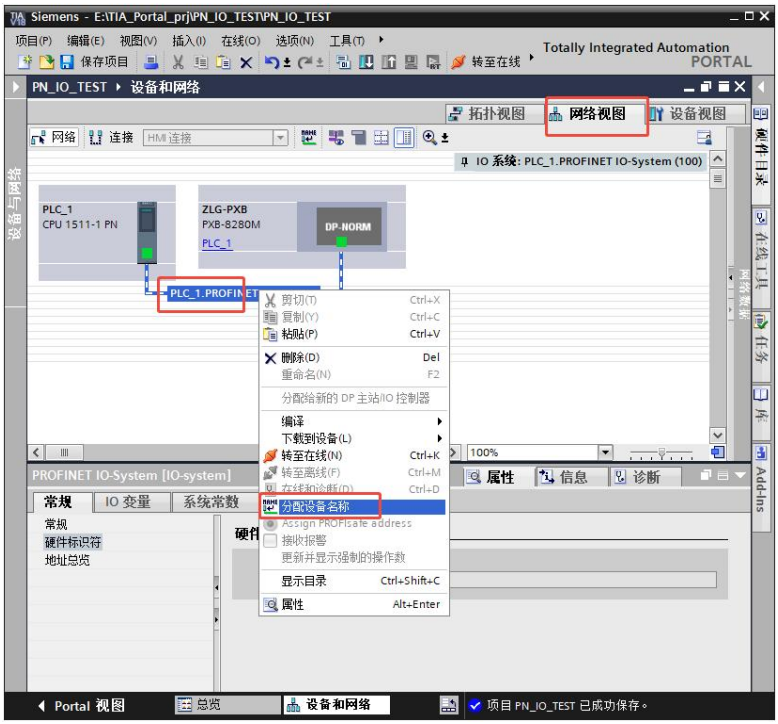



Figure 3.13 Select Connection

After selecting PXB for the PROFIENT device name, click on the update list. After the update is complete, if the status displays "Device Name Different", click on "Assign Name". If the status displays correctly, no name needs to be assigned, as shown in Figure 3.14.

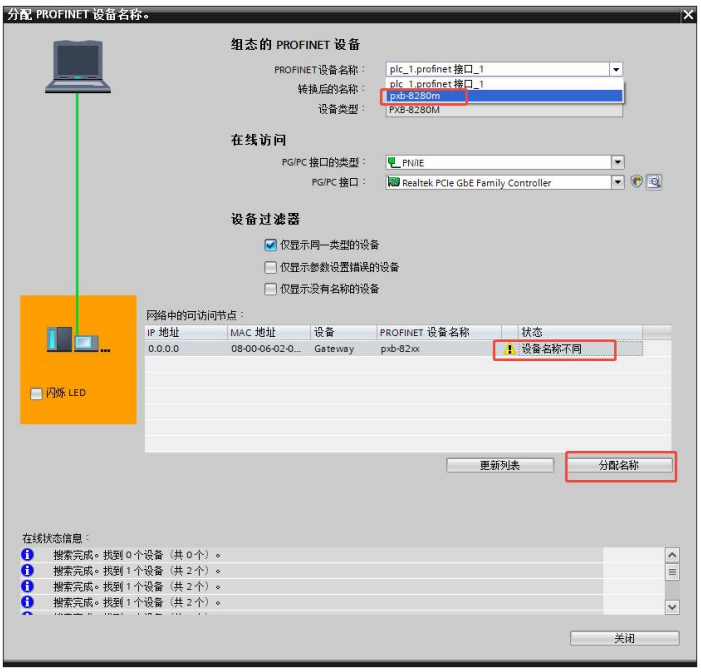



Figure 3.14 Allocation Name

After successfully assigning the name, click "Close" as shown in Figure 3.15.

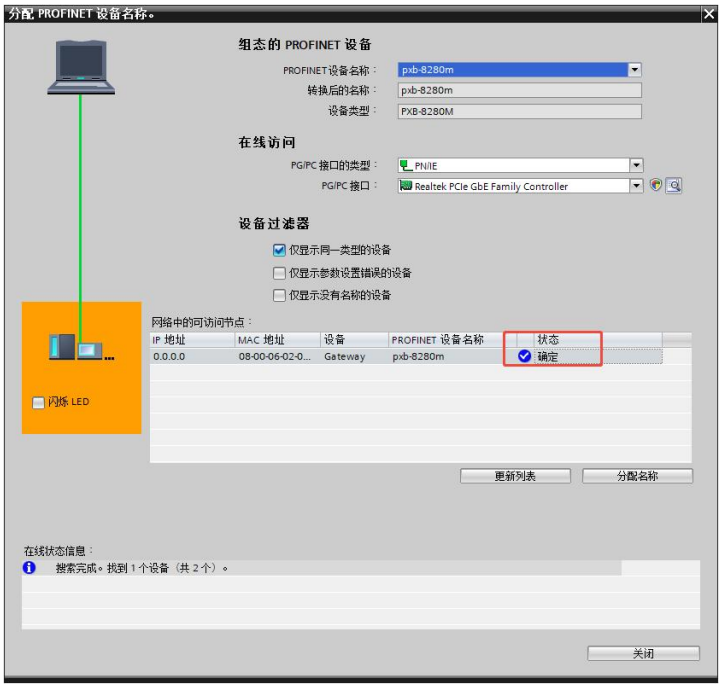



Figure 3.15 Allocation completed

### 3.1.6 Download Project

Click on the PLC project in the project tree, click save, and then click the download button to download the project, as shown in Figure 3.16.

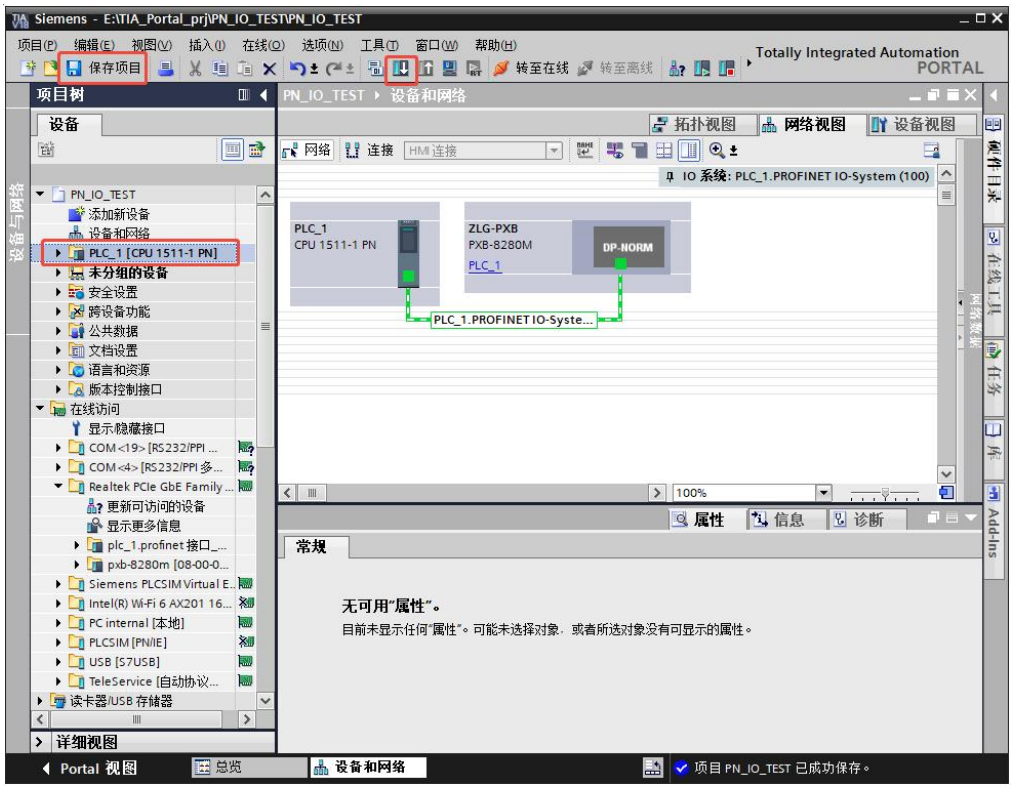



Figure 3.16 Click to download

Click "Start Search" on the pop-up download interface, and after finding the PLC, click "Download", as shown in Figure 3.17.

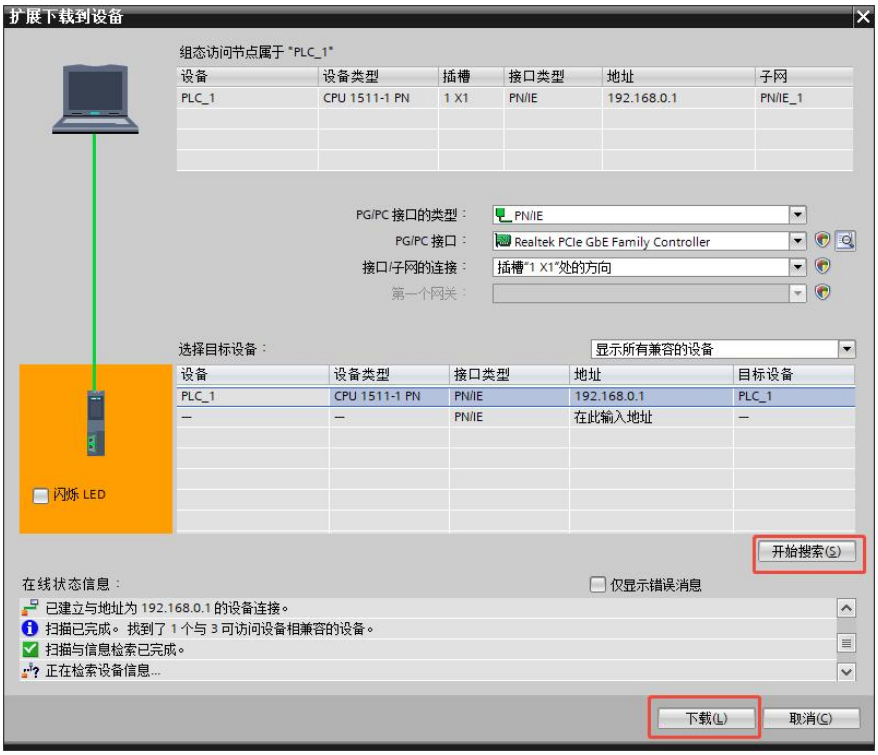



Figure 3.17 Download Project

Click 'Load', as shown in Figure 3.18.

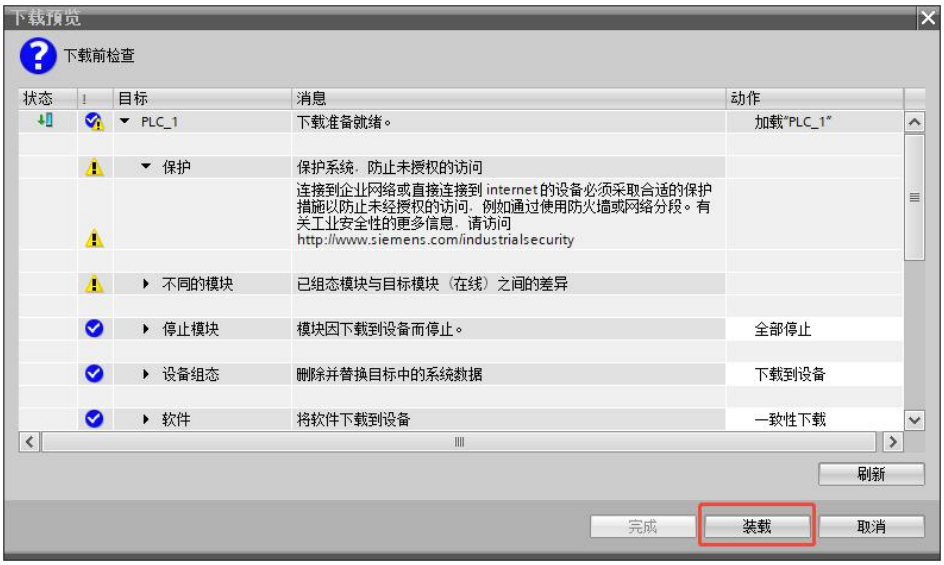



Figure 3.18 Click to load

Click Finish and wait for the PLC to restart, as shown in Figure 3.19.

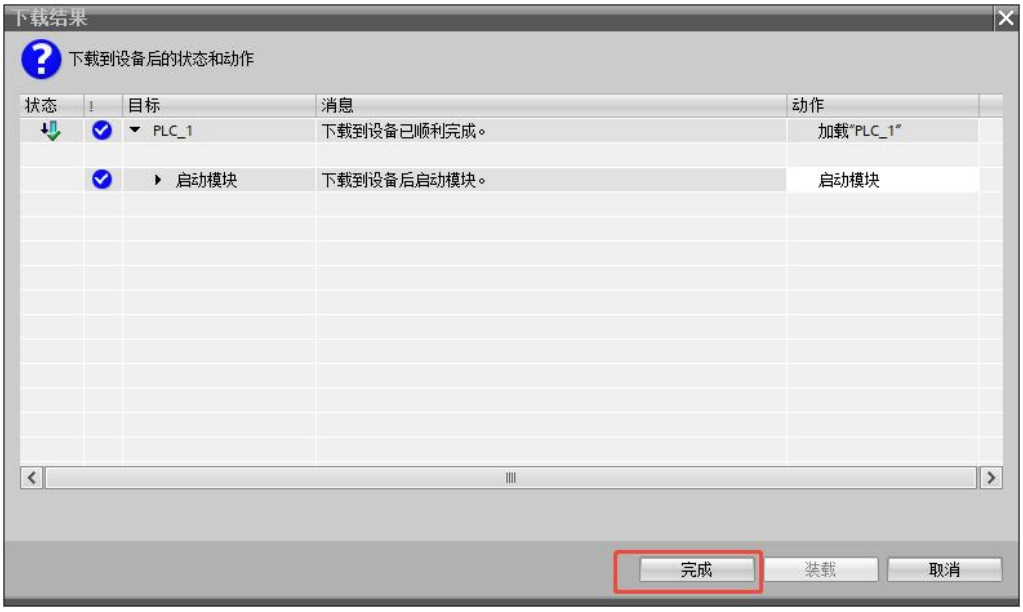



Figure 3.19 Download completed

Click on 'Go Online' and observe that all icons are green, indicating that there is no communication fault between PLC and PXB, as shown in Figure 3.20.

