Communication Manual for LX Series Programmable Logic Controllers

Version：2026.06

Copyright Notice

Address: Di Sheng Middle Road, No.2,
Economic-Technological Development Area, 100176, Beijing, China
Tel: +86 010-5898 1588
Consulting Hotline: 400-811-1999
Technology Request: +86 010-58981514
Fax: +86 010-5898 1558
Web: http://www.hollysys.com
FA Web: https://fa.hollysys.com/
Email: PLC@hollysys.com
Sina weibo: http://weibo.com/hollysysplc

Table of Contents

Chapter 1 Foreword

1.1 Purpose

This document will introduce the use of all hardware modules in the LX system, including product introduction, module components, technical parameters, indicators, wiring instructions, diagnostic information, etc. Please use it in conjunction with other supporting product materials to help you have a more comprehensive understanding of how to use the modules.

1.2 Object

This manual is intended for use by engineers or related professional technicians with expertise in automation and control systems.

1.3 Target Products

This manual is applicable to the following products:

- LX series PLC products

1.4 Statement of Use

- The contents of this manual have been tested according to regulations, and the diagrams are consistent with the software and hardware products described. However, errors are inevitable, and complete consistency cannot be guaranteed. If you have any suggestions for improving or correcting the contents of this manual, please let us know in time and we will make corrections in the next version.
- Due to product iteration and updates, the relevant parameters and information in this manual may change. If you have any questions, please consult the technical support staff.
- Since the devices described in this manual can be used in a variety of ways, the user and the person responsible for the use of the devices must ensure that each method is admissible. HollySys will not be legally responsible for any direct or indirect losses resulting from the use or misuse of these devices.
- Due to uncertainties in the actual application, HollySys assumes no responsibility for the direct use of the data provided in this manual.

1.5 Safety Precautions

1.5.1 Safety Statement

1. Please read and comply with these safety precautions before using this product.
2. To ensure personal and device safety, please strictly abide by the safety precautions on the product label and in the manual when using this product.
3. The words "Note", "Caution", "Warning" and "Danger" in this manual do not represent all the safety precautions that shall be observed, but only serve as a supplement to all precautions.
4. This product shall be used in an environment that meets the design specifications. Otherwise, it may cause faults. Device damage caused by failure to comply with relevant regulations is not covered by the product quality guarantee.
5. HollySys shall not be responsible for any personal safety accidents or property losses caused by any operation that does not comply with the provisions of this manual or does not meet the requirements.

1.5.2 Warning

For your safety and to avoid property losses, please pay attention to the warnings in this manual. The following warnings are indicated according to the hazard level from high to low. When there are multiple hazard levels, only the highest level is prompted. If the highest level is a warning that may cause personal injury, there may also be a warning that may cause property loss.

Warnings on personal safety: Indicated by a warning triangle  .

Warnings on property loss: Without any warning triangle.

 Danger
It indicates that death or serious personal injury may be caused if operations are not carried out as specified.

 Warning
It indicates that death or serious personal injury may be caused if operations are not carried out as specified.

 Caution
It indicates that minor personal injuries may be caused if operations are not carried out as specified.

Note
It indicates that property losses may be caused if operations are not carried out as specified.

Chapter 2 Document Guide

2.1 Related Manuals

The types of manuals for this product are shown in the table below. Please select and read according to the purpose of use.

Manual Name	Purpose	Content
Module Manual for LX Series Programmable Logic Controllers	Learn about the hardware modules of the LX system	Describes all hardware modules, including: Module function Hardware interface Wiring Indicator Technical parameters
Programming Manual for LX Series Programmable Logic Controllers	Learn about the use of AutoThink programming software	Describe the interface, functions, and related operations of AutoThink programming software, including: Software interface introduction Project creation Programming Online debugging function
Basic Instruction Manual for LX Series Programmable Logic Controllers	Learn about the use of basic control instructions of the LX system	Describe the use of basic control instructions, including: Instruction function Parameter Example
Motion Instruction Manual for LX Series Programmable Logic Controllers	Learn about the use of motion control instructions of the LX system	Describe the use of motion control instructions, including: Instruction function Parameter Example
PLCopen Instruction Manual for LX Series Programmable Logic Controllers	Learn about the use of PLCopen instructions of the LX system	Describe the use of PLCopen instructions, including: Parameter Instruction function Example Precautions See also
Feedback Control Algorithm Instruction Manual for LX Series Programmable Logic Controllers	Learn about the use of feedback control algorithm instructions of the LX system	Describe the use of feedback control algorithm instructions, including: Parameter Function Example Notes for Using

2.2 Usage Convention

The following symbols are used in this manual:

-  This symbol indicates helpful tips or suggestions for using the product more efficiently.

2.3 How to Get the Manual

If you need the electronic PDF file of this manual, please visit the official website of HollySys (https://www.hollyfa.com/LX-CU500.html#c_relevant_052-17326776315060) to download the file.

Chapter 3 Revision History

Version	Revision Date	Revision Content
V1.0.0	July 2023	Created
V1.1.0	October 2023	Modified content: added programming object description of master station and slave station in Configuring EtherNetIP Master Station and in EtherNet/IP Communication Example Programming Programming.
V1.1.1	December 2023	Modified content: In Configuring DeviceNet Master Station, changed the content of "In the DeviceNet Master Station tab, the master station MAC ID and baud rate can be modified" to "In the DeviceNet Master Station Configuration tab, the master station MAC ID, baud rate, and IO-cycle time (ms) can be modified"; in adding DeviceNet_Slave station step(3), changed the content of "DeviceNet_Slave module tree node" to "Double-click the added HollySys_LX-CM005 module tree node"
V1.1.2	June 2024	Added the module IM002 in the configuration example in Automatically Scanning Device
V1.1.3	August 2024	Added Hot Connection Configuration and Synchronization Unit Configuration
V1.1.4	September 2024	Added EtherCAT diagnostic error codes and EtherCAT slave ALStatusCode error codes in Fault Diagnosis
V1.2.0	May 2025	Add OPC UA Communication , Profibus-DP Network Communication and PROFINET Communication. Add a description of the starting byte offset for modules in the EtherCAT slave configuration. DeviceNet_Master added MacID communication tatus Diagnosis. EtherCAT added controller read/write parameter error codes for fault diagnosis.
V1.3.0	September 2025	Add EtherNet/IP Tag Communication and CANopen Communication. DeviceNet master diagnostics adds Network Power Fault (NetPowerFault).
V1.4.0	June 2026	Updated EtherCAT Master Configuration Cycle Time Description Added data retention setting description for when the task stops in the EtherCAT master configuration. LX-CU433 / LX-CU500 / LX-CU501 / LX-CU510 / LX-CU511 support EtherNet/IP Tag network communication. Updated error codes and added error handling methods for: EtherNet/IP Adapter fault diagnosis EtherNet/IP Scanner fault diagnosis EtherNet/IP Tag fault diagnosis Updated error codes and added error handling methods for: Modbus TCP fault diagnosis Modbus RTU fault diagnosis Added Ntwork Vriable

Chapter 4 Overview of LX Communication System

4.1 System Network Composition

The communication networks supported by the LX PLC are as follows:

- HLink network communication:

  HLink is a dynamic communication library, through which the connection with the controller can be realized to exchange data with the controller (reading and writing data, transferring files, clearing directories, deleting files, etc.). Specific functions need to be developed by third parties themselves.

- OPC UA network communication:

  It allows access to all previous objects using a single address space, making it easy to configure and use. Security features are integrated at both the protocol and application layers, enhancing overall security.

- EtherNetIP network communication:

  The LX PLC can be used as an EtherNet/IP slave station to communicate with other EtherNet/IP master stations.

  The LX PLC can be used as an EtherNet/IP master station to communicate with other EtherNet/IP slave stations.

  The LX series PLC can communicate with other controllers using tag variable names.

- DeviceNet network communication:

  The LX PLC support DeviceNet master and slave station protocols. The master station protocol is used to communicate with field instruments or sensors. The slave station protocol is used to communicate with other DeviceNet master station devices.

- EtherCAT network communication:

  The LX PLC can realize motion controller communication and distributed IO expansion through the EtherCAT network.

- Modbus TCP network communication:

  The LX PLC support Modbus TCP master and slave station protocols. The master station protocol is used to communicate with field instruments, sensors or other Modbus TCP slave stations. The slave station protocol is used for data exchange with other Modbus TCP master station devices.

- TCP/IP network communication:

  The LX PLC support TCP/IP free communication protocols, and support user-defined TCP/IP connection and data exchange formats.

- Serial port communication:

  The LX PLC can realize Modbus RTU master station, Modbus RTU slave station, or serial free communication through the RS485/RS232 interface.

- Profibus-DP Network Communication:

  The LX series PLC supports communication with controllers via EtherCAT, which is then converted to Profibus-DP for communication with lower-level mounted modules.

- PROFINET Network Communication:

  The LX series PLC converts the EtherCAT protocol to PN master protocol to enable communication with third-party device modules.

- CANopen Network Communication:

  The LX series PLC communicates with other CANopen slave devices on the field via its CAN interface.

4.2 Typical Network Applications

Applications	Supported Networks
Motion control	EtherCAT DeviceNet
Distributed IO control	EtherCAT
Data exchange with other devices, including accessing the controller via the AutoThink software	HLink OPC UA Profibus-DP TCP/IP EtherCAT Modbus TCP Modbus RTU Serial free protocol DeviceNet Ethernet/IP PROFINET RT CANopen

Chapter 5 HLink Network Communication

5.1 Network Overview

5.1.1 Introduction

- HLink is often used for secondary development by the user, and can operate the controller without the AutoThink software.
- HLink is a dynamic communication library that supports 32-bit and 64-bit versions. HLink can realize the connection with the controller, and it supports establishing 10 TCP connections with the controller. As long as the TCP connection maintains any valid link ID, it can exchange data with the controller before connection timeout (reading and writing data, transferring files, clearing directories, deleting files, etc.). Specific functions need to be developed by third parties themselves.
- HLink is based on the Boost library for cross-platform development. The Boost version used is 1.72.0 and the Boost library is used in static linking mode.
- For Boost Software License (Version 1.0), see the appendix or visit https://www.boost.org/LICENSE_1_0.txt.
- The communication medium is a standard RJ45 Ethernet interface.
- The HLink dynamic library offers a plethora of interfaces, such as those for establishing/disconnecting connections with controllers, reading/writing data, and starting/stopping tasks, which facilitate user-side secondary development. For comprehensive details, please refer to the "HLink Interface Documentation.pdf" and the "HLinkEx Interface Documentation.pdf". Both the interface documentation and the dynamic library files are housed within the "HLink" folder located in the installation directory of software. The dynamic library files encompass the following four components:
  - OnlineInterface.dll: dynamic library
  - OnlineInterface.lib: input library of the dynamic library
  - OnlineInterface.h: header file of the dynamic library
  - RsaDll.dll: Encryption dynamic library

5.1.2 Network Characteristics

- Standard RJ45 Ethernet interface.
- Support various standard Ethernet connection architectures with flexible configuration.

- HLink can realize the connection with the controller and supports establishing 10 TCP connections with the controller.

5.2 Permission Description

The HLink protocol facilitates access to the controller through AT software, auxiliary tools, and third-party software, with protocol permissions categorized into read-only, read-write, and full access. The permissions are described as follows:

- HLink supports up to 10 network connections, with no limit on the number of read-only permissions, no limit on the number of read-write permissions, and a maximum of only one full access permission. If more than one full access permission is requested, the controller will return the highest permissible level of access available.
- When a connection with full access permission is already established, subsequent connections can only obtain read-only or read-write permissions. If the current connection is read-only/read-write or the connection with full access permission is disconnected, subsequent connections may still request full access permission.
- When an HLink client with full access permission performs operations such as downloading, resetting, or cold resetting, the controller will not disconnect other connections.
- The login service returns the highest level of permission that the current client can obtain. For example, when the first HLink client connects, it may return the ability to obtain full access permission; when the second HLink client connects, it may return the ability to obtain read-write permission.

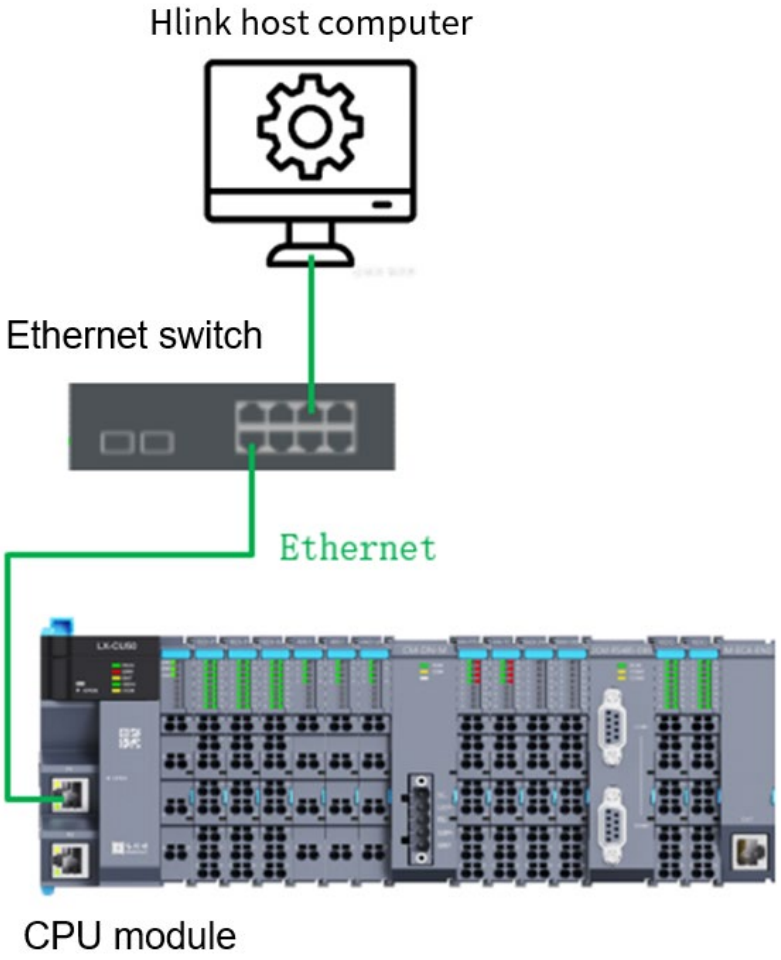
-  Note: The HLink API interface supports only 10 connections within the same process.

5.3 Network Topology

The user can directly connect the P1 or P2 interface of the CPU through a network cable, or realize the connection between the operating station and the PLC through a network topology such as star or ring composed of switches.

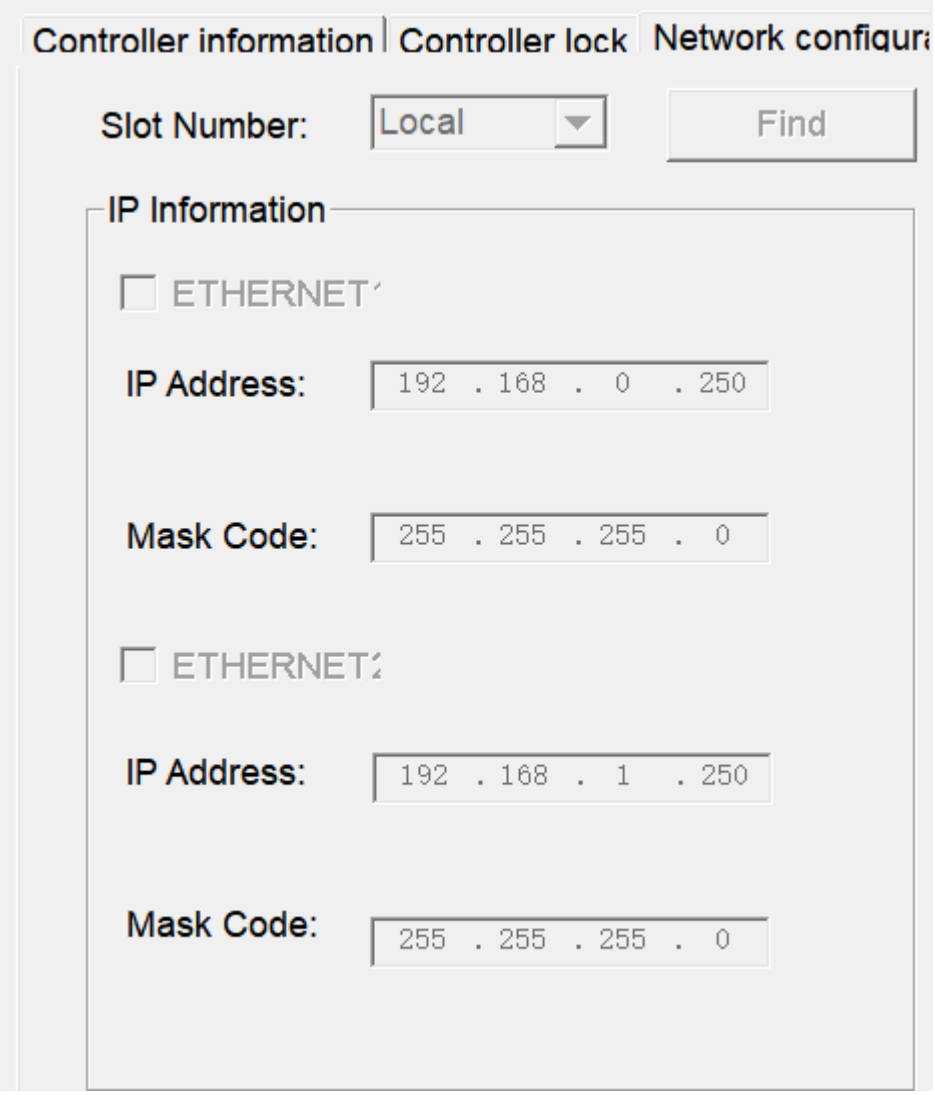
Ethernet Connection Topology

The network topology is shown in the figure below:



5.4 Configuration

To use HLink, the IP address of the CPU module shall be in the same network segment as that of the host. In the Controller Operation tool, check the Ethernet interface IP address configuration of the CPU module as follows:



The IP address modification function can be activated by checking the check box.

5.5 Program Development

Visual Studio, a Microsoft development tool, can be used for program development.

- Including header files

  #include "stdafx.h"

  #include "OnlineInterface.h"

  #include <iostream>

- Calling interface functions
  - HoliConnectController: establish a connection with the controller;

- HoliWriteAtBuffVar: write data to the controller;
- HoliDisConnectController: disconnect from the controller;

```
int _tmain(int argc, _TCHAR* argv[])
{
 unsigned int id_single; // link ID
 const char* ip = "192.168.0.250";
 const char* ucUserName = "1";
 const char* ucPassword = "1";
 unsigned int size_byte = 100;
 unsigned char buf[100];
 unsigned int ulOffset = 0;
 ControllerUserinfo* LoginUserinfo = new ControllerUserinfo(ucUserName,ucPassword);
ControllerLoginAccess*LoginAccess=newControllerLoginAccess(DEL_HOLI_APPLTYPE_PROGRAMMING_SYSTEM,DEL_HOLI_ACCESS_READ_WRITE);
 // s1 establishes link
tagEmDelHoliErrresult_connect=HoliConnectController(id_single,ip,1200,DEL_HOLI_CONTROLLER_LX,LoginAccess,LoginUserinfo);
 // s2 writes the first 100 bytes of area M as 0X55
 memset(buf, 0x55, 100);
tagEmDelHoliErrresult_w=HoliWriteAtBuffVar(id_single,DEL_HOLI_DR_MEMORY,ulOffset,size_byte,&buf);
 // s3 disconnects
 tagEmDelHoliErr result_disconnect= HoliDisConnectController(id_single);
 return 0;
}
```

Chapter 6 OPC UA Network Communication

OPC UA is an industrial communication protocol that incorporates previous protocols such as A&E, DA, OPC XML DA, and HDA. It allows access to all previous objects using a single address space, making it easy to configure and use. Security features are integrated at both the protocol and application layers, enhancing overall security. OPC UA supports configuration of the OPCUA_Server protocol under controllers, which mainly includes OPCUA_Server configuration, generation of OPC UA real-time library XML, conversion of XML to binary, and the OPC UA binary download process.

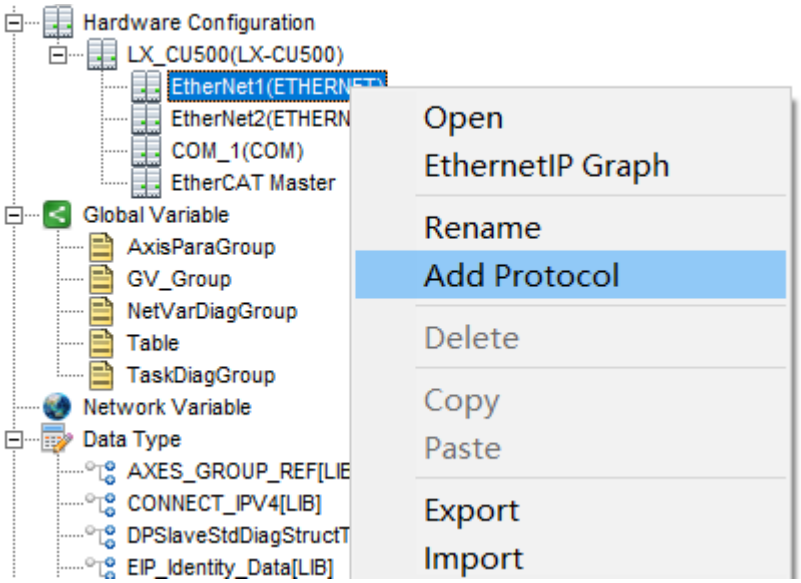
-  OPC UA is supported in versions LX-CU500-B02, LX-CU501-A02, LX-CU510-A01, LX-CU511-A01, LX-430-A02, LX-433-A01 and above.

6.1 Protocol Restrictions

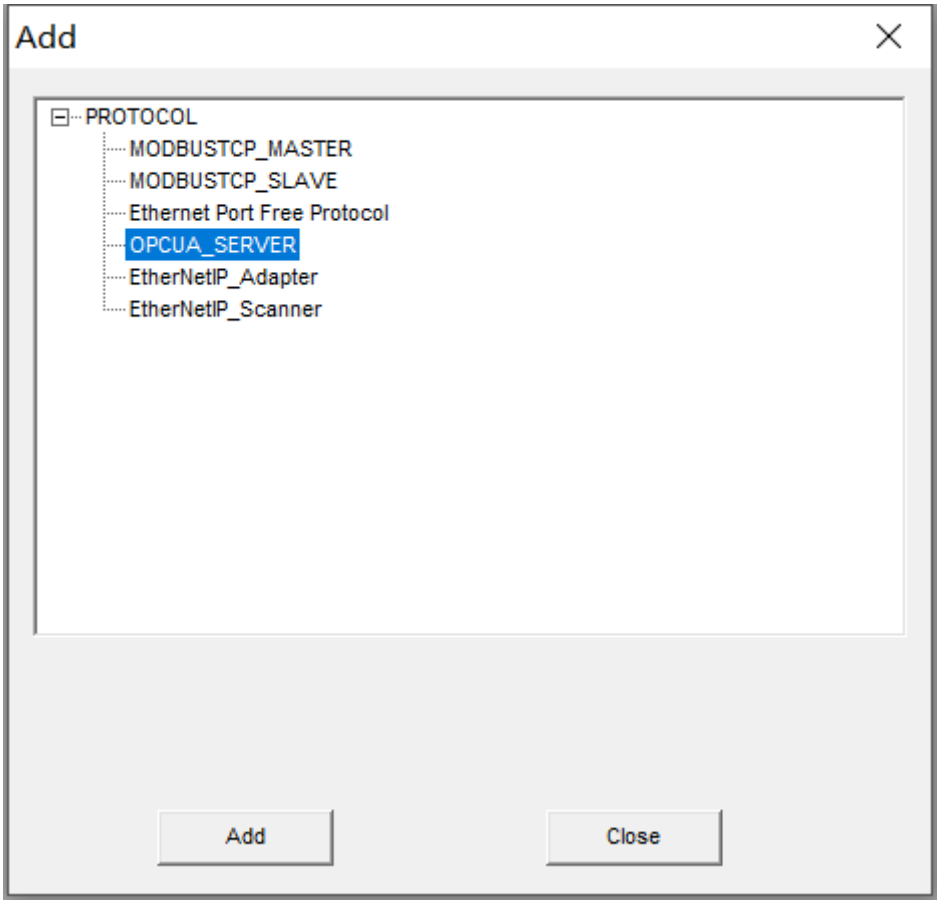
- The entire project supports the configuration of only one OPCUA_Server protocol, and any network interface can be chosen for configuration.
- The number of configurable OPC UA variables is limited by the total byte size occupied by all variables, with a maximum of 30,000 bytes for all configured OPC UA variables.
- It supports up to 10 OPC UA Clients connecting simultaneously.

6.2 OPC UA Configuration

(1) Right click the Ethernet 1 and Ethernet 2 nodes and select "Add Protocol";

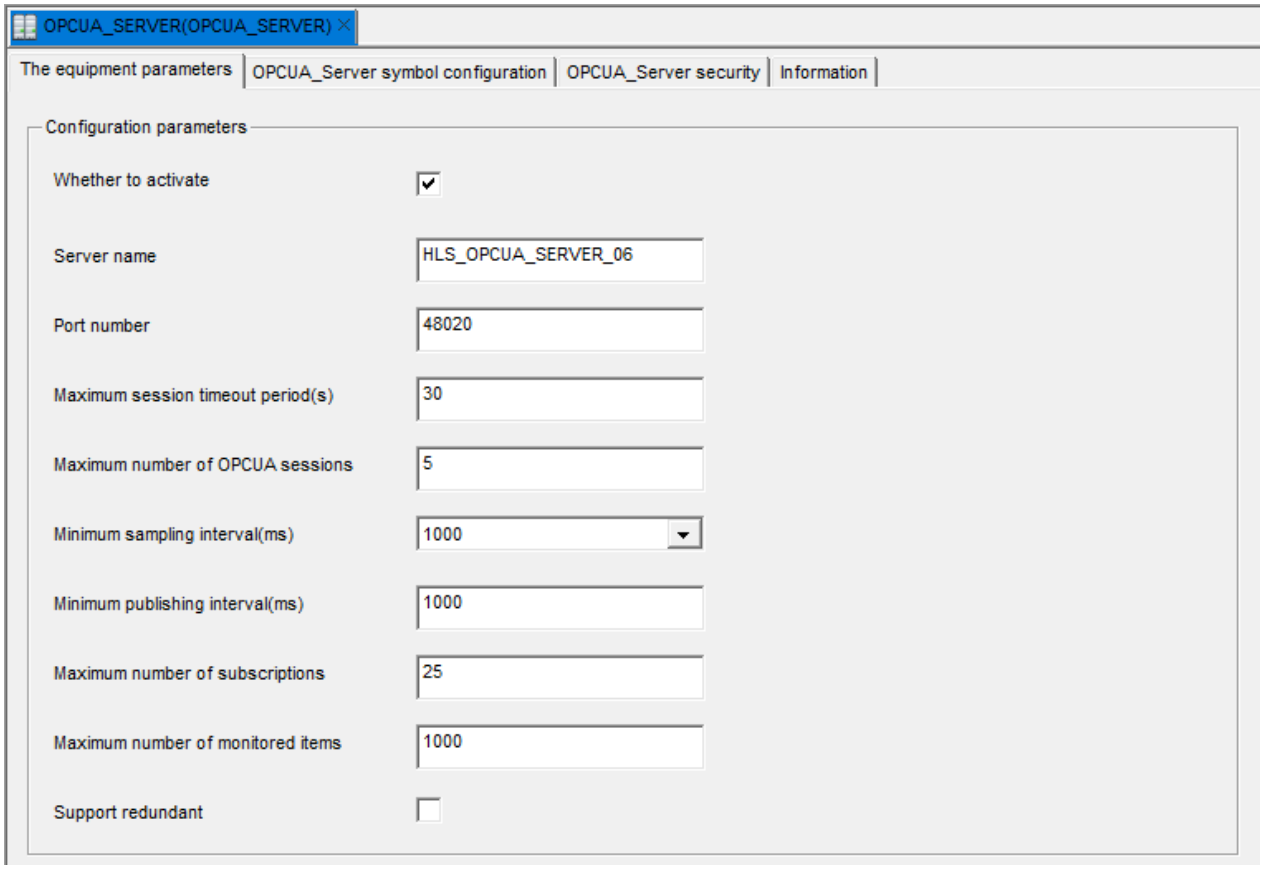


(2) Select " OPCUA_SERVER " protocol in the "Add" dialog box;



(3) Double click the " OPCUA_SERVER " tree node or right-click "Open" to open the " OPCUA_SERVER " protocol view.

The [Configuration Parameters] window is shown in the figure below, and "Whether to activate" is checked by default.



Corresponding default value and range of parameters:

- Port number (default value 48020, value range [1024.. 49151]).
- Maximum session timeout period (s): The server actively disconnects the session after the communication interruption exceeds the set time. The default value is 30, with a value range of [1.. 600000].
- Maximum number of OPC UA :The number of sessions that the client can initiate. The default value is 5, with a value range of [1.. 10]. Each session supports a maximum of 5 subscriptions, with a total maximum of 50 subscriptions supported simultaneously.
- Minimum sampling interval (ms) :The time interval at which the server samples data. The default value is 1000, with a value range of {50, 100, 250, 500, 1000, 2500, 5000, 10000}.
- Minimum publishing interval (ms):The time interval for the server to publish data. The default value is 1000, with a value range of [500.. 3600000].
- Maximum number of subscriptions: The maximum number of subscription tables supported by the server. The default value is 25, with a value range of [1.. 50].
- Maximum number of monitored items :The maximum number of sampling points supported by the server. The default value is 1000, with a value range of [100.. 13500].
- Support redundant: Check this option to enable redundancy with dual network interfaces.

6.2.1 OPC UA Variable Configuration

The data types supported by the OPCUA_Server protocol are as follows:

- Single elements: BOOL, BYTE, DINT, DWORD, INT, LREAL, REAL, SINT, UDINT, UINT, USINT, WORD, ENUM, STRING.
- Complex data types:FB, Struct, Array, Union.

OPC UA read-write permission is divided into "Only read" permission and "Read and write" permission.

By default, area S has read-only permission, which cannot be set to read-write permission.

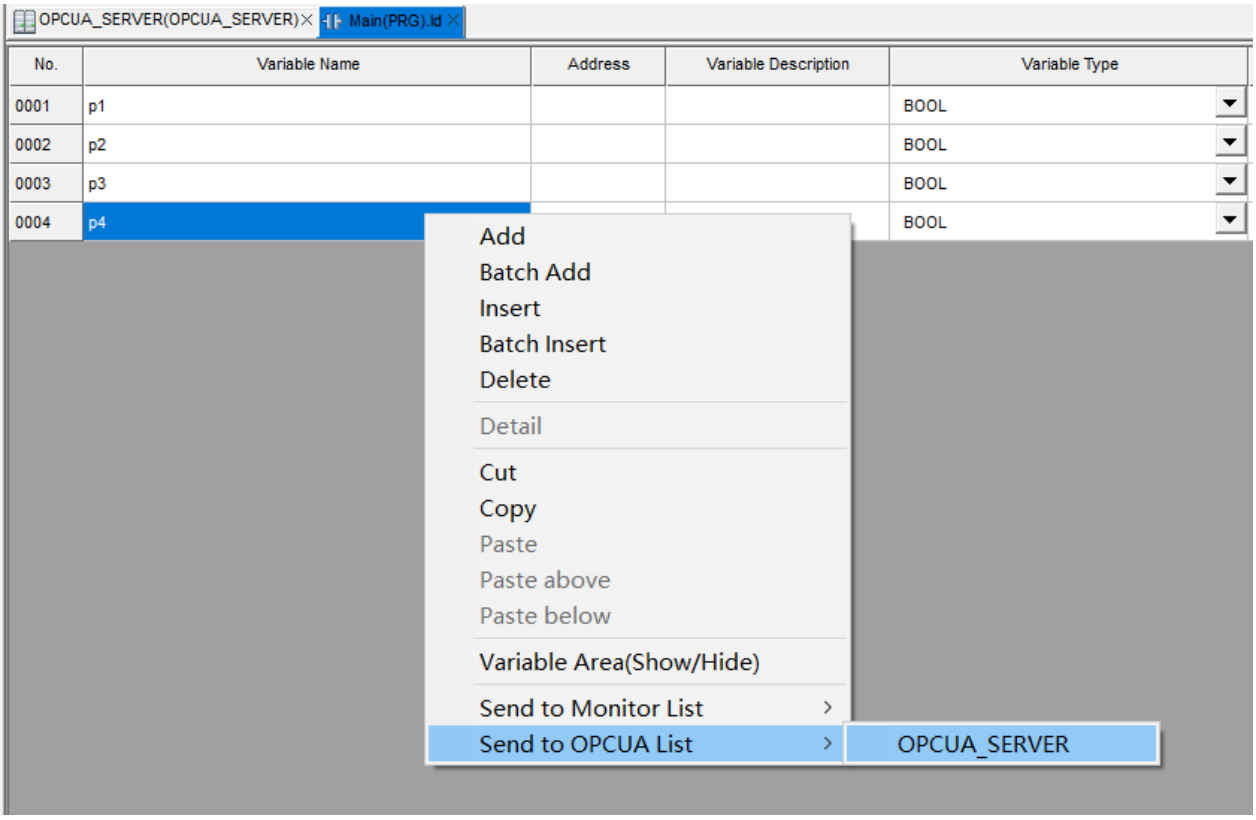
Variables in other data areas except the S area are “Read and write” permission by default.

Add variables to the “OPCUA_Server symbol configuration” tab by doing the following.

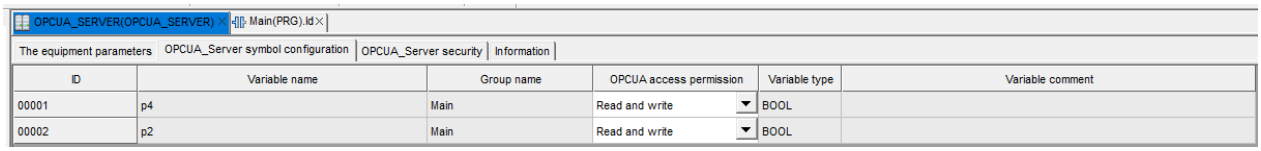
(1) Right click a variable in the POU, global variable group or I / O module channel tab;

(2) In the right-click menu, select [Send to OPCUA List] - [OPCUA_SERVER] command;

The output window shows that the variable is added to OPC UA successfully.



(3) Double click the "OPCUA_SERVER " tree node to open the configuration window. Click the " OPCUA_SERVER Symbol Configuration" tab to see the added OPC UA variables.



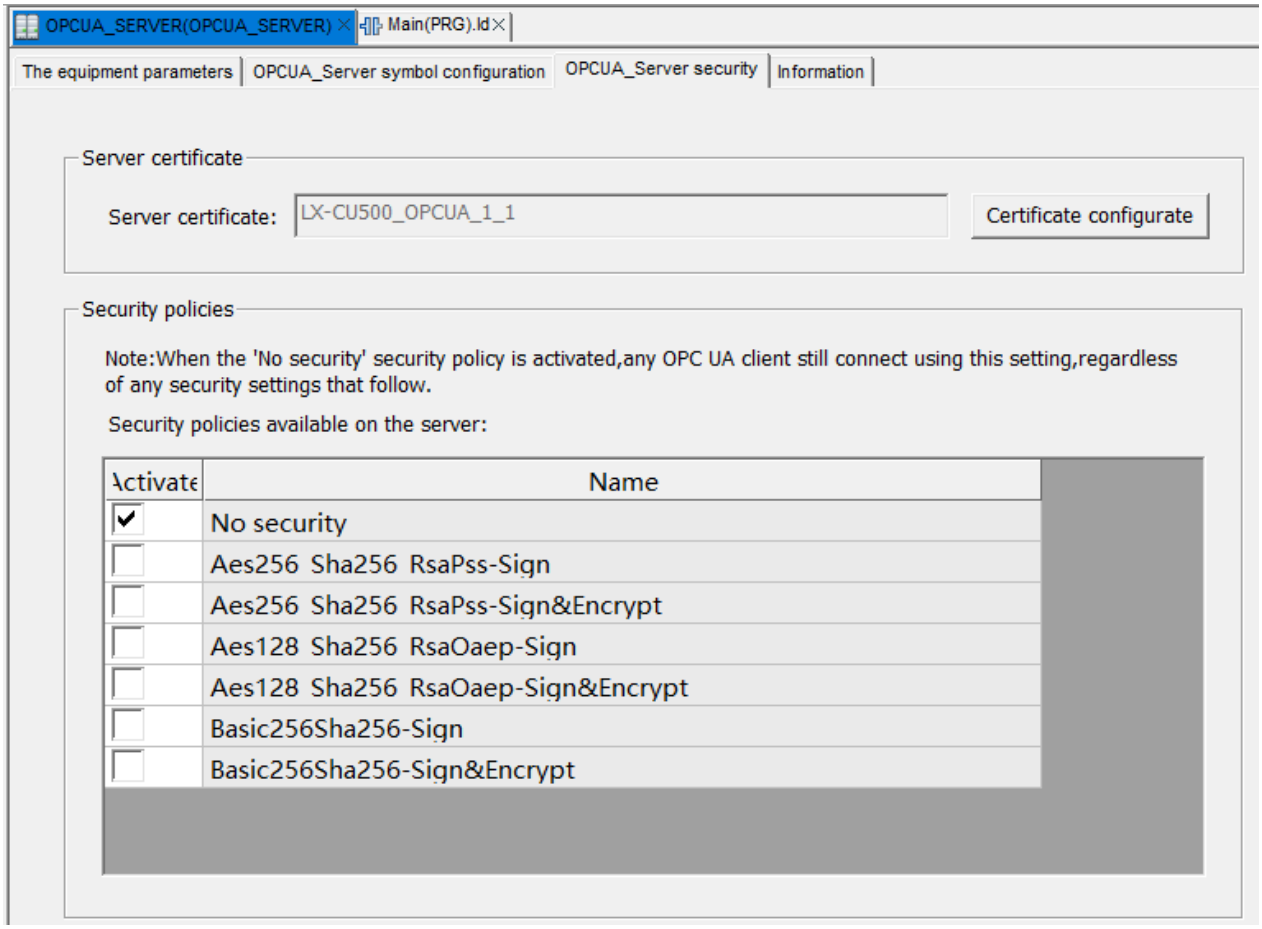
6.3 OPC UA Security

Users can assign OPCUA_Server three authentication permissions: anonymous access mode, user name password mode and certificate, which can be configured separately or at the same time.

The user selects one of the above methods on the client for authentication login, data access and interaction.

6.3.1 Server Certificate

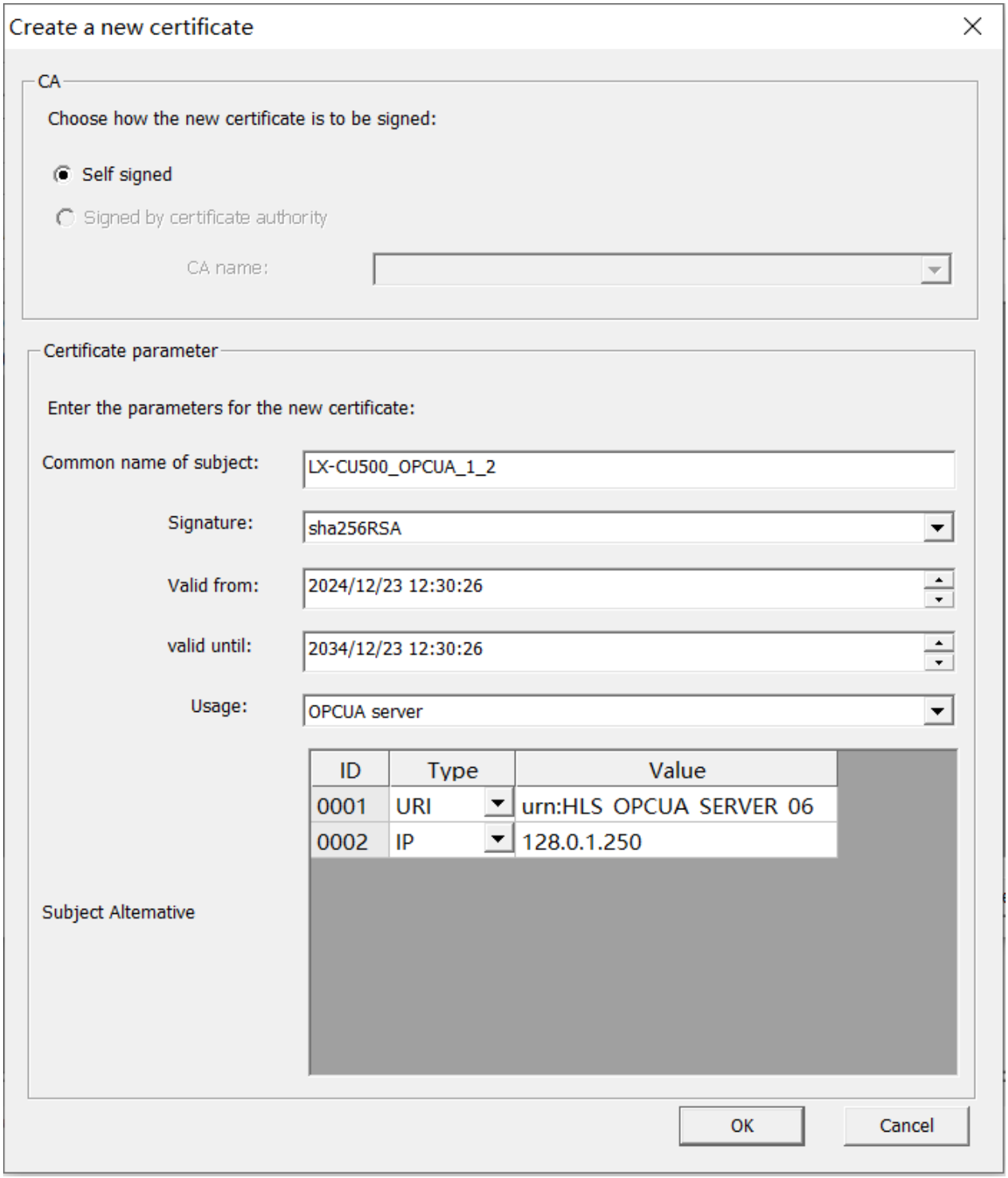
(1) Server certificate configuration and management.



- Certificate configuration provides the function of selecting certificates.
- Certificate configuration provides the function of adding new certificates.

- Certificate configuration provides the function of deleting certificates.
- Certificate configuration provides certificate configuration function.

Create a self-signed CA certificate (certification authority signature is not supported temporarily).



- Common name of subject is the certificate name.
- Signature: Support sha1RSA, sha256RSA.

- Valid from: The certificate takes effect from the specified date.
- Valid until: After the specified date expires, the certificate will become invalid.
- Usage: only "OPCUA server" certificate is supported.
- Subject Alternative Name.

(2) Server certificate generation and download

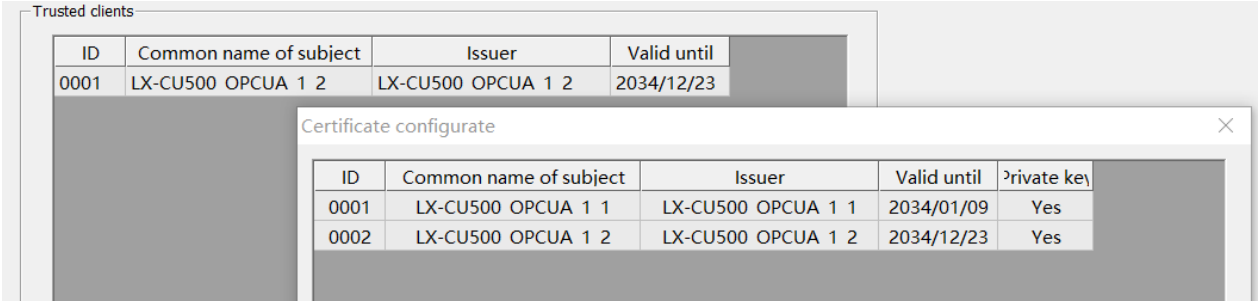
Integrate the open source code consistent with the protocol stack, generate and issue the certificate.

(3) New application name in device parameter property page

By default, the name of this application is generated according to the model of the main control and the serial number of the OPC UA module. The user OPC UA can modify it manually. It is 32 characters long and can contain upper and lower case letters, numbers and underscores, but it must start with letters or underscores. This parameter will be downloaded to the OPC UA server with the hardware configuration. The URI that generates the optional name of the server certificate contains this parameter.

6.4 Trusted Clients

Provide the trusted client certificate import function. After the user selects the corresponding client certificate to import, the certificate will be distributed to the server, and up to 64 trusted clients can be configured. Support certificate export and import functions, and support Der format without private key and P12 format certificate with private key.



6.4.1 Security Policy

Since Basic128Rsa15 and Basic256 do not recommend this security policy in the embedded version of OPCUA_Server protocol stack, so the following security policies are supported. Each security policy has two ways: "signature" and "signature and encryption".

Security Policy Table

Security Policy	URI
Aes256_Sha256_RsaPss	http://opcfoundation.org/UA/SecurityPolicy#Aes256_Sha256_RsaPss
Aes128_Sha256_RsaOaep	http://opcfoundation.org/UA/SecurityPolicy#Aes128_Sha256_RsaOaep

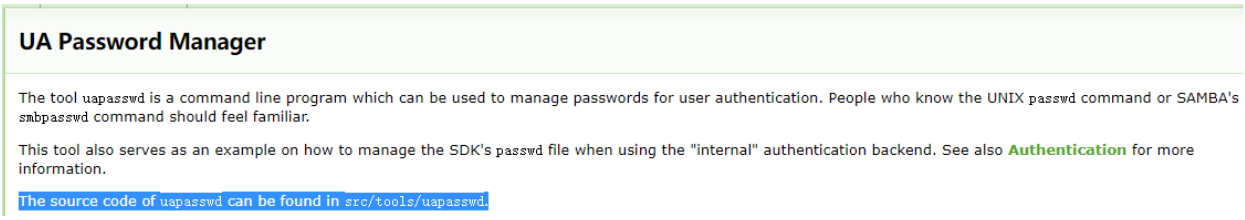
Basic256Sha256	http://opcfoundation.org/UA/SecurityPolicy#Basic256Sha256
None	http://opcfoundation.org/UA/SecurityPolicy#None

Each security policy can be configured independently. The upper computer is only configured to issue security policies.

6.4.2 User Authentication

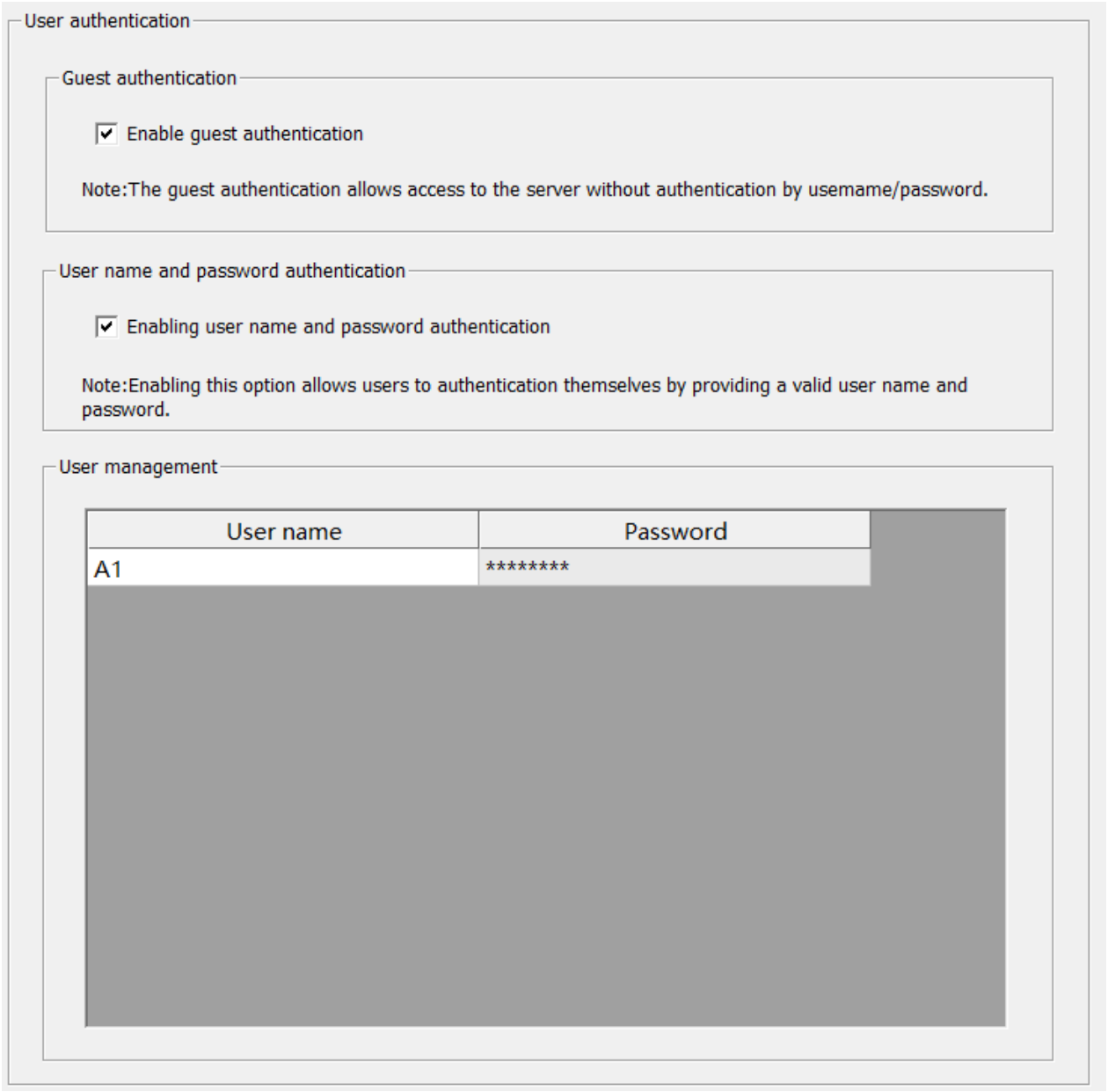
Users can be authenticated through "visitor authentication" and "user password authentication". The two authentication methods can be parallel, and the UA client selects the authentication method.

(1) Visitor authentication. Users do not need to prove their identity (anonymous access). The OPC UA server does not check the client user permissions.



(2) User password authentication (non anonymous access). User management needs to be configured after enabling. The OPC UA server will check whether the client user has the permission to access the server, authenticate through the user name and password, and support up to 20 users.

The user name is required to be a legal identifier, which is composed of uppercase and lowercase letters and numeric underscores, with a maximum length of 32 characters. The password is required to be 6 ~ 32 characters, which must contain letters, numbers and special characters, and must contain uppercase and lowercase characters.



6.5 Export OPC UA XML

After the project is compiled, the OPC UA XML file can be exported, and the generated XML can be exported to the folder specified by the user.

Perform the XML export operation as follows:

- Tree node: Right-click on [OPCUA_SERVER] - [Export XML].

This export function is only for users to view and export the OPC UA symbol point table of the current project in the format of OPC UA XML.

Chapter 7 Profibus-DP Network Communication

The LX-CM009 module communicates with the controller via EtherCAT and converts the communication to Profibus-DP to interact with the lower-level mounted modules.

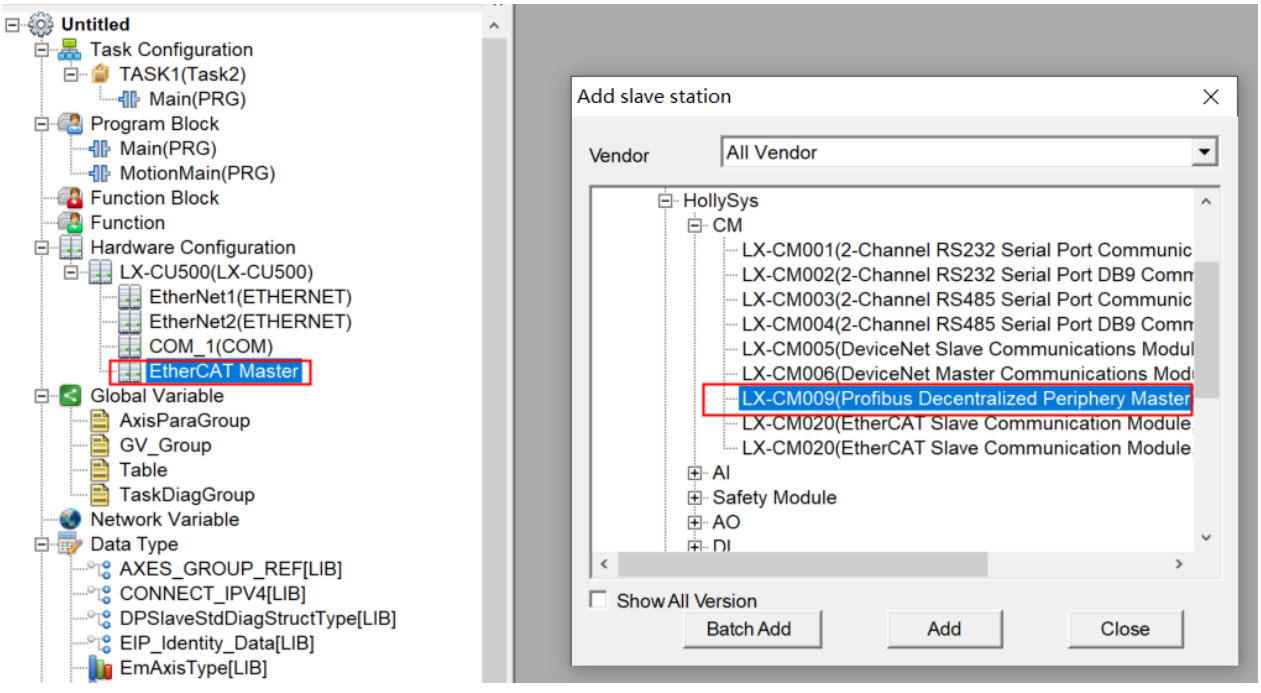
Each LX-CM009 module supports the configuration of up to 32 slave stations.

7.1 Add LX-CM009 Master

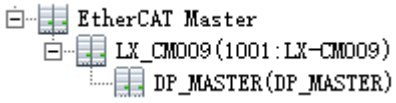
The add steps are as follows:

(1) Right-click EtherCAT Master under the Hardware Configuration node, and click Add Slave;

(2) After clicking "Add Slave," click on "CM" in the "HollySys" column, then select "LX-CM009" to add it.



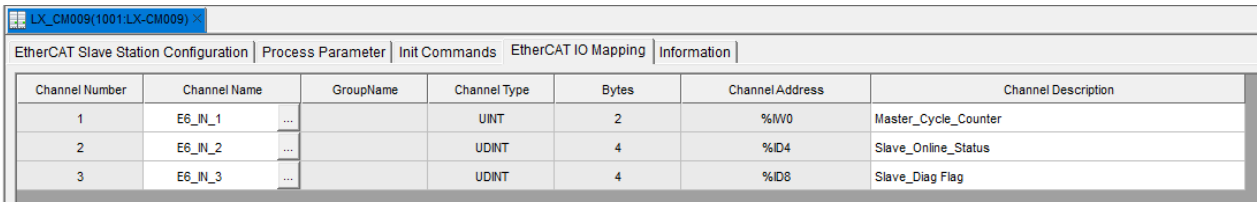
(3) The LX-CM009 and DP_MASTER have been added under the EtherCAT Master node.



7.2 Configuration LX-CM009

Double-click on the "LX_CM009" node in the left tree to open the module configuration interface.

By default, the parameters in "EtherCAT Slave Station Configuration", "Process Parameter", "Init CommandS" and "Information" in the module need not be changed. Module channel information can be viewed in the "Ether CAT IO Mapping", as shown in the following figure.



LX-CM009 EtherCAT IO Mapping

Parameter specification

Configuration Parameters	Configuration Description
Master_Cycle_Counter	Profibus-DP Master Diagnostics: The Profibus-DP bus loops the count, and if the slave is configured and the master is functioning properly, cyclic accumulation will be performed.
Slave_Online_Status	Slave Online Status (32bit): Bit0-bit31 corresponds to slave station 0-31, online is 1, offline is 0.
Slave_Diag Flag	Diagnostic identification (32bit): Bit0-bit31 corresponds to slave stations with numbers 0-31 (1 indicates that diagnostic data is valid and readable; 0 indicates that diagnostic data has not been refreshed. Diagnostic data include basic diagnosis (6 bytes) + extended diagnosis with a maximum limit of 244 bytes. Detailed diagnostic data corresponds to the sub-index of dictionary F103. For example: 1 index for # 0 slave station (F103: 1). And so on.

- Note: The slave station number here is AT configuration order, not the slave address.

7.3 Configuration Profibus-DP Master

Double-click "DP _ MASTER" on the tree node to open the Profibus-DP master device configuration page, as shown in the following figure.

Device Configuration | Information

Project	Content
Protocol name	DP_MASTER
Alias	DP_MASTER
Port	COM
Baud rate	1500.00kBits/s
TSL(0~65535)	400
Min TSDR(0~65535)	11
Max TSDR(0~65535)	150
TQUI failure/Repeater switching time(0~255)	0
TSET(0~255)	1
TTR	4449
Gap	10
Retry limit(0~255)	2
Slave interval(0~65535)	50
Poll timeout	500
Data control time	1200
Protocol	PROFIBUS-DP
Address range of slave station	2~125
Position of master/slave station	Profibus-DP master station

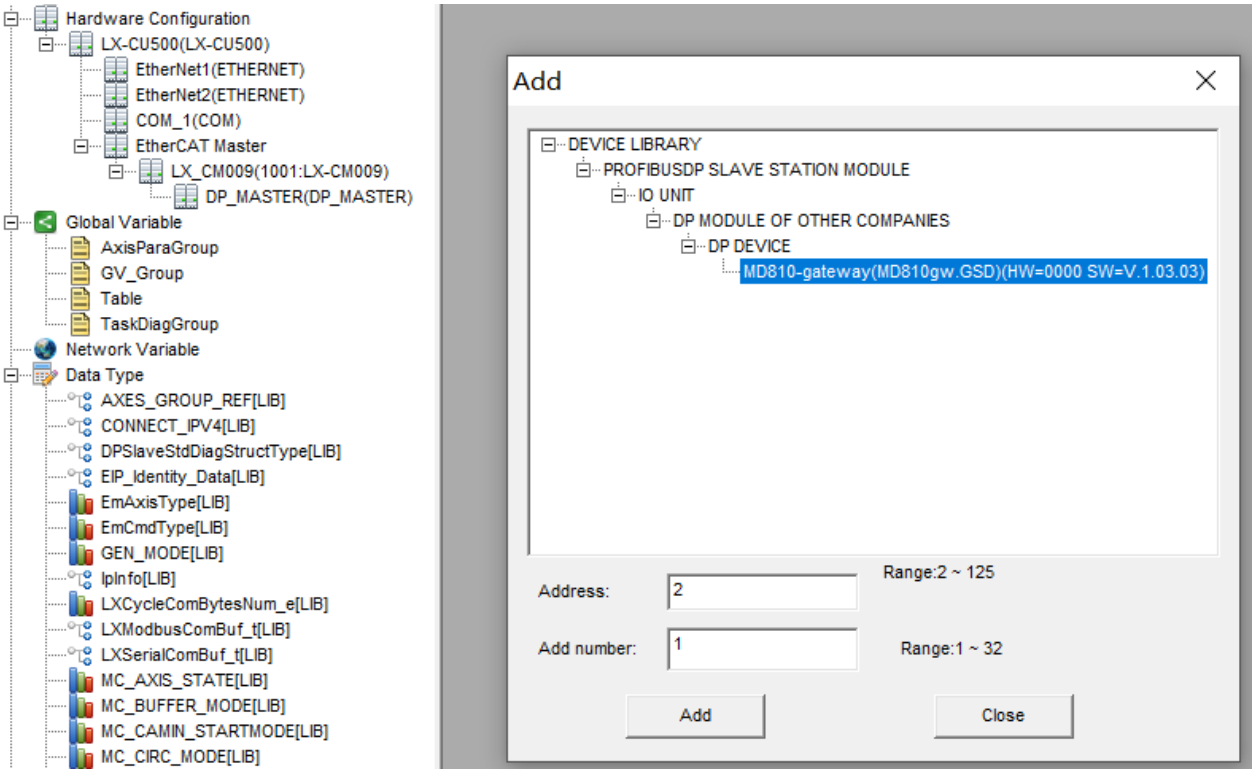
Profibus-DP Master device configuration page

- Protocol name: DP_MASTER.
- Alias: DP_MASTER.
- Baud rate: Support 9.6k,19.2k,31.25k,45.45k,93.75k,187.5k,500k,1.5M,3M.
- TSL (0 ~ 65535): The Profibus-DP polling time of a slave station; that is, retry or poll the next station if no response is received after this period of time.
- Min TSDR (0 ~ 65535):Minimum slave latency; The minimum delay from master request to the slave reply.
- Max TSDR (0 ~ 65535):Maximum slave latency; The maximum delay from the master request to the slave reply.
- TQUI failure/Repeater switching time(0~255):The switch time of Profibus-DP send failed to next resend.
- TSET(0~255):Profibus-DP connection establishment time.
- TTR:Maximum time available for token delivery.
- Gap:Number of token turns.
- Retry limit(0~255):Maximum number of retries after Profibus-DP reply timeout.
- Slave interval(0 ~ 65535):The time interval of polling the same slave twice cannot be less than this parameter, otherwise the slave may not be processed in time.
- Poll timeout:Maximum polling time for a token between master stations.
- Data control time:The maximum time for a master station to update all its slave stations, typically set to 1200ms.
- Protocol:PROFIBUS-DP.

- Address range of slave station:2~125.
- Position of master/slave station: Profibus-DP master station.

7.4 Configuration Profibus-DP Slave

(1) Right-click on the node tree "DP_MASTER", select Add Device Options, and the Add Device dialog box will pop up. Select the Profibus-DP device you want to add, as shown below.



Add Profibus-DP Device

- Before adding a Profibus-DP device, ensure that the project has imported the GSD file of the Profibus-DP Slave.

(2) Double-click the Profibus-DP slave device node to open the device information window, as shown in the figure.

The Profibus-DP slave station can be configured with an alias, device address, and device properties. The input/output starting byte offset can be configured after adding the module in the "Device Properties" dialog.

MD810_gateway(2:MD810-gateway)

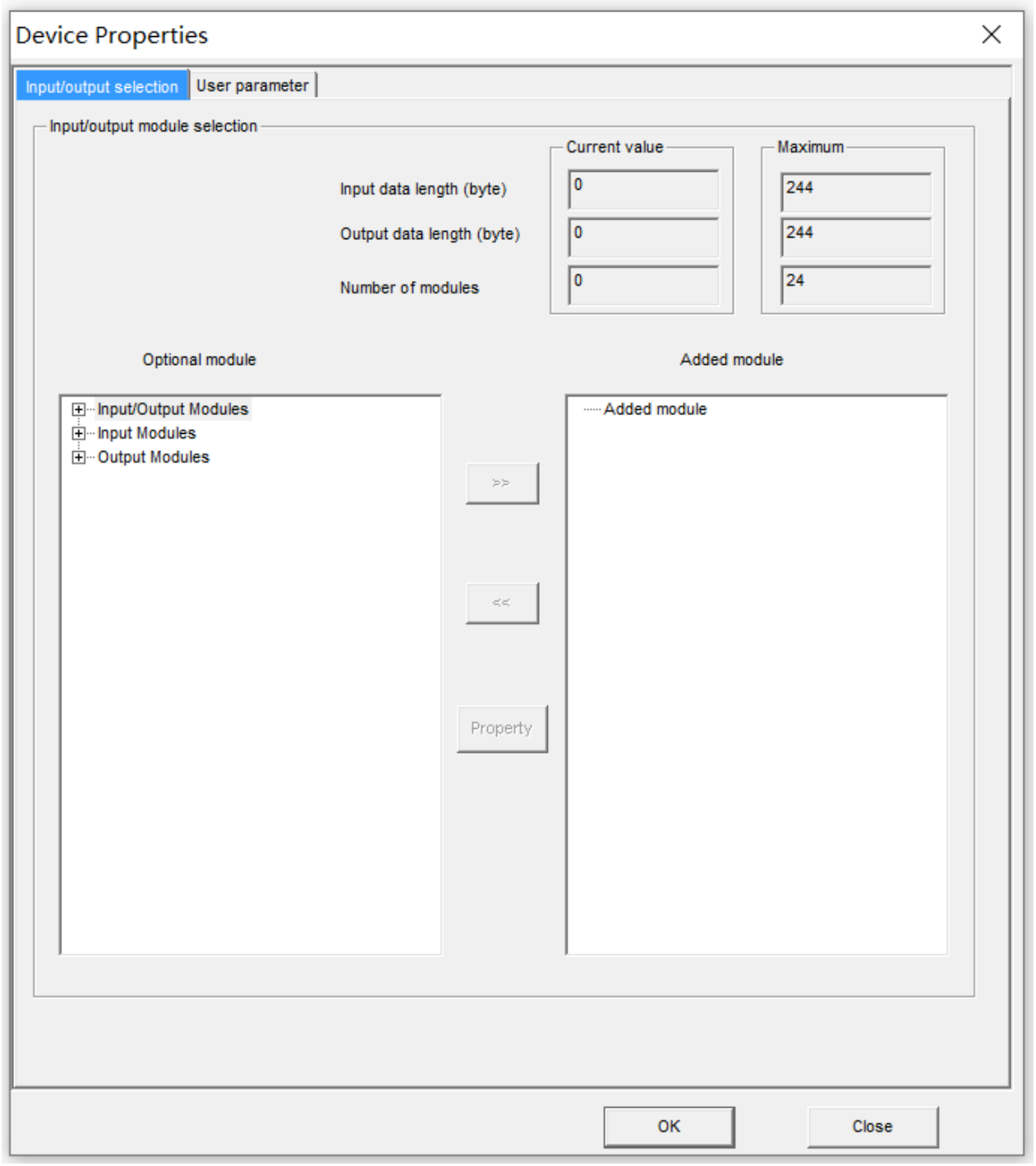
Device Configuration | Channel | Information

Project	Content
Module Type	MD810-gateway
Alias	MD810_gateway
Device Name	MD810-gateway
Description	MD810gw.GSD HW=0000 SW=V.1.03.03
GSD File Name	MD810gw.GSD
Device Address	2(Double-click Configuration)
Redundancy	NO
Input Starting Byte Offset(%IB)	Not configured
Output Starting Byte Offset(%QB)	Not configured
Device Property	Double-click Configuration
Vendor Name	INOVANCE
Version	1.06
HW Version	0000
SW Version	V.1.03.03
Device Type	9@INOVANCE/PROFIBUS@DP

The Profibus-DP slave supports up to 32 slave stations (no relay).

Can configure Alias, Device Address, Device Properties.

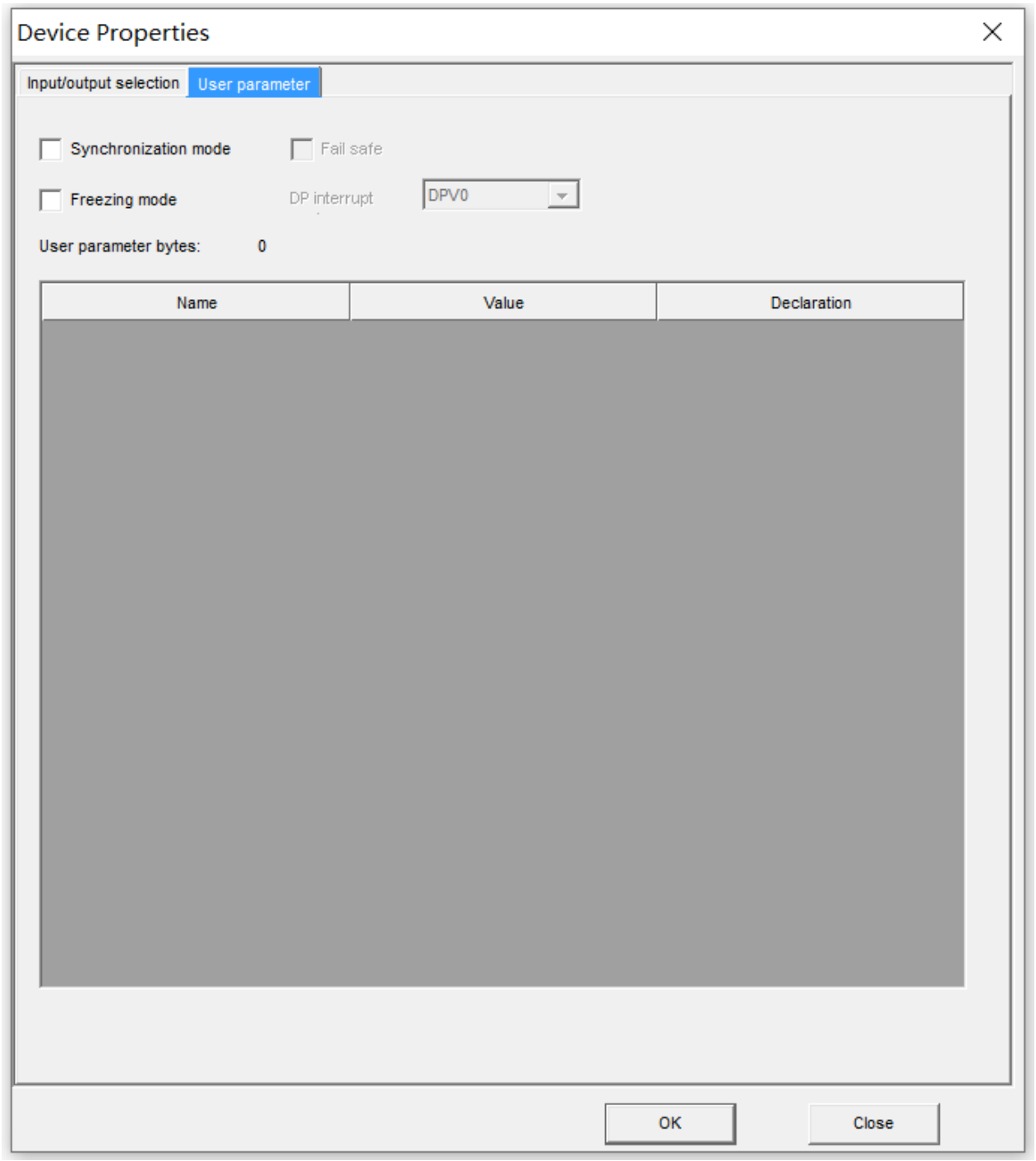
- Device Address: Range 2-125.
- Device Properties: input data configuration, output data configuration, synchronous mode, or frozen mode.



Device Properties-Input/Output selection

- Input/Output data length (Bytes): The number of bytes occupied by the added input/output modules, input modules, and output modules.
- Number of modules: The total number of added modules.

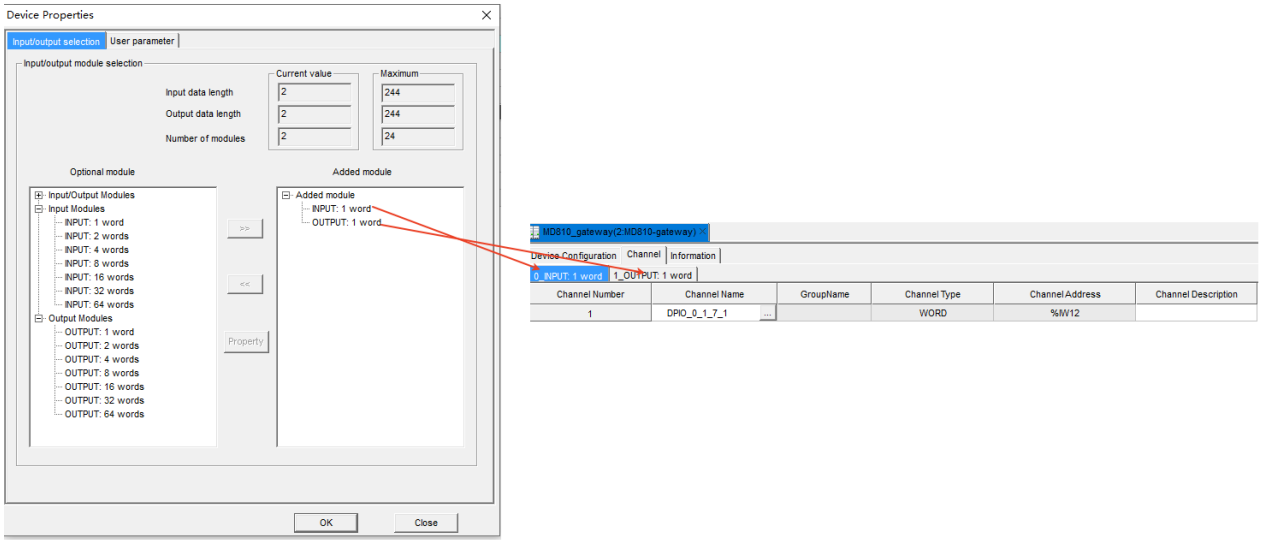
- Input data, output data, parameter configuration, or diagnostic data — each type can have a maximum of 244 byte per slave.
- The total combined input data across all Slaves must not exceed 1.2 Kbyte.
- The total combined output data across all Slaves must not exceed 1.2 Kbyte.
- The total number of input/output variables must not exceed 1,000 (excluding the module's internal 3-period data).



Device Properties-User parameter

- Synchronization Mode: Not recommended to check.
- Freezing Mode: Not recommended to check.

(3) The modules added in the "Device Properties" will be mapped to the "Channels" tab, as shown in the figure below.


Device Properties
Input/output selection
User parameter
Input/output module selection
Current value
Maximum
Input data length
Output data length
Number of modules
2
2
2
244
244
24
Optional module
Added module
Input/Output Modules
Input Modules
INPUT: 1 word
INPUT: 2 words
INPUT: 4 words
INPUT: 8 words
INPUT: 16 words
INPUT: 32 words
INPUT: 64 words
Output Modules
OUTPUT: 1 word
OUTPUT: 2 words
OUTPUT: 4 words
OUTPUT: 8 words
OUTPUT: 16 words
OUTPUT: 32 words
OUTPUT: 64 words
>>
<<
Property
Added module
INPUT: 1 word
OUTPUT: 1 word
OK
Close
MD810_gateway(2:MD810-gateway)
Device Configuration
Channel
Information
0_INPUT: 1 word
1_OUTPUT: 1 word
Channel Number
Channel Name
GroupName
Channel Type
Channel Address
Channel Description
1
DPIO_0_1_7_1
WORD
%IW12

Chapter 8 EtherNet/IP Network Communication

8.1 Network Overview

8.1.1 Introduction

The LX series controllers can communicate with third-party EtherNet/IP master stations as EtherNet/IP slave stations. Each controller can be configured with up to one EtherNetIP_Adapter. A maximum of 15 modules can be added under each EtherNetIP_Adapter.

8.1.2 Network Characteristics

- EtherNet/IP uses the physical layer network of Ethernet and is built on the communication protocol of TCP/IP or UDP. The application layer protocol is based on the Common Industrial Protocol (CIP).
- In a control system, a device that uses the Ethernet/IP protocol and has an independent IP address is counted as an EtherNet/IP node.
- EtherNet/IP adopts the Producer/Consumer communication mode. It allows all nodes on the network to access the same data from one data source at the same time, thus optimizing data transmission.
- It uses standard Ethernet communication media, applicable category 5, 5e or above shielded twisted-pair cables, and interface type RJ45.
- For the transmission of IO data, the UDP protocol and port 2222 are used.
- It supports standard Ethernet switches.

8.2 Network Topology EtherNet/IP

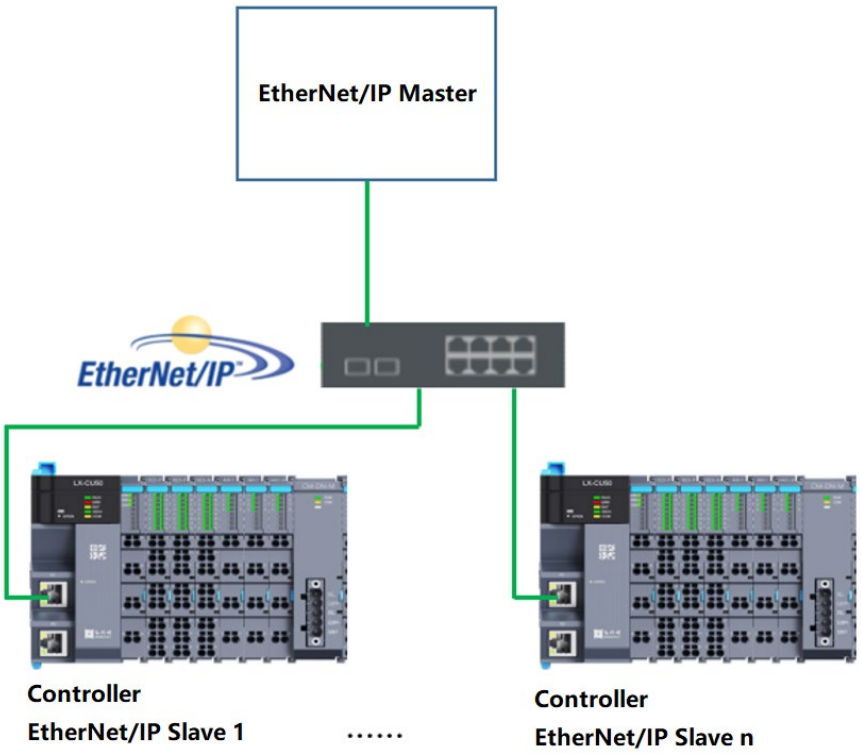
Since EtherNet/IP is based on the standard TCP/IP protocol, it supports a variety of topologies in terms of network topology, such as the most common star and ring topologies.

Requirements for communication cables:

Cable Model	Specifications	Communication Distance
Cat5e shielded twisted-Pair cable	T568B crossover cable	100 m

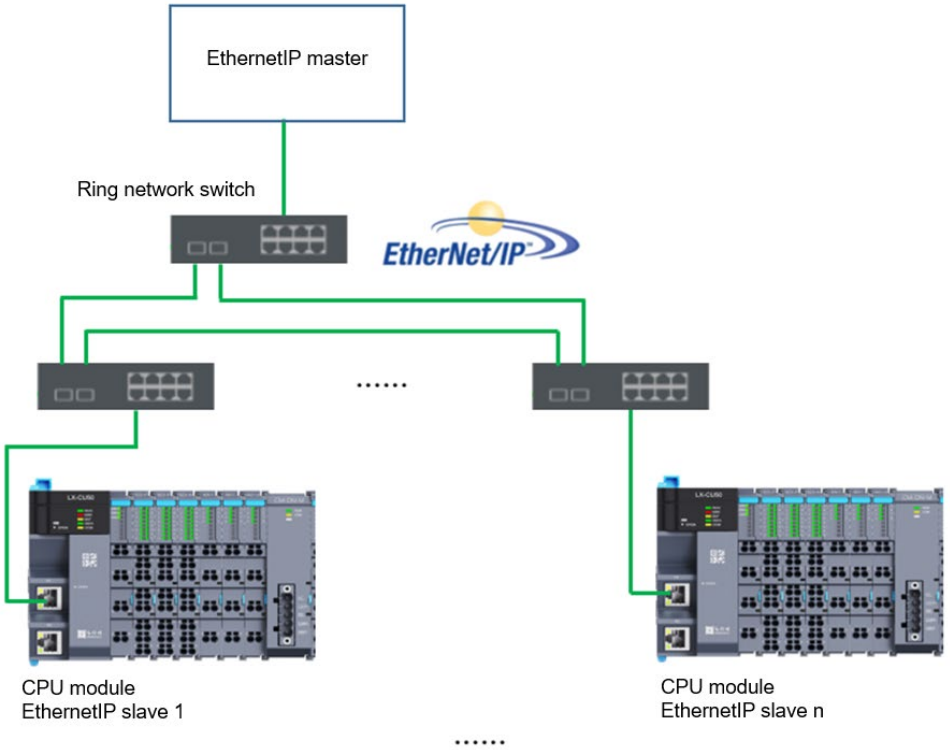
8.2.1 Star Network Topology

The Ethernet/IP star network connection topology is the most common connection method. In this case, all devices are connected to industrial-grade switches to ensure the security, stability and reliability of the system. When a problem occurs at one site, it will not affect the communication of other sites. The number of network nodes is determined by the characteristics of the Ethernet/IP master station. When the number of nodes is large, the network can be expanded through switch cascading. The star network topology is shown in the figure below:



8.2.2 Ring Network Topology

A ring network can be formed through ring network switches, and it allows a breakpoint. The number of nodes is determined by the characteristics of the Ethernet/IP master station. The ring network topology diagram is as follows:

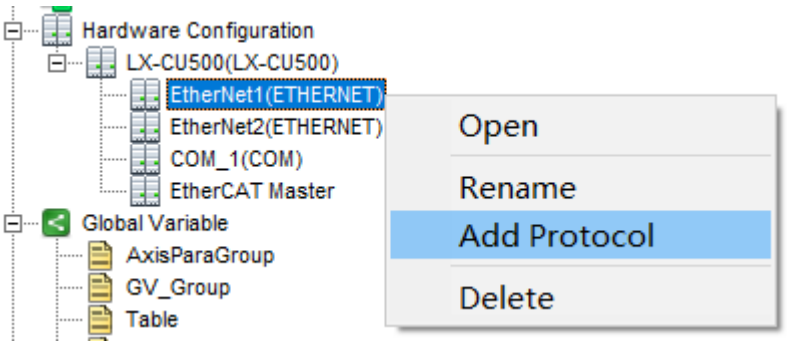


8.3 Configuration EtherNet/IP

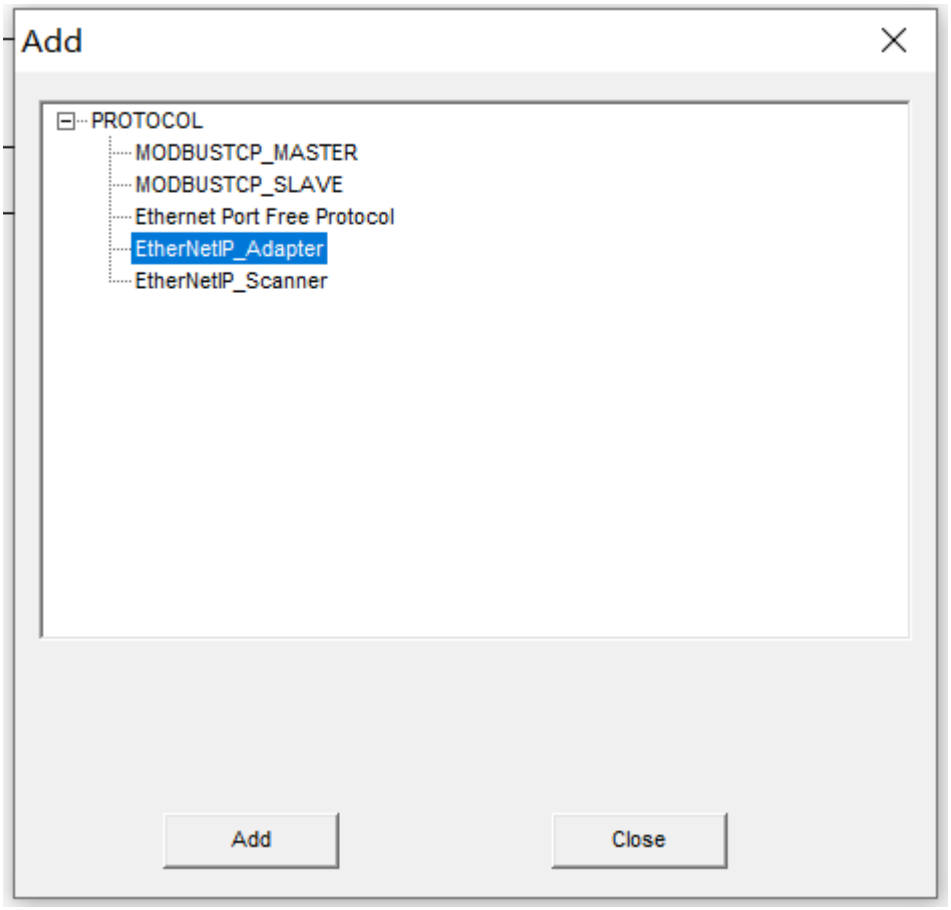
8.3.1 Configuring EtherNetIP Slave Station

To configure an EtherNetIP slave station, please follow these steps:

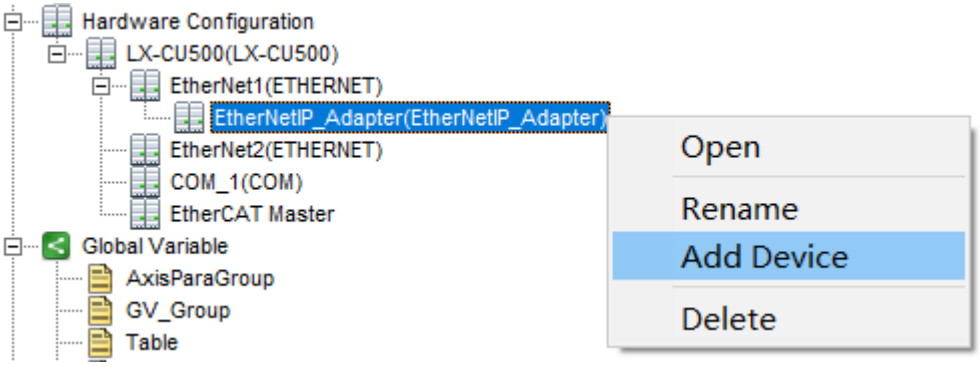
(1) In the right-click menu of the EtherNet tree node, select the Add Protocol command.