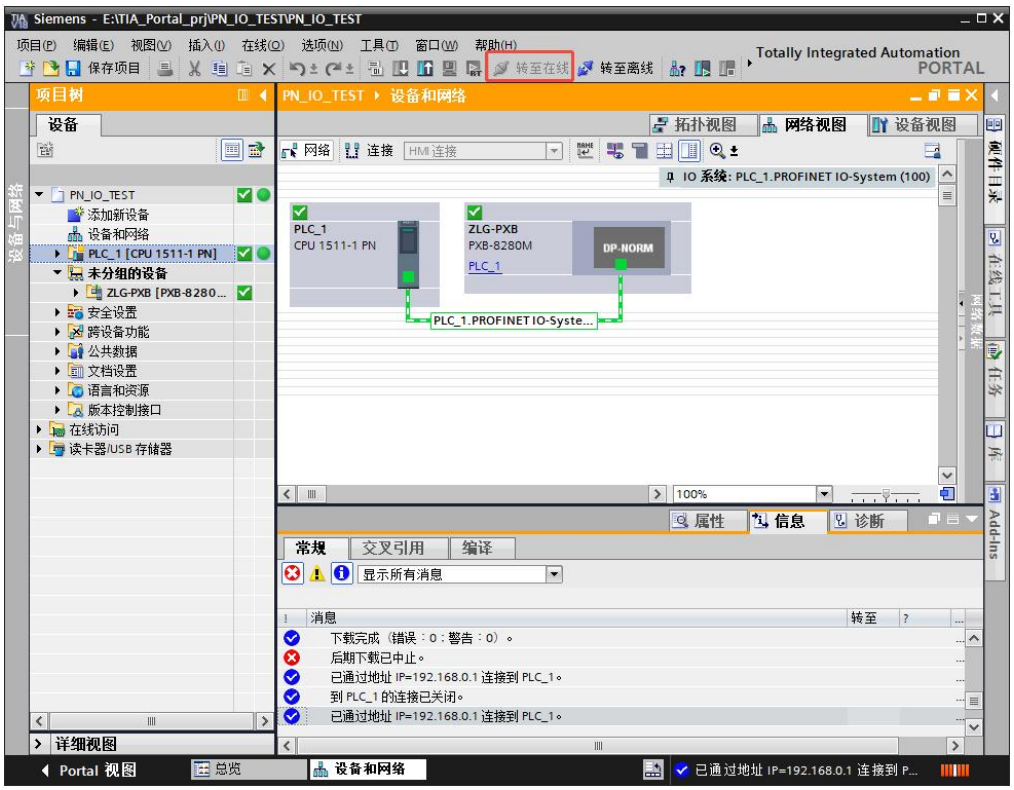



Figure 3.20 Communication is normal

A monitoring table can be established based on the I and Q addresses corresponding to the slots in the "Device View" for monitoring, as shown in Figure 3.21.The I address is the input process mapping address of the PLC, and the Q address is the output process mapping address of the PLC.

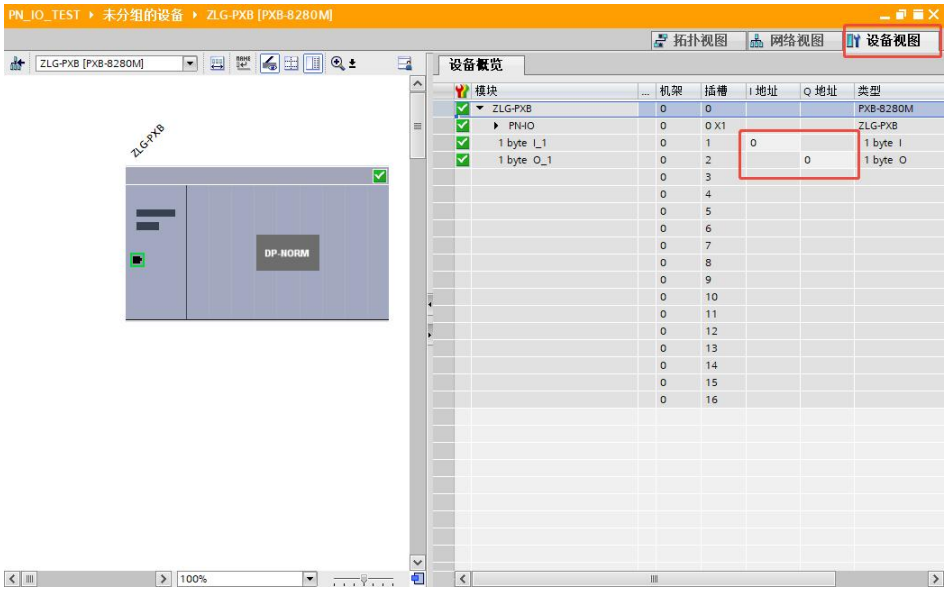



Figure 3.21 Slot Address

Double click "Add Monitoring Table" to add a monitoring table that can monitor and modify the data inserted into the PXB slot, as shown in Figure 3.22.The I address value is the PN IO data input to the PLC by PXB, and the Q address value is modified to enable the PLC to output PN IO data to PXB.

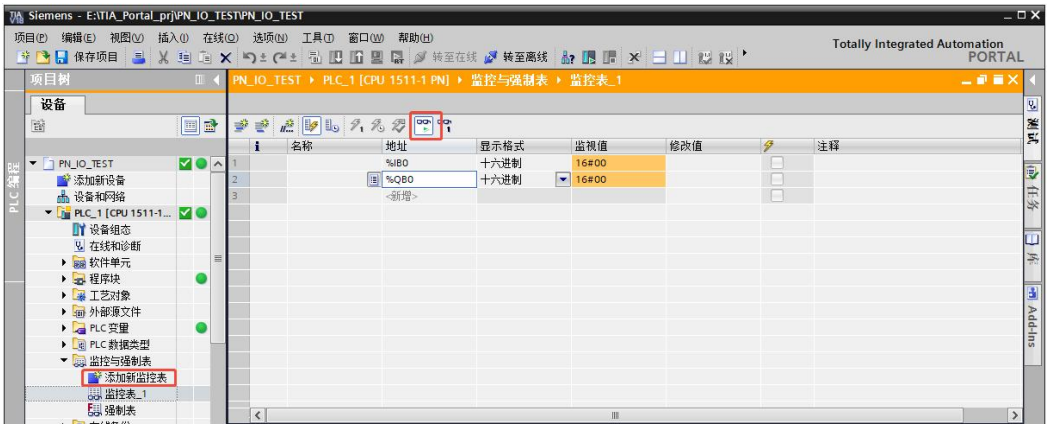



Figure 3.22 Add Monitoring Table

## 3.2 TwinCAT3 PROFINET Main Station Settings

Taking TwinCAT3 as an example, we will explore how to add a PXB-82xx protocol converter;

### 3.2.1 Import GSDML file

Copy and paste the GSDML file that comes with the PXB-82 protocol converter into the TwinCAT3 installation directory, for example:

C:\TwinCAT\3.1\Config\Io\Profinet；

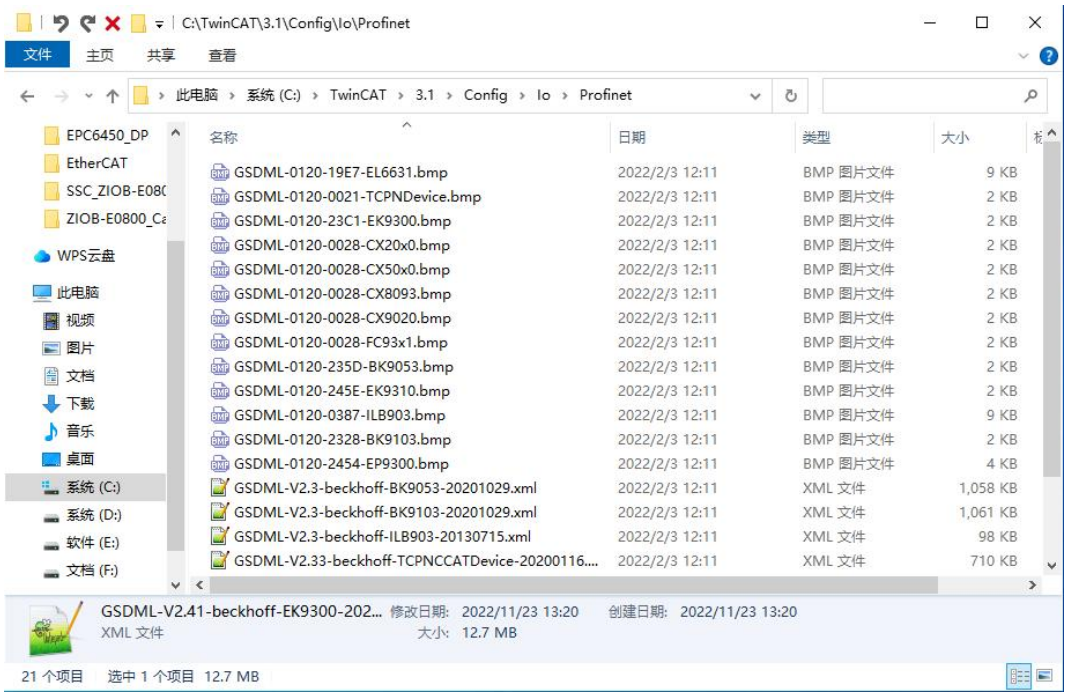



Figure 3.23 GSDML file placement directory

### 3.2.2 Configure PXB Protocol converter

1. New Project

Open the installed TwinCAT3 software and enter the development environment:

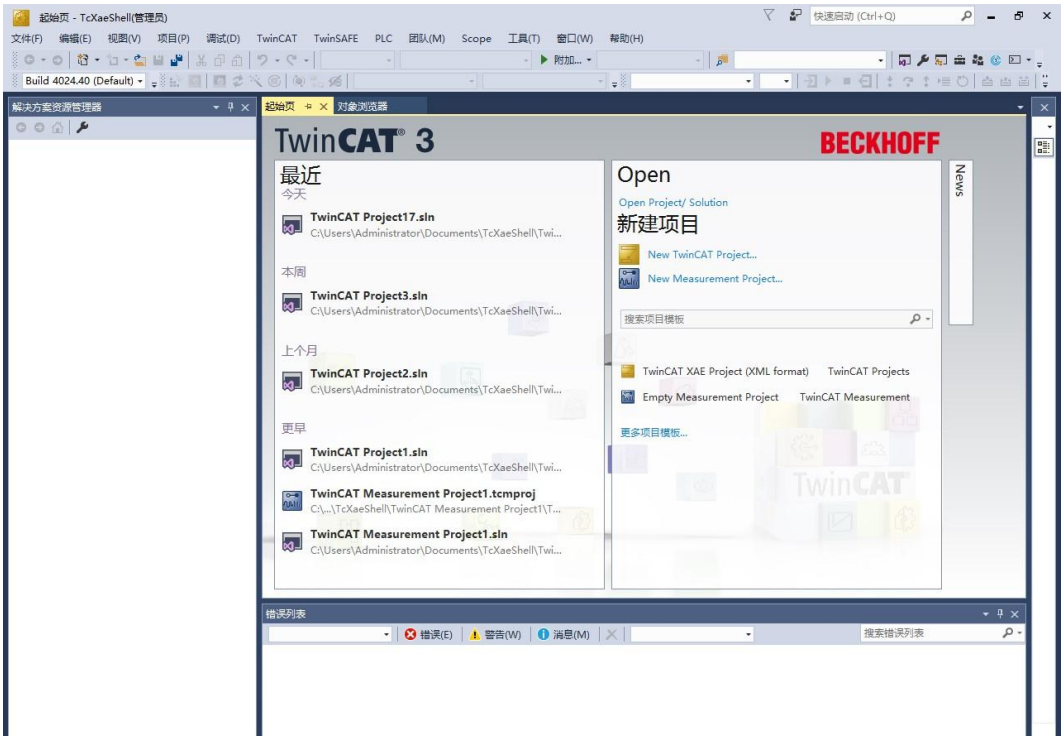



Figure 3.24 TwinCAT3 software development interface

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


Figure 3.25 New Project Interface

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

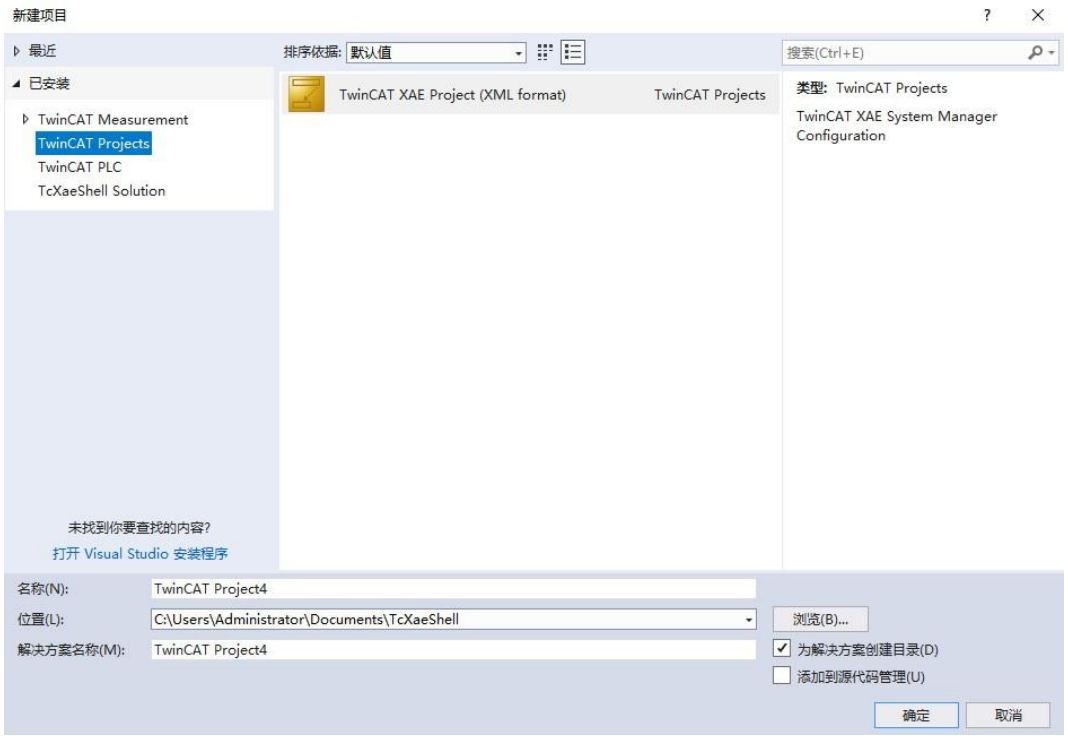



Figure 3.26 Engineering Information Modification Interface

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

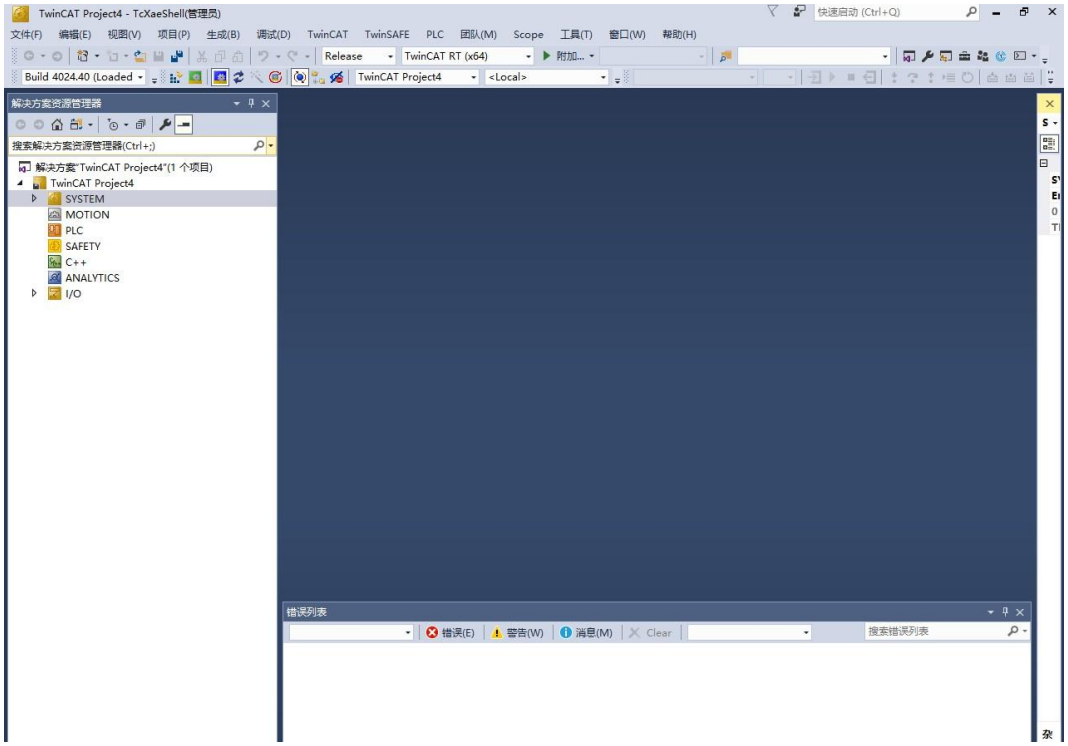



Figure 3.27 Project Creation Success Interface

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";