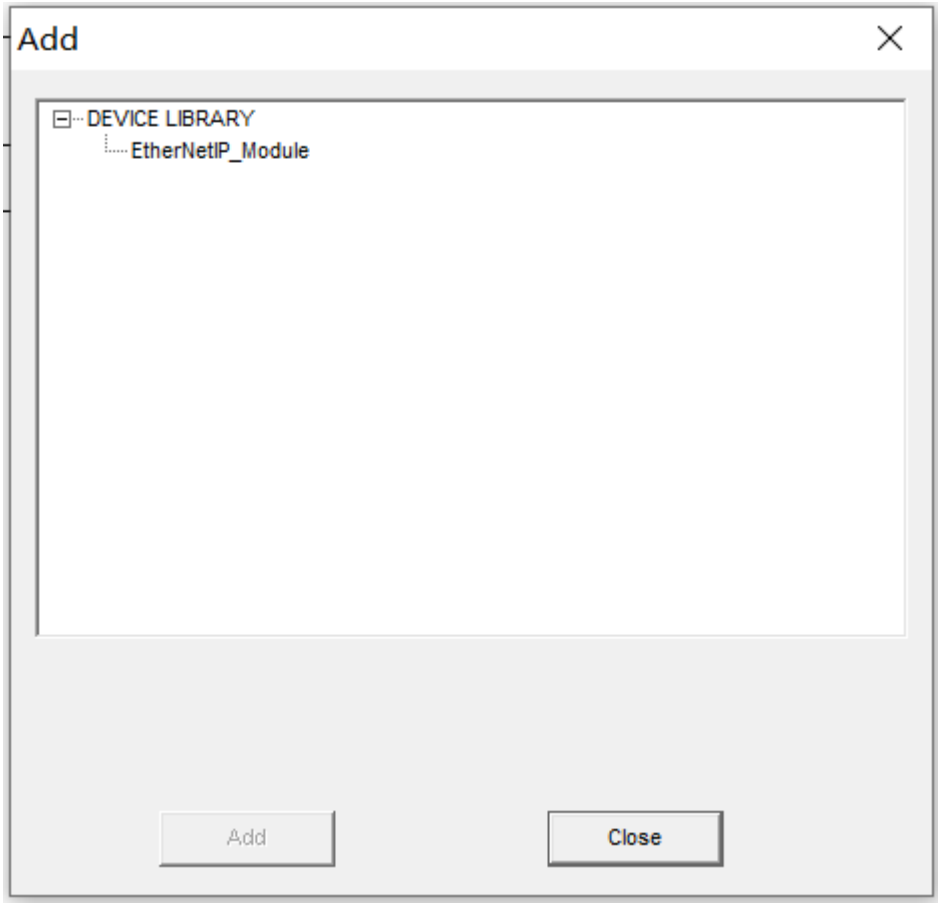
(2) When the Add dialog box is displayed, select the EtherNetIP_Adapter protocol, and click the Add button.



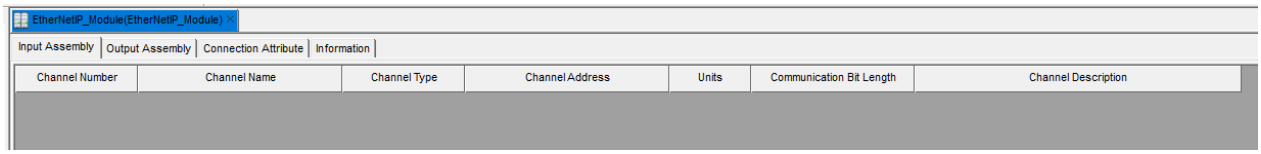
(3) Right-click the newly added EtherNetIP_Adapter tree node and select the Add Device command.



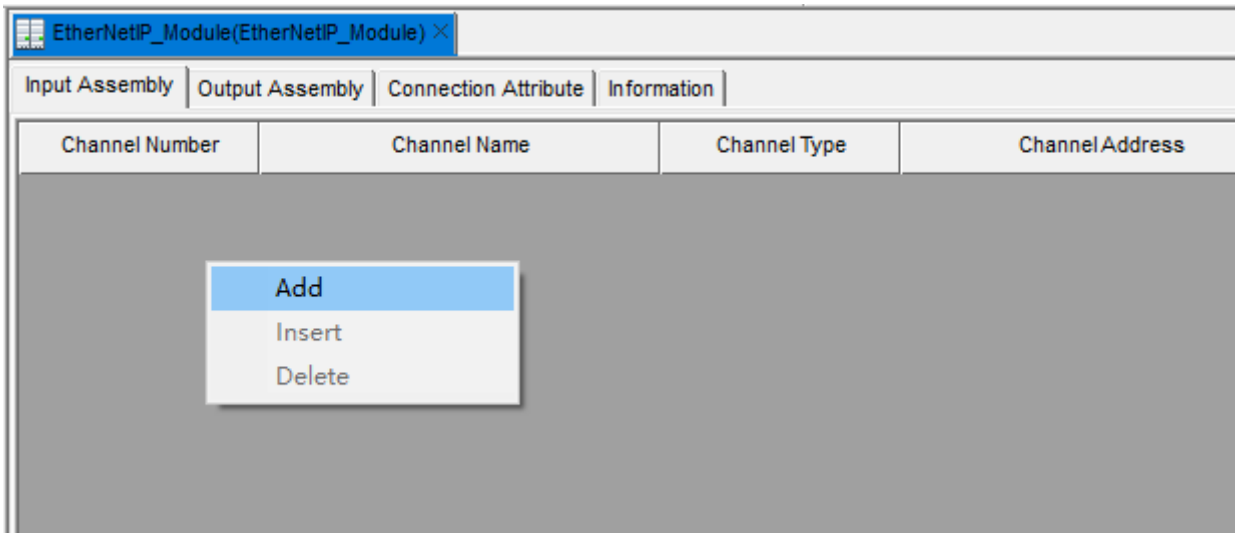
(4) When the Add page is displayed, select EtherNetIP_Module and click the Add button.



(5) Double-click the newly added EtherNetIP_Module tree node, and the module configuration interface is displayed.

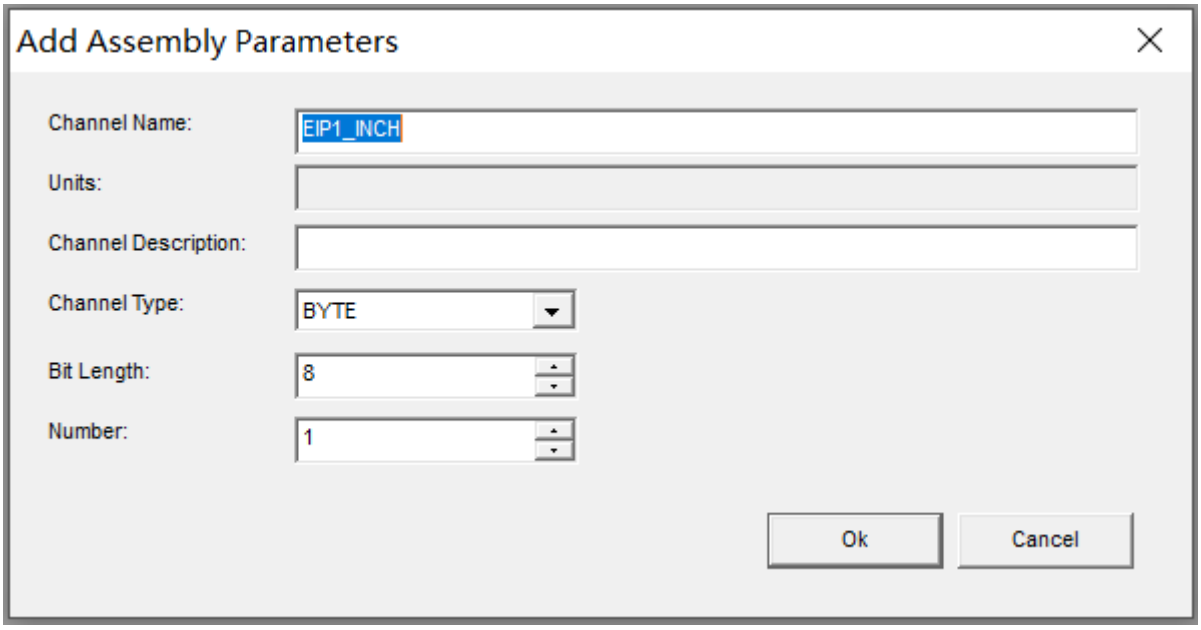


(6) Right-click the blank area and select Add from the right-click menu.



(7) The Add Assembly Parameters dialog box is displayed. The description of the parameters is as follows:

- Channel Name: When the number of channels is greater than 1, this name is the data name of the first channel, and then the channel name automatically adds 1 as a suffix.
- Data Type: Such basic data types as BYTE, BOOL, WORD, DWORD, USINT, UINT, UDINT, SINT, INT, DINT, REAL, and LREAL are supported.
- Bit Length: The type length is automatically matched according to the selected data type. The user can also manually configure the bit length. It is recommended that the configured bit length be consistent with that corresponding to the channel type.
- Number: Each EtherNetIP_Module supports adding up to 502 bytes of area I variables and 502 bytes of area Q variables.



(8) Click the OK button to complete the addition.

8.3.2 Exporting EDS File

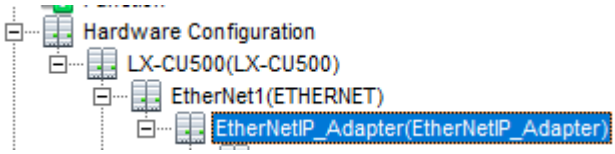
8.3.2.1 Introduction

When the controller communicates with the master station as an EtherNet/IP slave station, the user needs to import the EDS configuration file of the slave station into the master station and parse it into EtherNet/IP slave station communication data in the master station. The EDS file contains the configuration information of all modules, such as input and output data, device information, and connection information. The user needs to generate it manually after completing the EtherNetIP slave station configuration and passing the compilation.

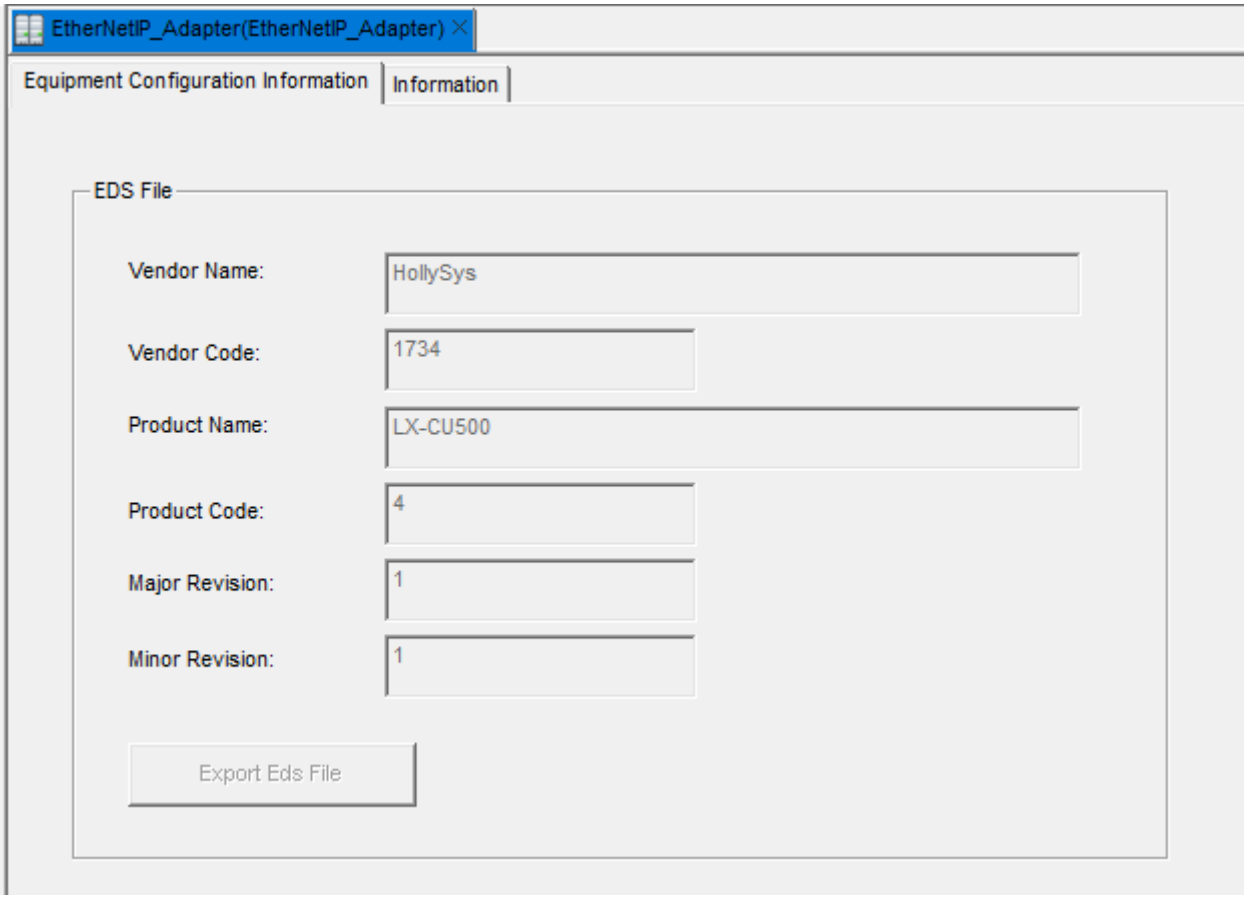
8.3.2.2 Steps

To export an EDS file, please follow the steps below:

(1) Double-click the EtherNetIP_Adapter tree node under the controller's Ethernet port,



The device configuration information page is displayed.



(2) Click the Export EDS File button, and the Save As dialog box is displayed.

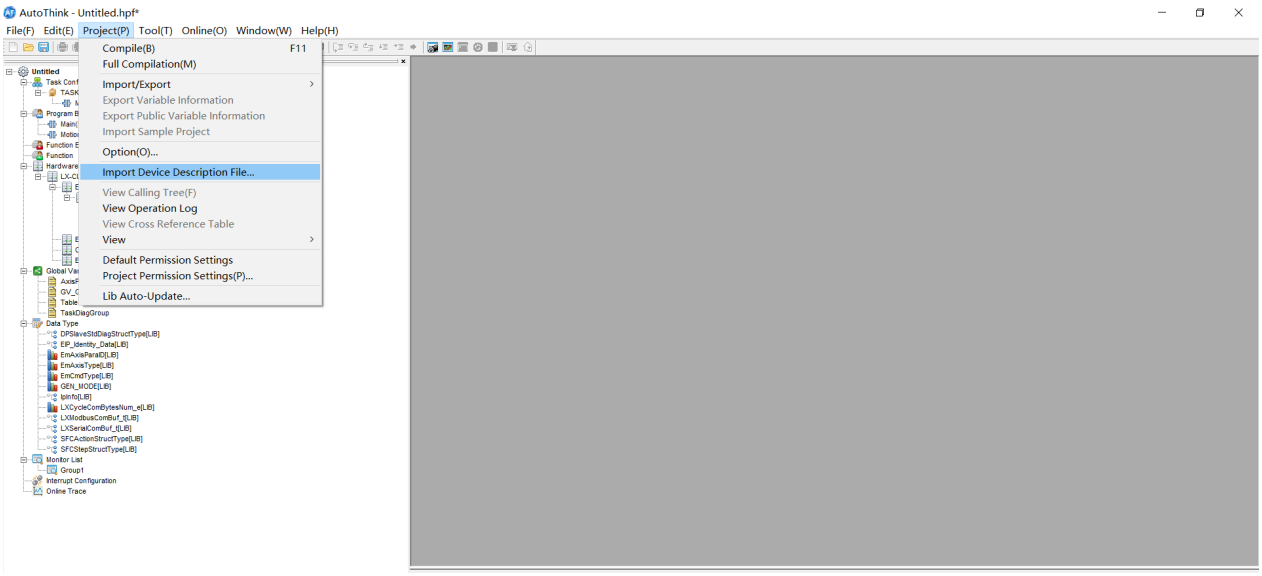
(3) Select a path to save the EDS file, and a .eds file is generated under the corresponding path, with the default file name as HollySys LX-CU500 EtherNet_IP EDS.eds, which can be modified as needed.

8.3.3 Configuring EtherNetIP Master Station

To configure an EtherNetIP master station, please follow these steps:

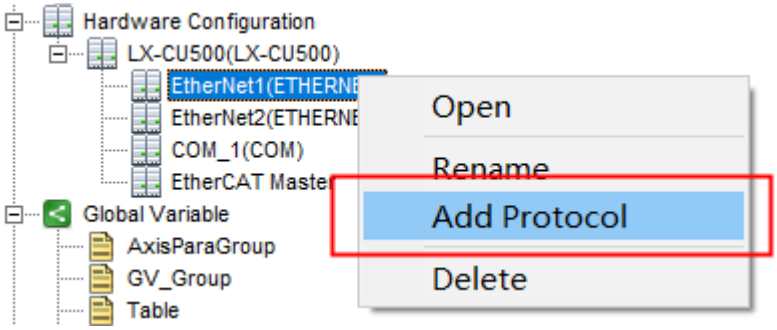
(1) Importing a device description file

Select Project - Import Device Description File

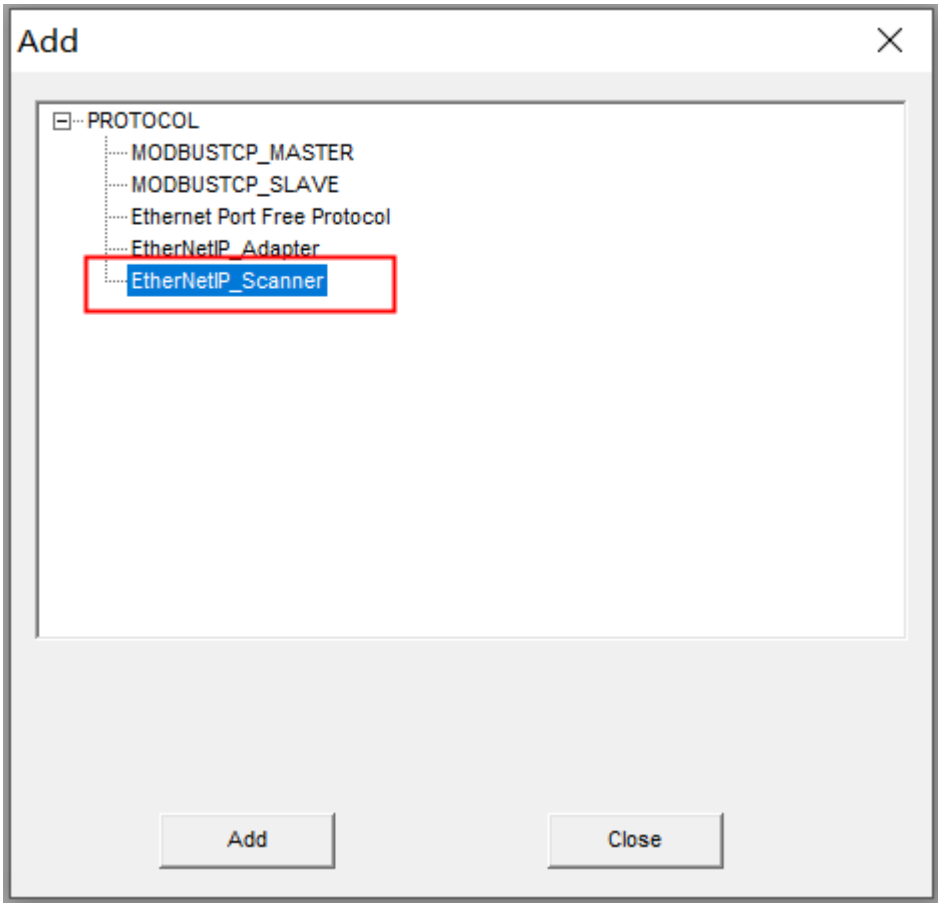


(2) Hardware configuration

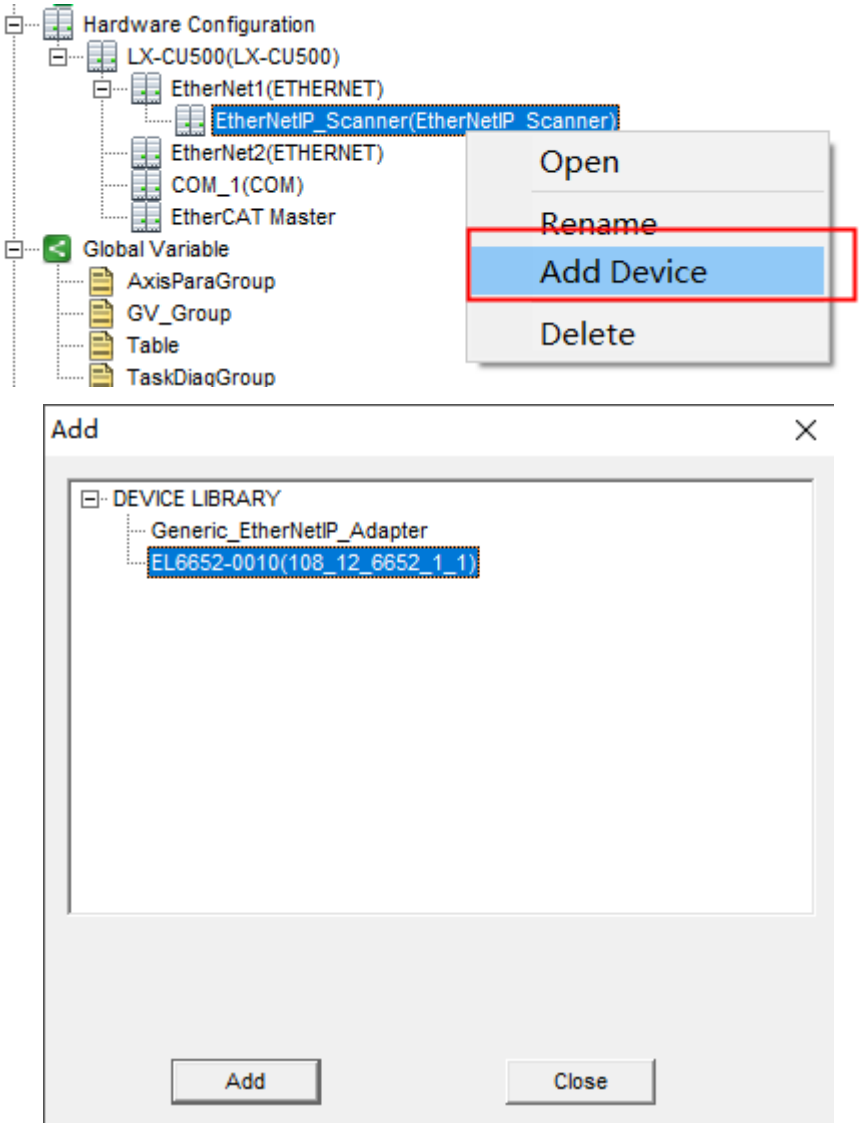
(a) Right-click EtherNet1 - Add Protocol



(b) Select EtherNetIP Scanner

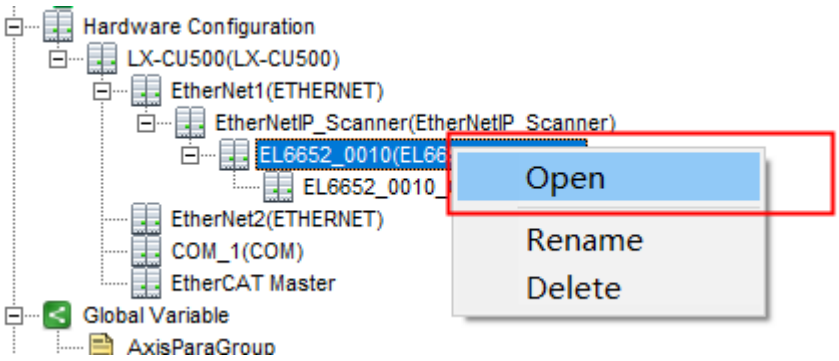


(c) Right-click EtherNetIP Scanner - Add Device

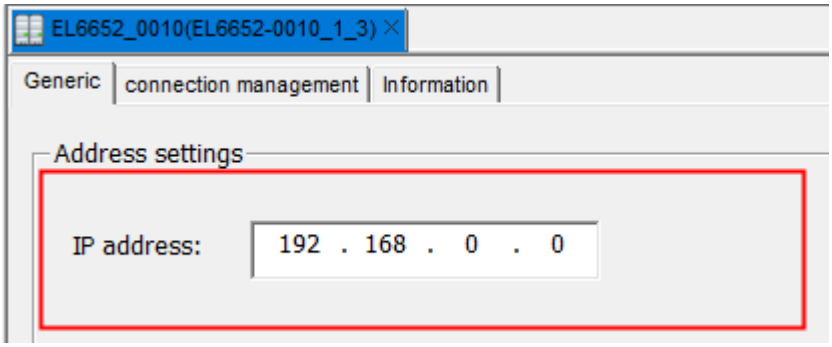


(3) Setting parameters

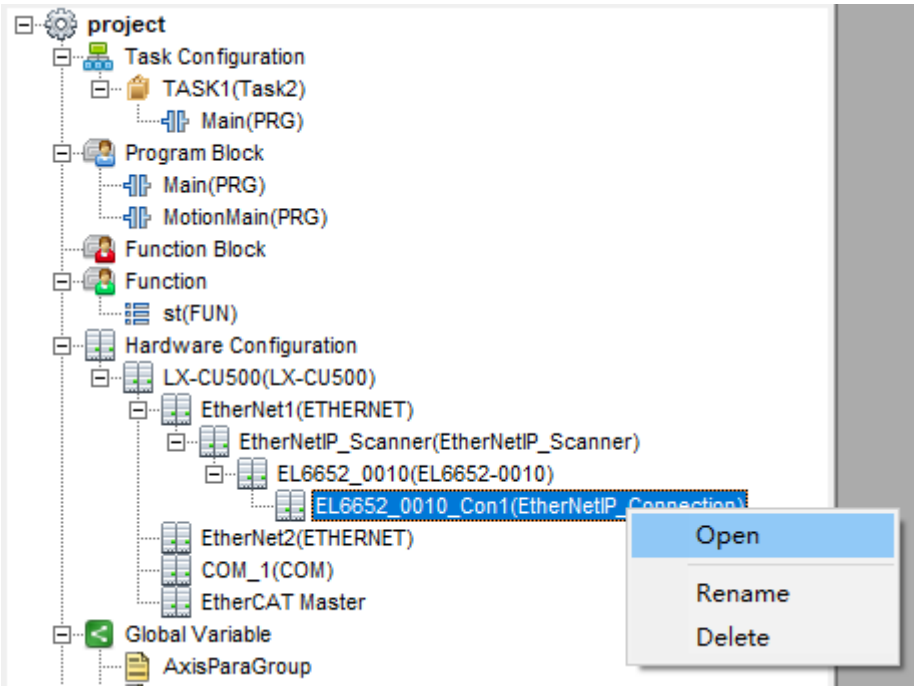
(a) Right-click EL6652_0010 - Open



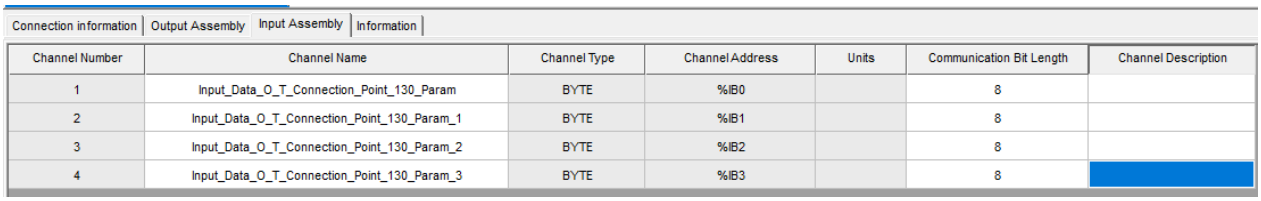
(b) Configure the slave station IP address



(c) Right-click EthernetIP_Connection – Open



(d) The input/output data can be viewed in the input/output assembly. It is necessary to ensure that the length of input/output data is consistent with that of the slave station configuration data. If it is inconsistent, it can be changed manually.


Connection information | Output Assembly | Input Assembly | Information

Channel Number	Channel Name	Channel Type	Channel Address	Units	Communication Bit Length	Channel Description
1	Input_Data_O_T_Connection_Point_130_Param	BYTE	%IB0		8	
2	Input_Data_O_T_Connection_Point_130_Param_1	BYTE	%IB1		8	
3	Input_Data_O_T_Connection_Point_130_Param_2	BYTE	%IB2		8	
4	Input_Data_O_T_Connection_Point_130_Param_3	BYTE	%IB3		8	

8.4 EtherNet/IP Communication Example

8.4.1 Example Function

This example uses the EIP master station communication module Codesys software PLC to implement

data communication with the EIP slave station communication module LX-CU500.

8.4.2 System Composition

- EIP master station: communication module Codesys software PLC.
- EIP slave station: communication module LX-CU500.

8.4.3 Example Topology

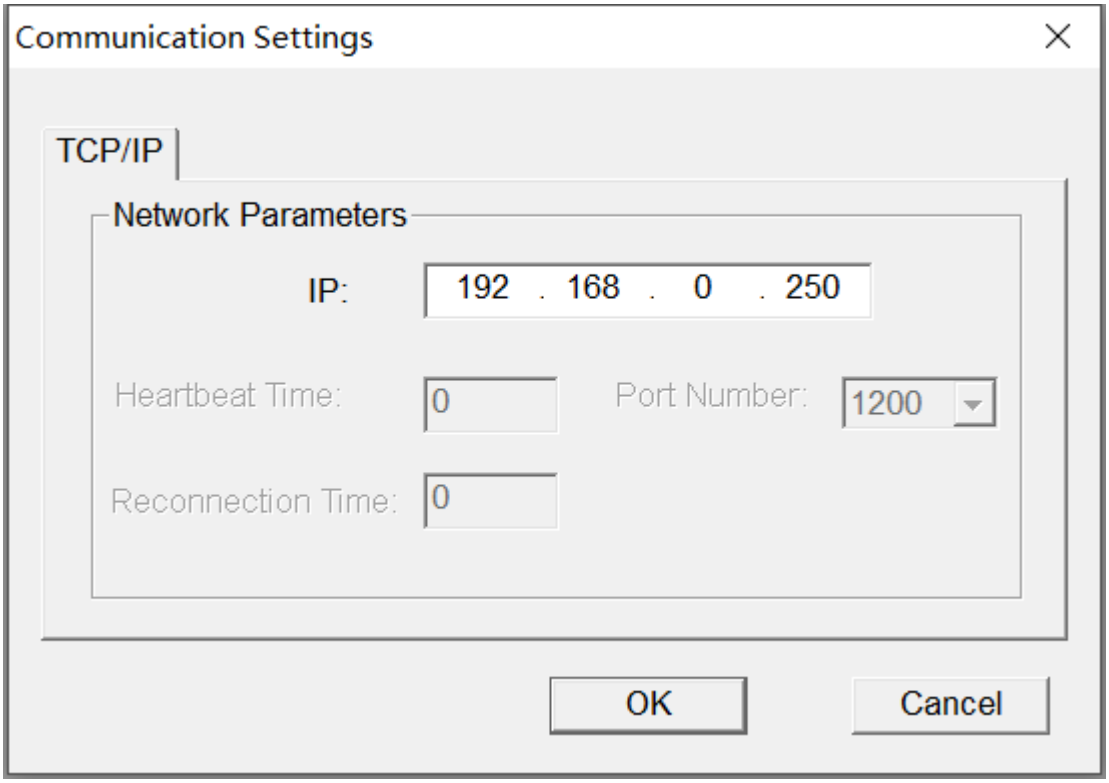
8.4.4 Programming

Create two projects, where the master station project uses the Codesys software, and the slave station project uses the AutoThink software.

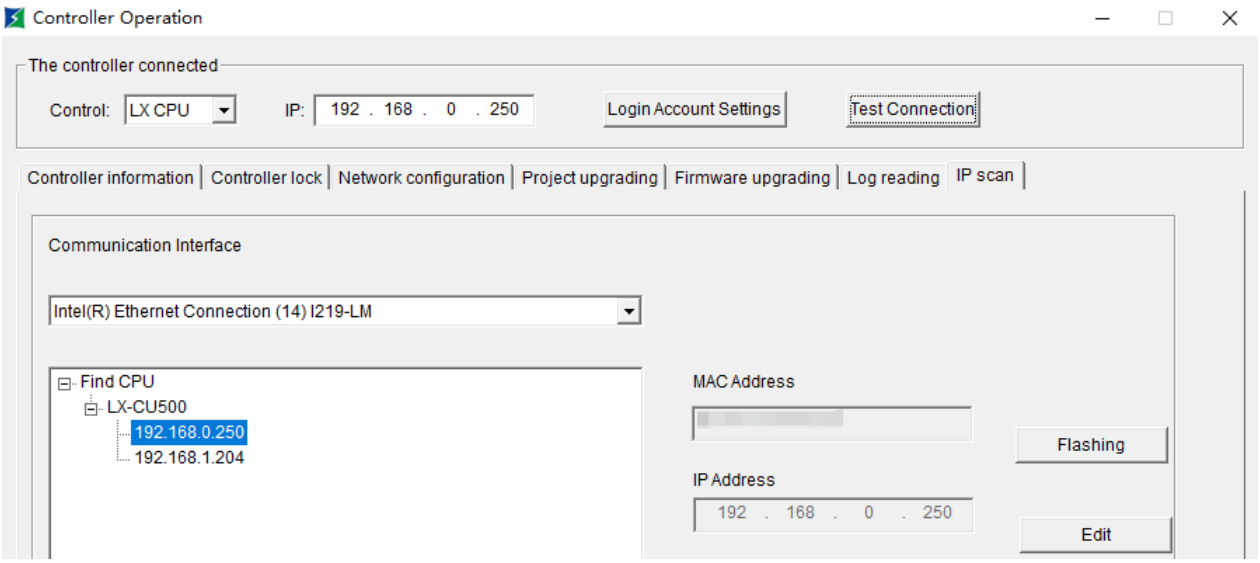
8.4.4.1 Slave station configuration

The specific configuration in the slave station program in this example is as follows:

(1) Ethernet communication IP settings: click AT - Online - Communication Settings.



(2) Click Tools - Auxiliary Tools - Controller Operation, and confirm again that the IP is correct.



(3) The Input Assembly and Output Assembly of the EIP slave station are as follows:

EtherNetIP_Module(EtherNetIP_Module)

Input Assembly | Output Assembly | Connection Attribute | Information

Channel Number	Channel Name	GroupName	Channel Type	Channel Address	Units	Communication Bit Length	Channel Description
1	EIP1_INCH		BYTE	%IB100		8	
2	EIP1_INCH_1		BYTE	%IB101		8	

EtherNetIP_Module(EtherNetIP_Module)

Input Assembly | Output Assembly | Connection Attribute | Information

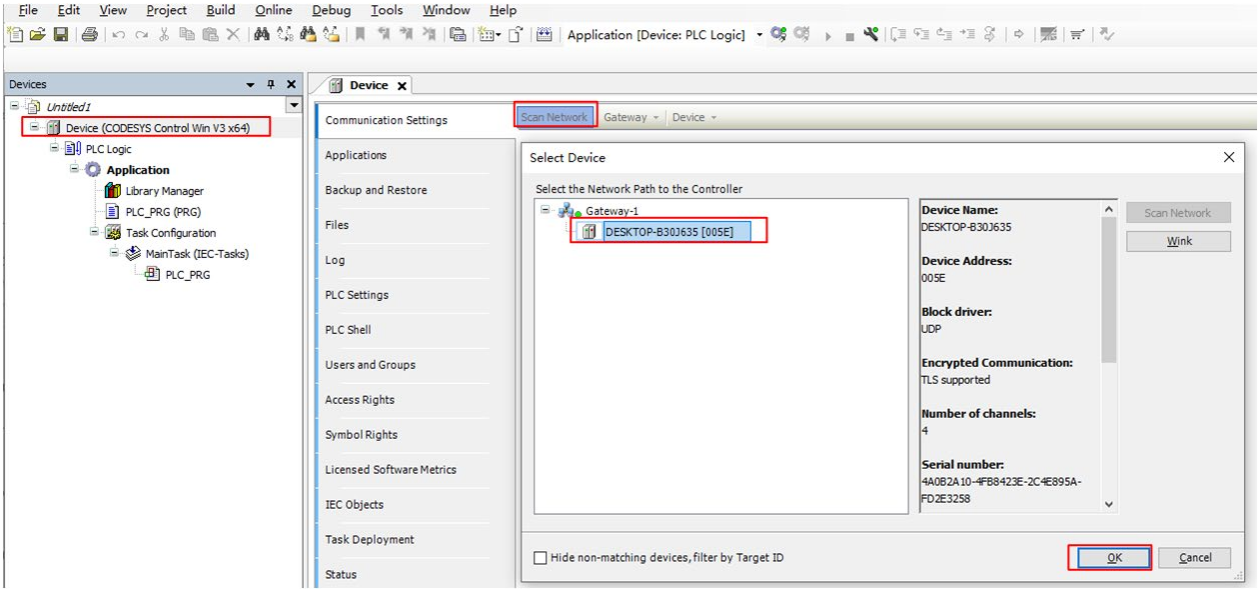
Channel Number	Channel Name	GroupName	Channel Type	Channel Address	Units	Communication Bit Length	Channel Description
1	EIP1_OUTCH		BYTE	%QB100		8	
2	EIP1_OUTCH_1		BYTE	%QB101		8	

For the communication program configuration of the EIP slave station, refer to Configuring EtherNetIP Slave Station.

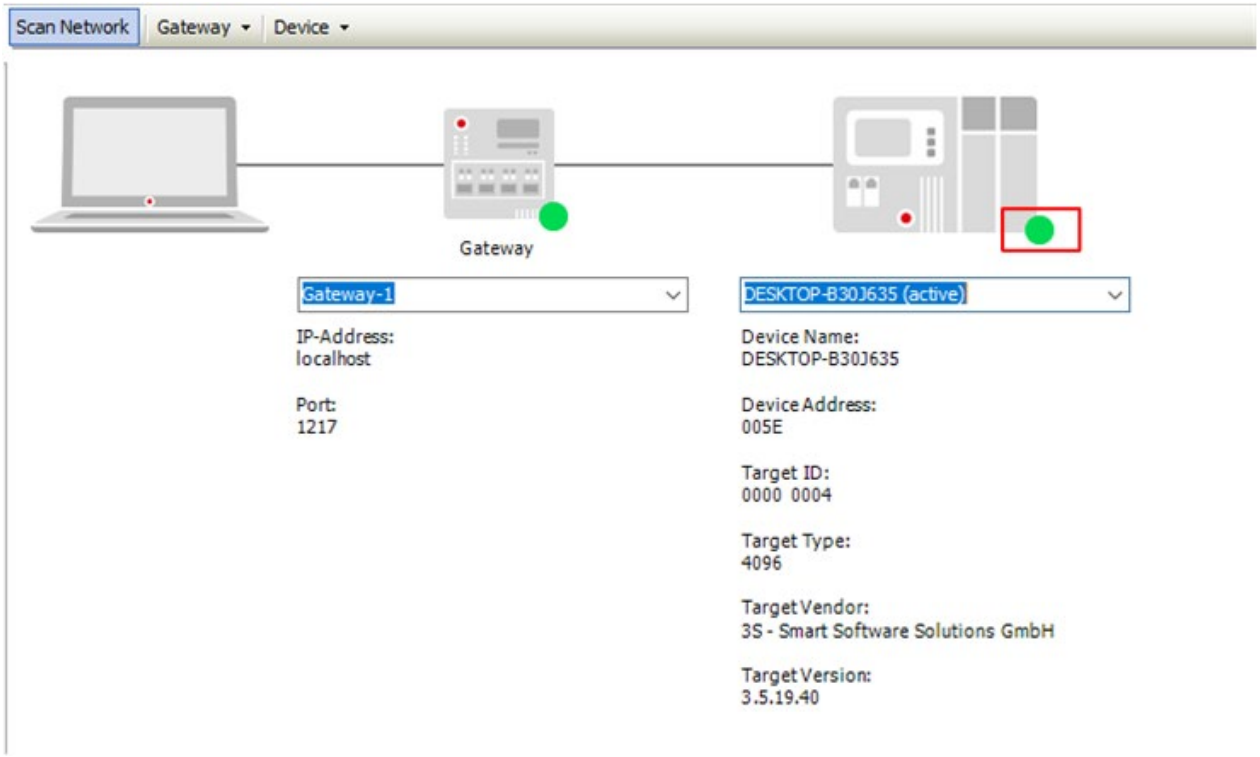
8.4.4.2 IP Settings of Master Station

(1) If the master station is the Codesys software PLC, click start PLC first.

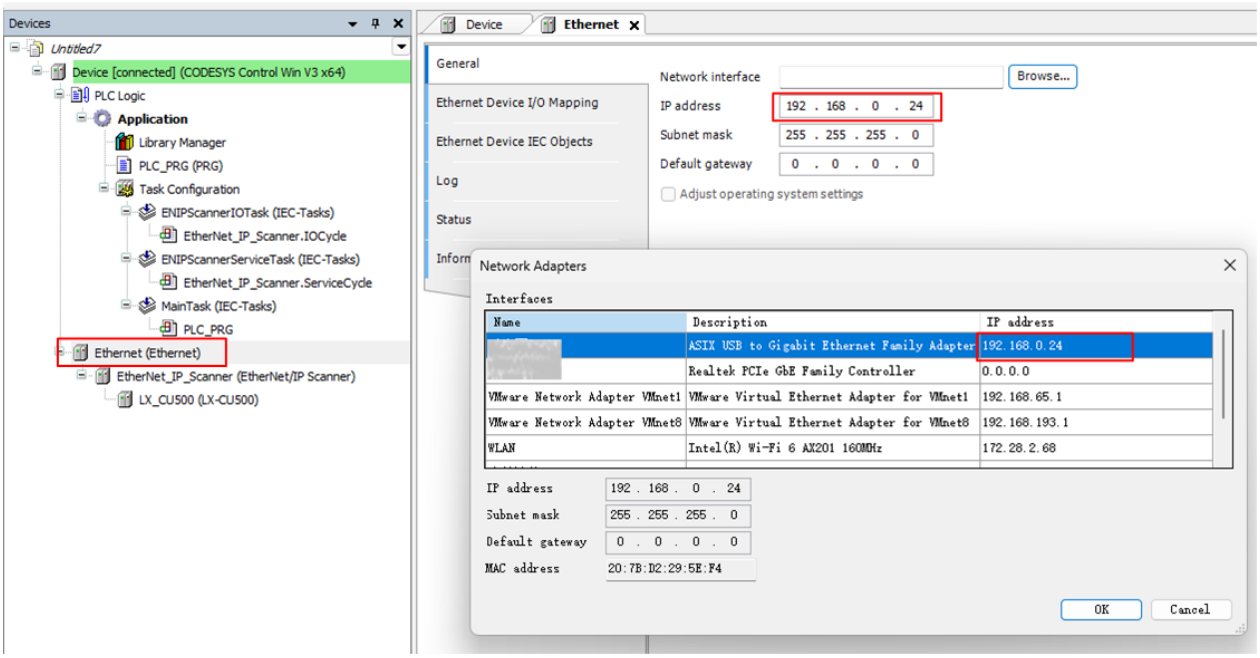
(2) After start, click Device - Scan Network to scan the master station device.



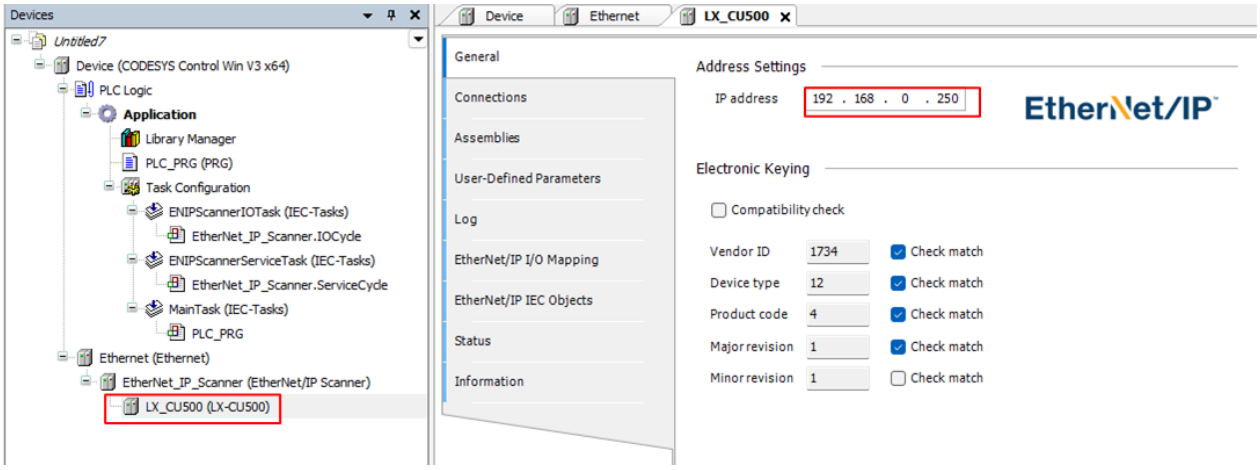
In the figure below, green means the scan is successful.



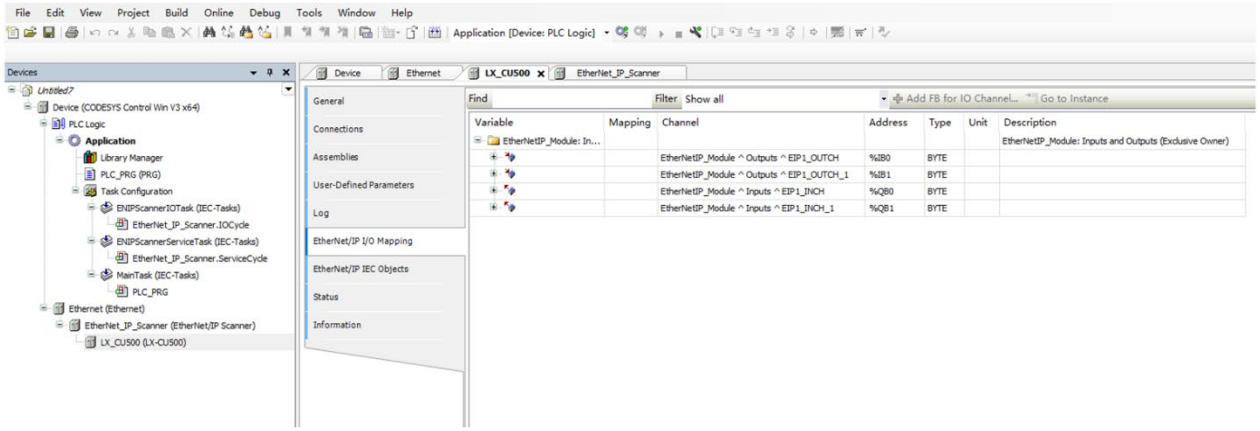
(3) Click Ethernet - General - Browse, and select the master station IP (the Ethernet IP for master-slave communication of your own computer).



(4) Click LX-CU500 – General - IP Address, and set the slave station IP.



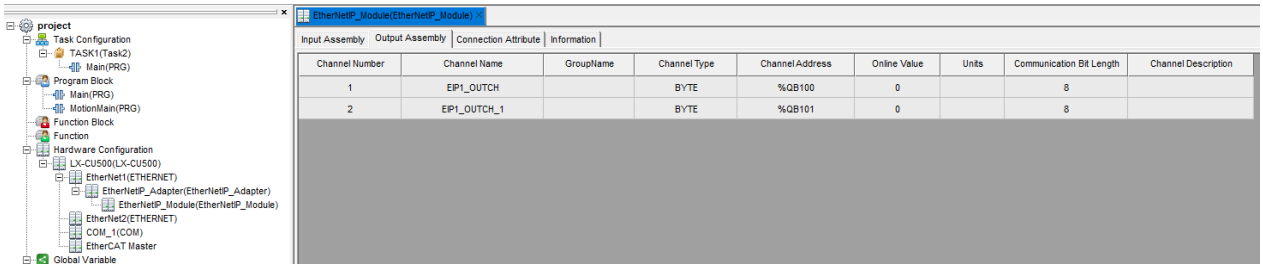
(5) Click LX-CU500 - EtherNet/IPI/O Mapping, and set the periodic task.



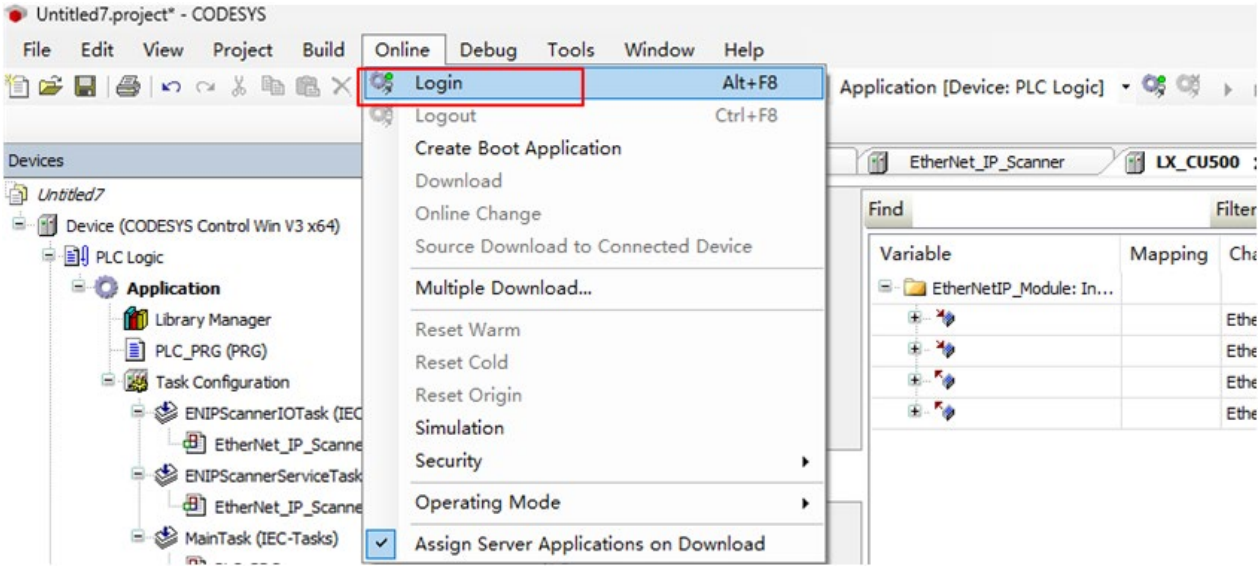
8.4.5 Online Debugging

After the configuration of AT and Codesys is completed, install the program to the master and slave station PLCs respectively, and conduct debugging in monitor mode.

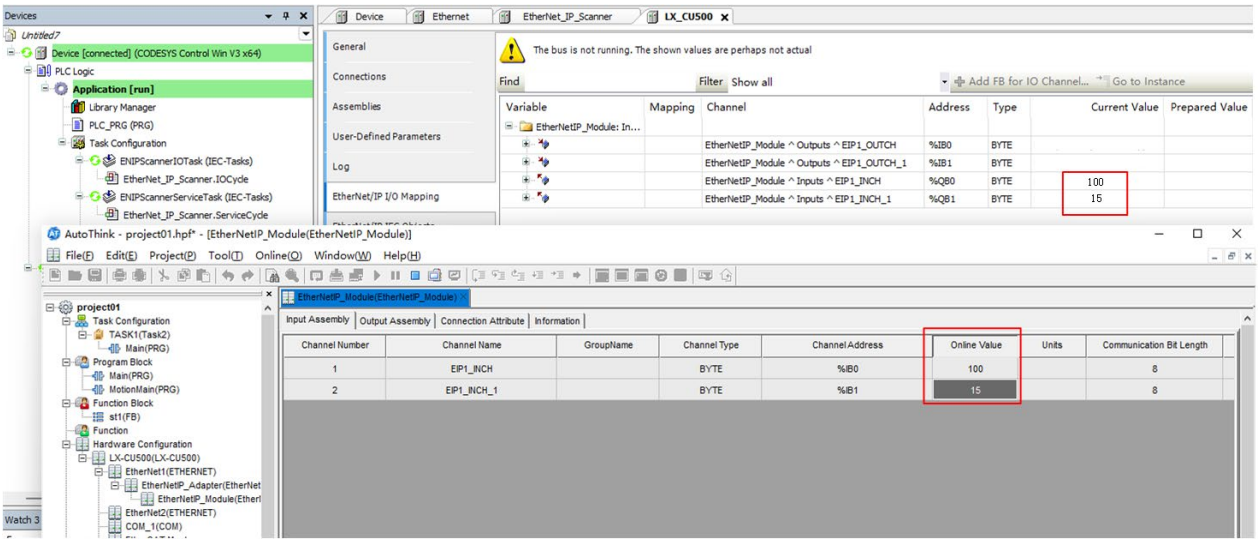
(1) Double-click EtherNetIP_Module on the slave station to view communication data. The specific page is as follows:

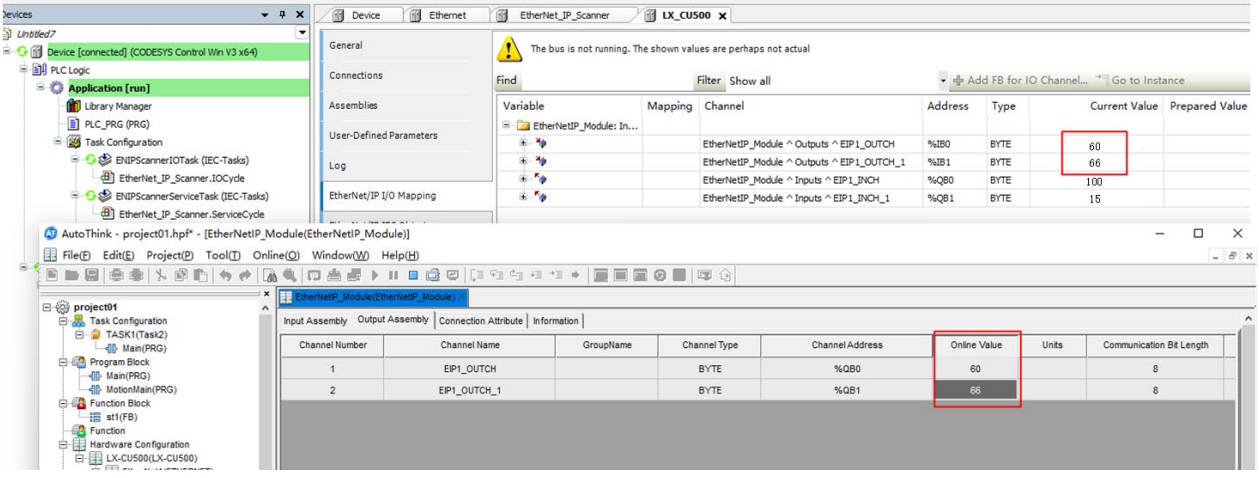


(2) Enter the monitoring module from Online – Login on the master station, and click on the page as follows:



(3) Double-click the output variable in area Q of the master station or slave station, enter the data, and view the input variable in area I of the slave station or master station. If the data transmission is successful, the corresponding display is consistent.





8.5 Fault Diagnosis

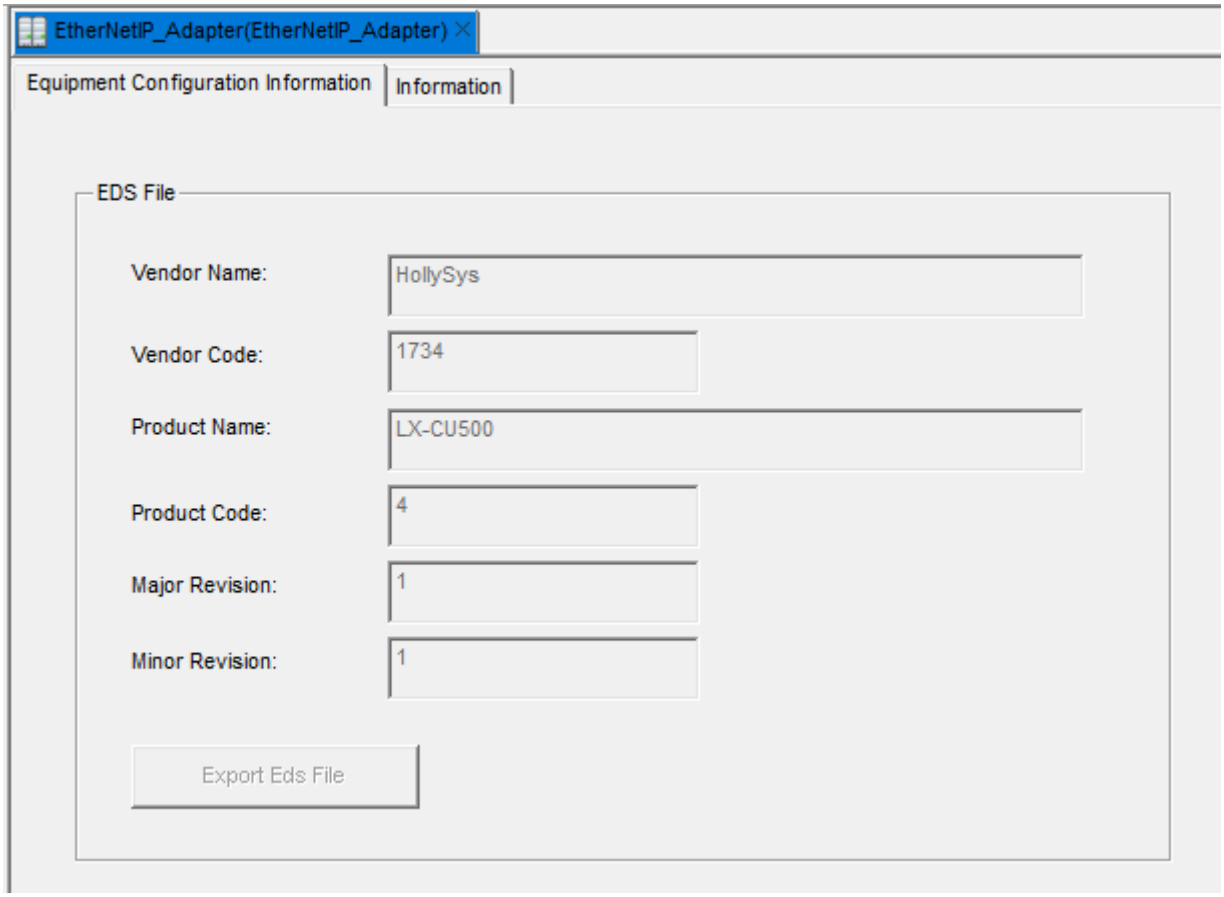
8.5.1 Slave Station Diagnosis

The diagnostic data of the EtherNet/IP slave station is reported and displayed through AutoThink. The diagnostic data includes device status diagnosis, slave station diagnosis and module diagnosis. The user needs to configure the input and output data in the slave station module. After the compilation is passed, the diagnostic variables of the module will be generated in the Diagnostic Information tab. Each module generates a corresponding diagnostic variable.

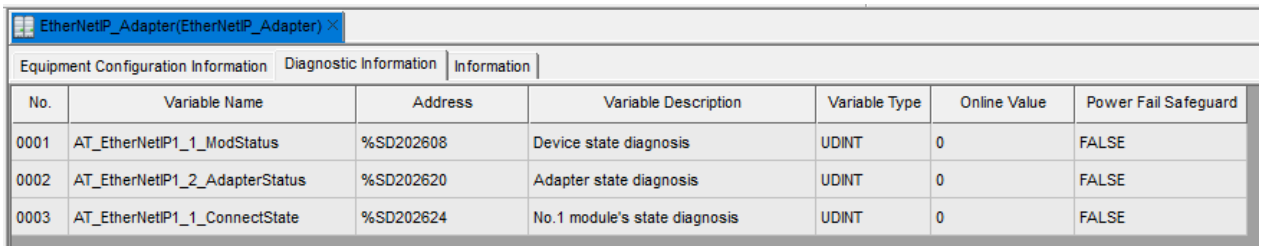
8.5.1.1 Steps

To view the EtherNet/IP slave station diagnostic information, please follow these steps:

(1) Double-click the EtherNetIP_Adapter tree node under the controller's Ethernet port, and the device configuration information page is displayed.



(2) Click the Diagnostic Information tab to view the diagnostic information status.


EtherNetIP_Adapter(EtherNetIP_Adapter)

Equipment Configuration Information | Diagnostic Information | Information

No.	Variable Name	Address	Variable Description	Variable Type	Online Value	Power Fail Safeguard
0001	AT_EtherNetIP1_1_ModStatus	%SD202608	Device state diagnosis	UDINT	0	FALSE
0002	AT_EtherNetIP1_2_AdapterStatus	%SD202620	Adapter state diagnosis	UDINT	0	FALSE
0003	AT_EtherNetIP1_1_ConnectState	%SD202624	No.1 module's state diagnosis	UDINT	0	FALSE

Diagnostic Variable	Diagnostic value	Description of Values	Processing Method
ModStatus (device status diagnosis)	1	No error	-
	0	Initial state	-
	4	Internal fault	■ In the AT engineering menu, select "Full Compilation" to re-download the user project and check whether the fault still exists. ■ Power cycle the controller and check whether the fault still exists. ■ If the above measures are ineffective, contact the equipment manufacturer.
	5	Internal fault	
AdapterStatus (slave station status diagnosis)	0	Initial state	-
	32	Connection timeout	Check the module with diagnostic status 4 under this slave station, then follow the module connection timeout troubleshooting procedure.
	48	Not connected	■ Check the device IP address: On the scanner side, verify whether the IP address of the target device is consistent with the IP address configured on the controller. ■ Network connection: Check whether the network connection between the scanner and the controller is normal. Use the ping command to test connectivity. ■ Check whether the scanner and controller are in normal working condition. ■ Check the configuration of switches or routers in the topology. Replace with an unmanaged switch and try again. ■ Use a network scanning tool to check whether the TCP/UDP port (default: 44818) of the controller is open. ■ Power cycle the controller and check whether the connection is successful. ■ In the AT engineering menu, select "Full Compilation" to re-download the user project, and check whether the device connects successfully. ■ Reduce the device scale on the scanner side. Remove other functioning adapter devices and retain only the target device for testing. ■ Check the scanner-side configuration: connection path, tag name, communication byte size, connection type, RPI, and other parameters. Refer to the scanner device troubleshooting manual for details. ■ If the above measures are ineffective, contact the equipment manufacturer.
	80	The slave device has a non-recoverable error	■ Power cycle the controller and check whether the fault still exists. ■ If the above measures are ineffective, contact the equipment manufacturer.
	96	Correct diagnostic code: At least one I/O	No action required. The slave connection is normal and the Scanner is in RUN state.

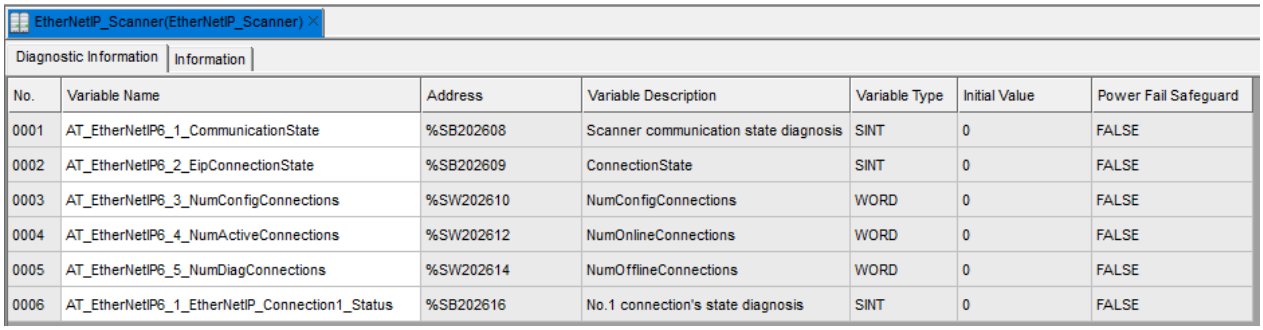
		connection is in Run mode	
	112	The master is in IDLE state	Check whether the master device is in IDLE state. When in this state, both sides transmit packets normally, but data is not updated. After switching to RUN state, data updates normally.
ConnectState (status diagnosis of the nth module) (n≤15)	0	Not connected, or the connection has been closed	■ Check the device status of the adapter. If the device shows a fault, troubleshoot according to device fault procedures. ■ Check the slave status of the adapter. If it shows that no connection has been established, troubleshoot according to slave fault procedures. ■ If both the adapter device status and slave status are normal, perform troubleshooting according to the following measures: □ Adjust the RPI or timeout settings of the current module. Increase the RPI time or the timeout multiplier (default *4) to a larger value. □ On the scanner side, check whether the RPI settings for the three connection types—Exclusive Owner, Listen Only, and Input Only—of the current module are consistent. □ Check whether the scanner-side O->T data size, target configuration size, proxy configuration size, and T->O data size are consistent with the configuration of the current module. □ Check whether the variable type and size of the output assembly on the scanner side are consistent with those in the input assembly of the current module. □ Check whether the variable type and size of the input assembly on the scanner side are consistent with those in the output assembly of the current module. □ Check whether the connection path between the scanner side and the current module is consistent. □ Check the conversion format: For the O->T direction, select 32-bit / Idle for the Exclusive Owner connection type, and select Heartbeat for other transmission types; for the T->O direction, select Pure Data. □ Check the connection type: For the O->T direction, point-to-point must be selected. For a Listen Only connection, multicast must be selected, and the corresponding module must contain a multicast Owner or Input Only connection.
	3	No error	-
	4	Connection timeout (watchdog / message timeout)	■ Check the device status of the adapter. If the device shows a fault, troubleshoot according to device fault procedures. ■ Check the slave status of the adapter. If it shows that no connection has been established, troubleshoot according to slave fault procedures. ■ If both the adapter device status and slave status are normal, perform troubleshooting according to the following measures: □ Adjust the RPI or timeout settings of the current module. Increase the RPI time or the timeout multiplier (default *4) to a larger value. □ Reduce the communication load on the adapter device, retaining only the current module for testing.

8.5.2 Master Station Diagnosis

The diagnostic data of the EtherNet/IP master station is reported and displayed through AutoThink. After the compilation is passed, the diagnostic variables of the module will be generated in the Diagnostic Information tab. Each module generates a corresponding diagnostic variable. To view the EtherNet/IP master station diagnostic information, please follow these steps:

(1) Double-click the EtherNetIP_Scanner tree node under the controller's Ethernet port.

(2) Click the Diagnostic Information tab to view the diagnostic information status


EtherNetIP_Scanner(EtherNetIP_Scanner)

Diagnostic Information | Information

No.	Variable Name	Address	Variable Description	Variable Type	Initial Value	Power Fail Safeguard
0001	AT_EtherNetIP6_1_CommunicationState	%SB202608	Scanner communication state diagnosis	SINT	0	FALSE
0002	AT_EtherNetIP6_2_EipConnectionState	%SB202609	ConnectionState	SINT	0	FALSE
0003	AT_EtherNetIP6_3_NumConfigConnections	%SW202610	NumConfigConnections	WORD	0	FALSE
0004	AT_EtherNetIP6_4_NumActiveConnections	%SW202612	NumOnlineConnections	WORD	0	FALSE
0005	AT_EtherNetIP6_5_NumDiagConnections	%SW202614	NumOfflineConnections	WORD	0	FALSE
0006	AT_EtherNetIP6_1_EtherNetIP_Connection1_Status	%SB202616	No.1 connection's state diagnosis	SINT	0	FALSE

Diagnostic Variable	Description of Values	Processing Method
CommunicationState (Scanner communication state diagnosis)	0: Init or unknown 1: Not config connection 2: Stop mode 3: Working mode	0: Please check the configuration status 1: Please config the connection 2: Please check the operating status of the master station 3: -
EipConnectionState (Connection State)	0: Unknown 1: All slave station are connected 2: Any one slave station is not connected	0: Contact Technical Support 1: - 2: check the network
NumConfigSlaves	Number of configured slave station	-
NumActiveSlaves	Number of online slave station	-
NumDiagSlaves	Number of offline slave station	check the network
EtherNetIP_Connectionn_Status (No. n Connection's state diagnosis)	1: Connection established and running normally	-
	0: Initial state	-
	-1: Adapter device IP address is invalid	Check IP input. In the device tree on the right side of the AT software, click on the Adapter device under Scanner to enter the "General" page, and check whether the IP address in the address settings is an invalid IP such as 0.0.0.0.
	-3: Controller system memory is insufficient	■ In the device tree on the right side of the AT software, delete any communication protocols or slave devices that have been configured but are not currently in use. ■ Contact the supplier to replace the controller model.

	-4: Connection parameter error, adapter device rejected the connection	■ For predefined connections: Re-export the EDS file of the adapter device and import it into the configuration project on the scanner side. After re-adding the device, compile and download. ■ For generic connections: Confirm the following parameters for each module: ☐ Check whether the RPI settings for the three connection types—Exclusive Owner, Listen Only, and Input Only—are consistent. ☐ Check whether the scanner-side O->T data size, target configuration size, proxy configuration size, and T->O data size are consistent with the configuration on the adapter side. ☐ Check whether the variable type and size of the output assembly on the scanner side are consistent with those in the input assembly on the adapter side. ☐ Check whether the variable type and size of the input assembly on the scanner side are consistent with those in the output assembly on the adapter side. ☐ If path connection is selected, check whether the connection path between the scanner side and the adapter is consistent. ☐ Check the conversion format of the connection on the scanner side: For the O->T direction, select 32-bit/Idle for the Exclusive Owner connection type, and select Heartbeat for other transmission types; for the T->O direction, select Pure Data. ☐ Check the connection type on the scanner side: For the O->T direction, point-to-point must be selected. For a Listen Only connection, multicast must be selected, and the corresponding module must contain a multicast Owner or Input Only connection.
	-7: Adapter device connection timeout	■ Check the device IP address: Click on the Adapter under Scanner, enter the "General" page, and verify whether the IP address matches the actual IP address of the physical device. ■ Network connection: Check whether the network connection between the controller and the Adapter device is normal. Use the ping command to test connectivity. ■ Check whether the controller module and the Adapter device are in normal working condition. ■ Check the configuration of switches or routers in the topology. Replace with an unmanaged switch and try again. ■ Use a network scanning tool to check whether the TCP/UDP port (default: 44818) of the Adapter device is open. ■ Power cycle the controller and check whether the Adapter device comes back online. ■ Test by modifying the connection types for T->O and O->T (test point-to-point and multicast separately). ■ Adjust the RPI or timeout settings of the Adapter device. Increase the RPI time or the timeout multiplier (default *4) to a larger value. ■ In the AT engineering menu, select "Full Compilation" to re-download the user project, and check whether the Adapter device comes back online. ■ Reduce the connection scale on the scanner side. Remove other functioning Adapter devices and retain

		only the current device for testing. ■ If the above measures are ineffective, contact the equipment manufacturer.
	-9: Configuration does not match the actually connected device	■ Check whether the configured Adapter device matches the actually connected Adapter device (vendor ID, device type, product code, etc.). ■ Uncheck the "Major Revision" and "Minor Revision" match check options, then re-download and check whether the fault still exists. ■ Contact the Adapter device manufacturer, obtain the EDS file corresponding to the connected device, re-import it, and download again.

Chapter 9 EtherNet/IP Tag Communication

9.1 Network Overview

9.1.1 Introduction

EtherNet/IP Tag Communication (also known as EtherNet/IP Tag Communication) is a tag-based data exchange method primarily designed for efficient communication between controller layers. Controllers can communicate directly using tag variable names without the need to associate actual memory addresses.

Supported data types for tag communication include: global variables, PRG local variables, structures, enumerations, ARRAY, BOOL, BYTE, DATE, DINT, DT, DWORD, INT, LREAL, POINTER, REAL, SINT, STRING, TIME, TOD, UDINT, UINT, USINT, WORD.

Class3 communication does not support structures, enumerations, complex type arrays, or nested arrays.

- EtherNet/IP Tag Communication is supported in LX-CU430_V1.1.1 and above versions, LX-CU433_V1.1.0 and above versions, LX-CU500_V1.4.0 and above versions, LX-CU501_V1.2.0 and above versions, LX-CU510_V1.1.0 and above versions, LX-CU511_V1.0.1 and above versions.

9.1.2 Network Characteristics

- A single controller supports one EtherNet/IP Tag Communication channel. Support communication between Class1 and Class3.
- A maximum of 64 tag groups, with each group limited to 1400 bytes. Up to 384 tags in total (sum of all tags across all groups).
- Tag group names and tag names with a maximum length of 255 bytes.
- A maximum of 32 connections. Support for Tag TYPE as the connection type.

9.1.3 Precautions

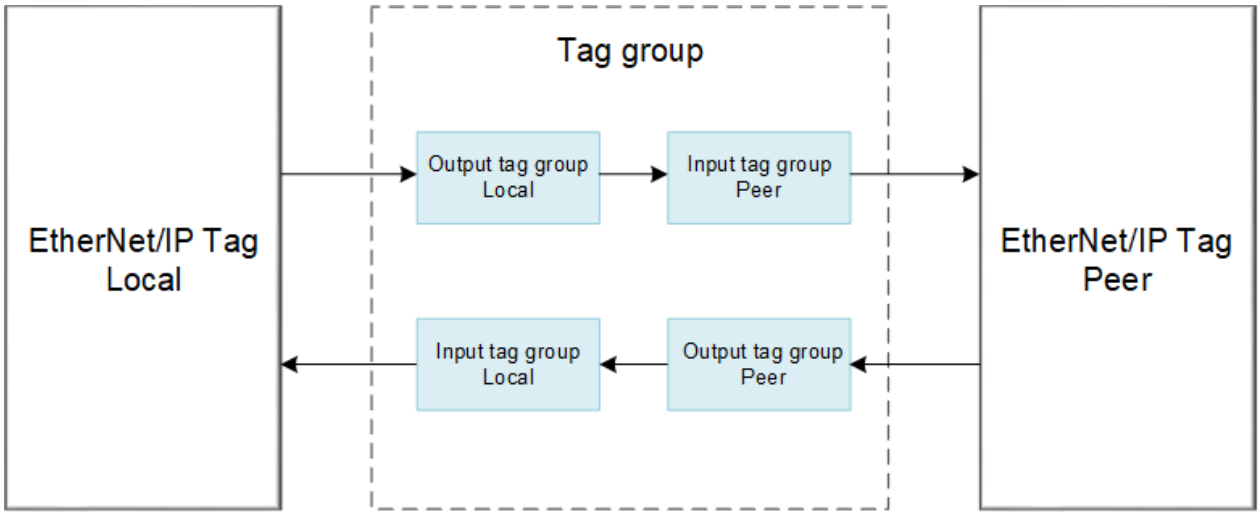
- When a variable's network exposure is set to "Public", it will be registered as an Input/Output (I/O) type in EtherNet/IP Tag Communication, variables of this type can be used for both output tags (writing) and input tags (reading).
- Do not configure two I/O-type tags to access each other, this may cause read/write conflicts, leading

to communication errors.

- Ensure that the memory addresses used for EtherNet/IP Tag Communication do not overlap with those used by other communication protocols. Address overlap may result in read/write conflicts and unpredictable behavior.
- Writing to an invalid (illegal) memory address via a pointer may cause controller exceptions or crashes. Use pointers with caution and ensure proper address validation.

9.2 Working Principle

The LX-CU430 sends output tag groups to the peer device and receives output tag groups from the peer, enabling data exchange between control layers.



Ensure the controller is properly connected before configuring EtherNet/IP Tag communication.

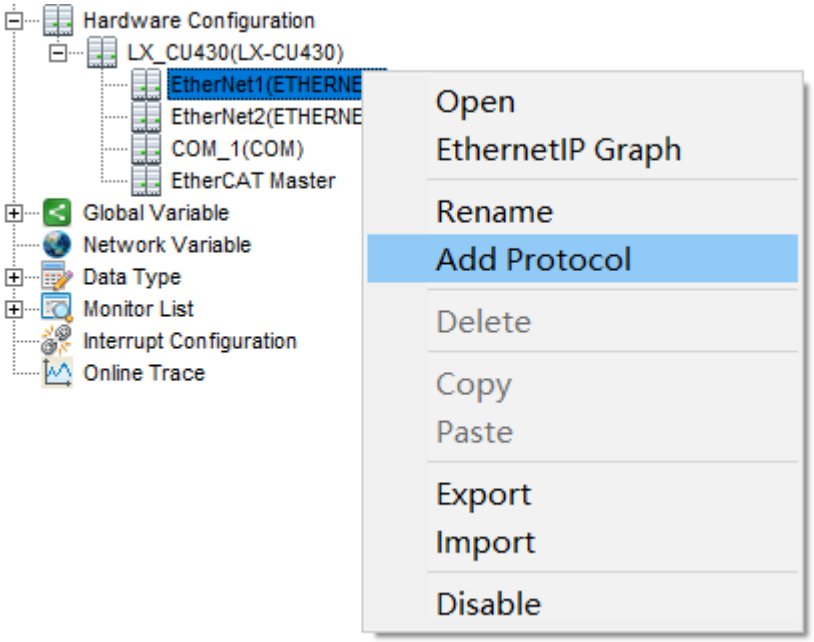
9.3 Configuration

Ensure the controller is properly connected before configuring EtherNet/IP Tag communication.

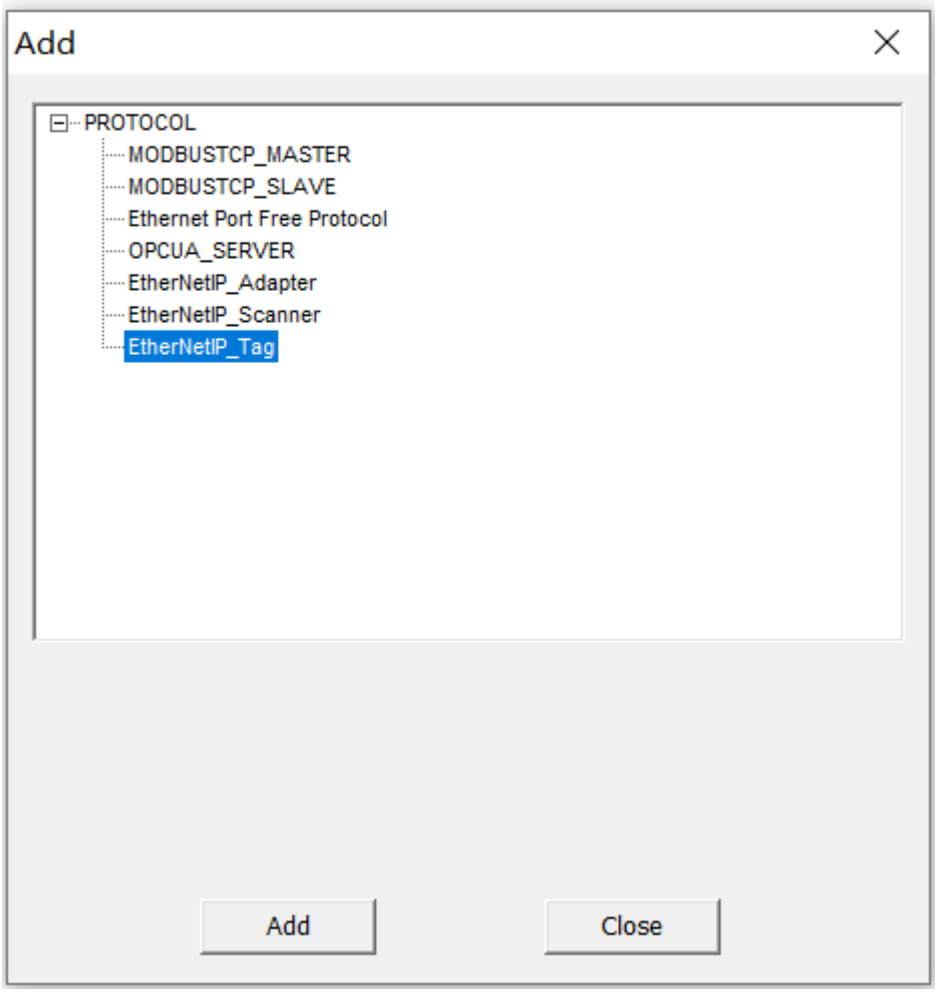
9.3.1 Add EtherNet/IP Tag

Follow these steps to add an EtherNet/IP Tag:

(1) Right-click on EtherNet1 or EtherNet2 in the configuration interface, select "Add Protocol" from the context menu.



(2) In the "Add" dialog box, choose "EtherNetIP_Tag".



(3) After completing the addition, click the "Close" button.