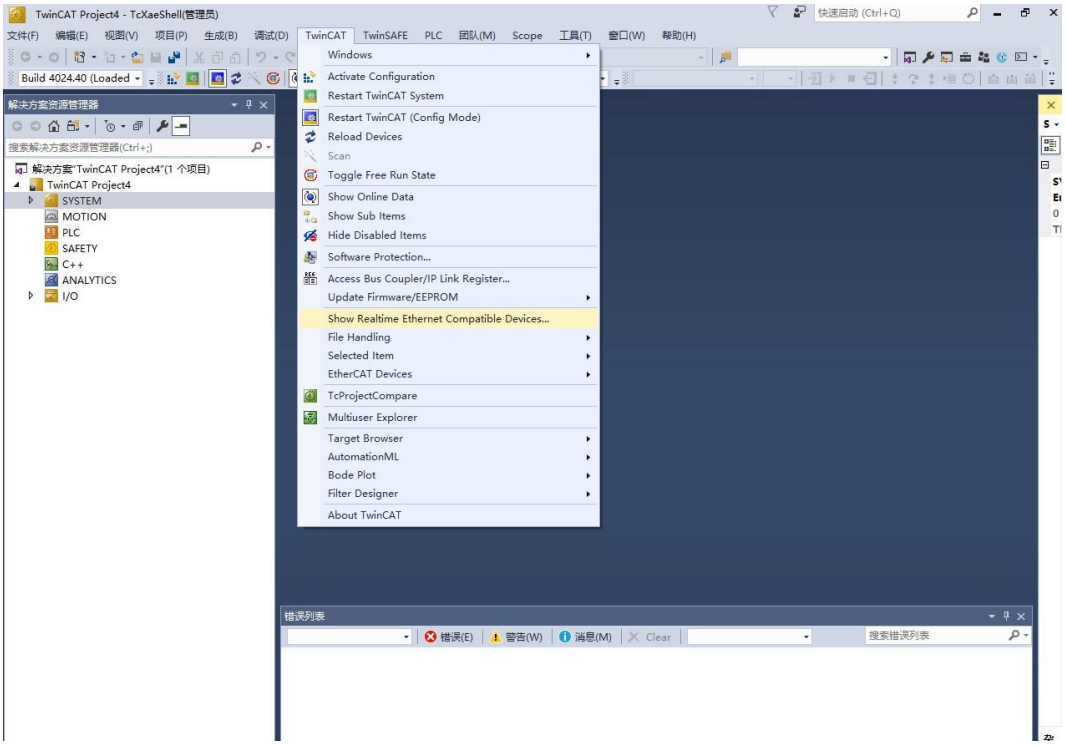



Figure 3.28 Network card installation interface

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 station 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.

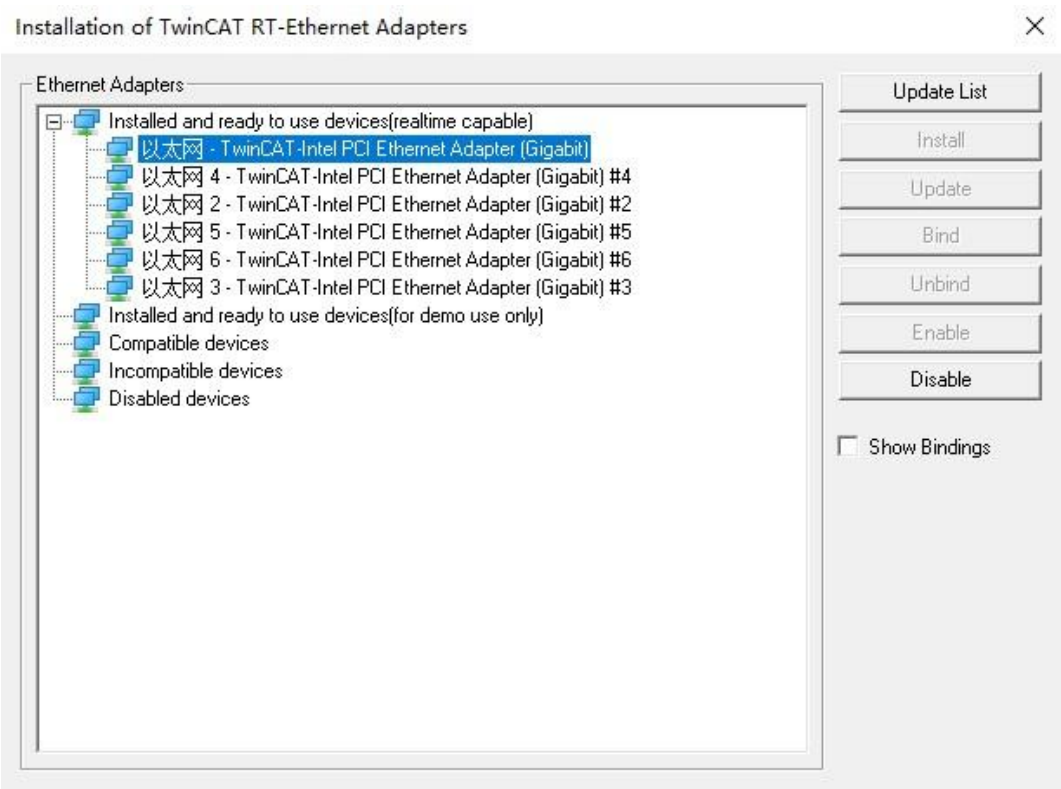



Figure 3.29 Network card driver normal interface

2. Scanning Device

In the project directory, select "I/O" to expand, choose "Device", right-click and select "Add New Item".

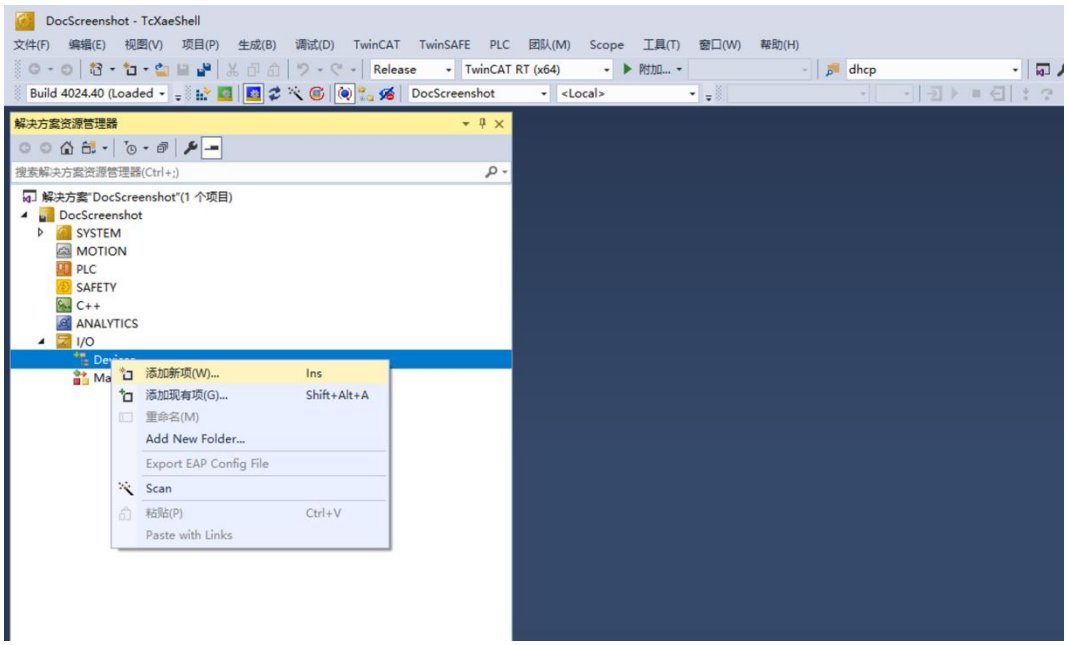



Figure 3.30 Add New Item Interface

Select Profinet I/O Controller (RT) and add PNIO master station.

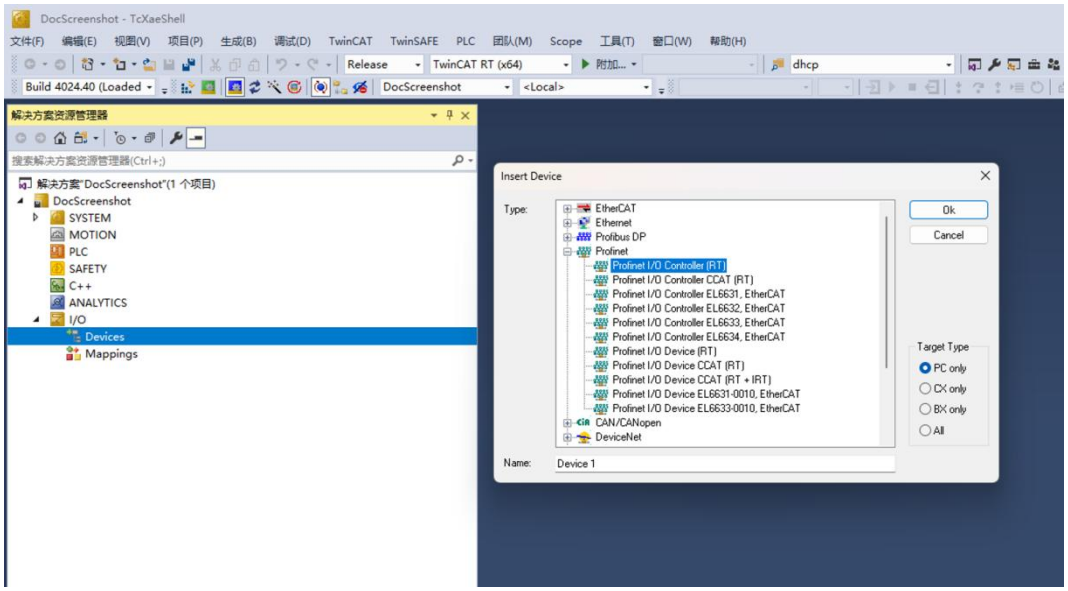



Figure 3.31 Add main station interface

Double click on Device 1 (Profinet Controller), select the Settings tab, and configure the master IP information.

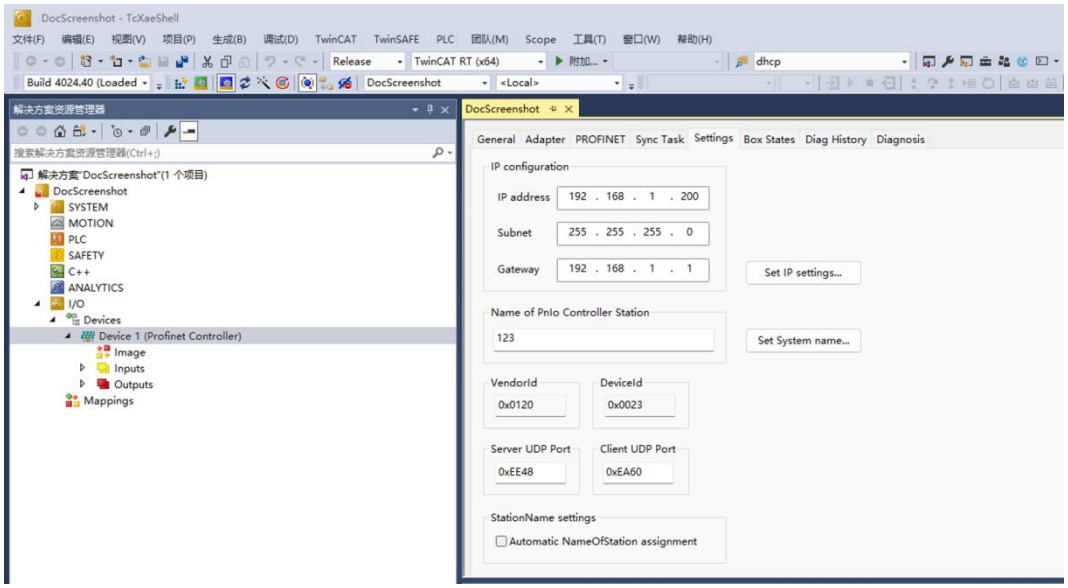



Figure 3.32 Configuring the main station network

Right click on Device 1 (Profinet Controller), select scan, and scan the slave station. This step requires connecting the PXB-82xx device to the corresponding network port on the PC in advance.

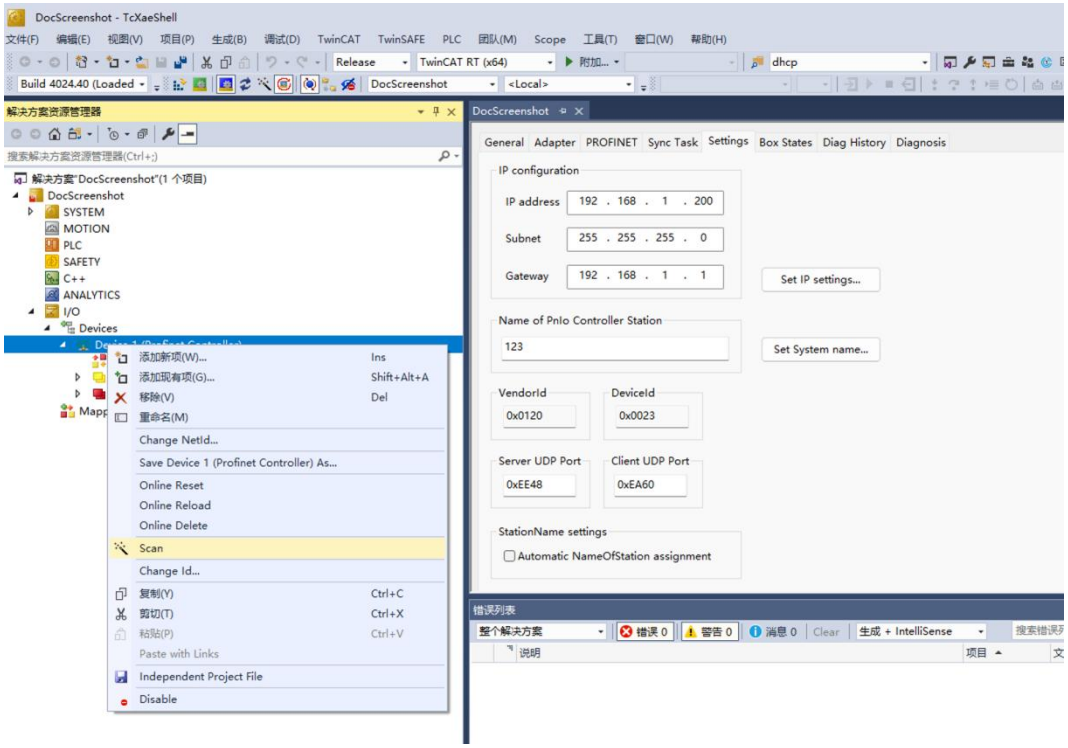



Figure 3.33: Selecting Scan Slave Devices

After successfully scanning the device, the following pop-up window will appear. Follow the instructions in the picture to configure it.