9.3.2 Configuring EtherNet/IP Tag

Double-click the "EtherNetIP_Tag" node to open the EtherNet/IP Tag communication configuration interface. This interface includes the following sections: Tag Groups, Target Device, Connection Settings, and Message Window. The details are as follows:

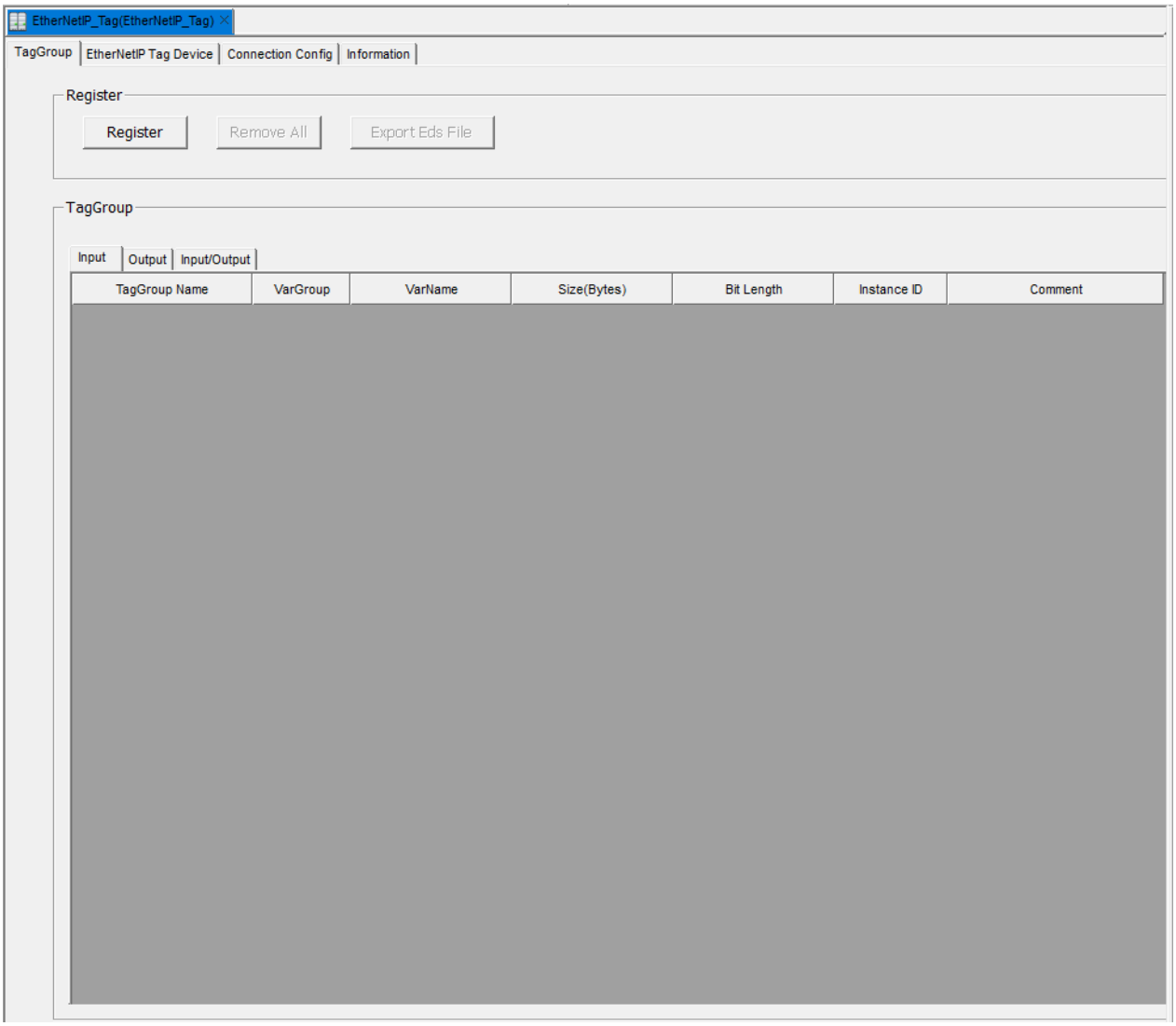
- TagGroup

  Register: Add tag groups.

  Remove All: Remove all tag groups and their associated tags.

  Export EDS File: Export the EDS configuration file for EtherNet/IP Tag (can be performed after project compilation).

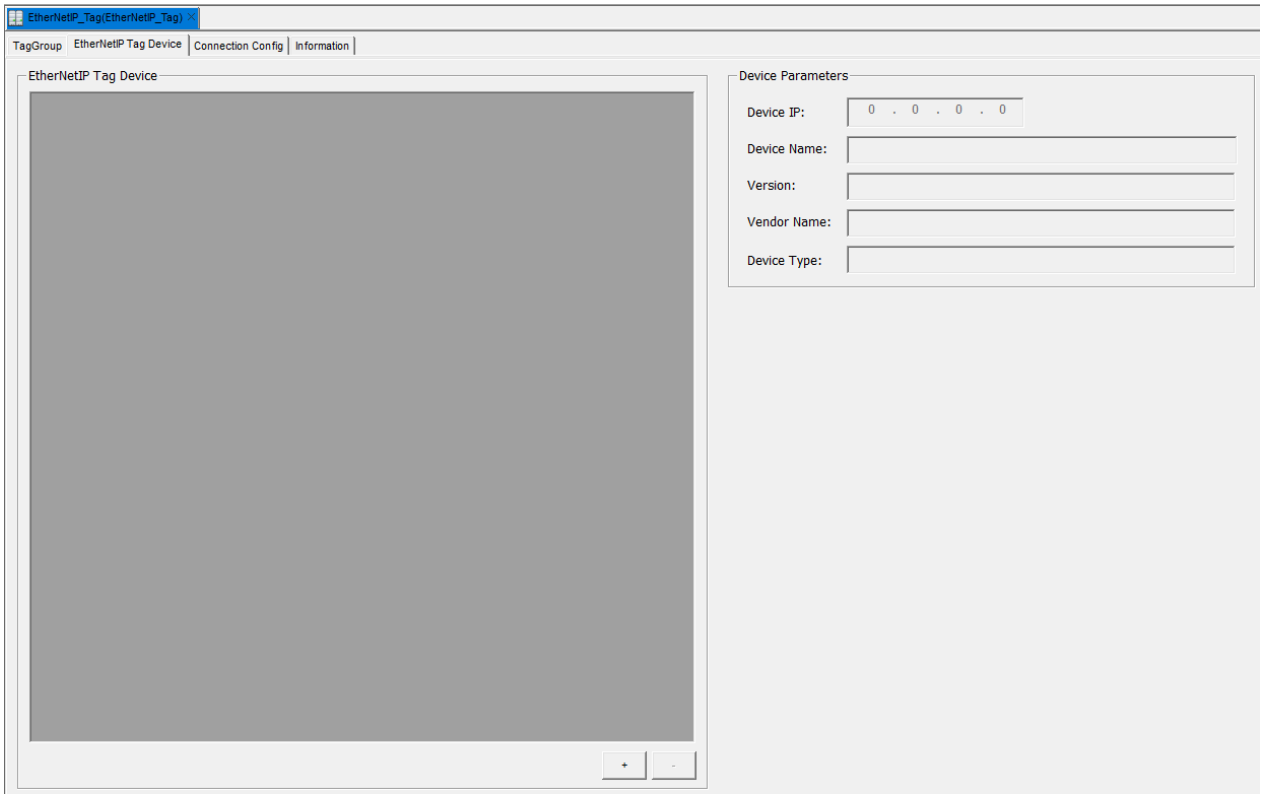
  TagGroup: Includes Input, Output, and Input/Output. Right-click in a blank area to add a tag group.



- EtherNetIP Tag Device

  Target Device Properties: Includes Device IP, Device Name, Version, Vendor Information, and Device Type.

  After adding a target device, only the Device IP can be configured; other fields are read-only.



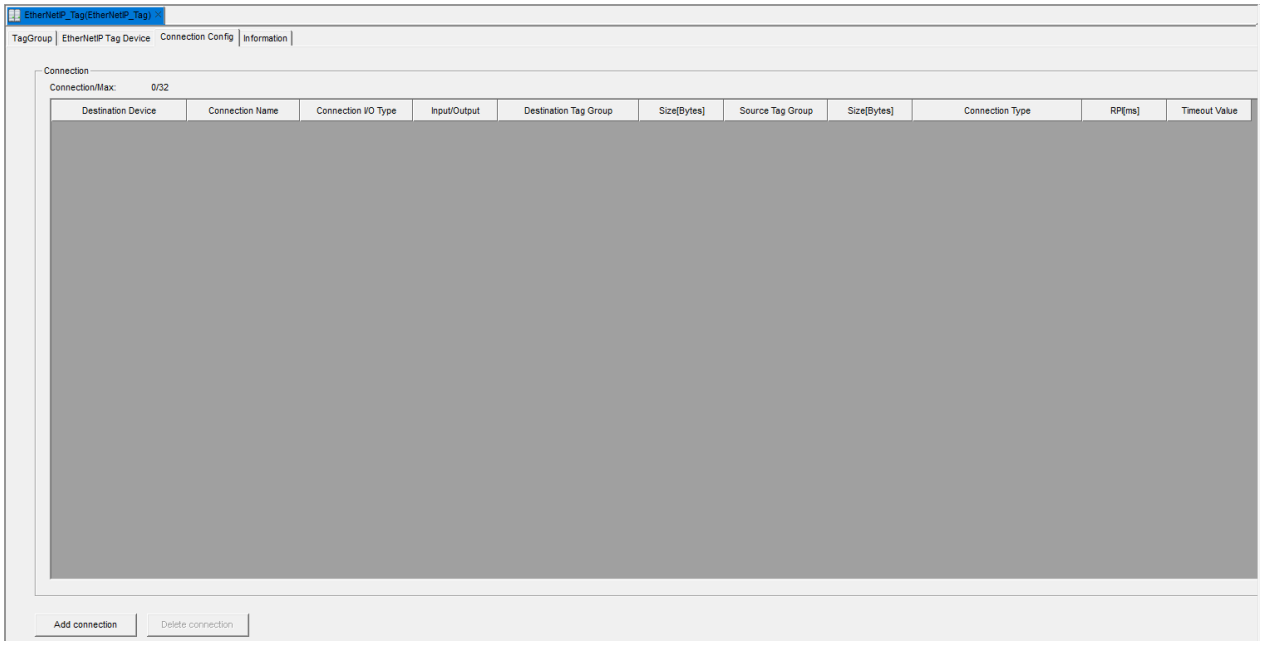
- Connection Config
  - Add Connection: Click the "Add Connection" button to create a new connection.
    - Destination Device: Select a pre-configured device from the "Target Device" dropdown as the connection peer.
    - Connection Name: Default: "default_xxx" (editable).
    - Connection I/O Type: Determined by the peer device.
    - Input/Output: Defined by the peer device (non-editable).
    - Destination Tag Group: Manually enter the peer device's tag group name.
    - Size [Bytes] 1: Manually specify the peer tag group's byte size.
    - Source TagGroup: Manually enter the local tag group name.
    - Size [Bytes] 2: Automatically determined by the source tag group (non-editable).
    - Connection Type:

    Point-to-Point: Direct data exchange between two devices.

    Multicast: One-to-many communication within a network segment (selected devices participate in read/write operations).

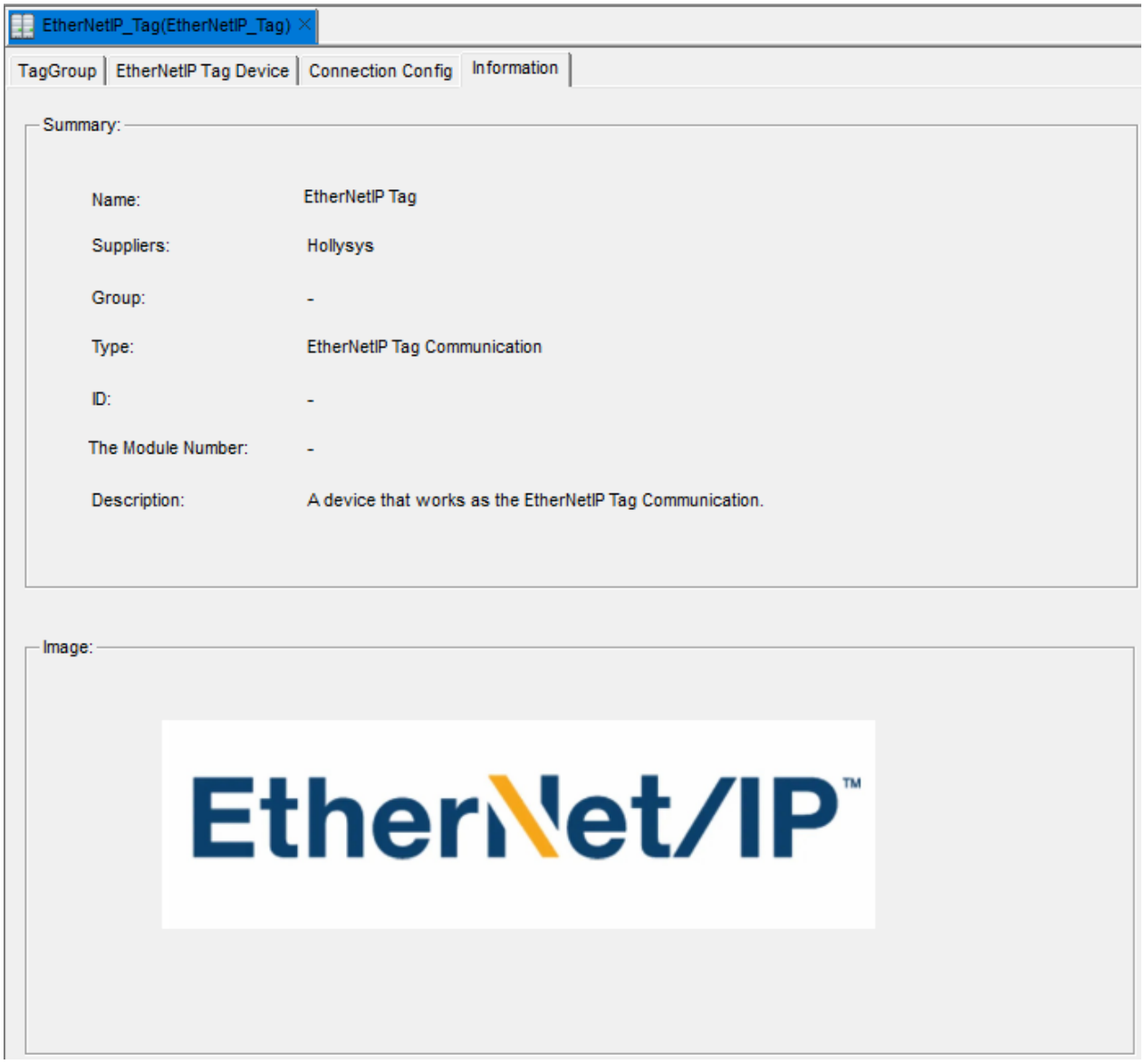
    - RPI [ms]: Sets the local device's polling interval for reading peer data. Range: 10–10,000 ms.
    - Timeout Value: Configures the connection timeout period, calculated as RPI × (4/8/16/32/64/128/256/512) (selectable multiplier).

□ Delete Connection: Select a connection and click "Delete Connection" to remove it.



■ Information

Displays EtherNet/IP Tag-related information.



9.3.3 EtherNet/IP Tag Configuration Example

1. Adding Tags

   Tags can be added in two ways:

   ■ Method 1: Adding via the "Tag Registration" Button

   (a) Add variables in the Global Variables or PRG Program section. Set the Network Exposure property to "Public" (Input/Output), "Input", "Output".

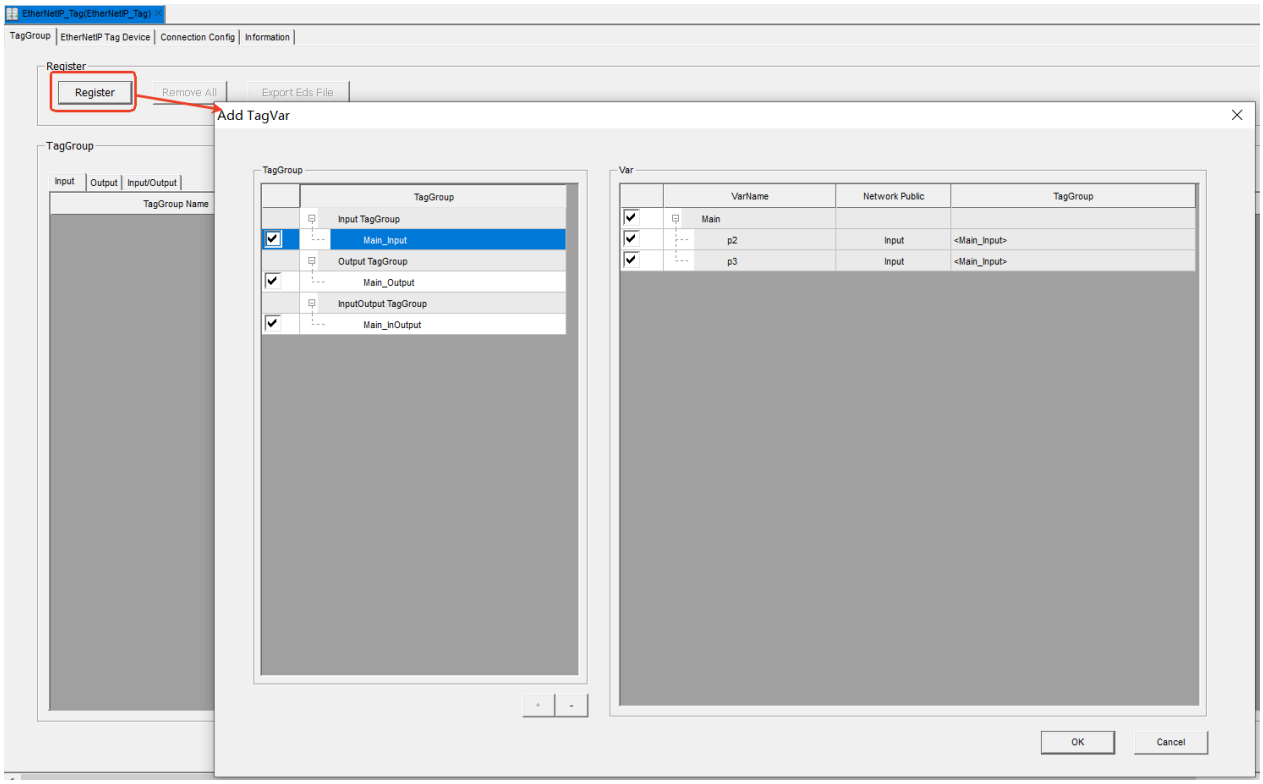
Main(PRG).kl

No.	Variable Name	Address	Variable Description	Variable Type	Initial Value	Power Fail Safeguard	NetWork Public	Enable constant
0001	p1			BOOL	FALSE	FALSE	Public	
0002	p2			BOOL	FALSE	FALSE	Input	
0003	p3			BOOL	FALSE	FALSE	Input	
0004	p4			BOOL	FALSE	FALSE	Output	

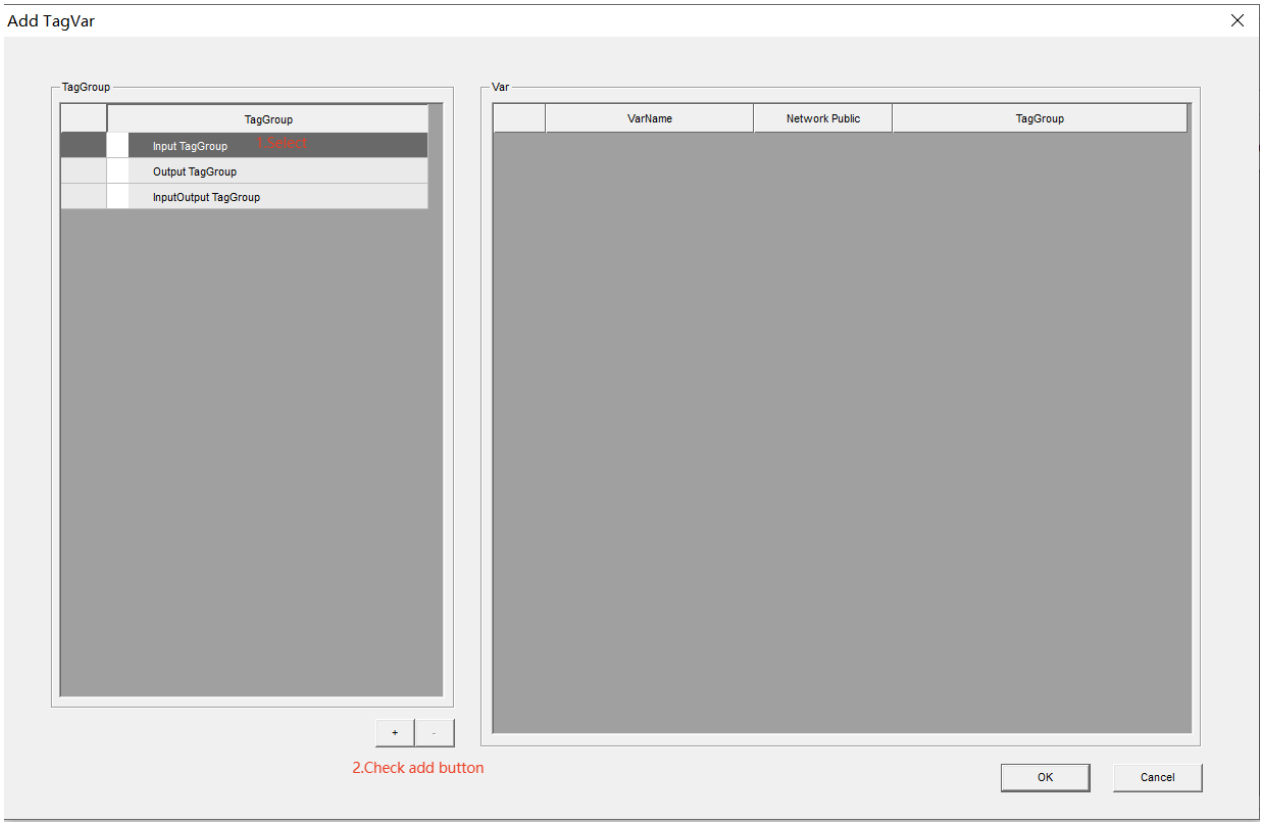
(b) Double-click EtherNetIP_Tag to open the configuration page. In the Tag Groups tab, click "Register".

In the "Add TagVar" window, select the desired tag group and click OK. Input variablesa added to the Input Tag Group. Output variables added to the Output Tag Group. Public variables added to the Input/Output Tag Group. See figure (a) below.

If no variables are pre-defined in Global/PRG sections, In the "Add TagVar" window, select a tag group and click the + button to create new tags. See figure (b) below.



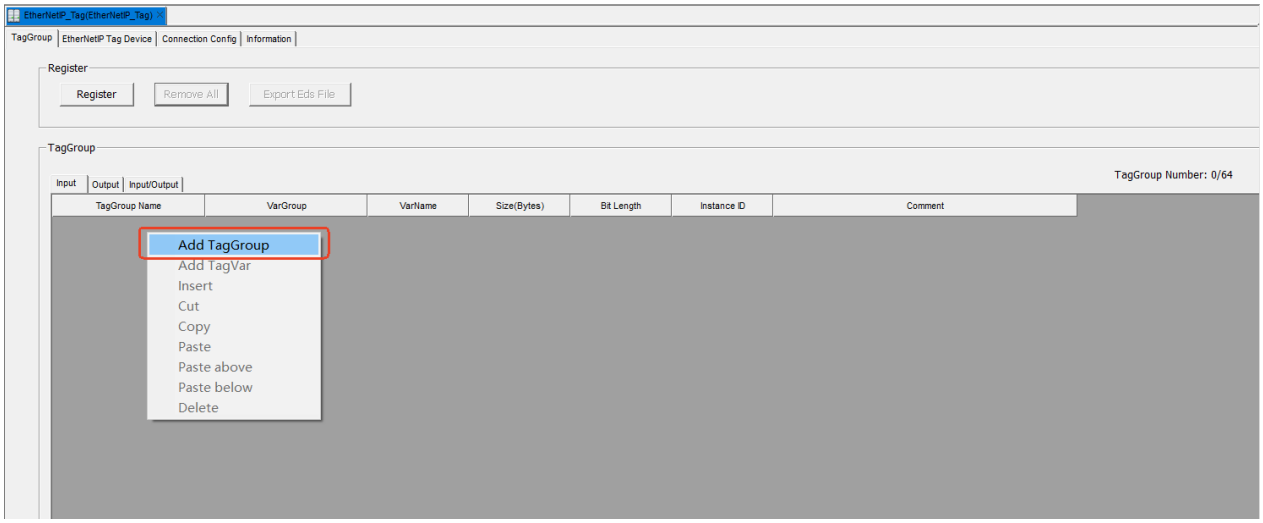
(a)



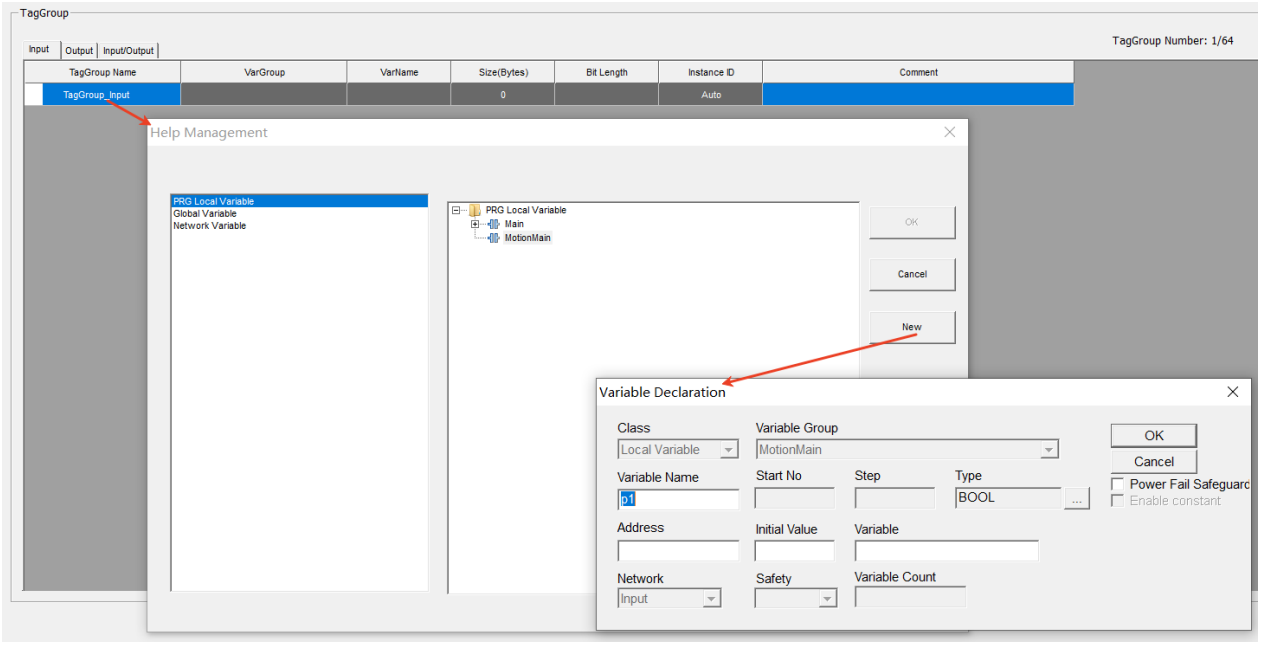
(b)

■ Method 2: Adding via Right-Click Menu in Tag Group I/O Section

(a) Right-click in the Input/Output area and select "Add TagGroup".



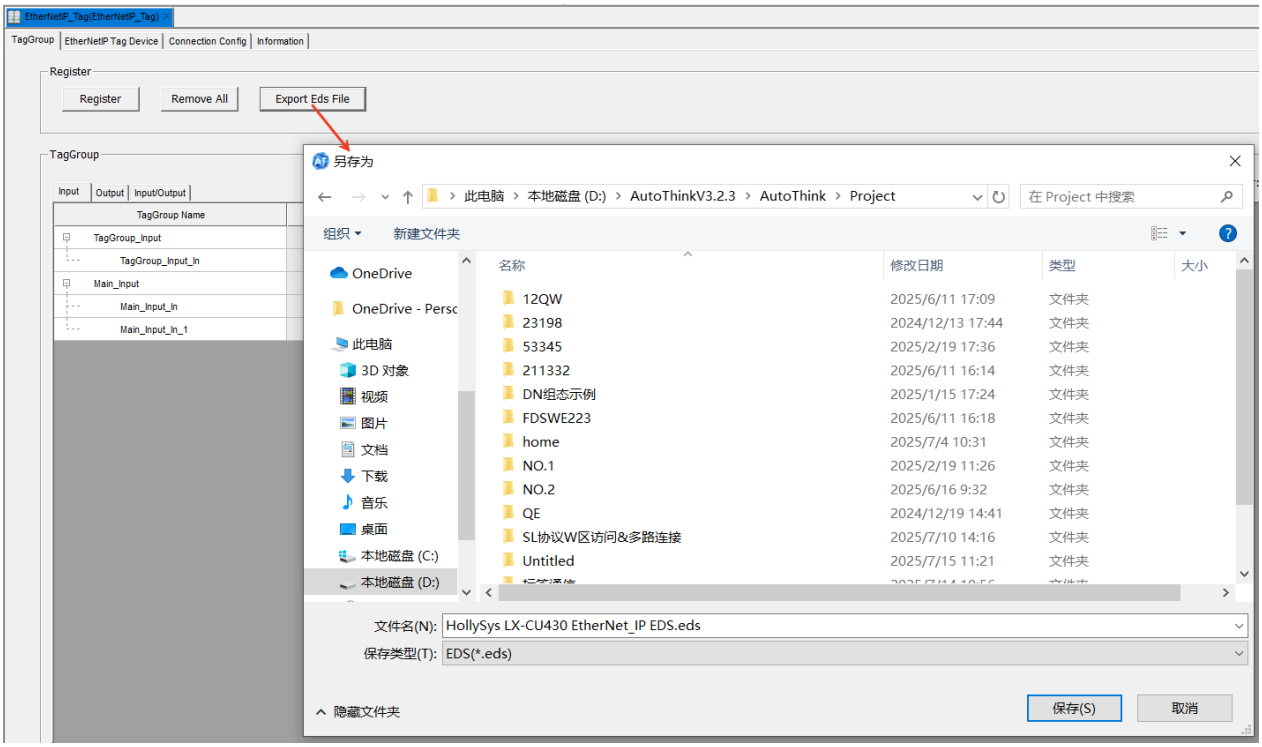
(b) Select the target tag group (Input/Output/Input-Output) in the configuration interface, right-click and choose "Add TagVar".Select a variable group (or click "New" to declare a new tag variable). Configure tag properties and confirm.



Select one or multiple tags, right-click and select "delete" or click the "remove all" button to remove all tags at once.

2. Exporting EDS File

After compiling the project, click the "Export EDS File" button, select a local save path to generate the EDS configuration file.



If it is only used as an EtherNet/IP slave station, import the exported EDS into the EtherNetIP_Tag master of the LX-CU430 or a third-party EtherNet/IP master. After configuring the IP and input/output

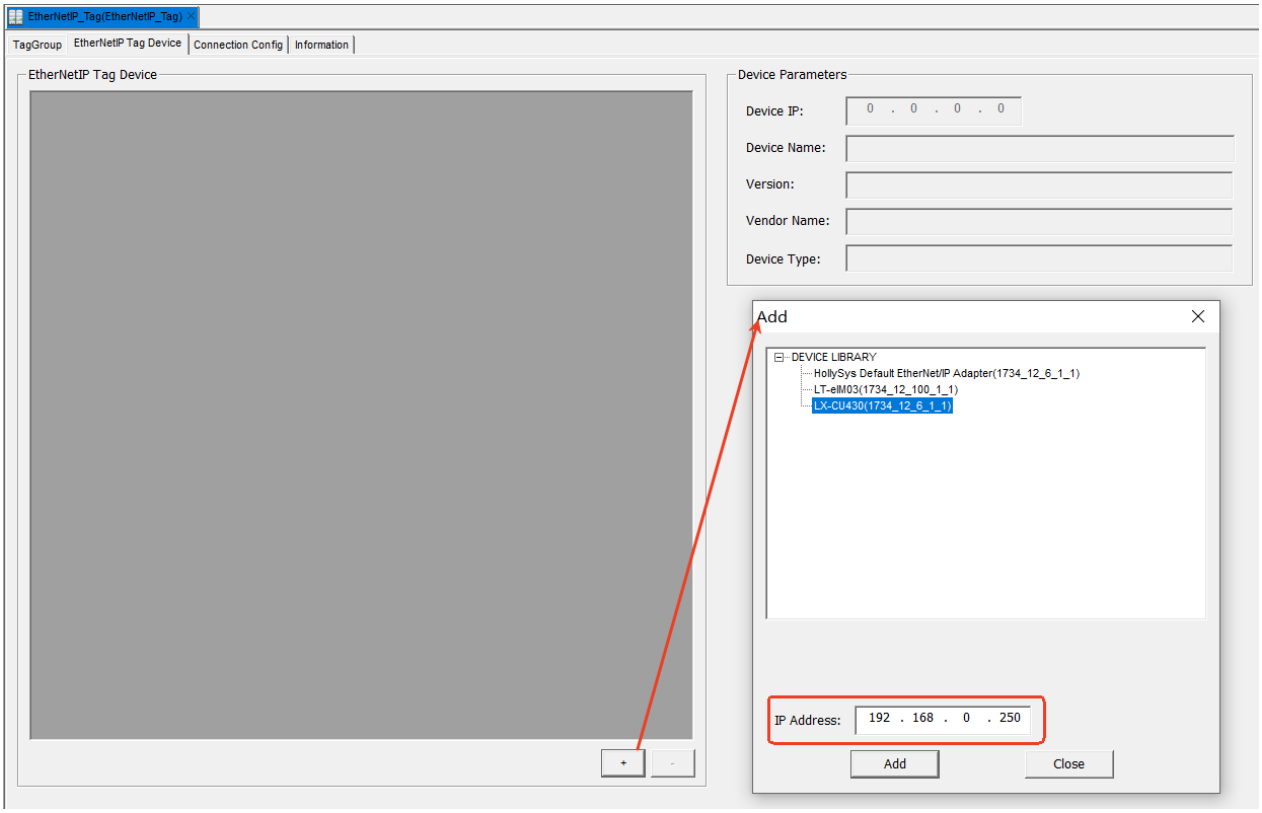
zone sizes, communication can be established.

If it is also used as an EtherNet/IP master station, proceed with the relevant configurations.

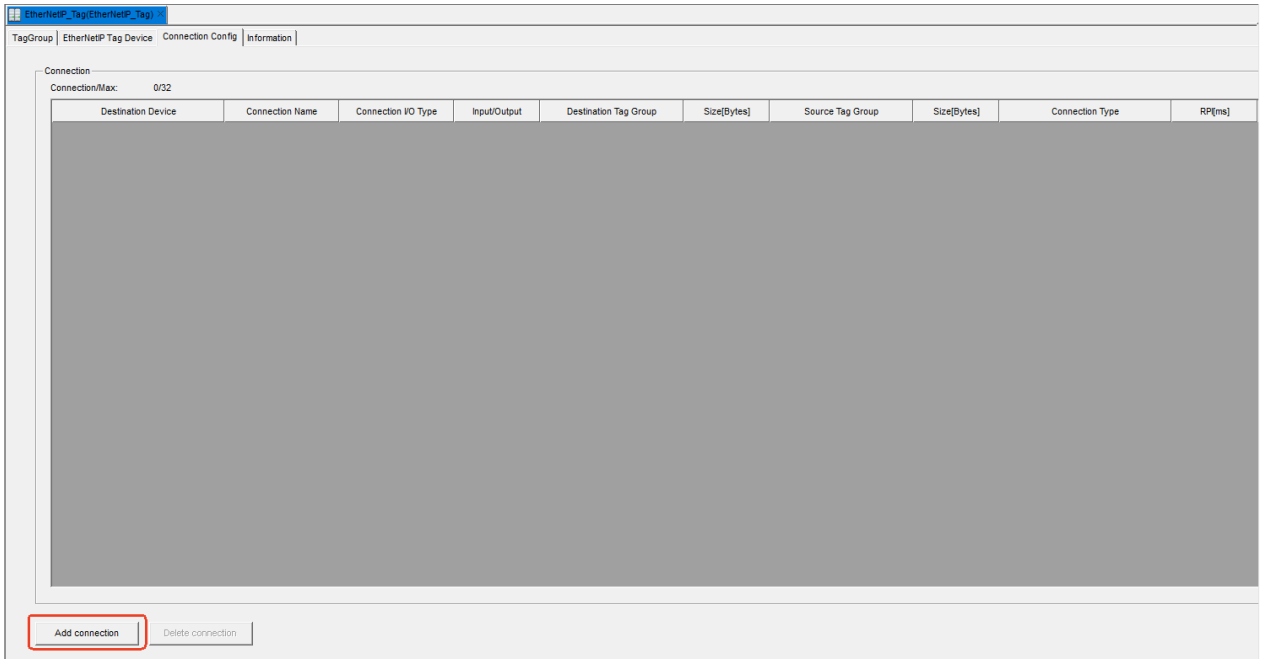
3. Peer Configuration

(1) Create a new LX-CU430 project, select "Project - Import Device Description File" from the menu bar, and import the device description file (HollySys LX-CU430 EtherNet_IP EDS.eds) corresponding to the EtherNet/IP slave station created in the previous step.

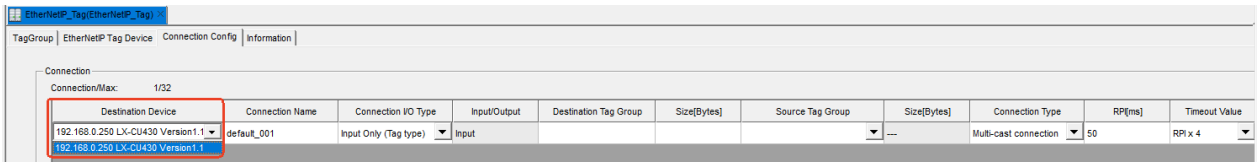
(2) On the "Target Device" tab, click the "+" button to add the imported slave station, and configure the device IP to match the EtherNet/IP slave station's IP. Click "Add" and then close the dialog.



(3) On the "Connection Config" page, click the "Add Connection" button to add a new connection.



(4) Click the drop-down list in the "Target Configuration" column and select the target device. Click the corresponding columns in the table to configure "Connection I/O Type", "Destination Tag Group" as the peer tag group name. "Source Tag Group" as the name of the current Tag group communicating with the peer, "Size [Bytes]" for the target tag group (the source and target tag group sizes must match). configure "Connection Type", "RPI [ms]", and "Timeout Value".



(5) Click the "Compile" button on the toolbar to compile. Once compilation succeeds, click the "Download" button to download the configuration.

(6) After a successful download, click the "Go Online" button on the toolbar to go online. You can then read and write data in the variable area.

9.4 EtherNet/IP Tag Communication Example

9.4.1 Example Function

A host computer uses Sysmac Gateway software to communicate with the PLC via the EtherNet/IP (EIP) network, enabling high-speed, large-volume data acquisition and control over Ethernet.

When communicating with Omron controllers, note the following:

A communication connection cannot be established if the data lengths of the tag groups on both sides do not match. Therefore, when communicating with Omron devices, if tag groups containing the following

data types are used, ensure that the total data length of the tag group meets the requirements. Alternatively, the data length can be padded by setting valid bits or adding variables.

(1) BOOL type: In Omron configuration, 1 variable occupies 2 bytes, while in AT, 1 variable occupies 1 byte. (Modifying bit communication is ineffective; the minimum occupancy for a BOOL variable is 2 bytes.)

(2) DATE type: In Omron configuration, 1 variable occupies 8 bytes, while in AT, 1 variable occupies 4 bytes.

(3) DT type (corresponding to DATA_AND_TIME in Omron): In Omron configuration, 1 variable occupies 8 bytes, while in AT, 1 variable occupies 4 bytes.

(4) STRING type: For a STRING type with a defined length of *n* bytes: Omron occupies *n* bytes, while AT occupies *n+1* bytes (AT includes '\0' in the length calculation).

(5) TIME type: In Omron configuration, 1 variable occupies 8 bytes, while in AT, 1 variable occupies 4 bytes.

(6) TOD type (corresponding to TIME_OF_DAY in Omron): In Omron configuration, 1 variable occupies 8 bytes, while in AT, 1 variable occupies 4 bytes.

- The above methods only ensure that the tag communication connection is established normally; however, the correctness of data for types (2)/(3)/(5)/(6) cannot be guaranteed.
- When communicating with a STRING type requiring *n* bytes in length, Omron must define a STRING type with a length of *n+1* bytes. The last byte is used to align with AT's '\0' terminator and does not participate in communication, so the actual effective length remains *n* bytes.

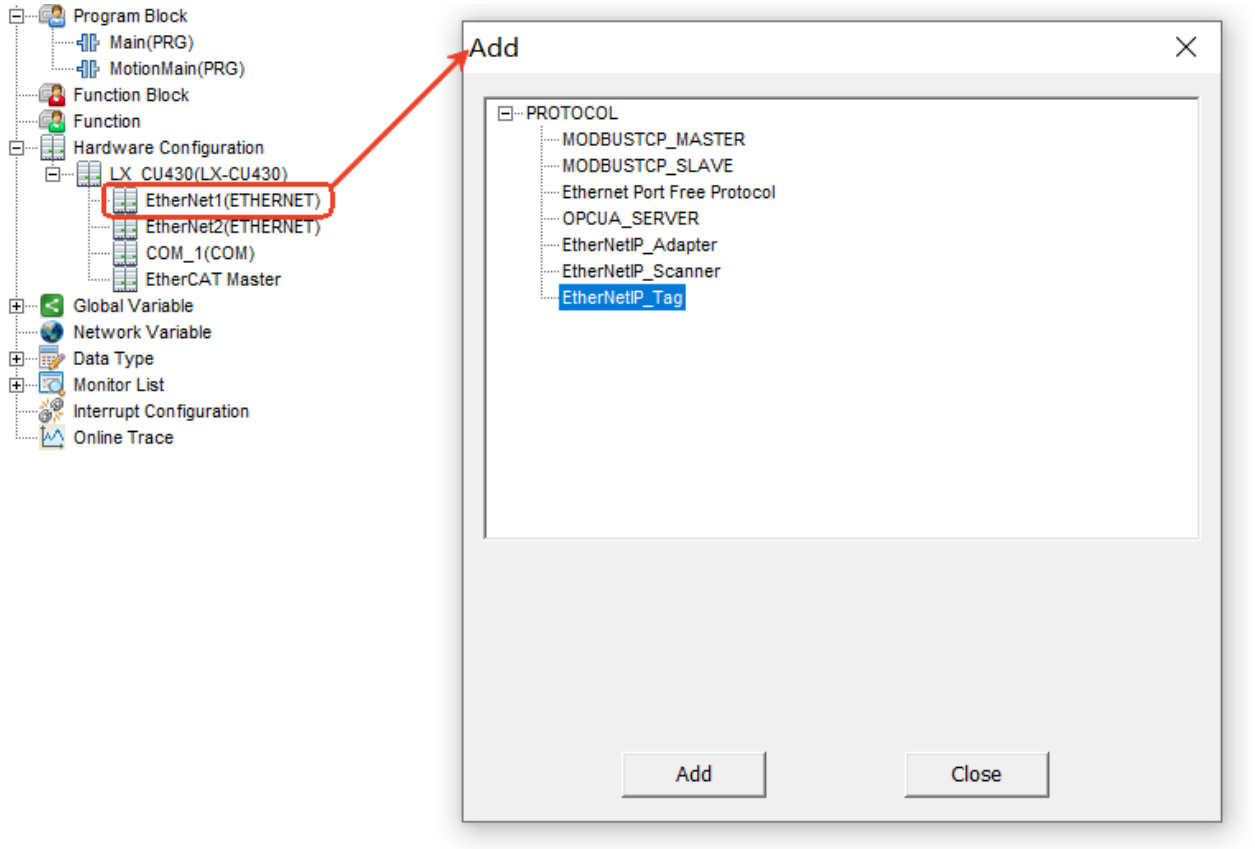
9.4.2 System Configuration

- Controller: LX-CU430, firmware version LX-CU430-A02_SP1.
- HollySys Software: AutoThink V3.2.3 SP2.
- Omron Software: SYSMAC Gateway 1.7.5.0.

9.4.3 Programming

9.4.3.1 AutoThink Software Configuration

(1) Right-click the "EtherNet1" node and select "Add Protocol". Choose the "EtherNetIP_Tag" protocol, which enables EIP tag-based communication, as shown in the figure below.

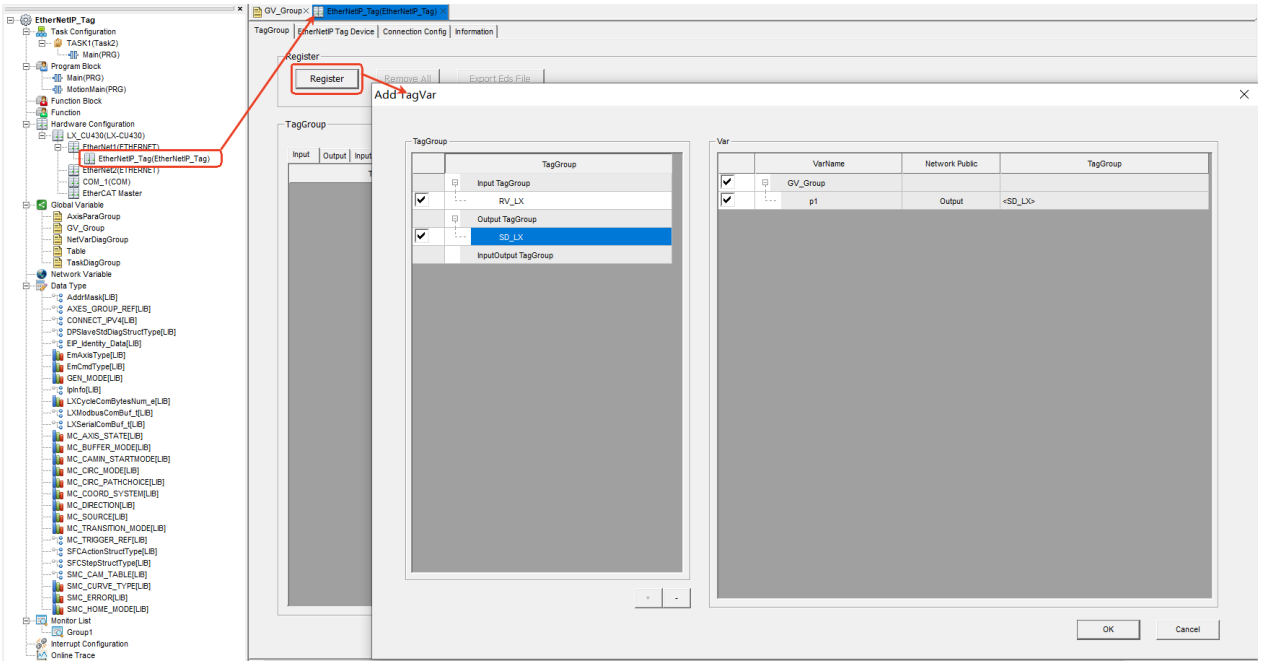


(2) Add two Word-type variables to the global variable "GV_Group", with their network public attributes set as output and input respectively.

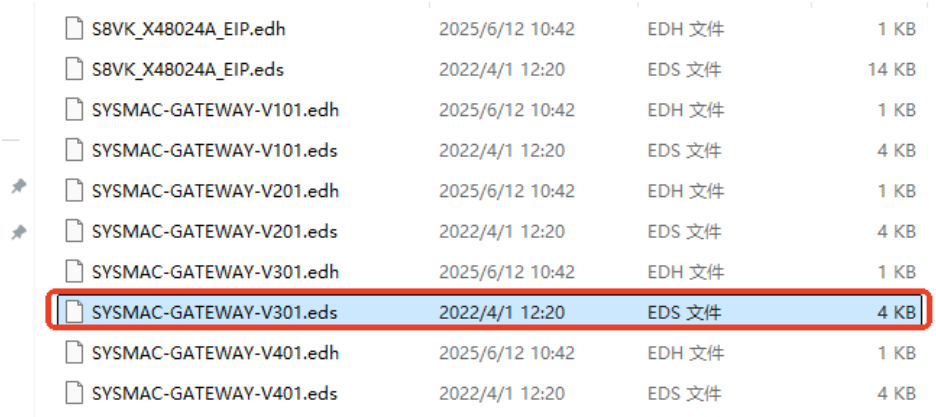
GV_Group

No.	Variable Name	Address	Variable Description	Variable Type	Initial Value	Power Fail Safeguard	NetWork Public	Enable constant
0001	p1			WORD	0	FALSE	Output	
0002	p2			WORD	0	FALSE	Input	

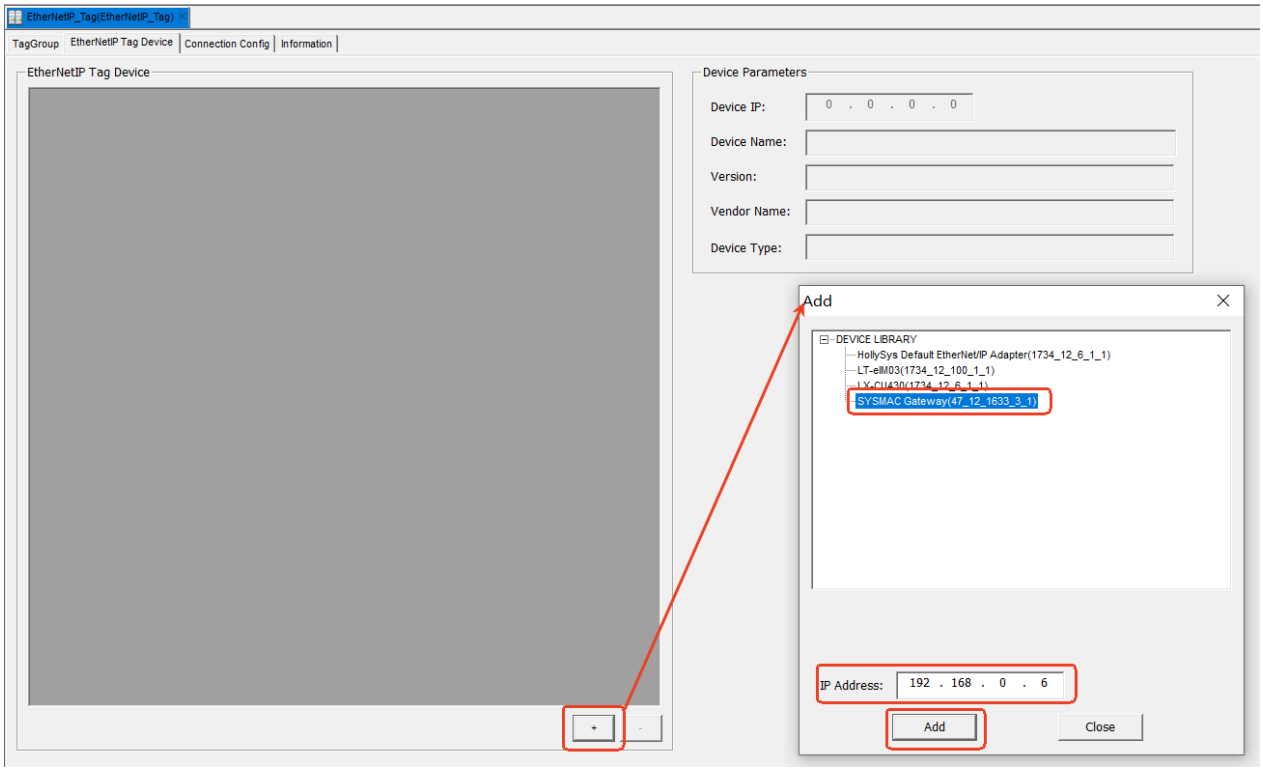
(3) Double-click the EtherNetIP_Tag node, then click the tag registration button to open the tag registration dialog. Change the input tag group name to RV_LX and associate it with the created variable p2. Modify the output tag group name to SD_LX and associate it with the created variable p1. The setup is as shown in the figure below.



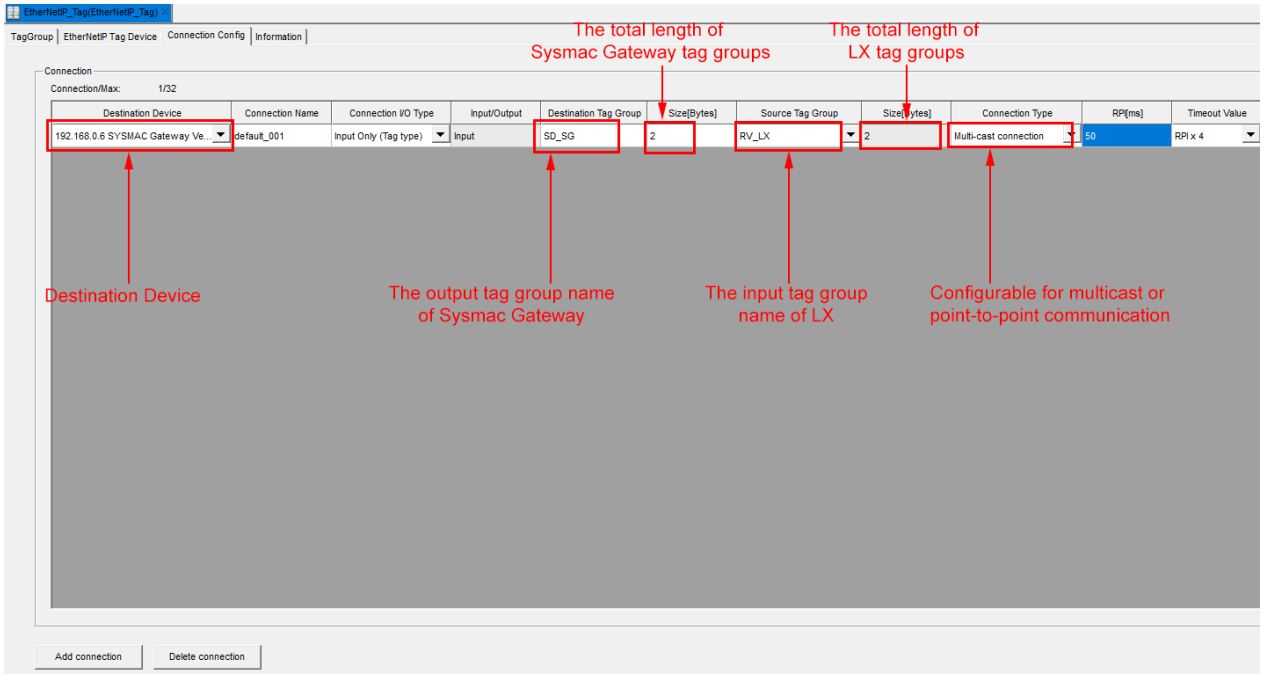
(4) Import the EDS file of Omron Sysmac Gateway software. The EDS file can be found in the installation directory of Sysmac Studio software: Sysmac \ Sysmac Studio \ IODeviceProfiles \ EipConnection \ Eds. In AutoThink, select "Project → Import Device Description File" to import the EDS file.



(5) In the target device window, click the "+" button to add the Sysmac Gateway device, and modify the IP address to match the computer's network card IP, as shown in the figure below.



(6) Add a connection as illustrated in the figure, setting the I/O type to Input Only. Configuration can be done as shown in the figure, ensuring the target tag group length matches the source tag group length.

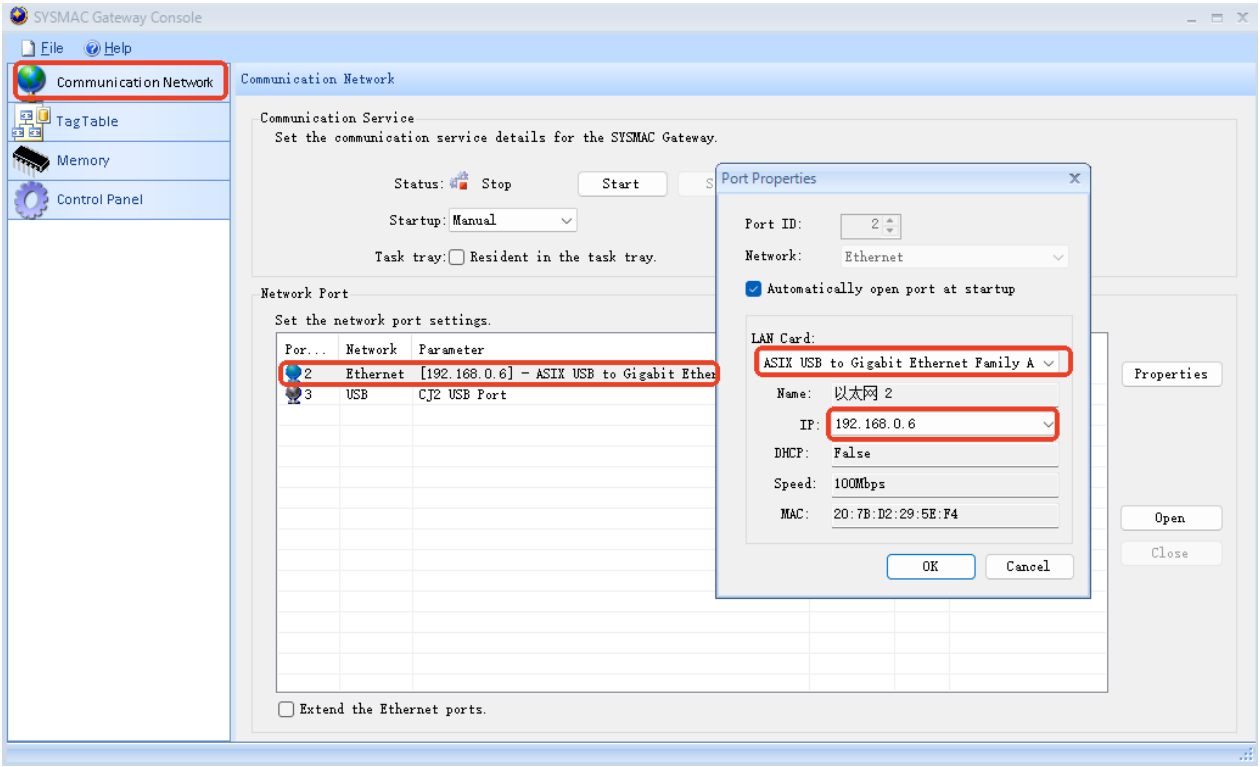


(7) At this point, the AutoThink project configuration is complete. Compile and download the program to proceed.

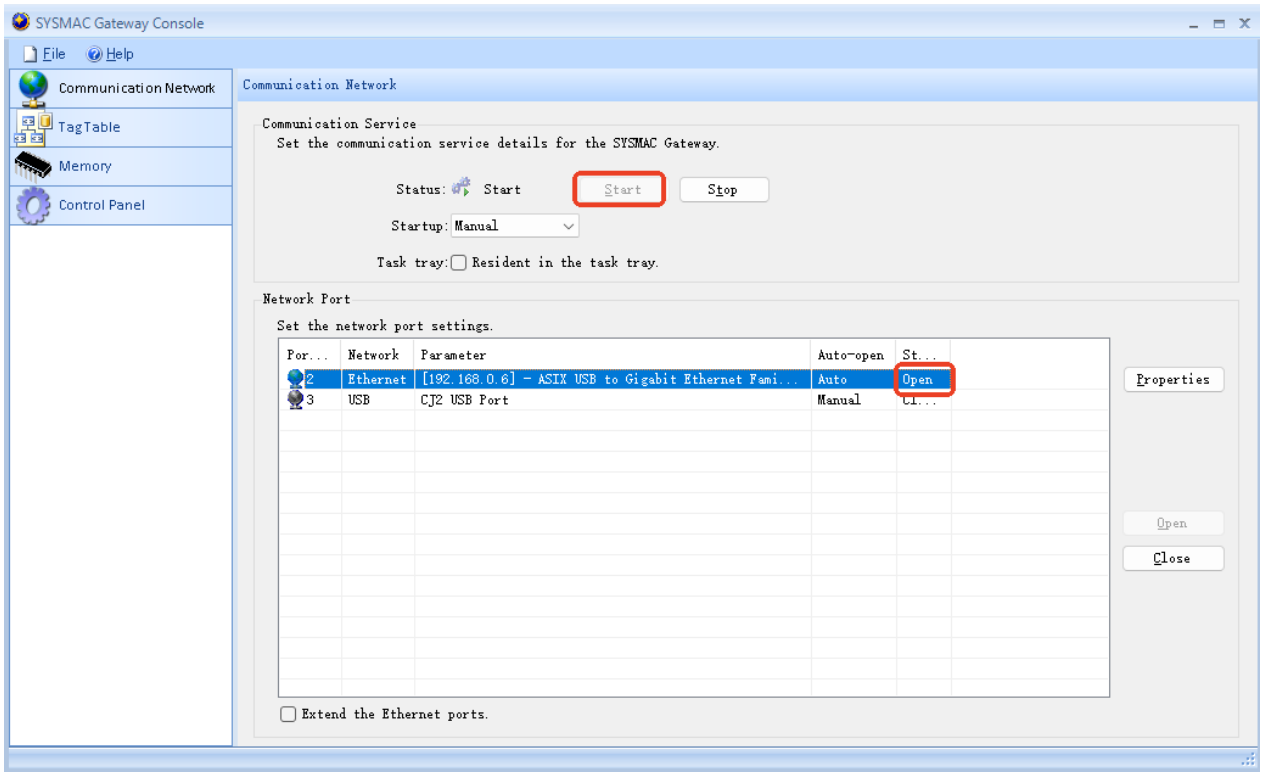
9.4.3.2 Sysmac Gateway Software Configuration

9.4.3.2.1 Creating a Sysmac Gateway Network

(1) Double-click "Communication Network" → "EtherNet", then select the computer's network adapter. Check whether the IP address is in the same subnet as the LX controller. If they are not in the same subnet, modify the computer's IP address accordingly.

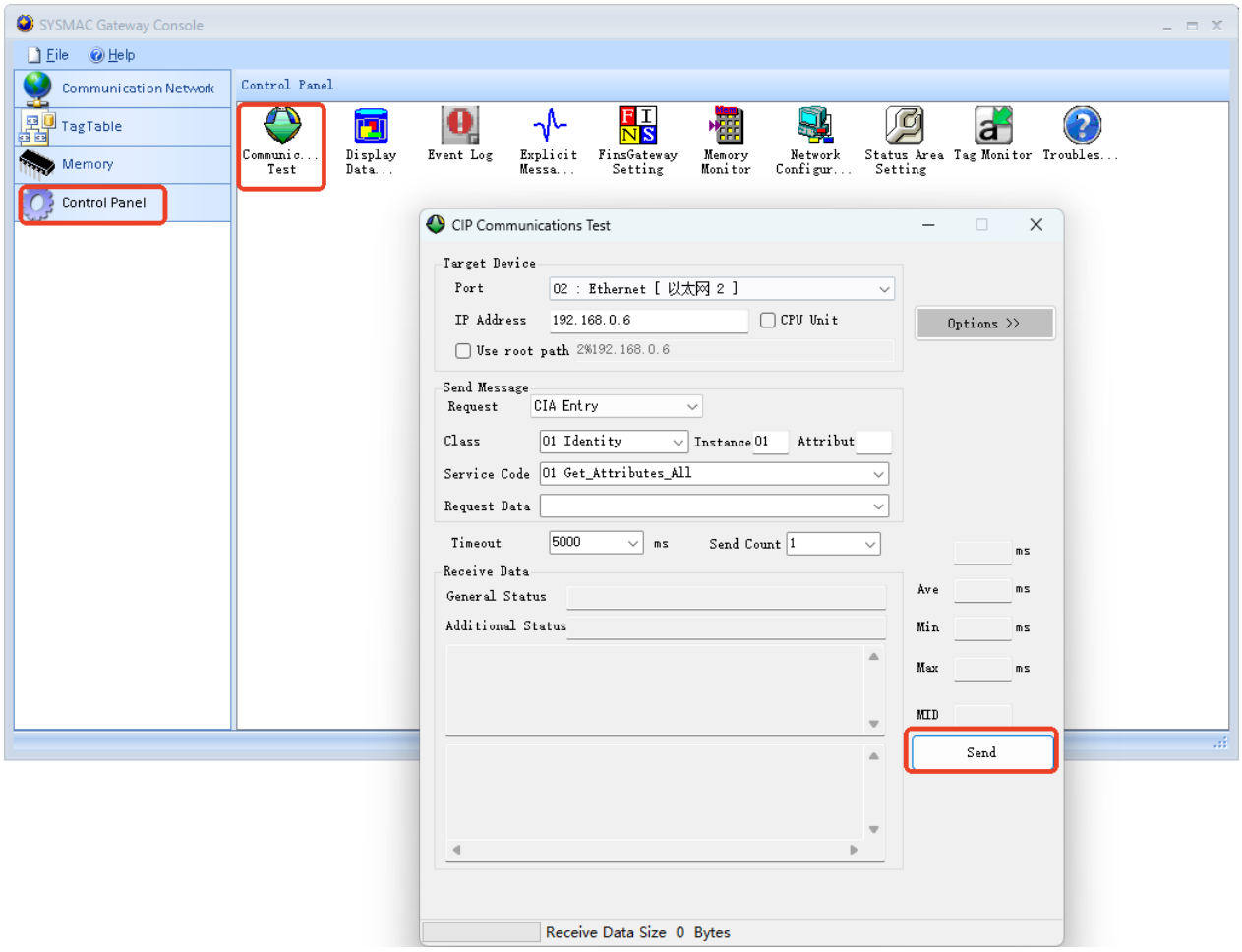


(2) Click "Start" to set the gateway to the Open state, as shown in the figure below.

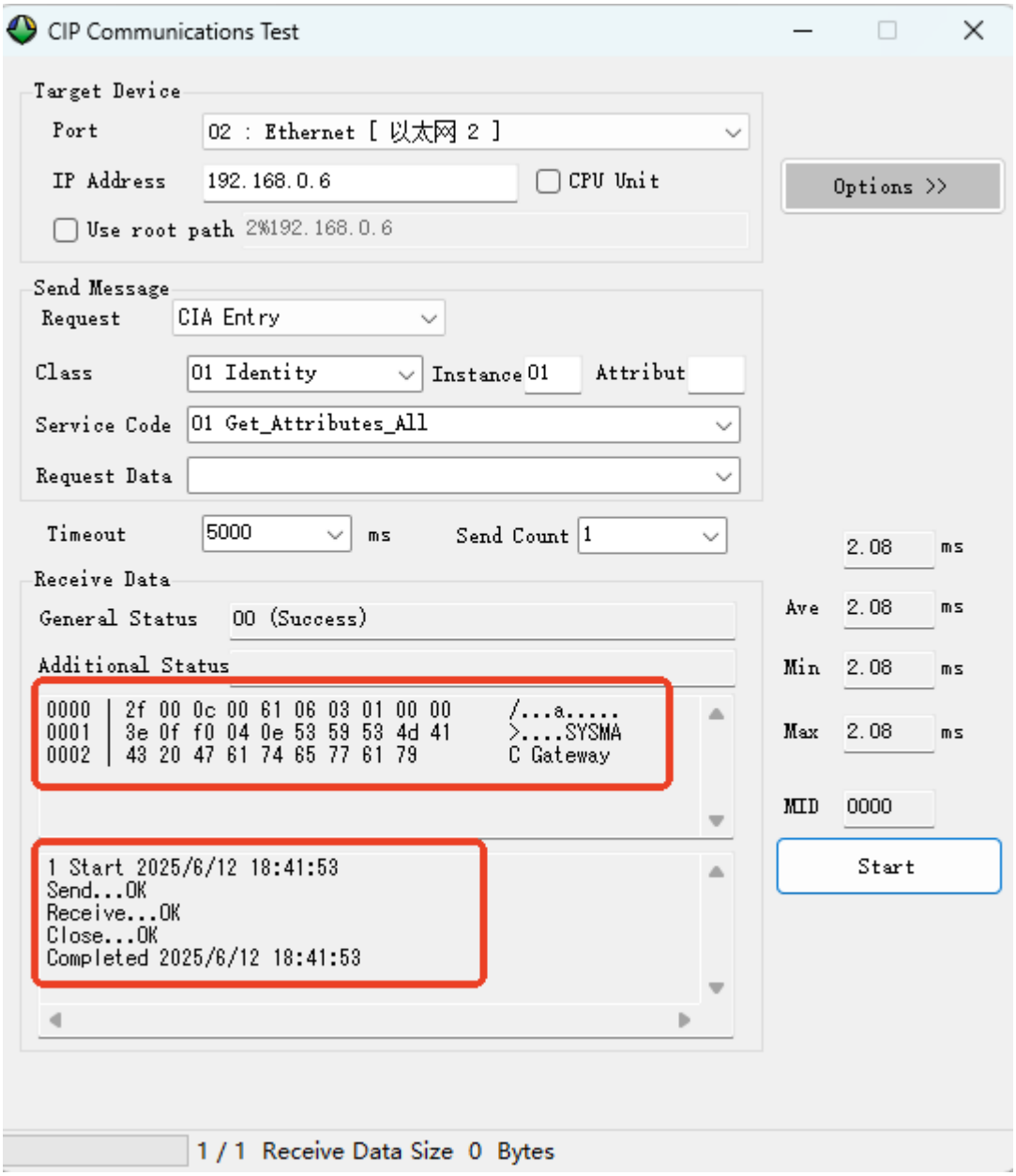


9.4.3.2.2 Testing Network Status

(1) Double-click "Control Panel" → "Communication Test" → "Send".



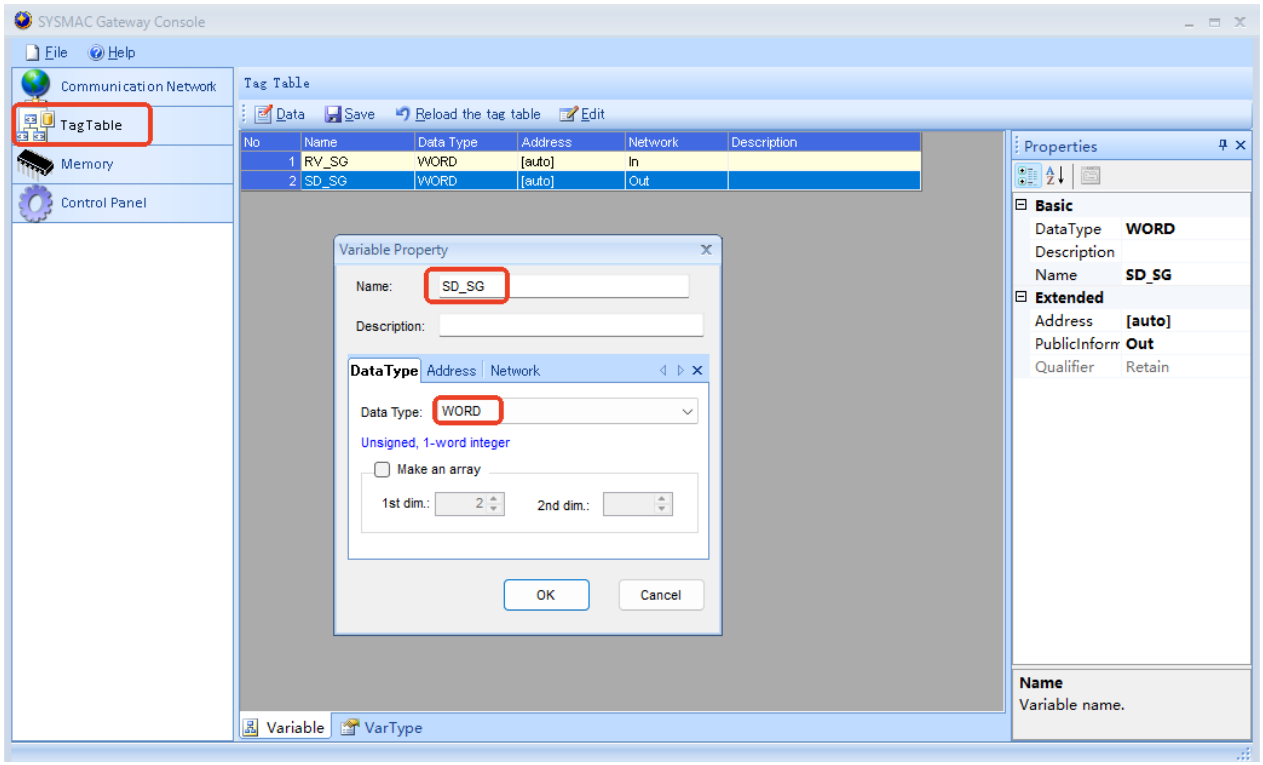
(2) View gateway information to confirm that the Sysmac Gateway is detected, with both SEND and RECEIVE statuses displayed as "OK."



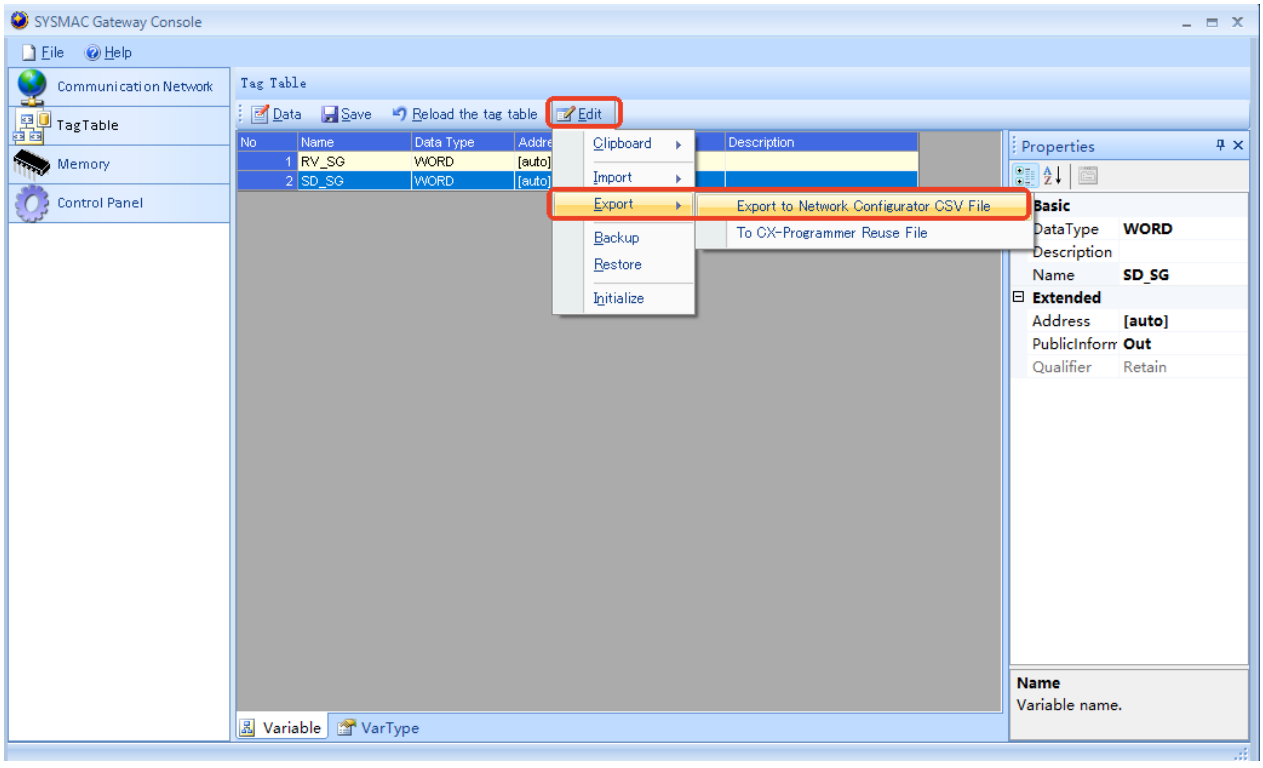
9.4.3.2.3 Create Tags

(1) Double-click the "Tag Table" menu, then right-click and select "Add Variable" to open a dialog box. Create a tag group name, set the data type to WORD, and assign the Network as IN or OUT. The tag group must align with the LX controller's byte configuration, with the corresponding relationship as follows:

LX	Tag Type	Sysmac Gateway	Tag Type
SD_LX	Output Tag	RV_SG	Input Tag
RV_LX	Input Tag	SD_SG	Output Tag



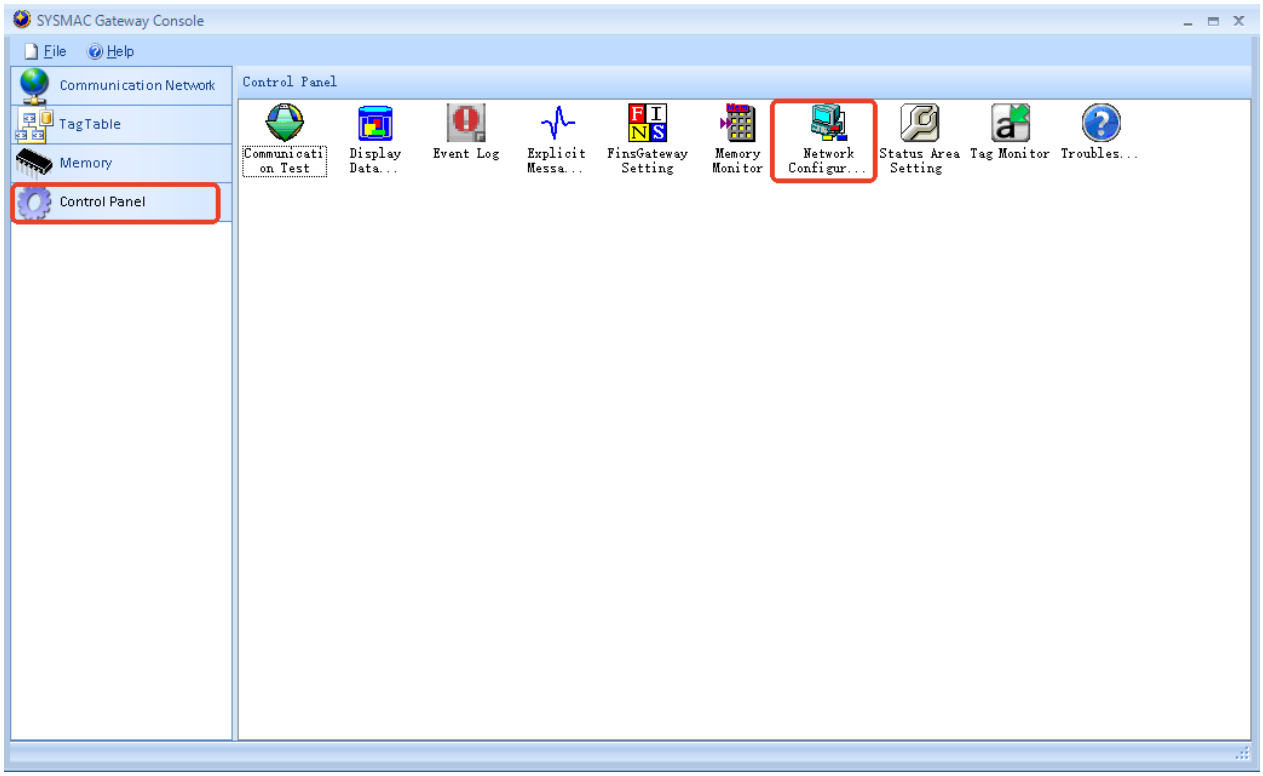
(2) Select "Edit" → "Export" → "Export to Network Configurator CSV File" to export all tag files.



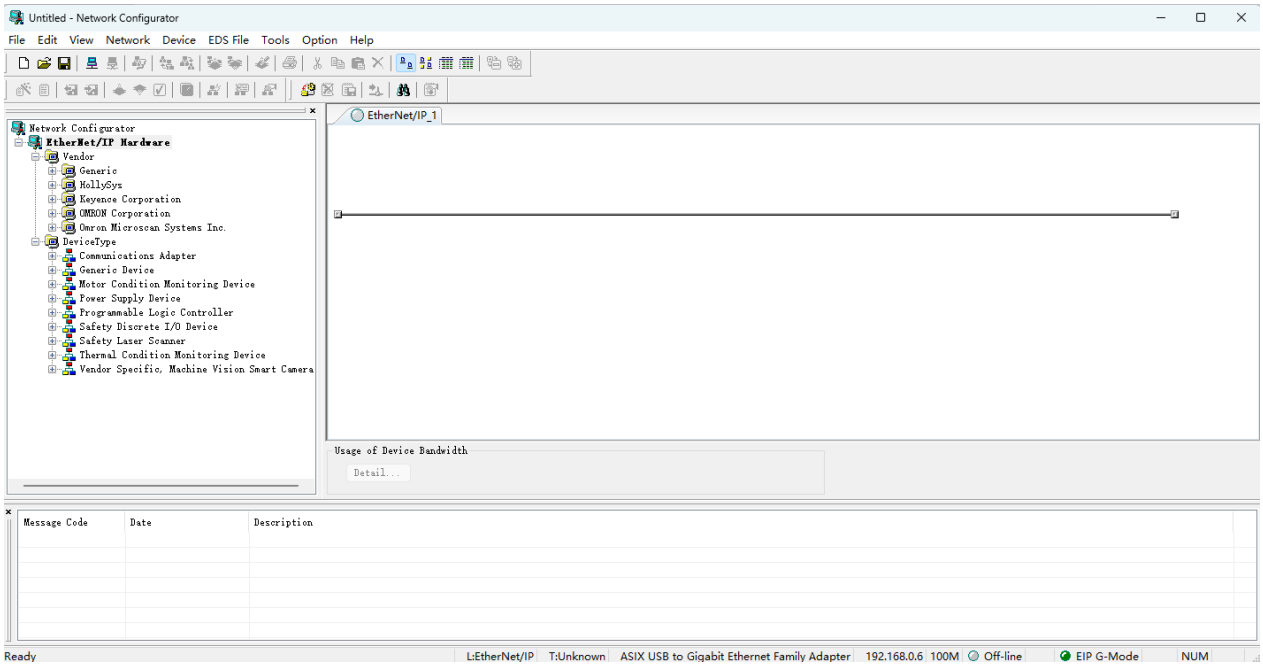
9.4.3.2.4 Network Configurator Configuration

(1) Double-click "Control Panel" - "Network Configurator" to open the Network Configurator software

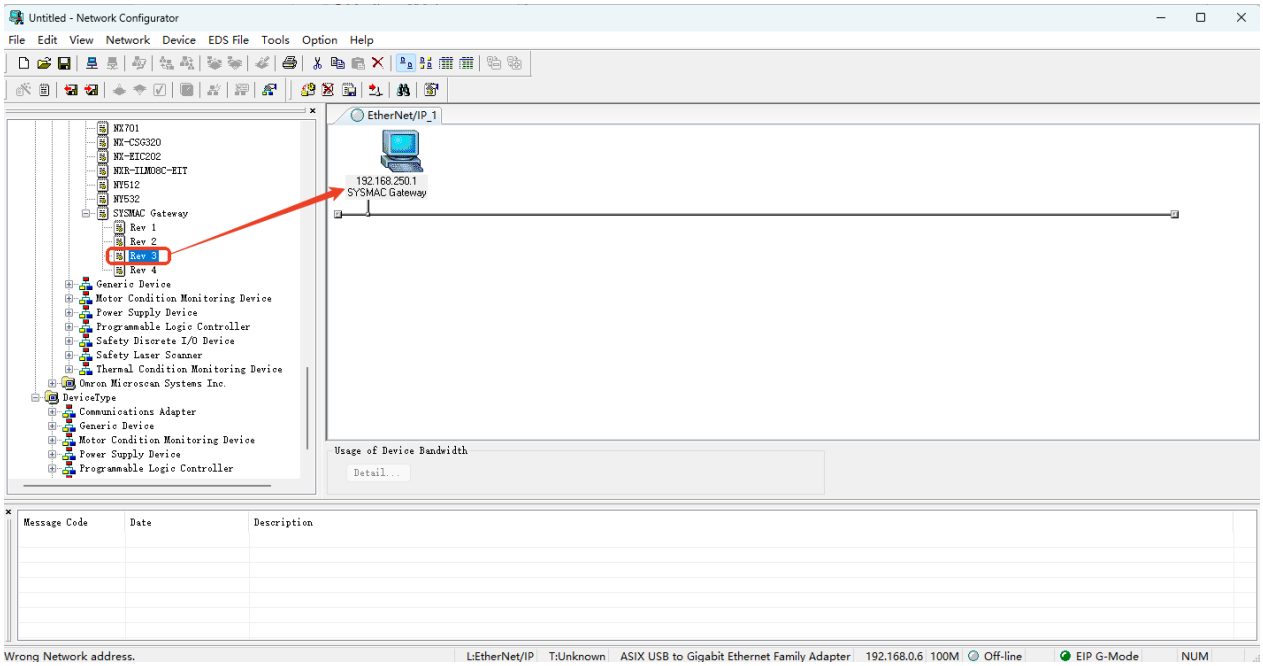
and configure the EIP network connection.