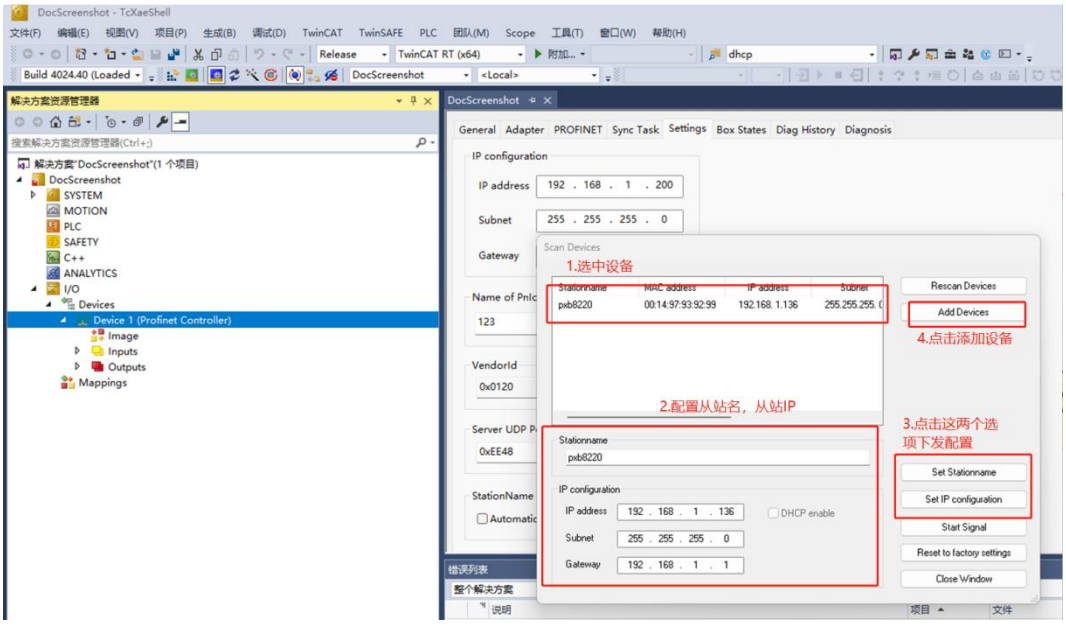



Figure 3.34 Configuring Slave Information

After successfully adding the device, there will be an additional option under Device 1 (Profinet Controller) that matches the slave name (as shown in pxb8220 in the figure below). Expand it, right-click on the API option, and select Add New Item.

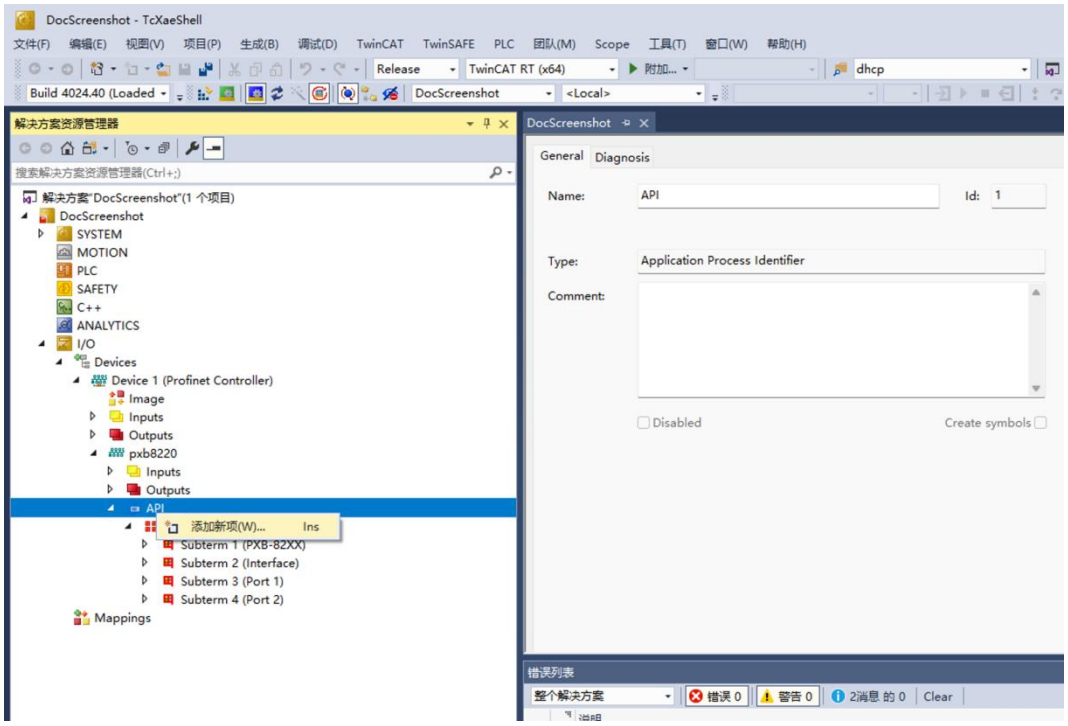



Figure 3.35 Device added successfully

After selection, the following window will pop up. Drag the corresponding module into slots 1-16 on the left according to the required number of bytes. One module can be reused in multiple slots.

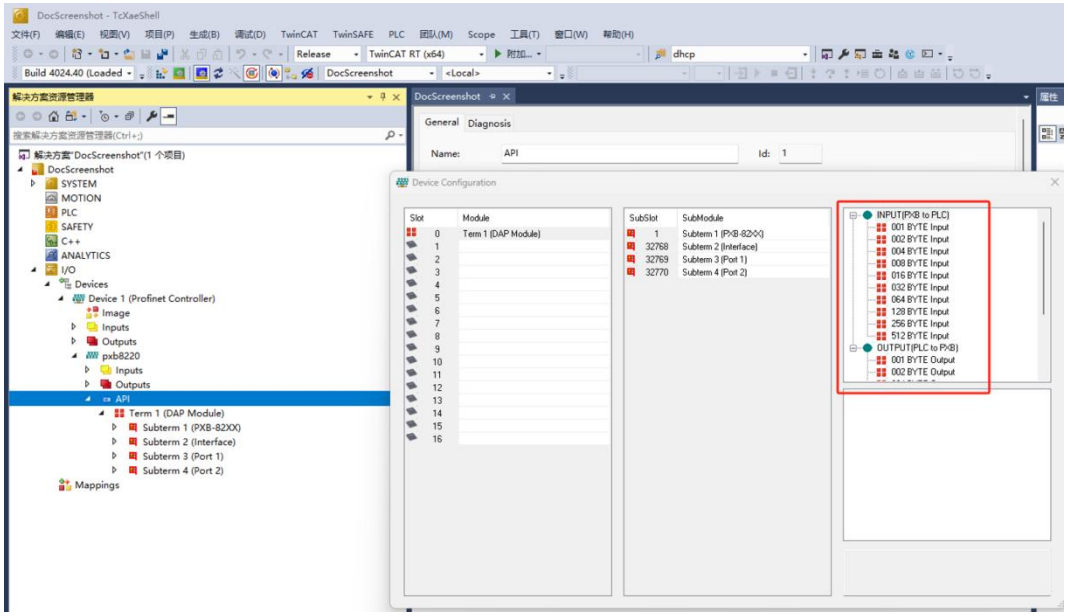



Figure 3.36 Insertion Module Interface

After successful addition, additional modules added by the user will be added under the API options (in the example shown in the figure, there are two modules for 8-byte input and 8-byte output). At this time, check the Show Online Data and Toggle Free Run State on the top, click

Reload Device, and wait for about 3-5 seconds for the device to successfully establish a connection with the main station and start exchanging data.

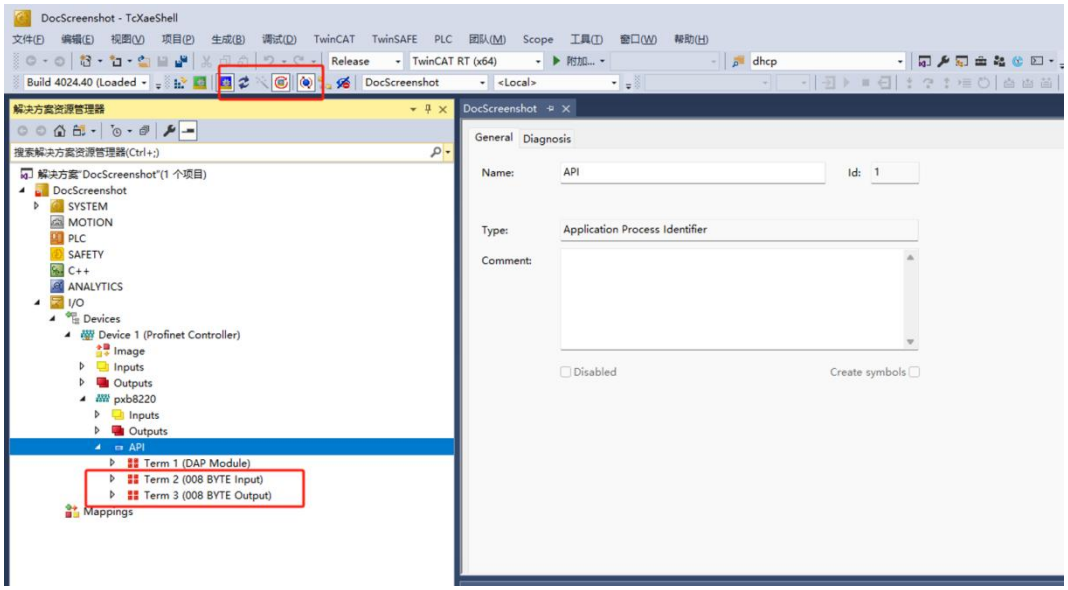



Figure 3.37 Establishing Connection

After expansion, as shown in the following figure, where input represents the data uploaded from the slave station to the master station, and output represents the data issued from the master station to the slave station.

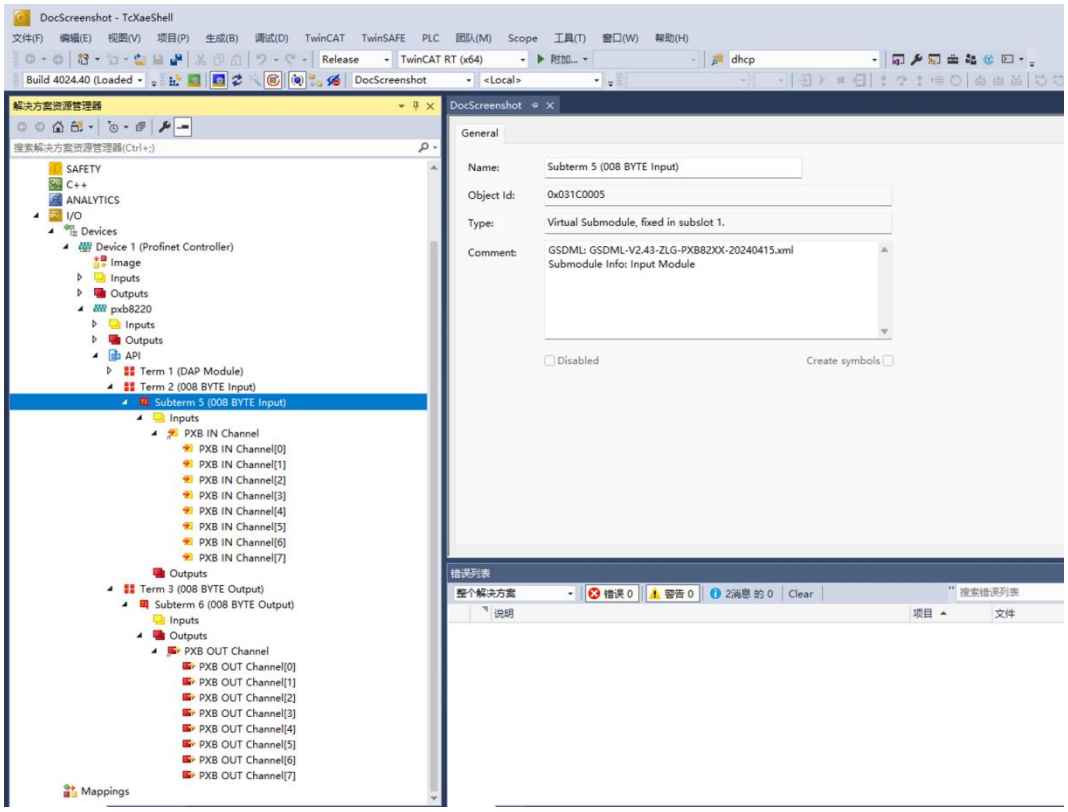



Figure 3.38 Expand Input Output Module

## 3.3 AWPX Configuration Software

### 3.3.1 Configure 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 3.39.

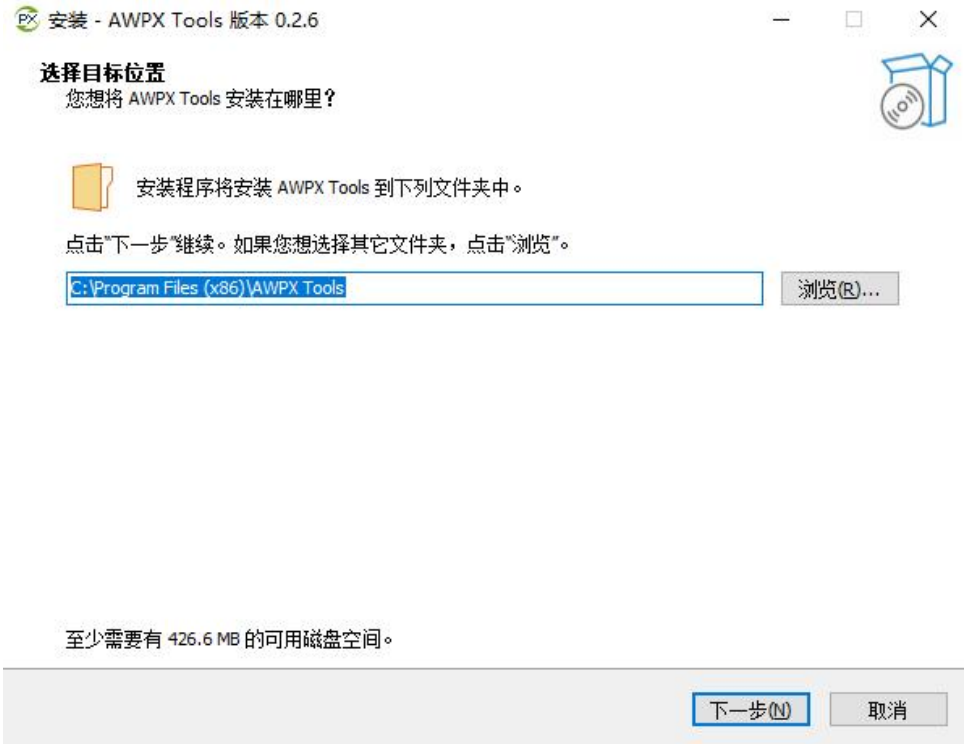



Figure 3.39 Start installing AWPX

After clicking on several 'Next' buttons, the installation interface is shown in Figure 3.40.