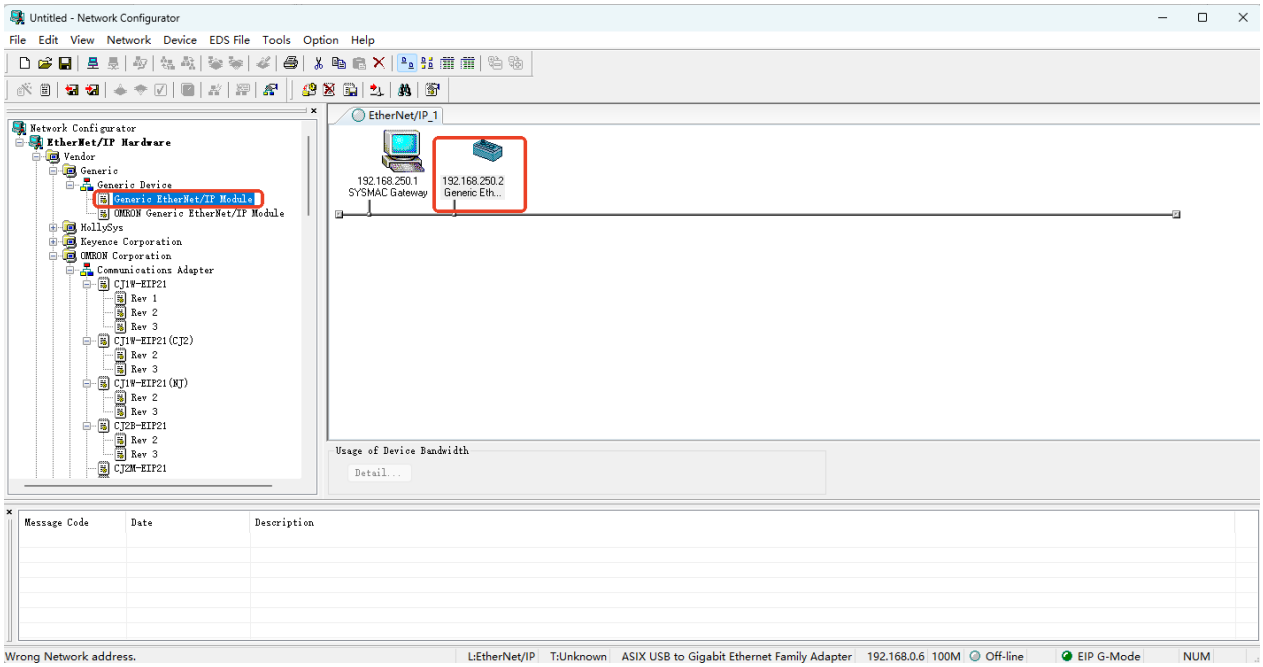
(2) The Network Configurator software is a utility tool for creating connections and downloading configurations. The interface is shown in the figure below.



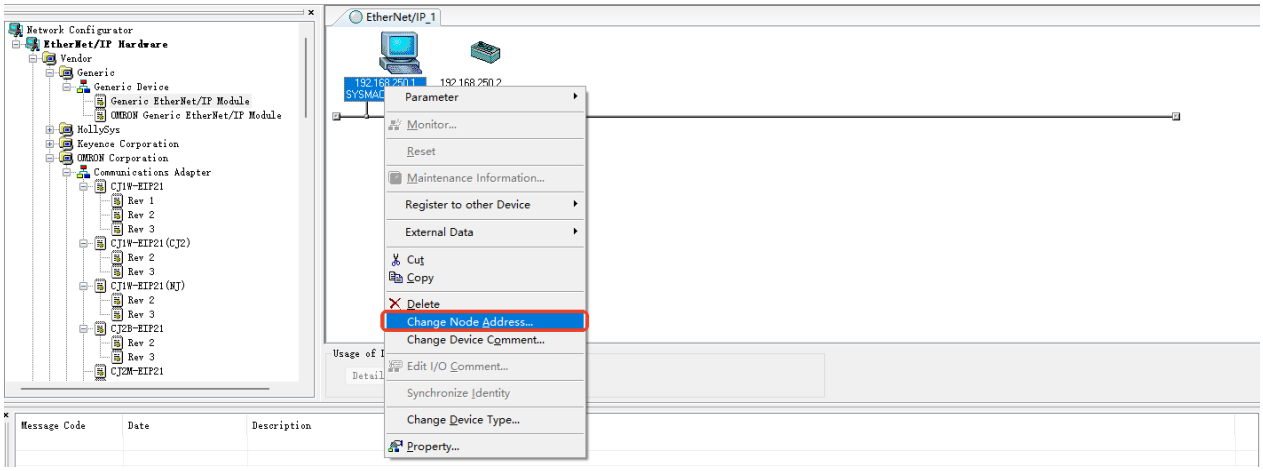
(3) Add SYSMAC Gateway Rev3 to the EtherNet/IP network as illustrated in the following diagram.



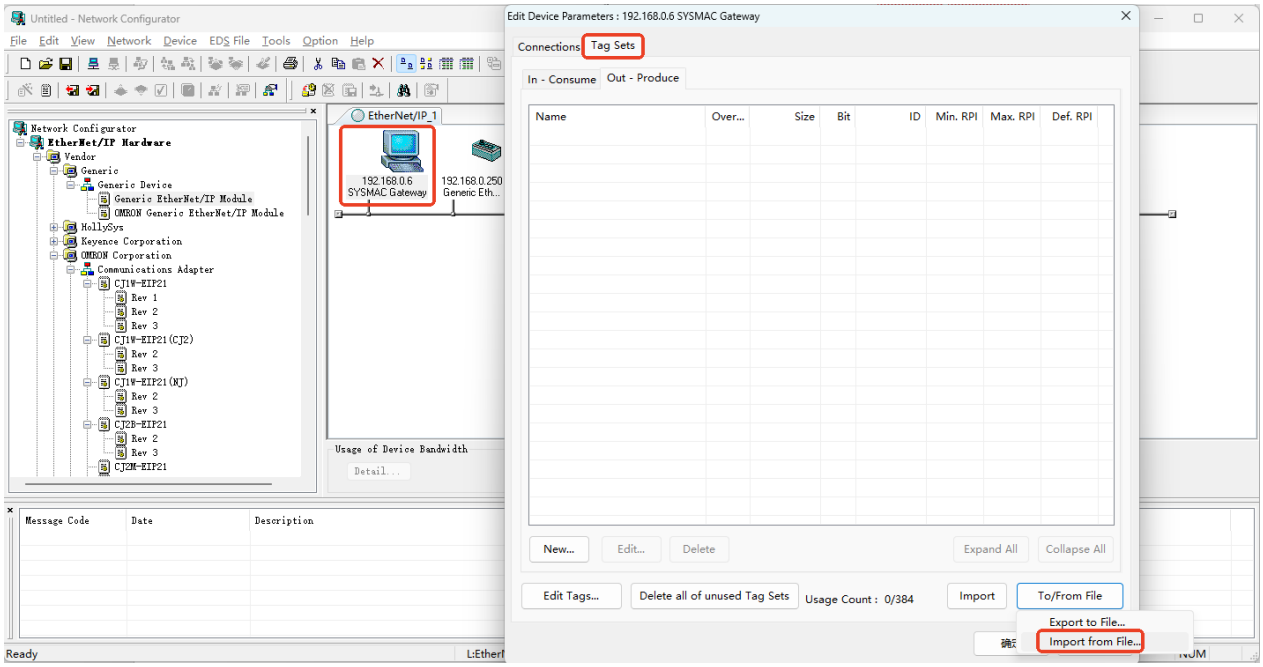
(4) Add a generic device, the Generic EtherNet/IP Module.



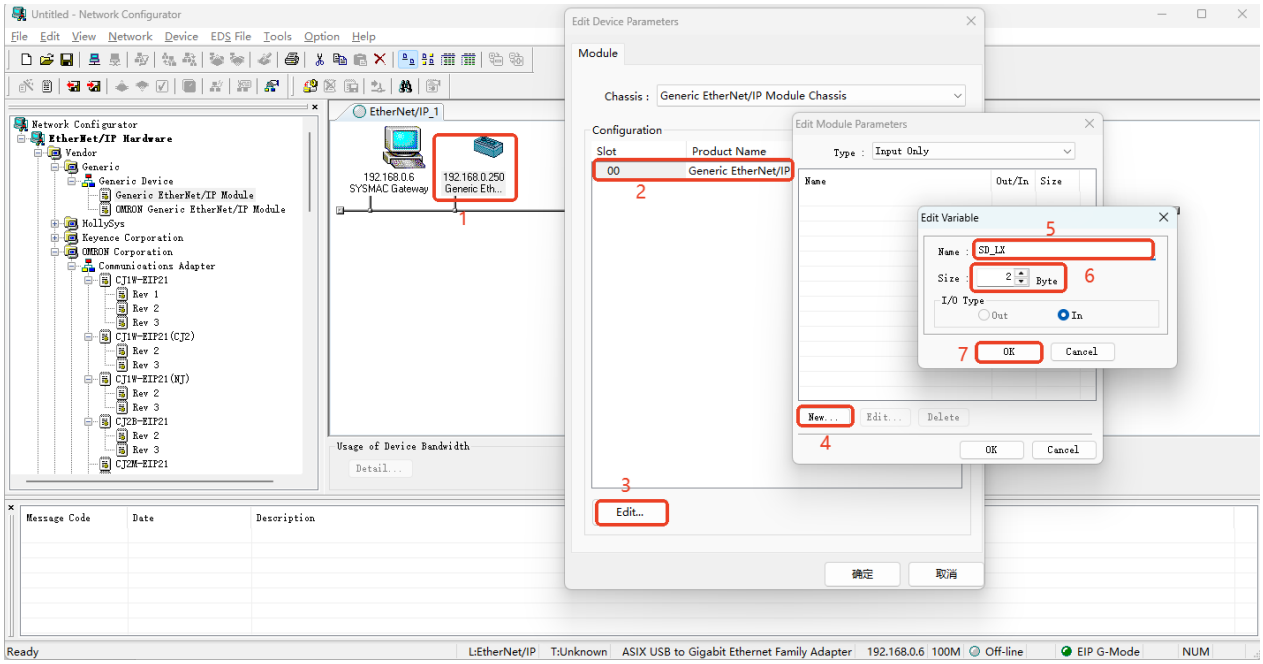
(5) Modify the IP addresses. For the SYSMAC Gateway software, change it to the computer's IP address. For the Generic EtherNet/IP Module, change it to the LX-CU430 controller's IP address. The method for modifying the IP addresses is as follows:



(6) Import the tags for the SYSMAC Gateway by loading the tags exported in section Create Tags into the SYSMAC Gateway. The operation steps are shown in the figure below.

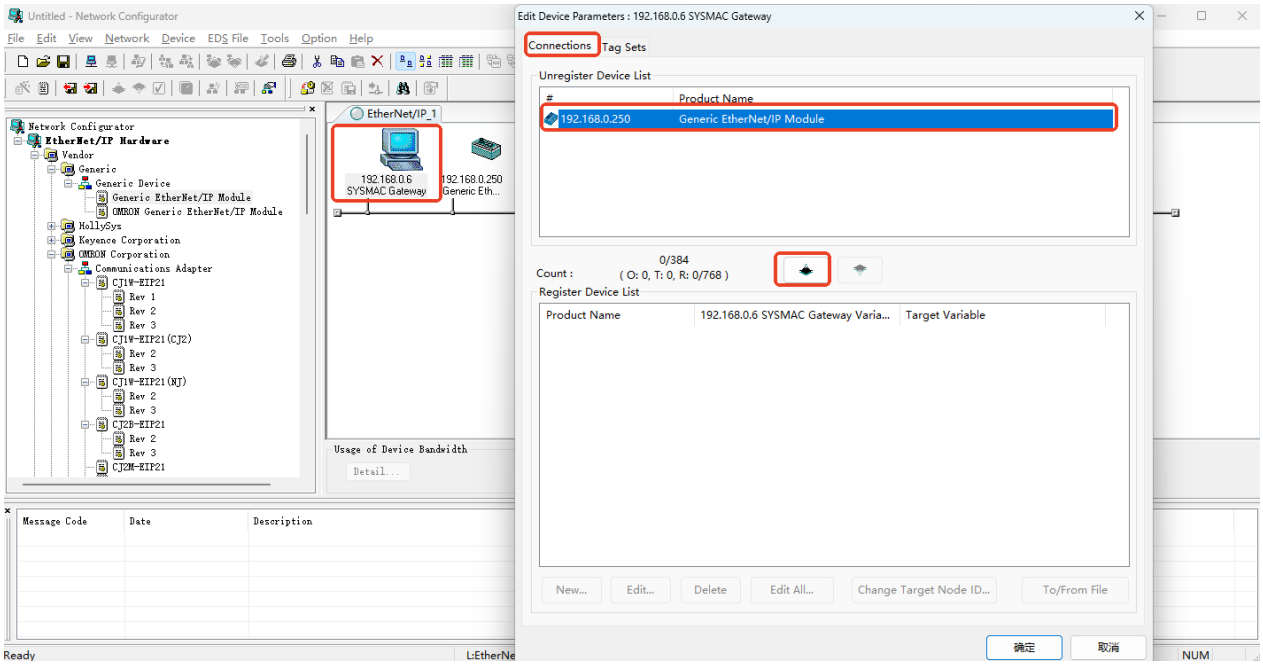


(7) Create output tags for the LX-CU430 controller. Since the input/output tags of the LX-CU430 controller have already been downloaded to the controller via AT, only the output tags need to be created here. The purpose is to associate the input tags with them in the SYSMAC Gateway connections. The method for creating the LX-CU430 controller's output tags is shown in the figure below.

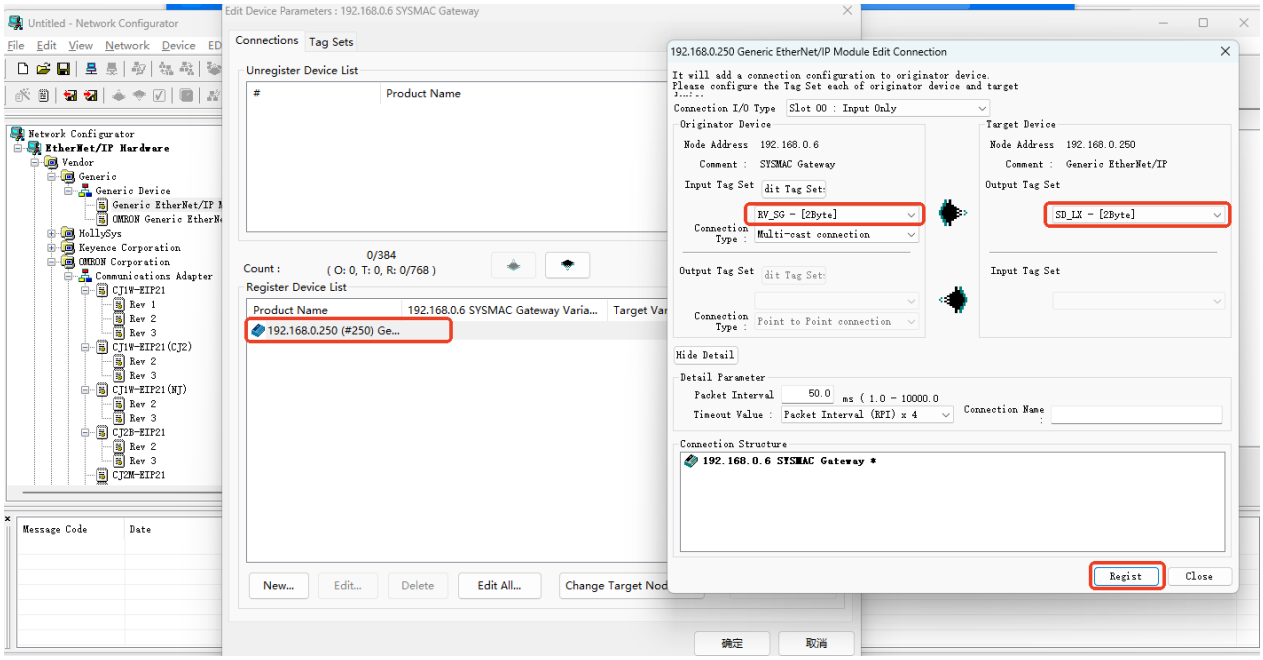


9.4.3.2.5 Create Connection

(1) Double-click the "SYSMAC Gateway" node and click "Connection" to establish a link.

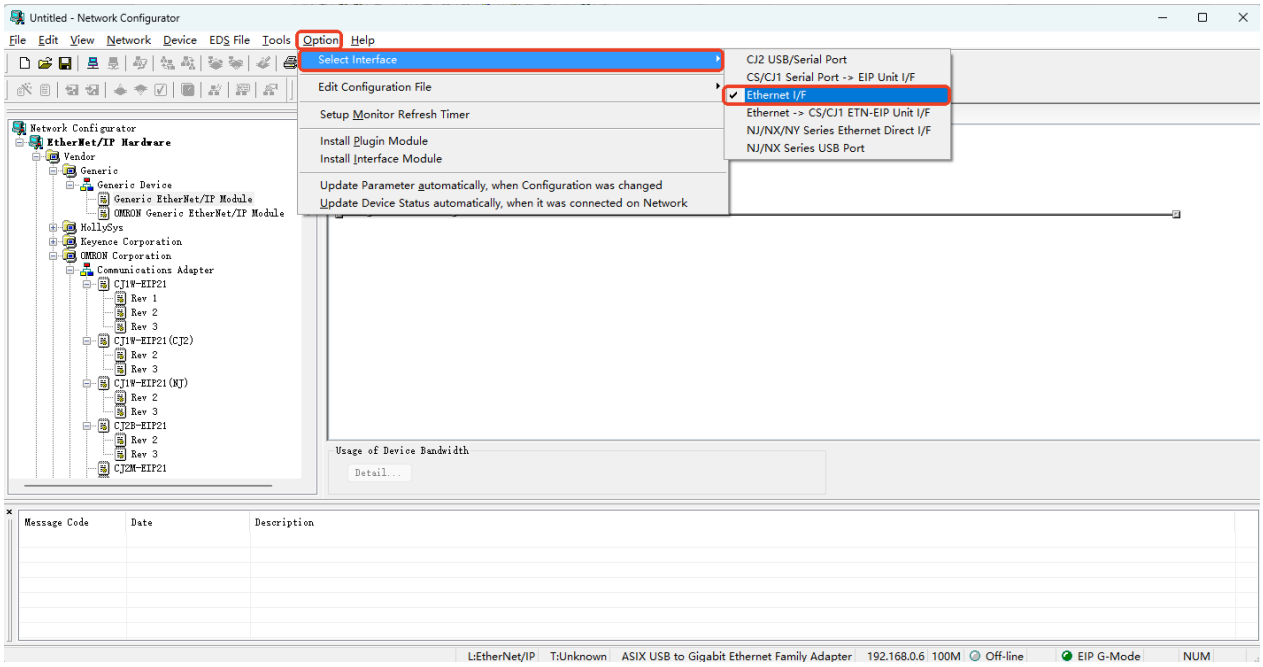


(2) Associate tags. Map the input tags of the SYSMAC Gateway to the output tags of the LX-CU430 controller.

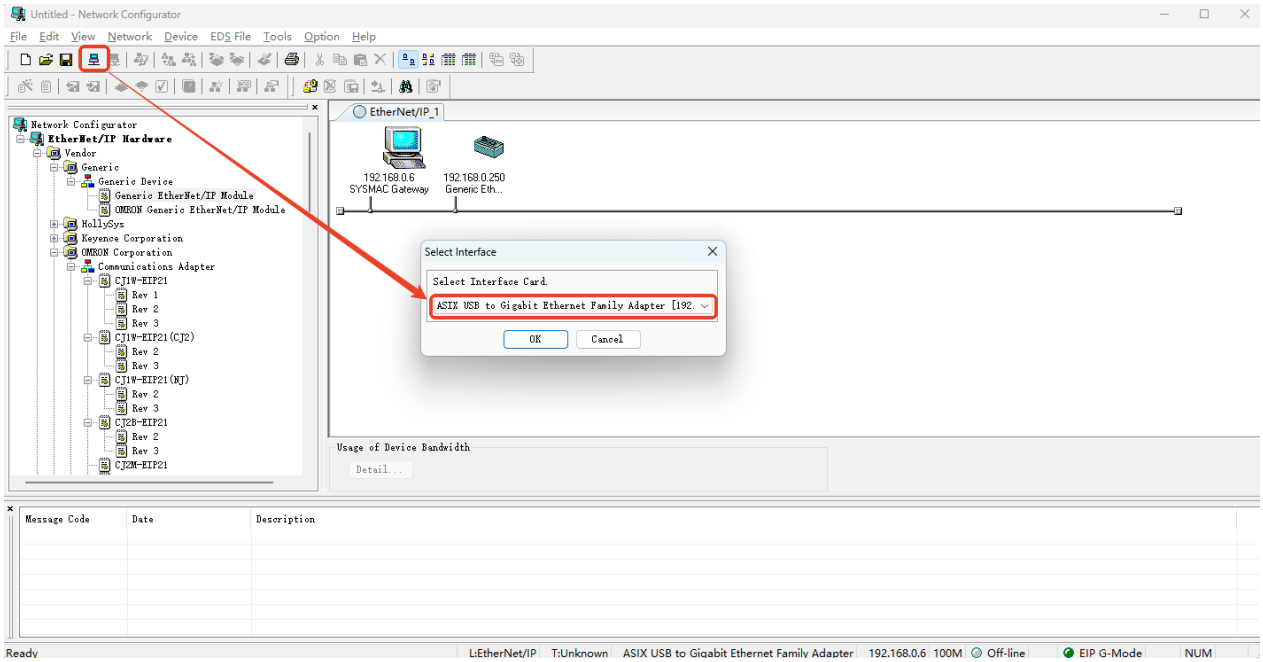


9.4.3.2.6 Download Network Configuration

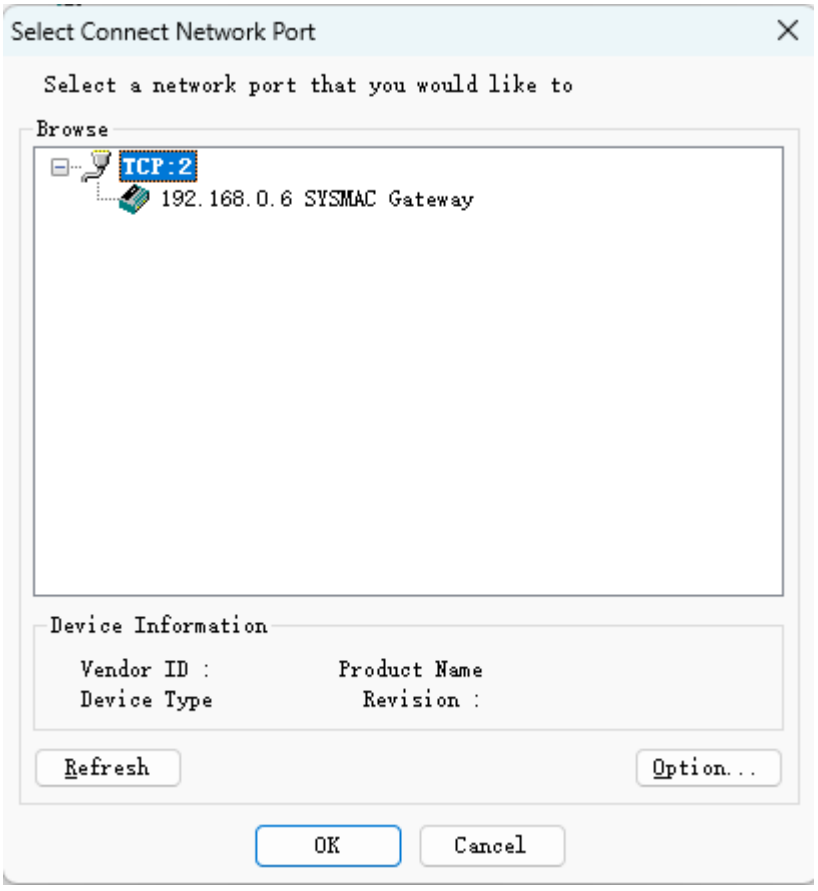
(1) Select the Interface and choose the Ethernet I/F mode.



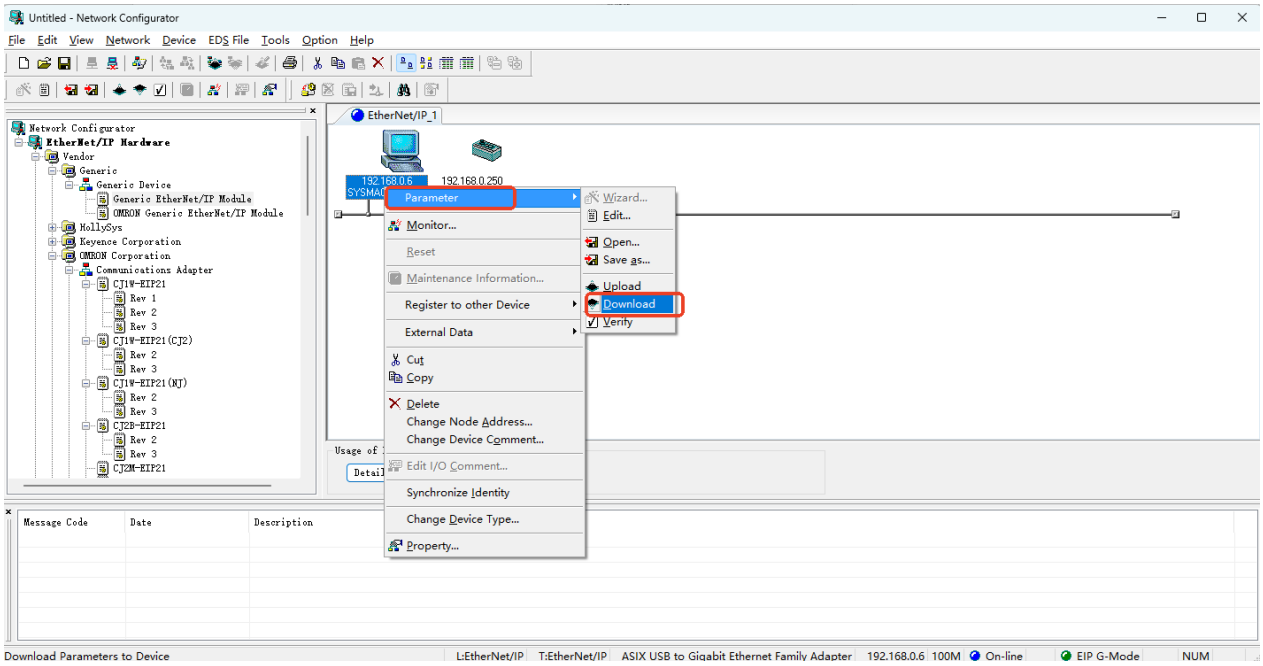
(2) Click "Connection" in the menu bar and select the computer's network adapter.



(3) After clicking "OK", the SYSMAC Gateway will be automatically scanned. If an Omron controller is present in the network, it will also be detected automatically.

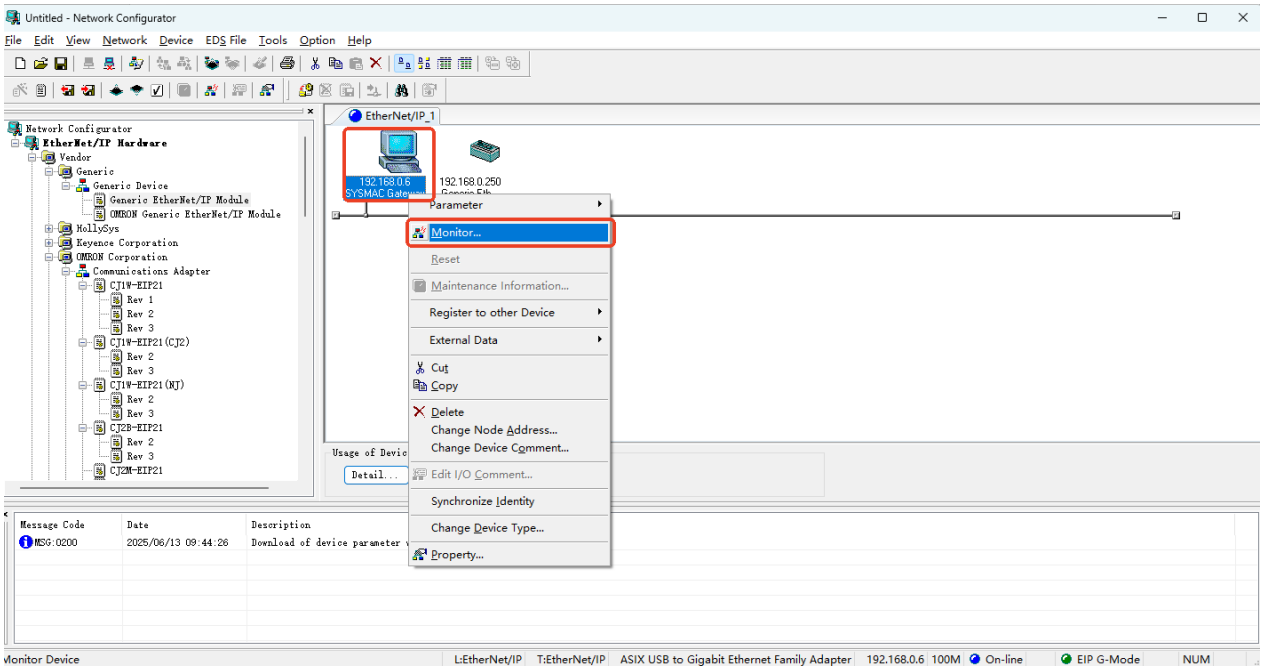


(4) Right-click on the "SYSMAC Gateway", select "Parameter" → "Download" to transfer the connection and tag configurations.

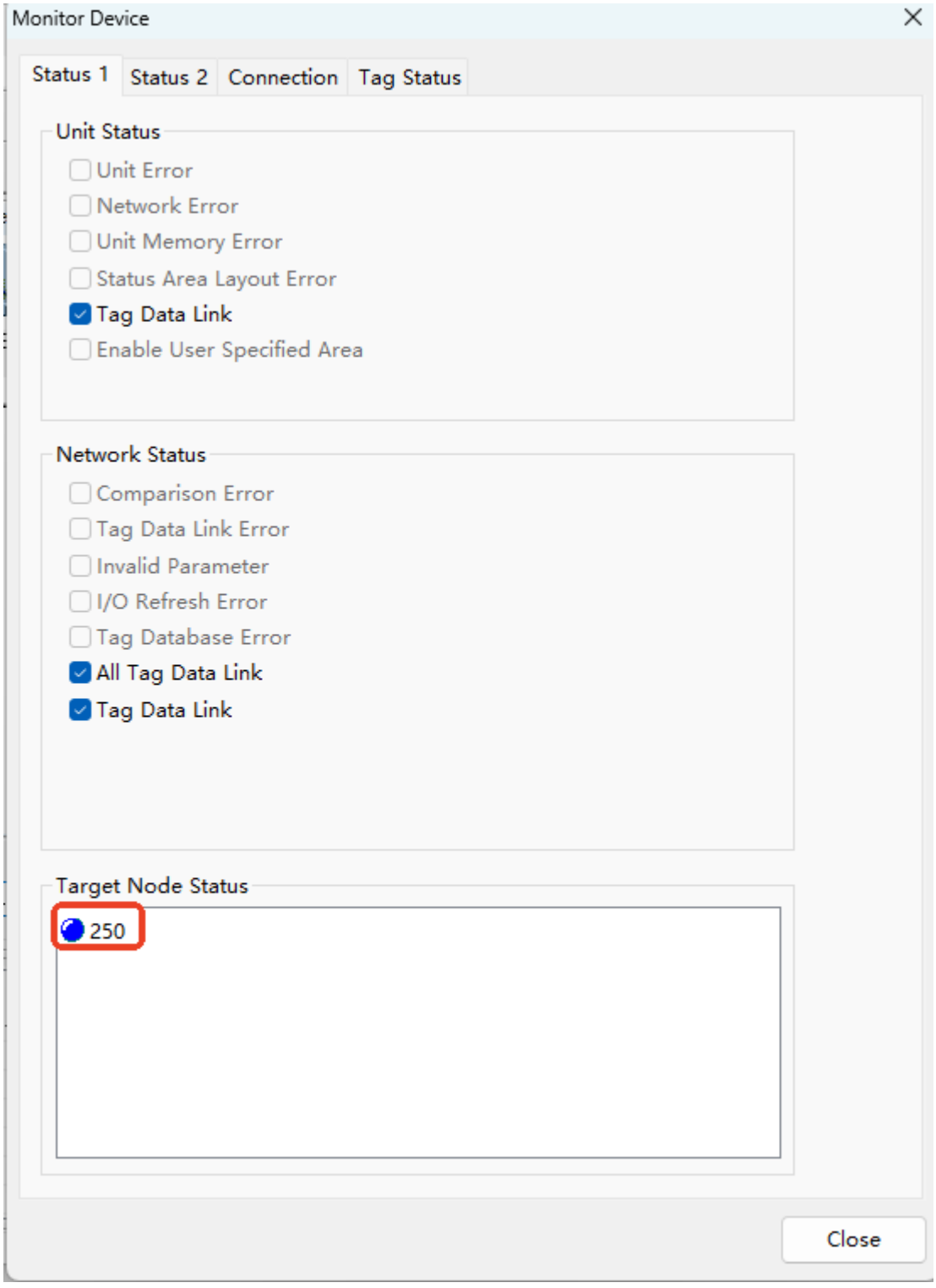


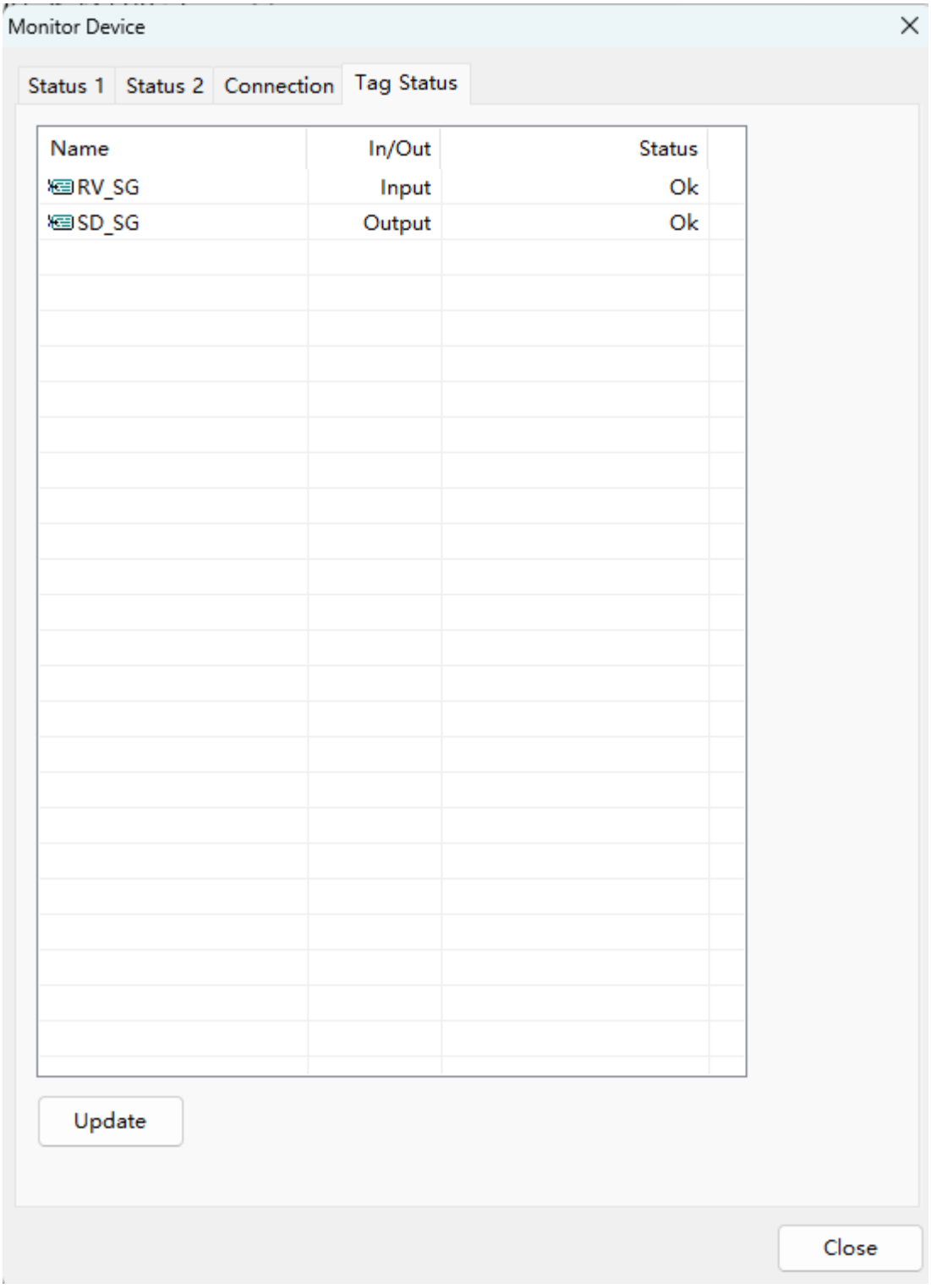
9.4.3.2.7 Diagnostics

(1) Right-click on "SYSMAC Gateway" and select "Monitor" to view diagnostic information for the connected target device.



(2) Blue indicates normal communication status. Connection status and tag status can both be checked in the diagnostic interface.