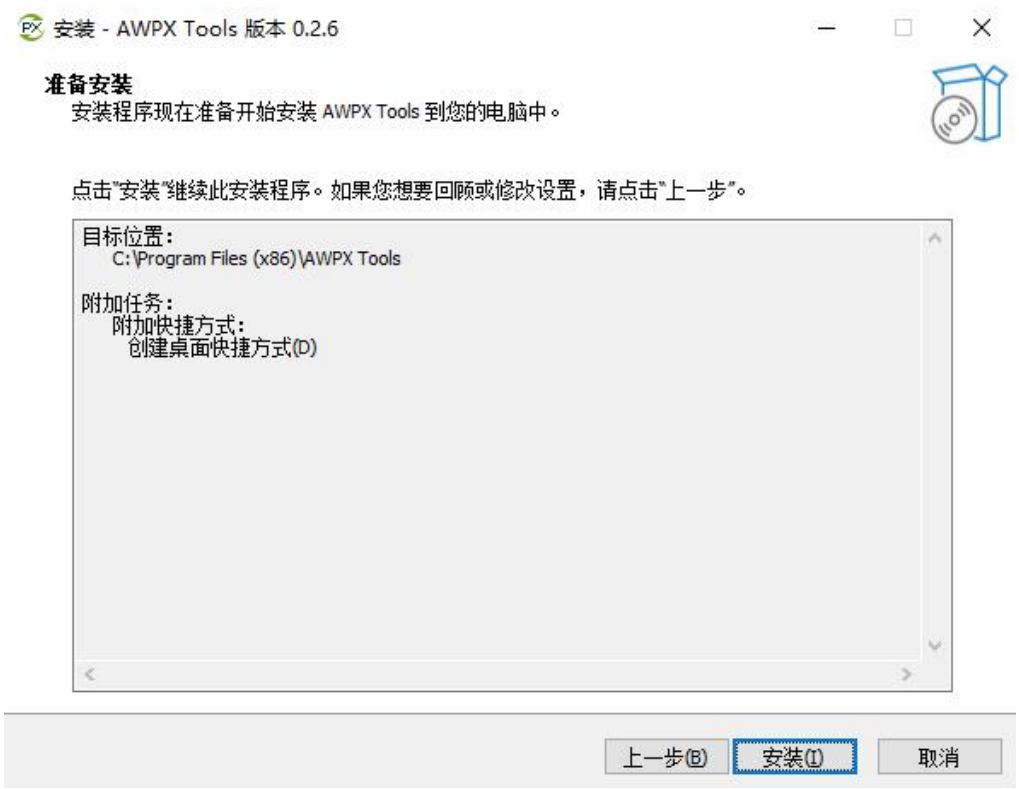



Figure 3.40 AWPX Installation Interface

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

### 3.3.2 Introduction To Configuration Software Functions

Run the AWPX software, and the interface is shown in Figure 3.41.

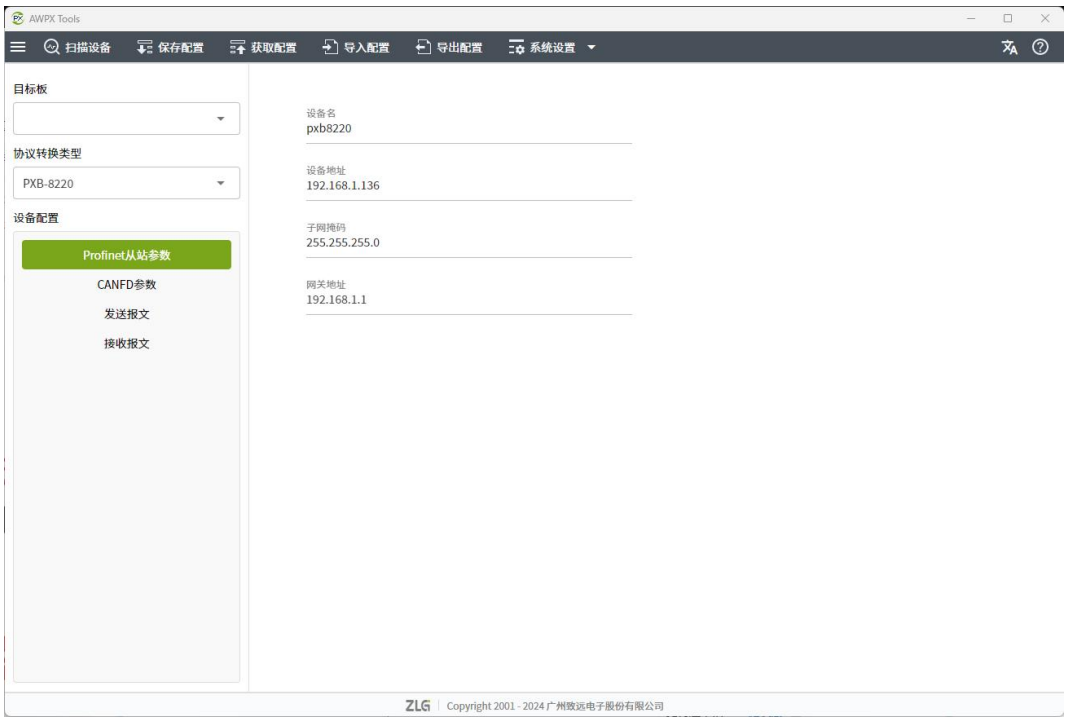


Figure 3.41 AWPX interface

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 3.42.


Figure 3.42 Menu bar buttons

1. Scanning Device

Click the 'Scan Device' button to connect the computer to the PXB82xx device using a USB cable. The AWPX software will search for the current USB connection to the PXB series device and display the MAC 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 send 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. Get 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.

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**: This setting is a useless option under PXB-82xx.

**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 REF _Ref154500755 \h 3.43.

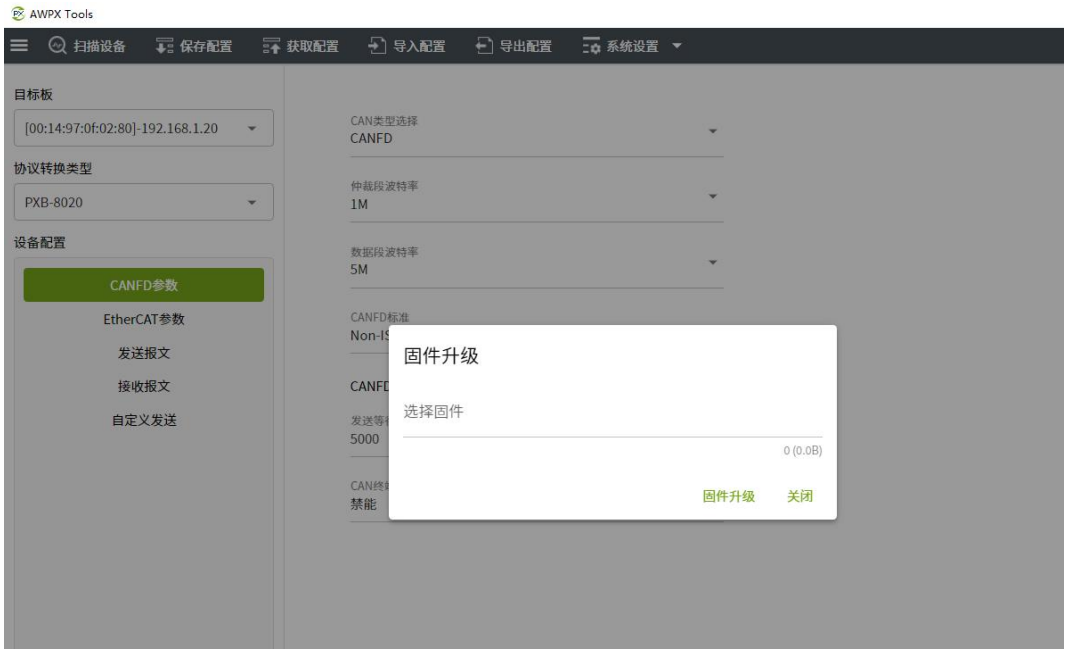



Figure 3.43 Firmware Upgrade

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

## 3.4 Device 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;

Step 2, configure parameters. You can configure the parameters in the "Device Configuration" column and network parameters according to your needs;

Step three, after completing the parameter configuration, click [Save Configuration] to save the configured parameters to the device, and wait for the device to restart to complete.

### 3.4.1 Label variable setting

Click on 'Tag Variables' in the' Device Configuration 'column of AWPX software to enter the tag variable configuration page. Tag variable configuration requires first creating a tag group, and then creating tag variables within the tag group.

1. Tag Group

Click the add button in the upper right corner of the 'Tag Variables' page to add tag groups.The maximum length of the group name for a label group is a string of 16 bytes (a-z, A-Z, 0-9), and the group name must be unique. When creating a new label group, the system defaults to the group name as "group", and AWPX can set multiple label groups.

2. Tag Variable

After the tag group is established, tag variables can be added within the tag group, and there is a button to add data on the tag group.When adding tags to a tag group, it is necessary to set the tag name, tag type, write permission, and initial value field of the tag variable.The tag name is a string with a maximum length of 16 bytes, and it needs to be unique within the tag group.Adding tag variables is incorrect!**No bookmark defined.**As shown.

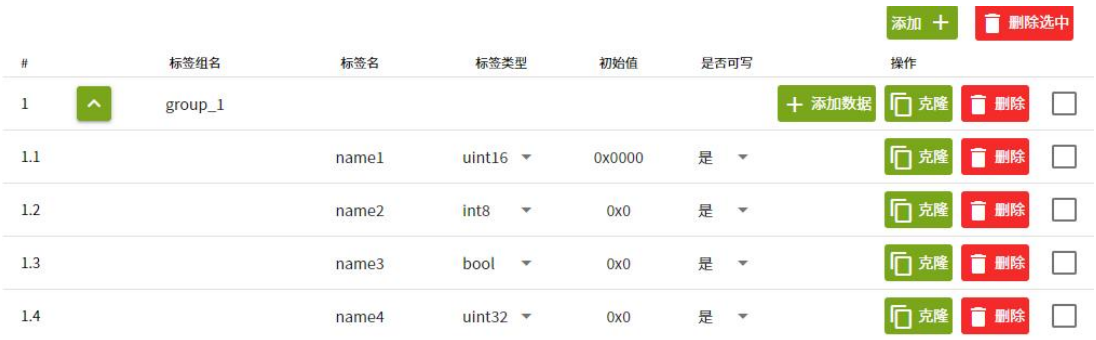



Figure 3.44: Creating a new label variable

### 3.4.2 PROFINET Protocol Settings

1. Sending Messages From The Station

The slave station sends messages to set the mapping relationship between PROFINET slave station sent messages and label variables. The set offset is the offset position in PROFINET sent messages, and the protocol converter associates the label variables with the current offset.The setting interface is shown in Figure 3.45.


Figure 3.45 PROFINET Message Sending Settings

2. Receive Messages from The Station

The slave station receives messages to set the mapping relationship between PROFINET slave station received messages and label variables. The set offset is the offset position in PROFINET received messages, and the protocol converter associates the label variables with the current offset.The setting interface is shown in Figure 3.46.

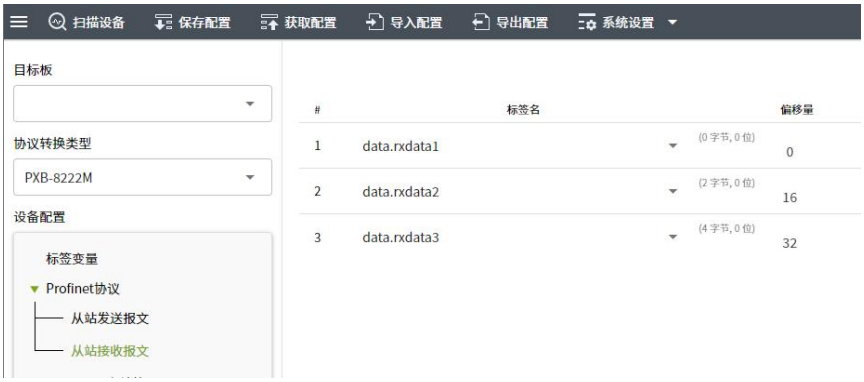



Figure 3.46 PROFINET Received Message Settings

### 3.4.3 CANFD protocol parameter configuration

1. CANFD parameters

Click on 'CAN FD Parameters' under' CAN FD Protocol 'to set the CAN FD parameters of the device, as shown in Figure 3.47.


Figure 3.47 CAN FD Parameter Interface

The CAN FD parameters are described in Table 3.1.

Table 3.1 Description of CAN FD Parameters

| parameter | Parameter Description |
|---|---|
| CAN type selection | The type selection can be CAN or CAN FD |
| Arbitration segment baud rate | Set the baud rate of the arbitration segment for CAN or CAN FD messages |
| Data segment baud rate | The type is valid for CAN FD, set the data segment baud rate for CAN FD messages |
| CAN FD standard | The type is CAN FD, which is valid. Set the standard used for CAN FD messages |
| Sending waiting time | Used to set how long to wait for the device to start sending messages after power on initialization is completed |
| CAN FD terminal resistor enable | Terminal resistance of CAN FD interface for enabling or disabling devices |

2. send message

The types of messages that can be sent depend on the setting of the CAN type selection. If the CAN type selection is set to CAN, the device can only send CAN messages;If the CAN type selection is set to CAN FD, the device can choose to send CAN, CAN FD, and CAN FD acceleration messages.

Click on 'Send Message' under the 'CAN FD Protocol' to set parameters such as the type and mode of message transmission by the device. The interface is shown in Figure 3.48.


Figure 3.48 Message Sending Interface

Add up to 128 sent messages 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 32 variables added per message.The 'Clone' button allows for batch cloning and adding tag groups or tags.Click 'Delete Selected' to delete selected items in bulk, or click 'Delete' to delete some items.The parameter description of the message sending interface is shown in Table 3.2.

Table 3.2 Parameter Description of Message Sending Interface

| parameter | Parameter Description |
|---|---|
| Message Name | The name of the message |
| 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 |
| CAN type | The types of frames sent are CAN, CANFD, and CANFD acceleration, which are optional |
| 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 the mode for the device to send CAN messages |
| Trigger frame ID | The CAN message ID that triggers the device to send CAN messages is valid when the triggering mode is selected as [Frame ID Trigger].Can be decimal or hexadecimal (starting with 0x) |
| Trigger frame type | The CAN message type that triggers the device to send CAN messages is valid when the trigger mode is selected as [Frame ID trigger] |
| Cycle Time | When the triggering mode is' periodic sending ', this time is the time for periodic message sending;<br>When the triggering mode is' Change Send ', this time is the cycle for checking the changes in tag values;<br>When the triggering mode is' single send ', this time is the waiting time for that single send;<br>When the triggering mode is' cyclic group sending ', this time is the sending interval between two CAN messages within one cyclic group;<br>When the triggering mode is [conditional sending], this time is the cycle time for triggering the device to send CAN messages periodically after the tag value corresponding to the conditional tag name is True |
| Loop group ID | The ID of a loop group, which can be understood as a list for sending messages. A loop group ID represents a sending list, and the messages in the list will be sent in a loop |
| Loop group number | The sequence number of the message sent within the loop group, which can be understood as the sequence number of each message sent within the loop group. The smaller loop group sequence number is sent first |

| parameter | Parameter Description |
|---|---|
| Condition label name | Check whether the value of the tag is a True tag name, which is valid when the trigger mode is selected as [conditional send], and the corresponding tag type should be set to boolean.When the tag value corresponding to the condition tag name is True, it triggers the device to periodically send CAN messages |
| variable name | Name of variable |
| tag name | The name of the label.Group_1. name1 represents the tag named name1 in the group_1 tag group |
| offset | The offset of CAN messages, measured in bits.Setting it to 1 means that starting from the first bit of the CAN message data segment, the tag value is mapped to the CAN message data segment, and the mapping size depends on the tag type |

The modes for triggering the device to send CAN messages are:

**Periodic transmission**: When this trigger mode is set, the device will cyclically send CAN messages based on the [cycle time].

**Change sending**: When this trigger mode is set, if the value of any tag in the message entry changes, the device will be triggered to send a CAN message.

**Single send**: When setting this trigger mode, only send the CAN message once.

**Frame ID trigger**: When this trigger mode is set, if the device receives a CAN message that matches the set 【 Trigger ID 】 and 【 Trigger Frame Type 】, it triggers the device to send a CAN message.

**Loop group sending**: When this trigger mode is set, messages with the same loop group ID will be sent in a loop according to the loop group number. The message with the smaller loop group number will be sent first, and the sending interval between two CAN messages is set by the [cycle time].

**Conditional sending**: When this triggering mode is set, if the value of the label corresponding to the [conditional label name] is True, the device will be triggered to periodically send CAN messages.

Example of message sending configuration: Configure the message sending interface as shown in Figure 1.3.

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

Sending message count: The device supports counting the sending of successfully sent CAN messages, and the count value is saved in the sending count tag.The setup steps are as follows: Create a canfdsys tag group ->Create a send count tag in the format of tc_+frame ID. The frame ID must be in hexadecimal and aligned with 8 bytes. The tag type also needs to be set to uint64.For example, when counting 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 3.49.


Figure 3.49 Sending Message Counting Configuration

3. Receive message

The types of messages that can be received depend on the setting of the CAN type selection. If the CAN type selection is set to CAN, the device can only receive CAN messages;If the CAN type selection is set to CAN FD, the device can receive CAN, CAN FD, and CAN FD acceleration messages.

Click on 'Receive Message' under the 'CAN FD Protocol' to set parameters such as frame ID and frame type for the device to receive messages. The interface is shown in Figure 3.50.


Figure 3.50 Receiving Message Interface

Add up to 128 received messages 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 32 variables added per message.The 'Clone' button allows for batch cloning and adding tag groups or tags.Click 'Delete Selected' to delete selected items in bulk, or click 'Delete' to delete some items.The parameter description of the message receiving interface is shown in Table 3.3.

Table 3.3 Description of Interface Parameters for Receiving Messages

| parameter | Parameter Description |
|---|---|
| Message Name | The name of the message |
| Frame ID | Set the frame ID for receiving messages, which can be decimal or hexadecimal (starting with 0x) |

| parameter | Parameter Description |
|---|---|
| frame type | Set whether the received message is a standard frame or an extended frame |
| variable name | Name of variable |
| tag name | The name of the label.Group_1. name1 represents the tag named name1 in the group_1 tag group |
| offset | The offset of CAN messages, measured in bits.Setting it to 1 means that starting from the first bit of the received CAN message data segment, the content of the CAN message data segment is mapped to a label, and the mapping size depends on the label type |

Example of receiving message configuration: Configure the receiving message interface as shown in Figure 3.50.

When the device receives a CAN standard frame with a frame ID of 0x2 and frame data of 11 22 33 44 55 66 77 88 (hexadecimal), the device will start from the second byte of the CAN frame data segment and sequentially map the data of the two bytes to the label flow, and the value of the label flow will become 0x4433.

Received message counting: The device supports receiving and counting successfully received CAN messages, and the count value is saved in the received count tag.The setup steps are as follows: Create a canfdsys tag group ->Create a receive count tag in the format of rc_+frame ID. The frame ID must be in hexadecimal and aligned with 8 bytes. The tag type also needs to be set to uint64.For example, when counting the reception of messages with frame IDs 0x1 and 0x1A000000, the tag names should be set to rc_00000001 and rc_1A000000, as shown in Figure 3.51.


Figure 3.51 Receiving Message Counting Configuration

### 3.4.4 CANopen master station protocol configuration

1. CANopen master station parameters

Click on 'CANopen Master Station Protocol' in the 'Device Configuration' column of the AWPX software, and select 'CANopen Master Station Configuration' to configure the relevant parameters of the CANopen master station. The configuration interface is shown in Figure 3.52.


Figure 3.52 CANopen Master Station Parameter Configuration Interface

The CANopen master station parameters are described in Table 3.4.

Table 3.4 CANopen Master Station Parameter Description

| parameter | Parameter Description |
|---|---|
| Node ID | Main station node ID |
| Baud rate | Set the CAN baud rate of the main station (10K, 20K, 50K, 100K, 125K, 250K, 500K, 800K, 1M) |
| Main station synchronization mechanism | Set whether to activate the main station synchronization message mechanism |
| Delay start time | Do you want to delay starting the main station |
| Terminal resistance enable | Is the terminal resistor enabled |

2. CANopen Slave Parameters

After clicking on the 'CANopen Master Protocol' in the 'Device Configuration' column, and then clicking on the 'CANopen Slave List', you can view the established CANopen slave nodes and add or remove them by right clicking.The interface is shown in Figure 3.53 and Figure 3.54.

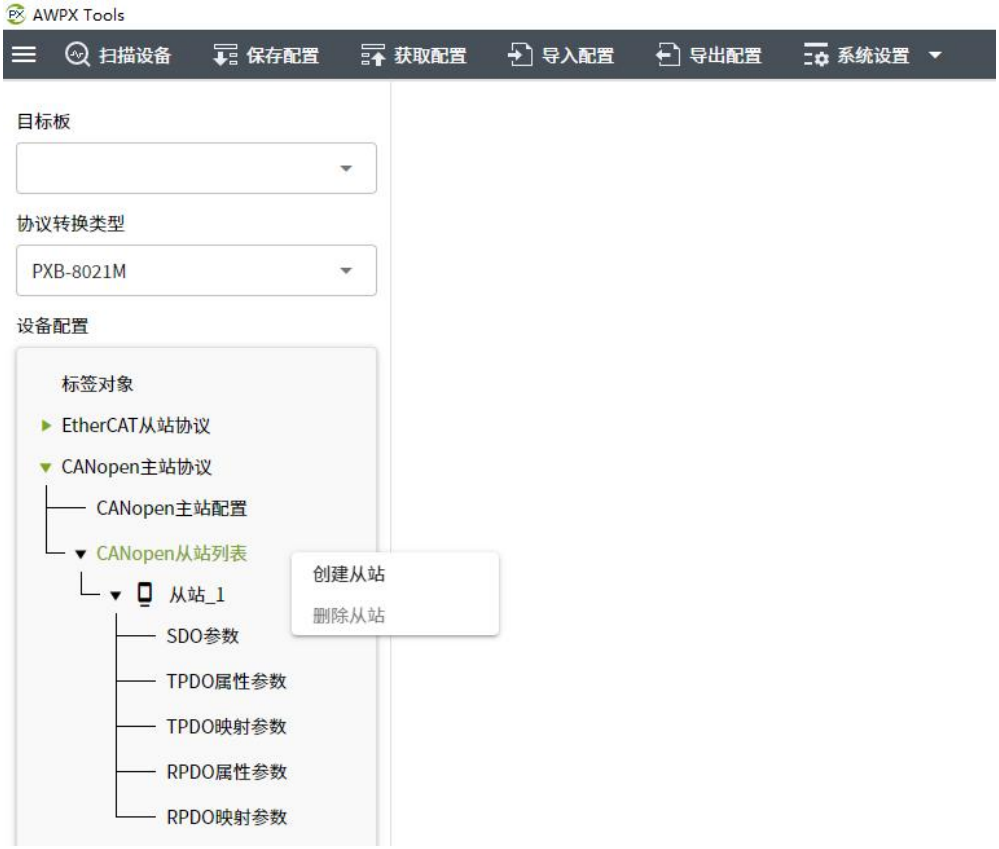



Figure 3.53 Creating a Slave Node


Figure 3.54 Slave Station Configuration Interface

The CANopen slave parameters are described in Table 3.5.

Table 3.5 Slave Configuration Interface

| parameter | Parameter Description |
|---|---|
| Station name | Slave node name, user mnemonic |
| Node ID | Slave node ID |
| Slave synchronization mechanism | Whether to configure the slave synchronization message mechanism |
| Slave detection mode | Can the detection mode of the slave station be enabled |
| Detection Mode | Under the premise of enabling detection mode, configure node guardian mode or heartbeat mode |
| inspection cycle | Detection time period (ms) |

1. SDO parameter configuration

After clicking on 'SDO Parameters' in the' Device Configuration 'column, the interface is shown in Figure 3.55.

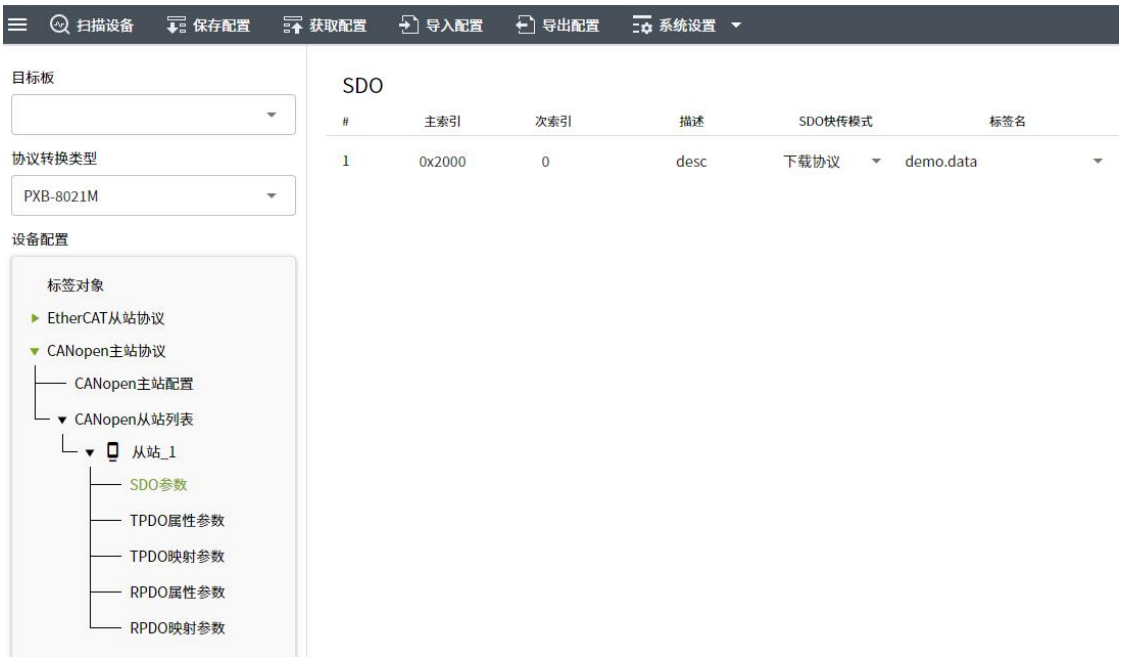



Figure 3.55 SD parameter configuration

The parameter description of CANopen slave SDO is shown in Table 3.6.

Table 3.6 SDO Configuration Parameter Description

| parameter | Parameter Description |
|---|---|
| main index | The main index of the substation dictionary can be decimal or hexadecimal (starting with 0x) |
| Secondary Index | Sub index from the site dictionary |
| description | User note information |
| SOD transmission mode | Currently only supports downloading protocols |
| tag name | The data load transmitted by SDO |

2. TPDO attribute parameters

After clicking on the 'TPDO attribute parameters' in the' Device Configuration 'column, the interface is shown in Figure 3.56.

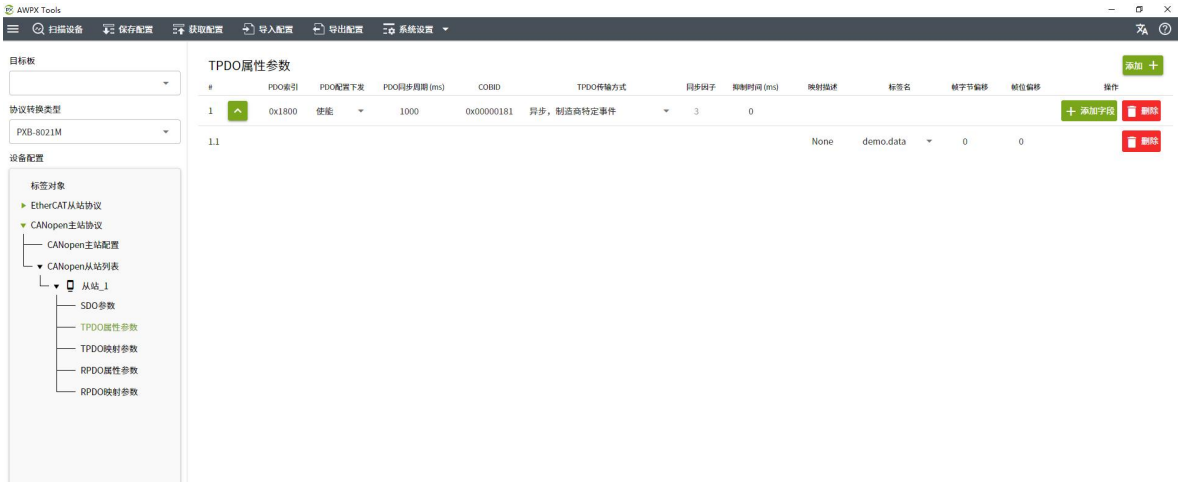



Figure 3.56 TPDO attribute parameters

The description of CANopen slave TPDO attribute parameters is shown in Table 3.7.

Table 3.7 TPDO Attribute Parameter Description

| parameter | Parameter Description |
|---|---|
| PDO index | TPDO main index from the site dictionary<br>Can be decimal or hexadecimal (starting with 0x) |
| Has PDO configuration been issued | Do you want to issue the current PDO configuration for the slave station |
| PDO synchronization period | Scheduled sending cycle from the station<br>(0 represents data change sending) |
| COBID | COBID of TPDO attribute parameter in the substation dictionary<br>Can be decimal or hexadecimal (starting with 0x) |
| TPDO transmission method | Selection of TPDO transmission mode from the substation |
| Synchronization factor | In synchronous loop mode, triggering the slave station to send data once when receiving synchronous messages N times |

| parameter | Parameter Description |
|---|---|
| Inhibition time | The minimum cycle for sending messages twice from the station |
| Mapping description | User remarks |
| label | TPDO message data distribution object |
| Frame byte offset | PDO byte offset<br>(Distribute the byte where the offset is located to the corresponding label) |
| Frame offset | PDO bit offset<br>(The tag type is only meaningful when it is a bit) |

3. TPDO mapping parameters

After clicking on the 'TPDO Mapping Parameters' in the' Device Configuration 'column, the interface is shown in Figure 3.57.

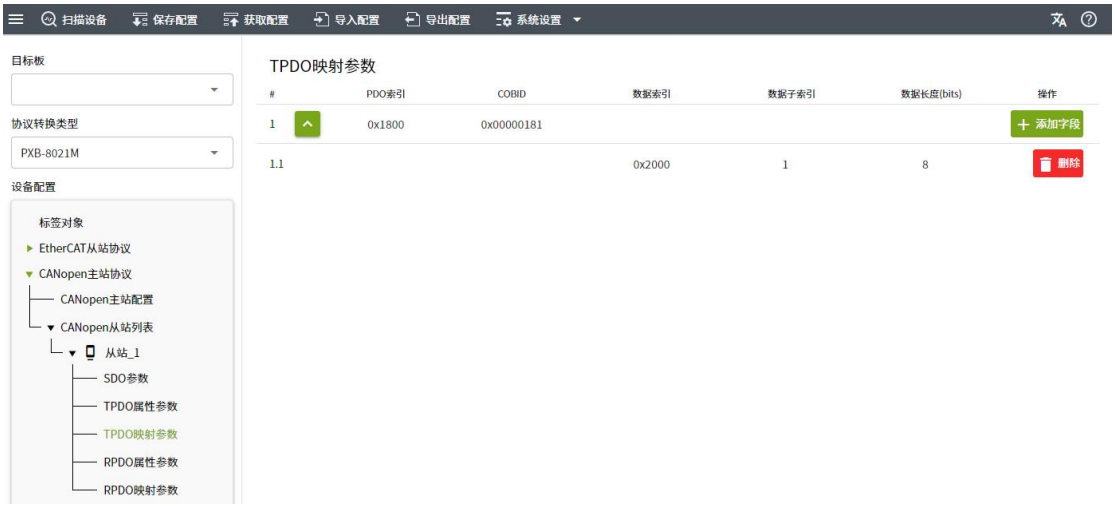



Figure 3.57 TPDO Mapping Parameters

The mapping parameters of CANopen slave TPDO are shown in Table 3.8.

Table 3.8 TPDO Mapping Parameter Explanation

| parameter | Parameter Description |
|---|---|
| PDO index | AWPX auto fill, user cannot change |
| COBID | AWPX auto fill, user cannot change |

| parameter | Parameter Description |
|---|---|
| data index | Mapping parameter master index from station dictionary<br>Can be decimal or hexadecimal (starting with 0x) |
| Data sub index | Mapping parameter sub indexes from the station dictionary |
| Data length (bits) | Mapping data length<br>(1~64) |

4. RPDO attribute parameters

After clicking on the RPDO attribute parameters in the "Device Configuration" column, the interface is shown in Figure 3.58.


Figure 3.58 RPDO attribute parameters

The description of CANopen slave RPDO attribute parameters is shown in Table 3.9.

Table 3.9 RPDO Attribute Parameter Description

| parameter | Parameter Description |
|---|---|
| PDO index | RPDO main index from the site dictionary<br>Can be decimal or hexadecimal (starting with 0x) |
| Has PDO configuration been issued | Do you want to issue the current PDO configuration for the slave station |
| PDO synchronization period | Main station timed transmission cycle<br>(0 represents data change sending) |

| parameter | Parameter Description |
|---|---|
| COBID | COBID of RPDO attribute parameter in the slave dictionary<br>Can be decimal or hexadecimal (starting with 0x) |
| RPDO transmission method | Selection of RPDO transmission method from the substation |
| Synchronization factor | retain |
| Inhibition time | retain |
| Mapping description | User remarks |
| label | Fill tag data into the corresponding PDO message |
| Frame byte offset | PDO byte offset<br>(Fill the label data into the corresponding PDO message) |
| Frame offset | PDO bit offset<br>(The tag type is only meaningful when it is a bit) |

5. RPDO mapping parameters

After clicking on 'RPDO Mapping Parameters' in the' Device Configuration 'column, the interface is shown in Figure 3.59.

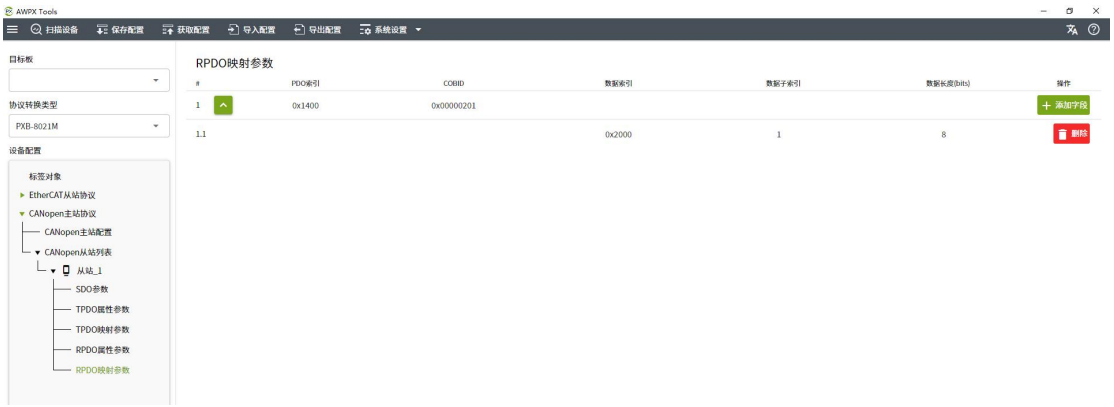



Figure 3.59 RPDO Mapping Parameters

The parameter description of CANopen slave RPDO mapping is shown in Table 3.10.

Table 3.10 RPDO Mapping Parameter Description

| parameter | Parameter Description |
|---|---|
| PDO index | AWPX auto fill, user cannot change |
| COBID | AWPX auto fill, user cannot change |
| data index | Mapping parameter master index from station dictionary<br>Can be decimal or hexadecimal (starting with 0x) |
| Data sub index | Mapping parameter sub indexes from the station dictionary |
| Data length (bits) | Mapping data length<br>(1~64) |

### 3.4.5 DevieNet main station protocol parameter configuration

1. DeviceNet master station parameters

Click on 'DeviceNet Master Station Protocol' in the 'Device Configuration' column of the AWPX software to configure the relevant parameters of the DeviceNet master station. The configuration interface is shown in Figure 3.60.

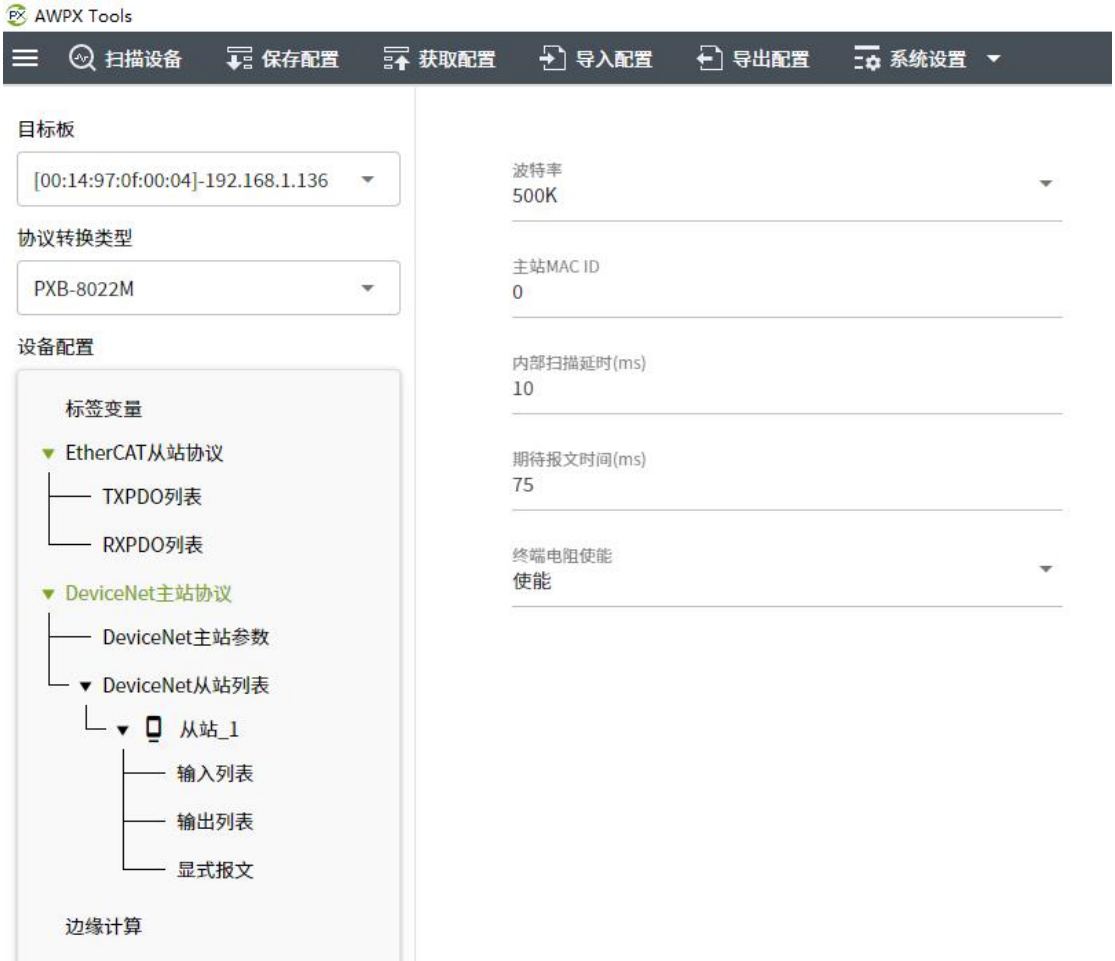



Figure 3.60 DeviceNet Master Station Parameter Interface

The parameter description of DeviceNet master station is shown in Table 3.11.

Table 3.11 Description of DeviceNet Master Station Parameters

| parameter | Parameter Description |
|---|---|
| Baud rate | CAN baud rate of DeviceNet |
| Main station MAC ID | Device address of DeviceNet master station |
| Internal scanning 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 message time | Expecting message rate, determining the timeout time for bit gating and polling messages |
| Terminal resistance enable | Enable or disable the terminal resistance of CAN interface |

2. DeviceNet Slave List

After clicking on the 'DeviceNet Master Protocol' in the 'Device Configuration' column, and then clicking on the 'DeviceNet Slave List', you can view the established DeviceNet slave nodes and add or remove them by right clicking.The interface is shown in Figure 3.61.

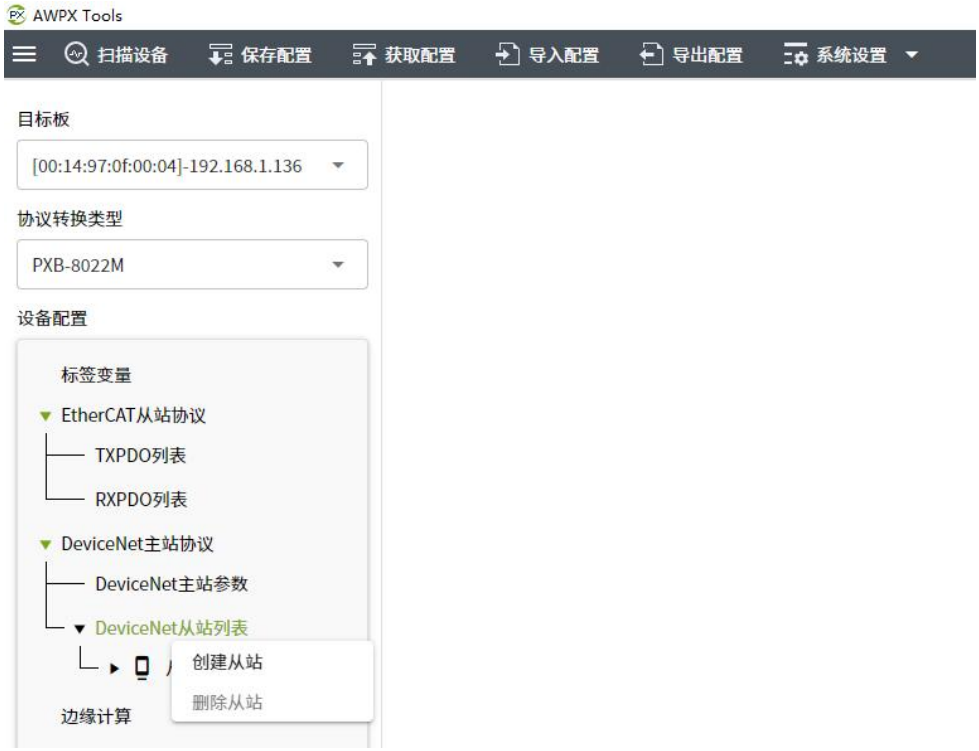



Figure 3.61 DeviceNet Slave List Interface

3. DeviceNet Slave Properties

Click on 'Create Slave' to bring up the DeviceNet Slave Configuration interface, which is used to configure the relevant attribute information of the DeviceNet Slave that needs to be connected.The interface is shown in Figure 3.62.


Figure 3.62 DeviceNet Slave Attribute Interface

On the left side of Figure 3.62, click on the corresponding slave name to configure the relevant attribute information of the DeviceNet slave. The description of the DeviceNet slave attribute information is shown in Table 3.12.

Table 3.12 Description of DeviceNet Slave Attributes

| parameter | Parameter Description |
|---|---|
| Station name | The name of this DeviceNet slave can be changed for mnemonic purposes |
| Slave MAC ID | The device address of this DeviceNet slave station |
| Data update interval | 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 station 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 station needs to be configured, which can be configured to be 1-64Byte;

2. Polling: The input and output buffer sizes of DeviceNet slave stations 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 3.63.


Figure 3.63 Mode of State Change

The parameter description of this interface is shown in Table 3.13.

Table 3.13 Description of State Change Mode Parameters

| parameter | Parameter Description |
|---|---|
| Input buffer size | The maximum size of I/O data that can be input to this DeviceNet slave station 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 |
| Response timeout | The time from sending a message to receiving a response from the DeviceNet master station when the DeviceNet slave station status changes |
| Heartbeat message time | The interval between when DeviceNet sends heartbeat messages from the slave station.When using the state change mode, the heartbeat message can query the device status at regular intervals to prevent DeviceNet from disconnecting from the slave station |
| Message production time | The time for generating DeviceNet slave messages should be less than the time for heartbeat messages |

4. Cycle: When selecting the cycle type, the interface is shown in Figure 3.64.


Figure 3.64 Periodic Mode

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

Table 3.14 Explanation of Periodic Mode Parameters

| parameter | Parameter Description |
|---|---|
| Input buffer size | The maximum size of I/O data that can be input to this DeviceNet slave station 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 |
| Response timeout | The time from sending a message from the DeviceNet slave station to receiving a response from the DeviceNet master station |
| cycle time | The time interval of DeviceNet's slave station cyclic communication can reduce unnecessary network traffic |

4. Input/output list

In the 'DeviceNet Slave List' interface, click on the 'Input List' or 'Output List' below the slave name. The interface is shown in Figure 3.65.Click the【 ▼ 】button to the left of the slave station name to hide its output, input list, and explicit messages.The output list and input list are relative to the DeviceNet master station, where the output list is a mapping entry list for the output I/O data of the DeviceNet master station, and the input list is a mapping entry list for the input I/O data of the DeviceNet master station.


Figure 3.65 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 3.15

Table 3.15 Input/Output List Parameter Description

| parameter | Parameter Description |
|---|---|
| variable name | The name of this mapping entry can be used for mnemonic purposes |
| IO type | Select the selected I/O connection type |
| tag name | Specify the label name of the I/O data mapping for the current offset.The tag needs to exist in the tag variable, and the output list data is set from the tag specified by the tag name to the DeviceNet I/O data, while the input list is set from the DeviceNet I/O data to the tag specified by the tag name. |
| offset | What is the position of the current I/O type input/output data |

### 5. Explicit Message

In the 'DeviceNet Slave List' interface, click on 'Explicit Message' below the slave name, as shown in Figure 3.66.Click the 【 ▼ 】 button to the left of the slave station name to hide its output, input list, and explicit messages.After establishing an explicit information connection, configure various parameters and exchange information for DeviceNet slave stations through this interface by setting explicit messages.


Figure 3.66 Explicit Message 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 explicit message parameter description is shown in Table 3.16.

Table 3.16 Explanations of Explicit Message Parameters

| parameter | Parameter Description |
|---|---|
| Message Name | The name of this message can be used for mnemonic purposes |
| Service code | Specify the service code corresponding to the explicit message |
| Class ID | Specify the ID of the class corresponding to the attribute obtained through explicit message configuration |
| Instance ID | Specify the ID of the instance corresponding to the attribute obtained through explicit message configuration |
| Request data length | The unit is Byte, specifying the length of the service data that sends the explicit message |
| Request timeout | The unit is ms, and after sending the explicit message, the DeviceNet slave must respond correctly within the time set here, otherwise it is a request timeout |
| Request Type | Specify the method for sending the explicit message, which can be either a timed request or a single request.<br>The timed request is to send the explicit message in a loop at the time interval set by the request cycle after the system is started.<br>[Single Request] refers to sending the explicit message only once after the time set in the [Request Cycle] after the system is started |
| Request cycle | Unit is ms, set the time interval for sending 【 timed request 】 and 【 single request 】 |

By clicking [Add Data], a mapping relationship can be established between explicit message request data or response data and label variables. The data type is used to select whether it is request data or response data, the label name is used to set the mapped label variable, and the offset is used to select the position of the data.

6. From station status

The status of the slave stations managed by the DeviceNet master station can be obtained through label variables. Users can create a label group named "dvnmst_stys" in the label variable entry of the configuration software, and create a uint16 type label variable named "slvX_status" within the label group (where X represents the MACID of the slave station, for example, when MACID is 2, the label name is slv2_status).The status values of DeviceNet slave stations are shown in Table 3.17.

Table 3.17 Description of DeviceNet Slave Status Values

| Status value (decimal) | Description of Status Values |
|---|---|
| 0 | error-free |
| 57 | Other undefined internal errors |
| 61 | The sub station does not exist |
| 62 | Sending data failed |

| | |
|---|---|
| 63 | No data |
| 66 | The main station is not online |
| 67 | Connection does not exist |
| 72 | The device stops communicating |
| 77 | Data length mismatch |
| 78 | The device is in the scanning list but has not responded |
| 84 | The device has not been initialized (main station status) |
| 86 | The device enters an idle state on its own |

### 3.4.6 Modbus protocol parameter configuration

1. Modbus settings

Click on "Modbus Protocol" in the "Device Configuration" column of AWPX software, and then click on "Modbus Settings" to configure Modbus parameters. The configuration interface is shown in Figure 3.67.


Figure 3.67 Modbus Settings

The Modbus protocol supports four working modes, namely "Modbus RTU Master", "Modbus RTU Slave", "Modbus TCP Master", and "Modbus TCP Slave". These four modes can be selected through the "Working Mode" drop-down list box, and each working mode has corresponding parameters to configure.

- Modbus RTU master station

When the working mode is Modbus RTU master, the Modbus setting parameters are shown in Figure 3.68.The "baud rate", "data bit", "stop bit", and "checksum" are used to set the communication parameters of RS485 and may need to be set;Terminal Resistance Enable "is used to set whether the onboard RS485 terminal is connected to the network, and" Timeout Time "is used to set the timeout time for RTU commands, in milliseconds. Modbus RTU Master mode supports multiple RTU slave access.


Figure 3.68 RTU Master Station Parameters

- Modbus RTU slave station

When the working mode is Modbus RTU slave, the Modbus setting parameters are shown in Figure 3.69.The "baud rate", "data bit", "stop bit", and "checksum" are used to set the communication parameters of RS485 and may need to be set;The 'Local Slave ID' is used to set the Modbus Slave ID of the device as a slave, while the 'Terminal Resistance Enable' is used to set whether the onboard RS485 terminal is connected to the network.


Figure 3.69 RTU Slave Parameters

- Modbus TCP Master

When the working mode is Modbus TCP master, the Modbus setting parameters are shown in Figure 3.70.Modbus TCP master mode supports up to 8 TCP slaves for access. Click "Add" to add slave information.The 'sub station number' is used to distinguish the sub station list in the future and cannot be repeated.


Figure 3.70 TCP Master Station Parameters

- Modbus TCP Slave

When the working mode is Modbus TCP slave, the Modbus setting parameters are shown in Figure 3.71, and the "local slave number" is used to set the ID as the slave.


Figure 3.71 TCP Slave Parameters

2. From Station List

Right click on the "Slave List" of the Modbus protocol, and select "Create Slave" or "Delete Slave" from the pop-up menu. When Modbus is in master mode, multiple slaves can be created, while when Modbus is in slave mode, there can only be one slave list.The parameter settings and function code settings for slave stations in master station mode are different from those in slave station mode.

- Main station mode

When the working mode is the master station, the parameter settings of the slave station are shown in Figure 3.72, where the "slave station number" is used to set the slave station ID for connecting to the slave station. For RTU, it can be set as needed, and for TCP, it needs to correspond one-to-one with the slave station number added in the "Modbus settings";The "polling cycle" is used to set the frequency of commands issued by the main station, which can be configured based on the data volume of the "function code list" and the real-time requirements of the application;The "send mode" is used to control the way to write to the slave station. There are two configuration options: "trigger send" and "polling send". The behavior of the "trigger send" option is to write to the slave station when the data associated with the function code changes, while the behavior of the "polling send" option is to write to the slave station according to the time set by the polling cycle;Function code retry count "is used to set the retry count after the function code fails to send;'Local data after read failure' is used to configure whether the variables associated with the function code after Modbus read failure should remain at their original values or be reset to zero;Conditional Enable "is used to control whether the slave station is enabled. When the selected conditional variable value is" True ", the slave station works normally, and when the value changes to" False ", the slave station does not work.


Figure 3.72 Slave Settings in Master Station Mode

The function code list for the main station mode is shown in Figure 3.73. The main station mode supports "01", "02", "03", "04", "05", "06", "15", and "16" function codes. Users can configure the "function code", "starting address", and "number of registers" as needed.After adding the function code, click the "Add Data" button to bind the data variable for the current function code. The "Tag Name" is used to select the variable, and the "Offset" is used to set the position of the variable associated with the current function code.

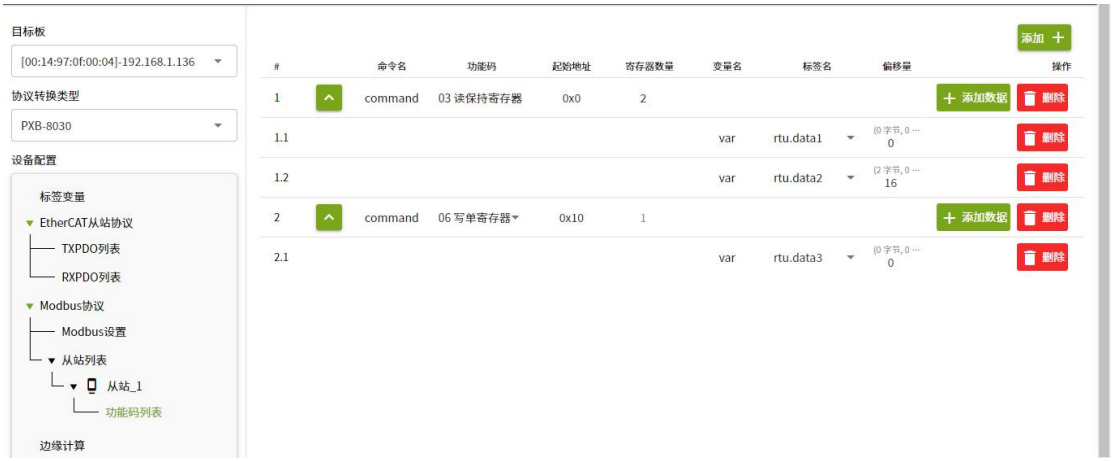



Figure 3.73 List of Function Codes in Main Station Mode

- Slave mode

When the working mode is slave station, the parameter settings of this slave station are shown in Figure 3.74. It can be seen that the parameter settings are relatively simple, only the size end parameters can be configured.


Figure 3.74 Slave Settings in Slave Mode

The function code list for the slave mode is set as shown in Figure 3.75.The register types can be selected as "coil", "discrete quantity", "hold register", and "input register". Users can configure the "register type", "starting address", and "register quantity" as needed. After adding the function code, click the "add data" button to bind the data variable for the current function code. The "label name" is used to select the variable, and the "offset" is used to set the position where the variable is associated with the current function code.

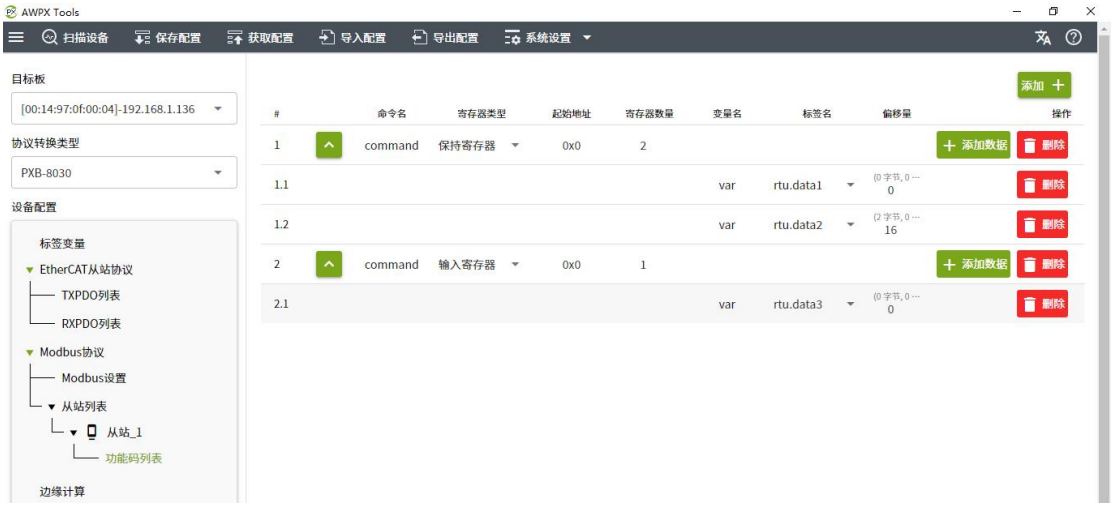



Figure 3.75 Function codes in slave mode

### 3.4.7 Edge computing Parameter Configuration

Click [edge computing] in the "Device Configuration" column of AWPX software to configure the relevant parameters of edge computing. The configuration interface is shown in Figure 3.76.

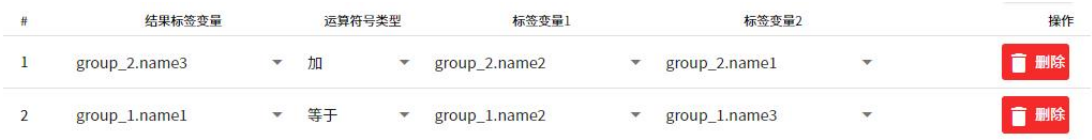



Figure 3.76 Configuring edge computing

There are four configuration items in the edge computing configuration form: "Result Label Variable", "Operation Symbol Type", "Label Variable 1" and "Label Variable 2". "Result Label Variable", "Label Variable 1" and "Label Variable 2" must be label variables that already exist in the label variable list.The types of operation symbols and operation rules are shown in Table 3.18.

Table 3.18 Description of edge computing

| Type of operation symbol | Explanation of operation rules |
|---|---|
| plus | Result label variable=label variable 1+label variable 2 |
| reduce | Result label variable=label variable 1- label variable 2 |
| ride | Result label variable=label variable 1 * label variable 2 |
| exactly divisible | Result label variable=label variable 1/label variable 2 |
| greater than | Result label variable (Boolean type)=label variable 1>label variable 2 |
| greater than or equal to | Result label variable (Boolean type)=label variable 1>=label variable 2 |
| equal to | Result label variable (Boolean type)=label variable 1==label variable 2 |
| not equal to | Result label variable (Boolean type)=label variable 1!=Tag variable 2 |
| less than or equal to | Result label variable (Boolean type)=label variable 1<=label variable 2 |
| less than | Result label variable (Boolean type)=label variable 1<label variable 2 |
| maximum | Result label variable=label variable 1>label variable 2?Tag variable 1: Tag variable 2 |
| minimum | Result label variable=label variable 1>label variable 2?Tag variable 2: Tag variable 1 |
| Modulo division | Result label variable (shaping)=label variable 1 (shaping)% label variable 2 (shaping) |
| shift left | Result label variable (shaping)=label variable 1 (shaping)<<label variable 2 (shaping) |
| Shift right | Result label variable (shaping)=label variable 1 (shaping)>>label variable 2 (shaping) |
| and | Result label variable (shaping)=label variable 1 (shaping)&label variable 2 (shaping) |
| or | Result label variable=label variable 1 (shaping) \| label variable 2 (shaping) |
| not | Result label variable (shaping)=not (label variable 1) (shaping) |
| XOR | Result label variable (shaping)=label variable 1 (shaping) xor label variable 2 (shaping) |

| | |
|---|---|
| exponentiation | Result label variable (shaping)=label variable 1 (shaping) ^ label variable 2 (shaping) |
| root extraction | Result label variable=√ Label variable 1 (shaping) |
| Logarithmic operation | Result label variable=ln (label variable 1) |
| Conditional operation | Result label variable=label variable 1 (Boolean type)?Tag variable 2: Tag variable 3 |

# 4. Product Installation

## 4.1 Mechanical dimensions

to be modified

PXB-82 series product dimensions: 125.00mm × 76.00mm × 28.00mm (bare metal).

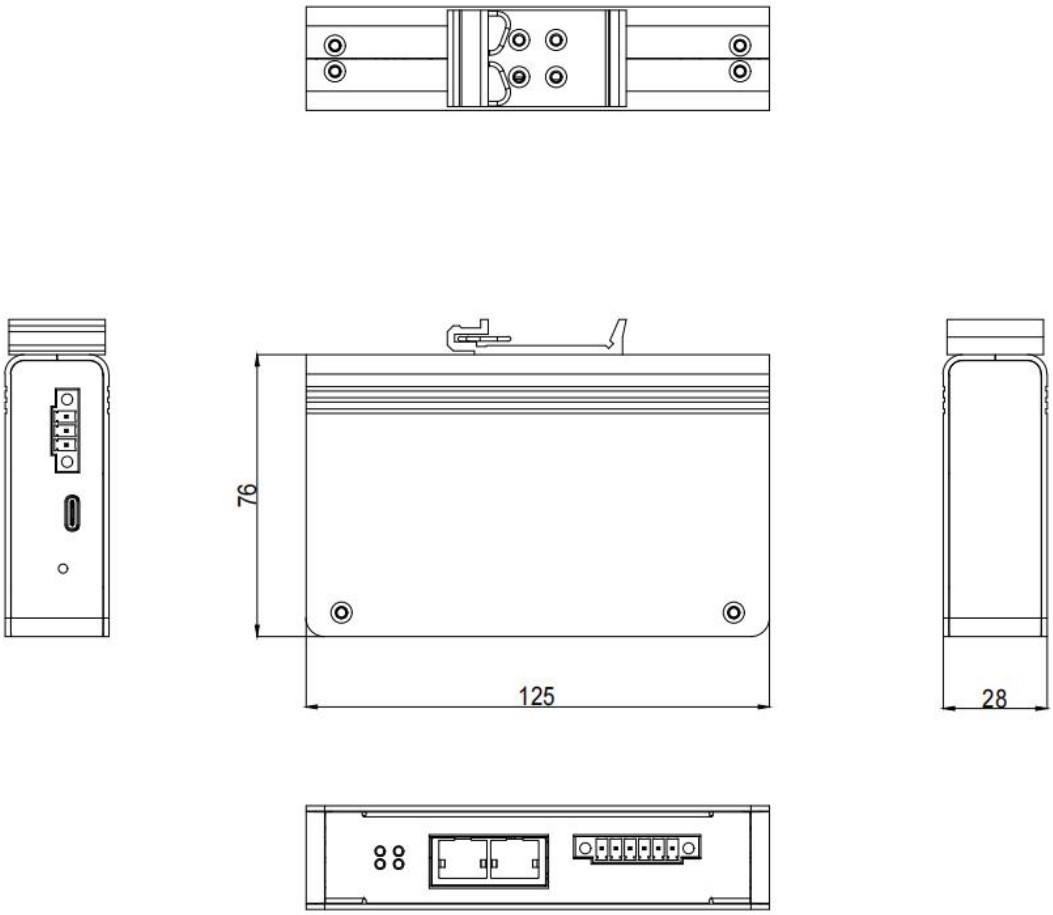



Figure 4.1 Product Dimensional Diagram

## 4.2 Installation method

The installation method of the product can refer to the official website material "Installation Instructions for Rail type PXB Series Products".

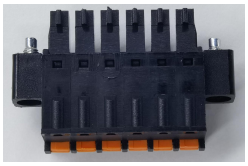
# 5. Product Maintenance And Precautions

- 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

| serial number | name | quantity | unit | physical object image |
|---|---|---|---|---|
| 1 | PXB-82xx Protocol Converter | 1 | platform |  |
| 2 | 3P wiring terminal | 1 | only |  |
| 3 | 6P wiring terminal | 1 | only |  |
| 4 | certificate of conformity | 1 | Zhang |  |

# 7. Disclaimers

In accordance with the principle of providing better services 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.To obtain the latest version of information, please visit the official website of Zhiyuan Electronics regularly or contact Zhiyuan Electronics staff.Thank you for your tolerance and support!

Integrity and win-win, continuous learning, customer first, professional focus, only do first

**Guangzhou Zhiyuan Electronics Co., Ltd**

For more details, please visit **www.zlg.cn**

Welcome to call the national service hotline **400-888-4005**