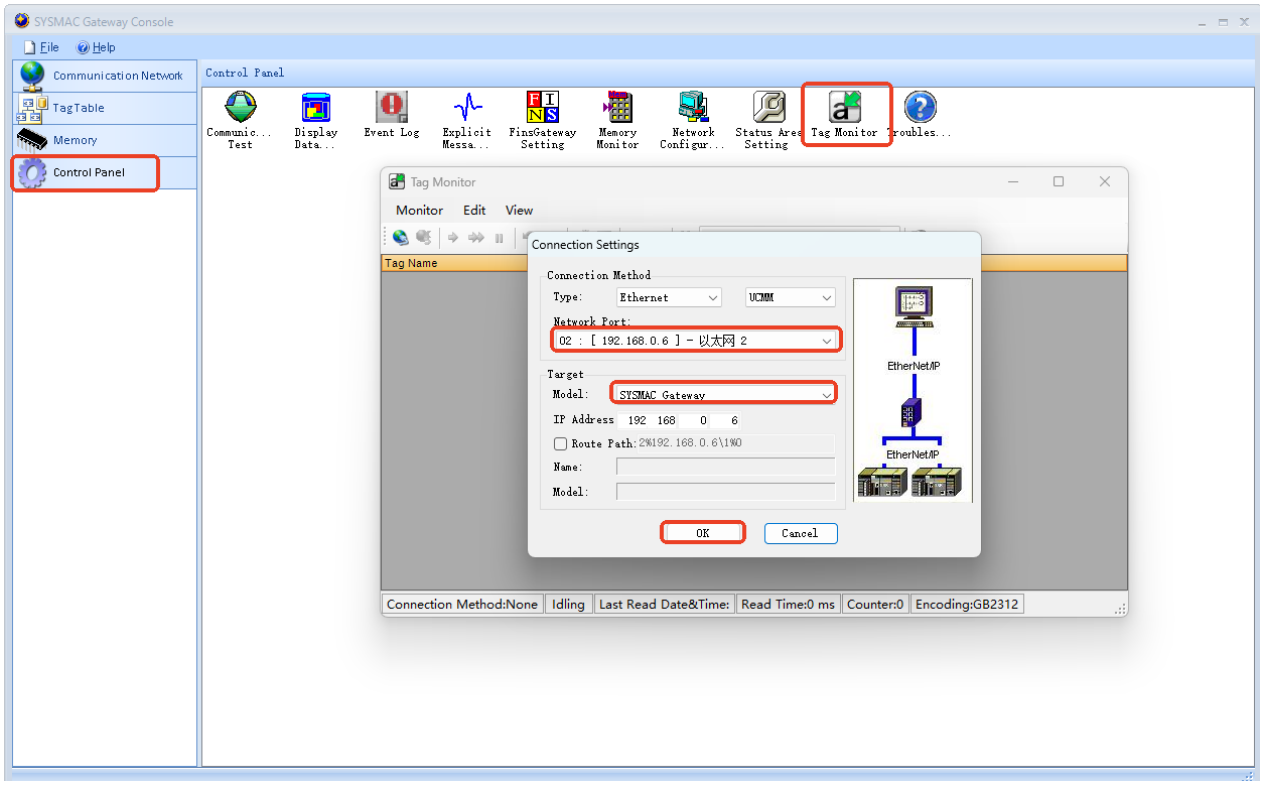

Monitor Device
Status 1
Status 2
Connection
Tag Status
Unit Status
Unit Error
Network Error
Unit Memory Error
Status Area Layout Error
Tag Data Link
Enable User Specified Area
Network Status
Comparison Error
Tag Data Link Error
Invalid Parameter
I/O Refresh Error
Tag Database Error
All Tag Data Link
Tag Data Link
Target Node Status
250
Close



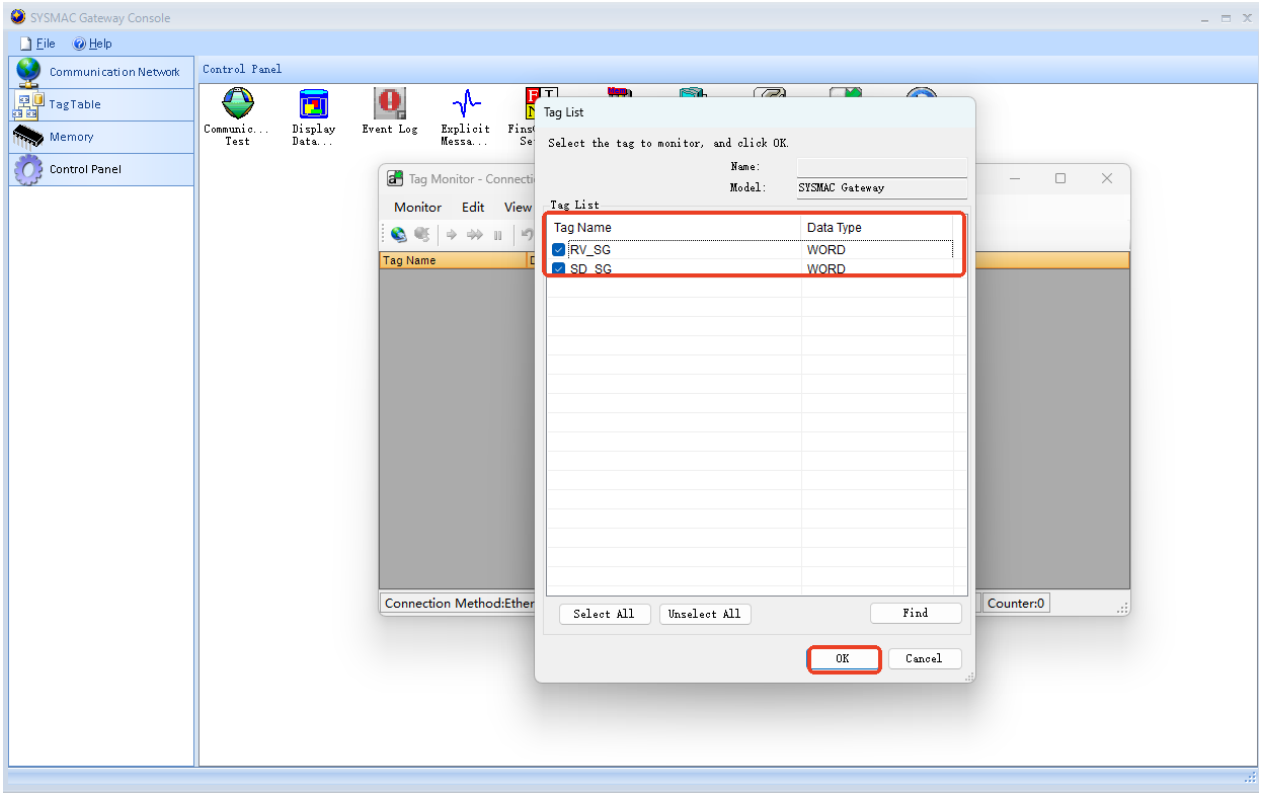
9.4.3.2.8 Data Read/Write

(1) Open the SYSMAC Gateway Console software, double-click the "Control Panel" menu, and the "Connection Settings" dialog box will pop up. Select the computer's network adapter and

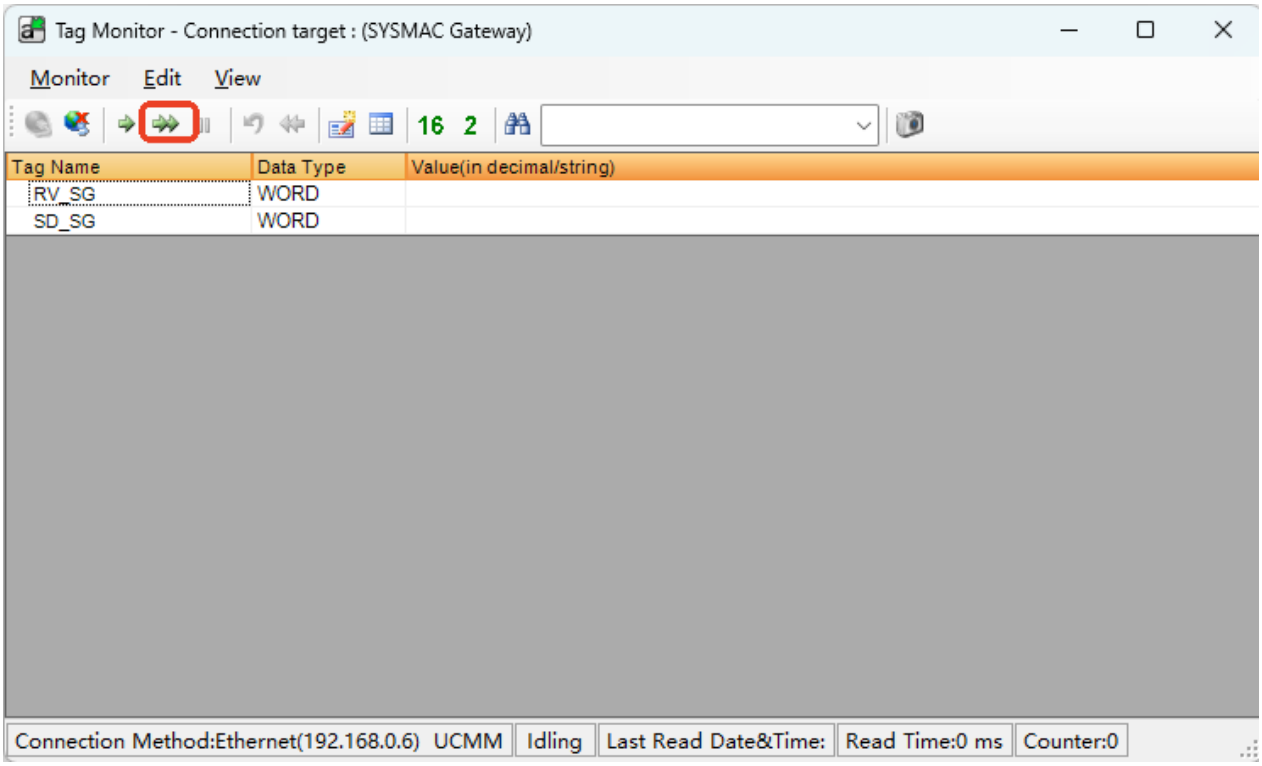
SYSMAC Gateway, then verify the IP address.



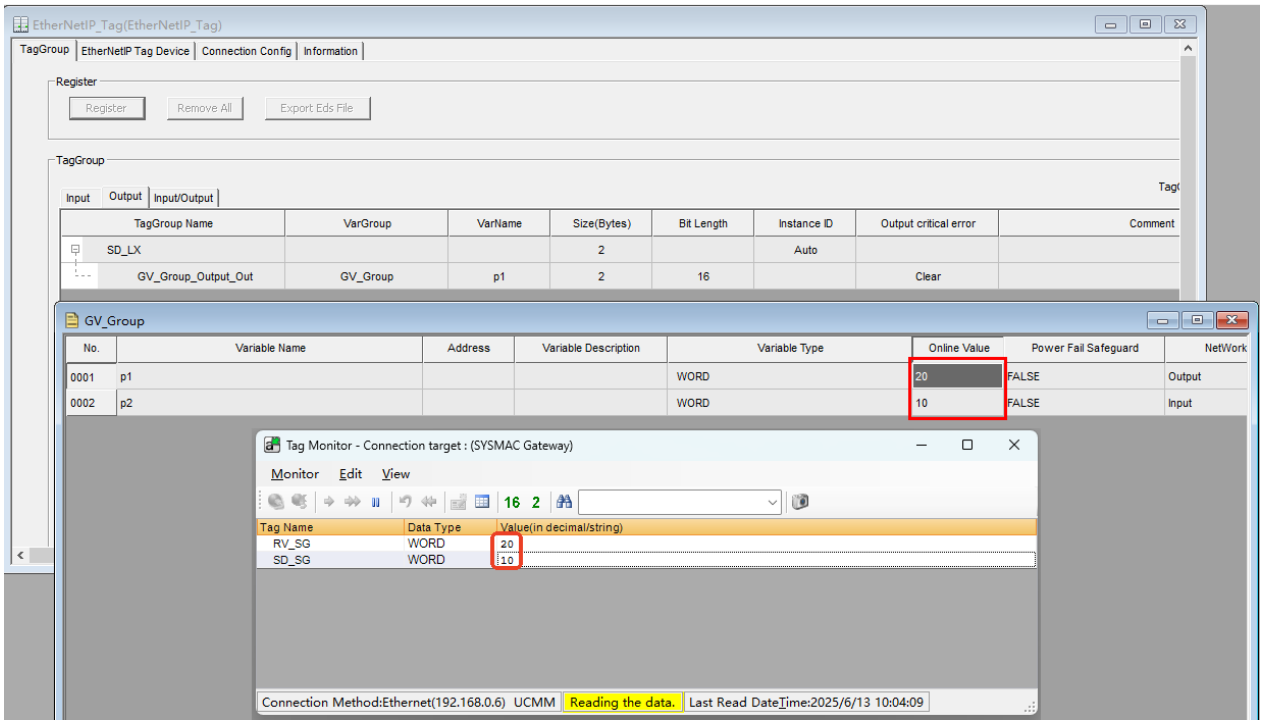
(2) The software will automatically scan the tags already created in the SYSMAC Gateway software. If the scanned tags do not match the created tags, save the created tags under the "Tag Table" tab. If the tags are correct, click "OK" to proceed.



(3) Click the icon , Read Tag in 1s intervals (as shown in the figure below) to read data every 1 second.

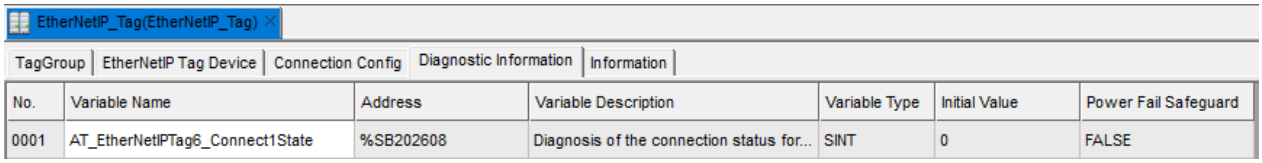


(4) In the Tag Monitor software, enter a value (e.g., 10) in the SD_SG output tag, enter a value (e.g., 20) in the SD_LX variable P1 (output tag in AT) for verification.



9.5 Fault Diagnosis

EtherNet/IP_Tag Hardware Configuration includes diagnostic information for connections, which can be viewed online.



The diagnostic descriptions are as follows:

Diagnostic Variable	Diagnostic value	Description of Values	Processing Method
AT_EtherNetIPTagN_ConnectNState	1	Normal, no diagnostic messages	-
	0	Initial state	-
	-3	Controller system memory is insufficient	■ In the device tree on the right side of the AT software, delete any communication protocols or slave devices that have been configured but are not currently in use. ■ Contact the supplier to replace the controller model.
	-4	Connection parameter error, adapter device rejected the connection	■ Predefined connection: Re-export the EDS file of the target device and import it into the device library. Delete the old device, then re-add the device. After performing a full compilation and download, check the connection status again. ■ Confirm the connection parameters:

			☐ Verify that the target tag group name and byte size on the input side are consistent with the tag name and T->O byte size on the output side. ☐ Verify that the transmission type on the output side is set to Input Only. ■ Power cycle the target device and check the connection status again. ■ Confirm whether the target device's IP address is valid. ■ If the above measures are ineffective, contact the equipment manufacturer.
	-5	Adapter device IP address is invalid	Check IP input: In the device tree on the right side of the AT software, click on the "Target Device" page under EtherNetIP_Tag, and check whether the configured IP address is an invalid IP such as 255.255.255.255.
	-7	Adapter device connection timeout	■ Check the target device IP address: Click on the "Target Device" page under EtherNetIP_Tag and verify whether the configured IP address matches the actual IP address of the physical device. ■ Network connection: Check whether the network connection between the controller and the target device is normal. Use the ping command to test connectivity. ■ Check whether the controller module and the target device are in normal working condition. ■ Check the configuration of switches or routers in the topology. Replace with an unmanaged switch and try again. ■ Use a network scanning tool to check whether the TCP/UDP port (default: 44818) of the target device is open. ■ Power cycle the controller and check whether the target device comes back online. ■ Adjust the RPI or timeout settings of the current connection. Increase the RPI time or the timeout multiplier (default *4) to a larger value. ■ Test by modifying the connection type of the current connection (test point-to-point and multicast separately). ■ In the AT engineering menu, select "Full Compilation" to re-download the user project, and check whether the target device comes back online. ■ Reduce the connection scale on the controller side. Remove other functioning connections and retain only the current connection for testing. ■ If the above measures are ineffective, contact the equipment manufacturer.
	-8	Internal error	Continuously observe whether the fault automatically recovers. If it does not: ■ Perform a full compilation of the project and then download it again. ■ Check whether the controller's own IP settings are correct.

			■ If the above measures are ineffective, please contact the equipment manufacturer.

Chapter 10 DeviceNet Network Communication

10.1 Network Overview

10.1.1 Introduction

LX-CM006/ LX-CM106 is a DeviceNet master station protocol module that communicates with other DeviceNet slave devices (slave stations, instruments, sensors, etc.) on site through the CAN interface. It supports up to 63 slave stations and supports Poll services.

LX-CM005 is a DeviceNet slave station protocol module that communicates with the DeviceNet master station through the CAN interface.

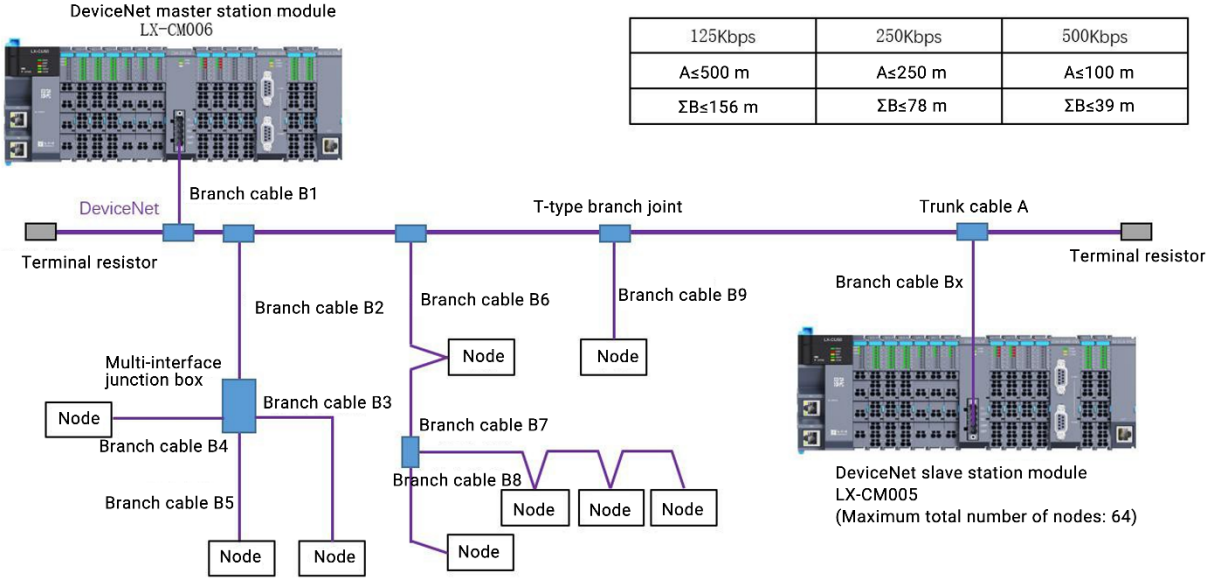
10.1.2 Network Characteristics

- Based on the Controller Area Network (CAN), it follows the physical layer and data link layer specified by the CAN protocol, and the application layer protocol uses the Common Industrial Protocol (CIP).
- There can be up to 64 nodes in a single network, with a range of node address (called MAC IDs in DeviceNet) from 0 to 63.
- It supports point-to-point, multi-master or master/slave communication.
- It supports four data transmission methods: bit-strobed, polling, status change and periodic.
- A terminal resistor (120 ohms, 1/4 W) is required.

10.2 Network Topology

The DeviceNet network topology is very flexible and can support trunk-branch topology, tree topology, daisy chain topology, etc., or a combination of them.

The figure below shows the DeviceNet composite network structure. In actual application, attention must be paid to the selection of relevant components and the length limitations of communication cables. Reasonable network planning must be carried out in strict accordance with the requirements of the table in the figure.



125Kbps	250Kbps	500Kbps
A≤500 m	A≤250 m	A≤100 m
ΣB≤156 m	ΣB≤78 m	ΣB≤39 m

Requirements for communication cables:

Cable Model	Specifications	Communication Distance
3082A trunk cable	Power cable specifications: 22AWG(0.34mm^2) Data cable specifications: 24AWG(0.22mm^2)	When the rate is 125 Kbps, the maximum network distance is 500 meters. When the rate is 250 Kbps, the maximum network distance is 250 meters. At the highest rate of 500 Kbps, the maximum network distance can reach 100 meters.
3084A branch cable	Power cable specifications: 15AWG(1.65mm^2) Data cable specifications: 18AWG(0.83mm^2)	The maximum distance of a single branch cable is 6 meters. When the rate is 125 Kbps, the maximum cumulative distance of all branch cables is 156 meters. When the rate is 250 Kbps, the maximum cumulative distance of all branch cables is 78 meters. When the rate is 500 Kbps, the maximum cumulative distance of all branch cables is 39 meters.

10.3 Configuration

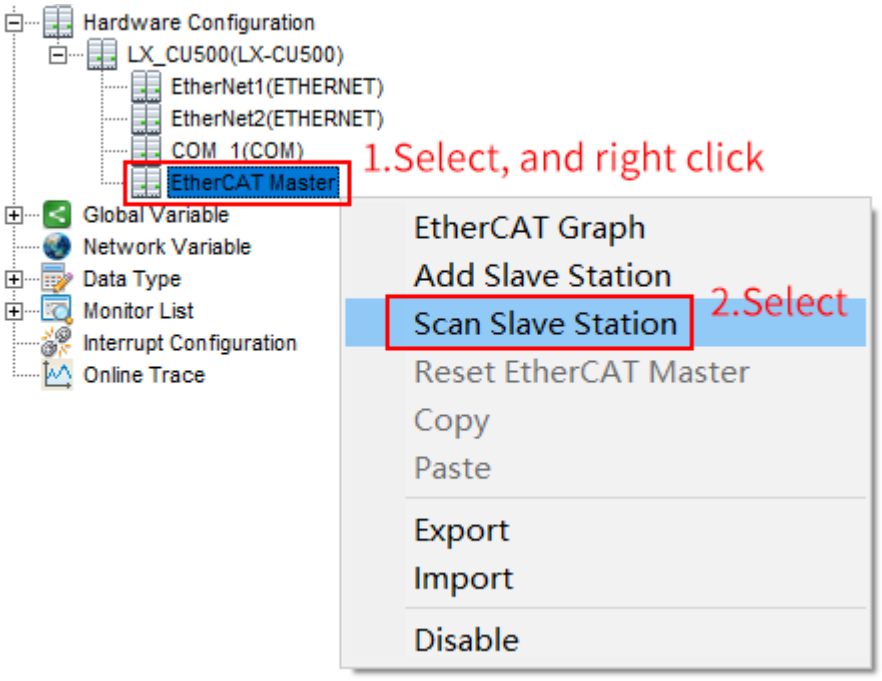
10.3.1 Configuring DeviceNet Master Station

Taking LX-CM106 as an example.

10.3.1.1 Adding LX-CM106 module

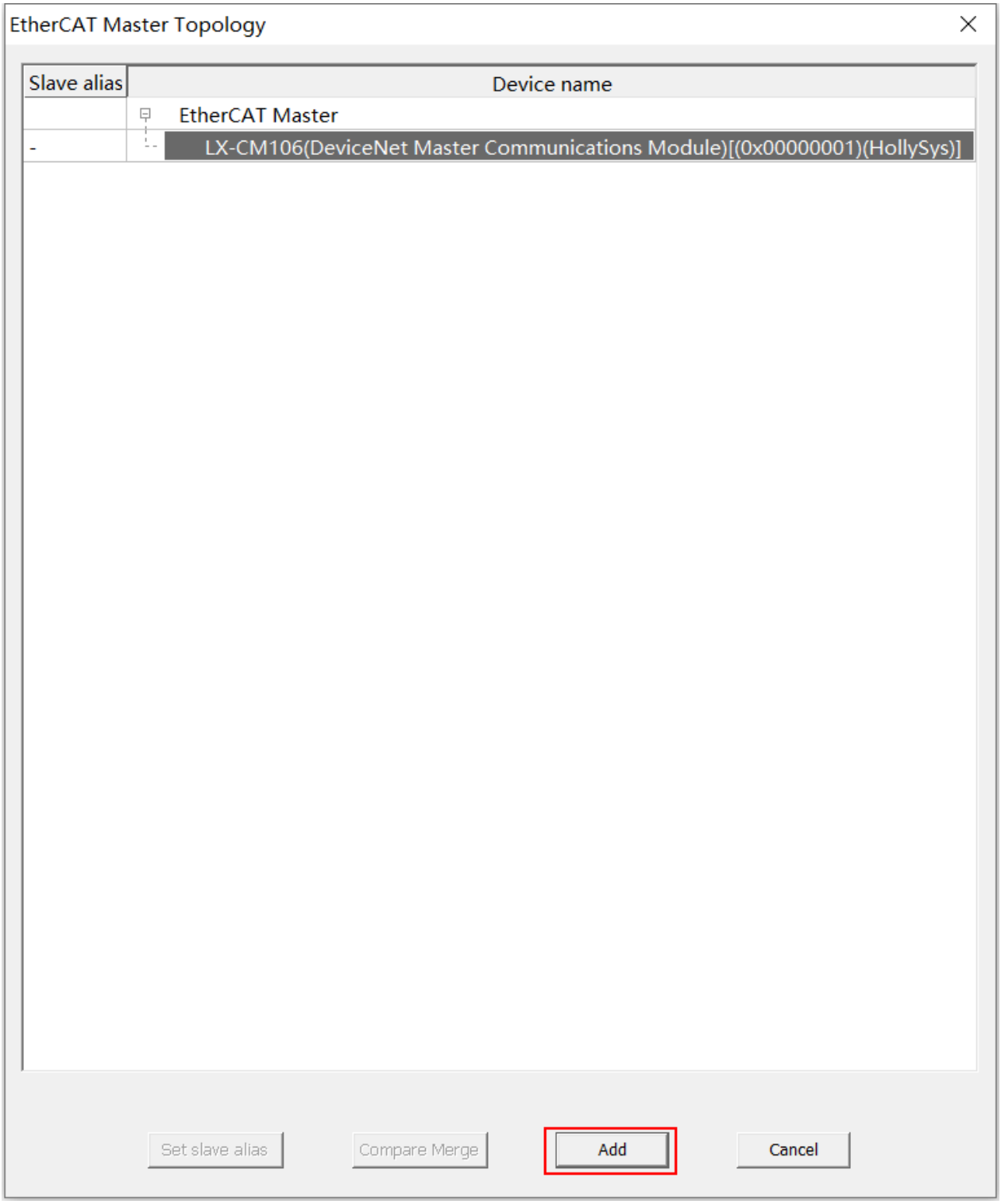
- Method I

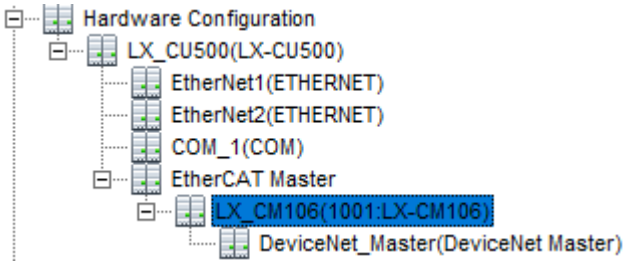
(1) Right-click the EtherCAT Master node and select Scan Slave Station.



The Scan Slave Station dialog box is displayed as shown in the figure.

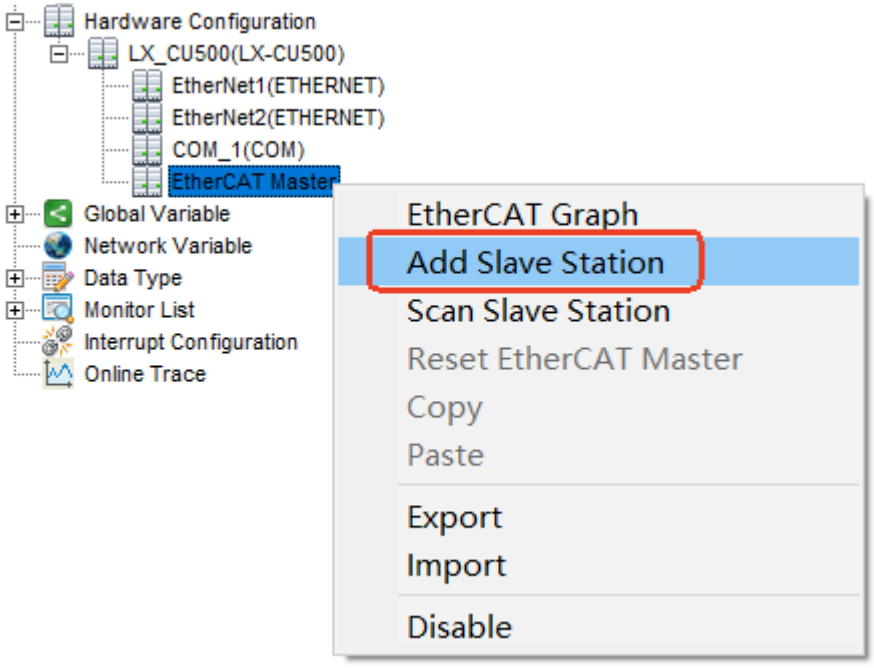
(2) Select "Start Scan" and enter the controller username and password. After the scan is completed, the "EtherCAT Master Topology" dialog box will pop up. Under the EtherCAT Master node, the LX-CM106 module is scanned. Select the "Add" button, and the module along with the DeviceNet master station will be displayed under the EtherCAT Master node.


EtherCAT Master Topology
Slave alias
Device name
EtherCAT Master
-
LX-CM106(DeviceNet Master Communications Module)[(0x00000001)(HollySys)]
Set slave alias
Compare Merge
Add
Cancel

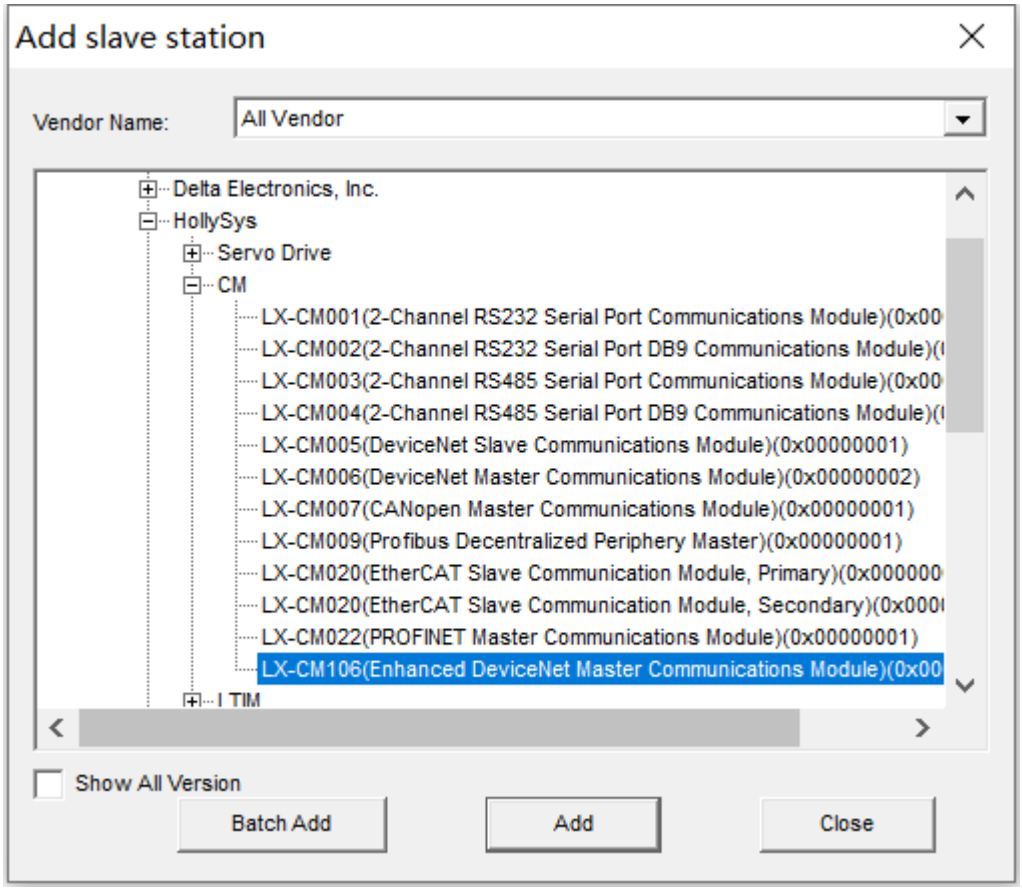


■ Method II

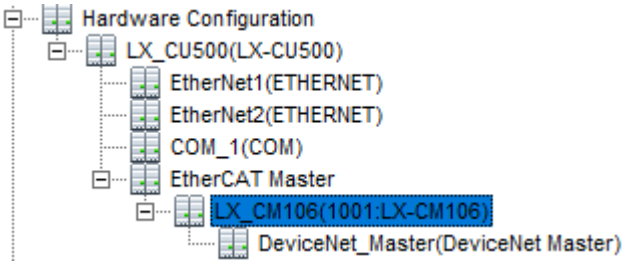
(1) Right-click the EtherCAT Master node and select Add Slave Station.



The Add Slave Station dialog box is displayed as shown in the figure.



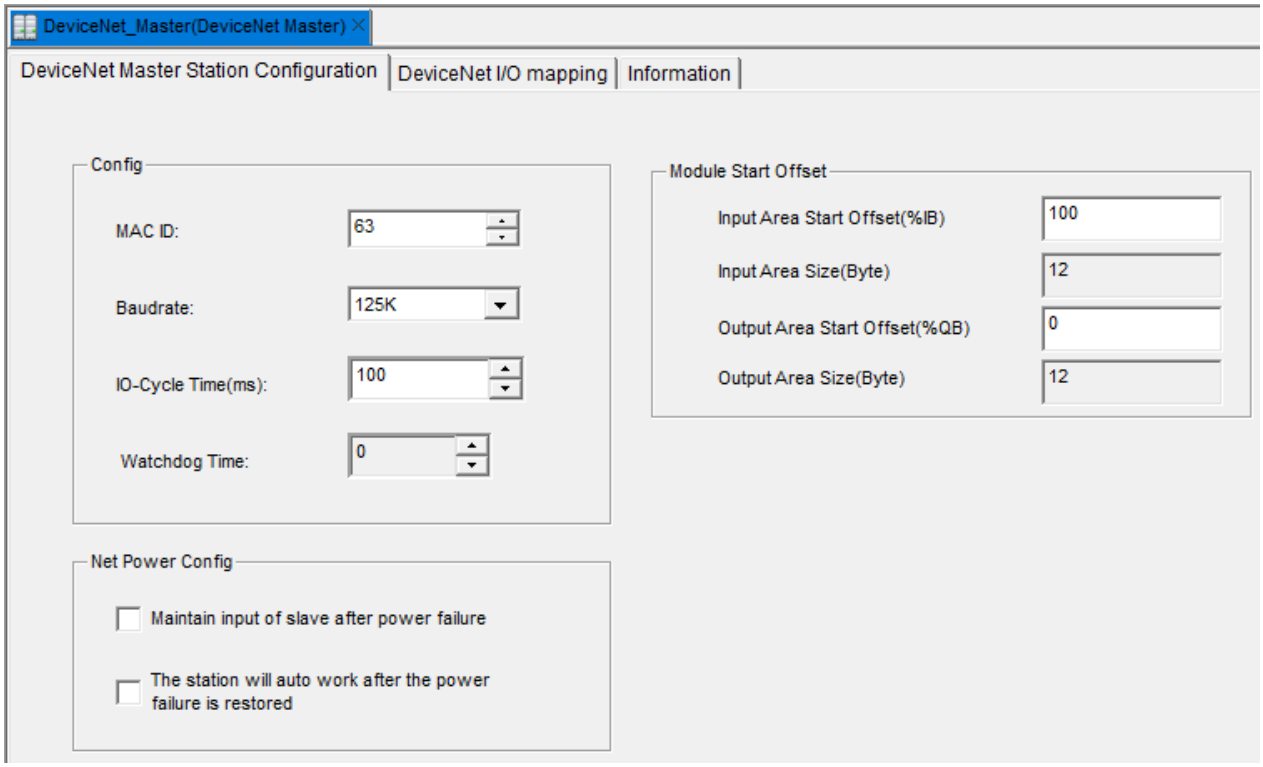
(2) Select LX-CM106, click the Add button, the EtherCAT Master node displays modules and the DeviceNet master station.



10.3.1.2 Configuring DeviceNet Master Station

Double-click the added DeviceNet_Master module tree node to configure the master station parameters.

The configuration interface includes DeviceNet Master Station Configuration, DeviceNet I/O Mapping and Information. as shown in the figure:



■ Information description for the 'DeviceNet Master Station Configuration tab:

□ Configuration: The master station MAC ID, Baud Rate, and IO-Cycle Time (ms) can be modified.

Configuration Parameters	Function Description	Parameter Range	Configuration Description
MAC ID	ID identification	0~63	Each node in the DN network must be assigned with a unique ID identification
Baud Rate	Set communication rate	125K 250K 500K	Configure baud rate based on end-to-end network distance 500 m, baud rate 125 Kbp/s 250 m, baud rate 250 Kbp/s 100 m, baud rate 500 Kbp/s
IO-cycle time (ms)	Communication cycle time	1~65535	Default values used when adding slave stations
Watchdog time		0~100000	Not used, no configuration required

□ Net Power Config (configuration not available for LX-CM006)

□ Maintain input of slave after power failure: Unchecked by default. If checked, slave inputs will be retained in case of network power failure.

□ The station will auto work after the power failure is restored: Unchecked by default. If checked, the master station will automatically resume operation upon network power recovery.

□ Module Start Offset

□ Input Area Start Offset (%IB): In offline mode, the starting address of the DeviceNet I/O mapping input channels can be modified.

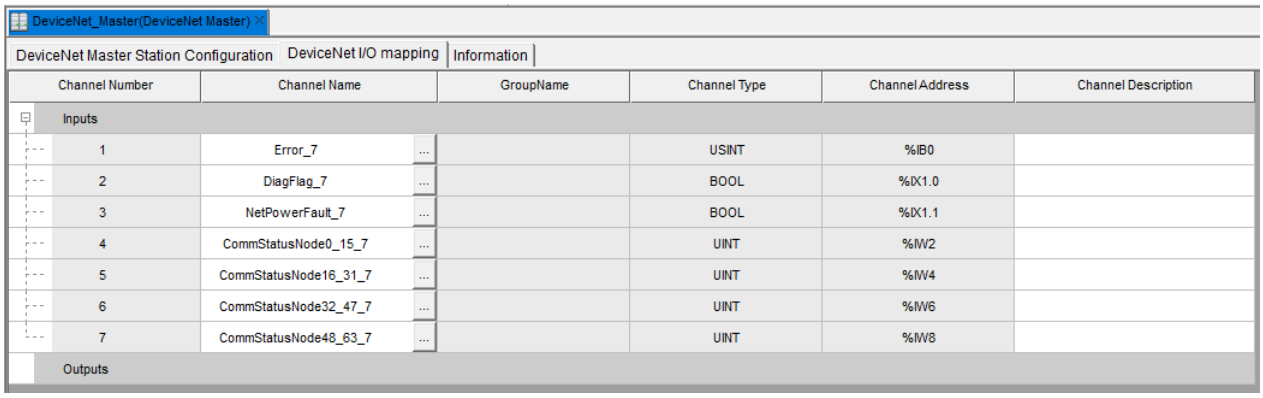
□ Input Area Size (Byte): The number of bytes in the input area.

□ Output Area Start Offset (%QB): In offline mode, the starting address of the DeviceNet I/O

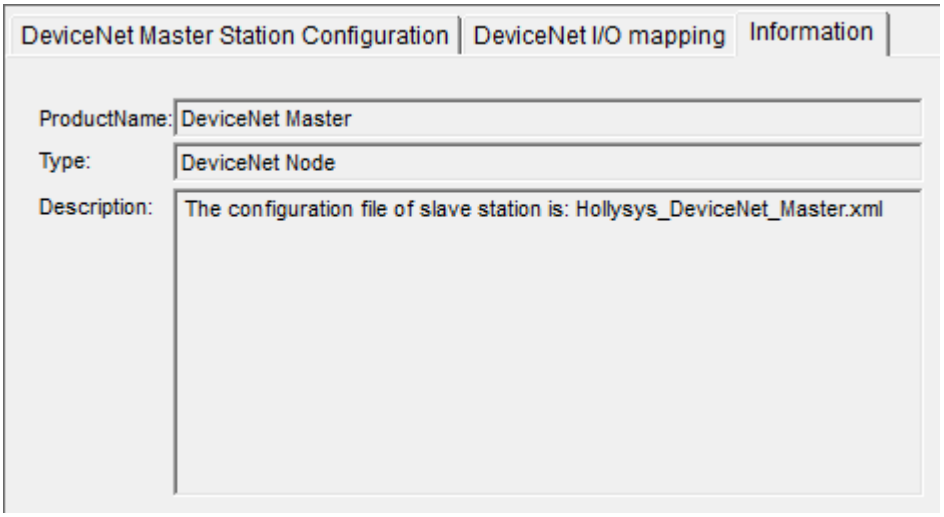
mapping output channels can be modified.

- Output Area Size (Byte): The number of bytes in the output area.

- The diagnostic information of the master station status reported in the "DeviceNet I/O Mapping" is shown in the figure below. For detailed instructions, please refer to the DeviceNet master module diagnostic.

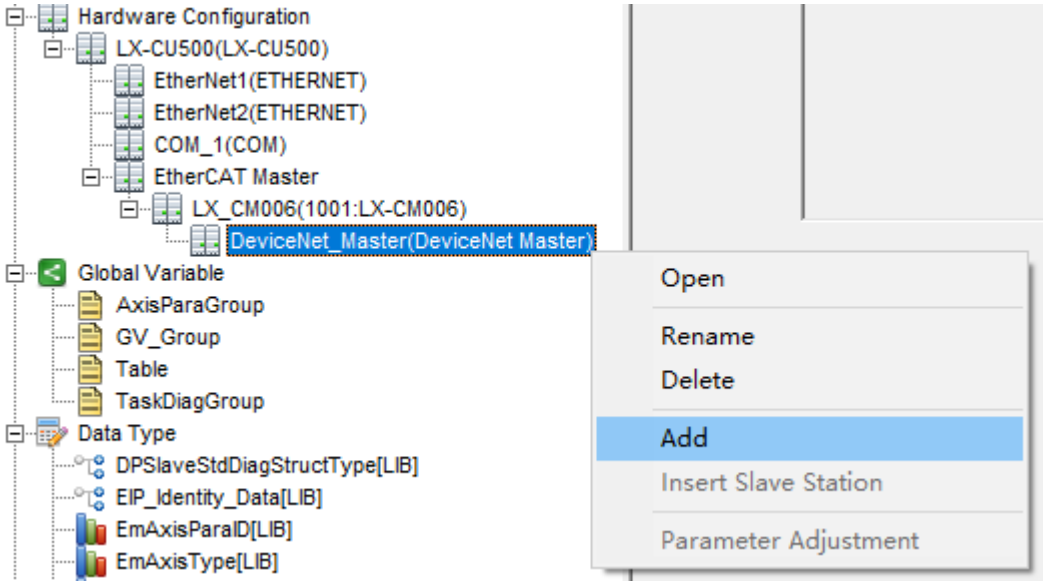


- The Information interface displays the product information of the current master station module.

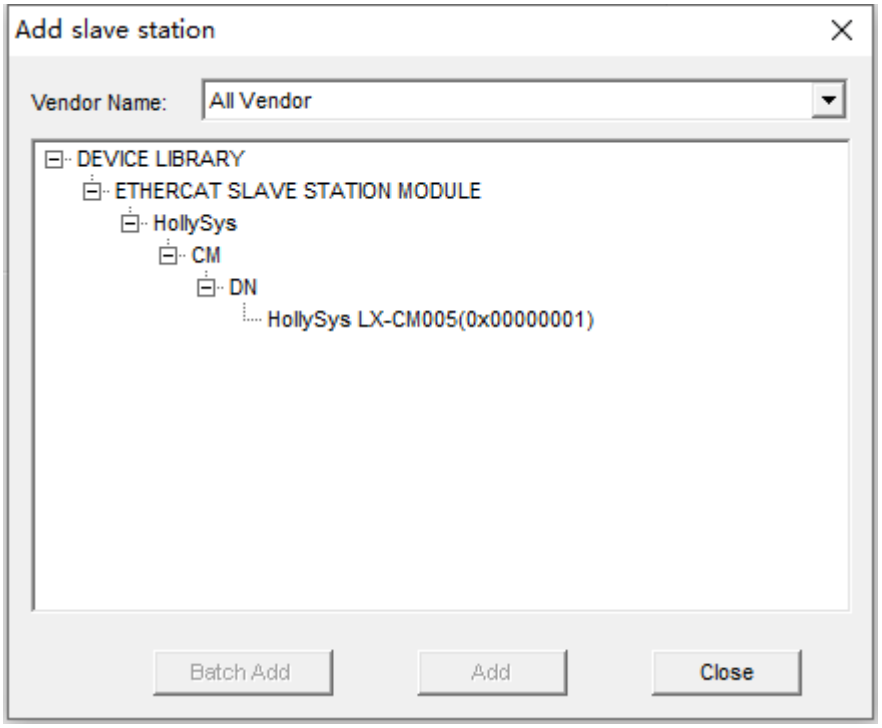


10.3.1.3 Adding DeviceNet_Slave station

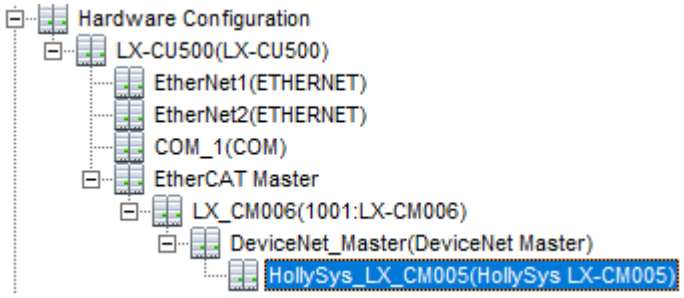
(1) Right-click the DeviceNet_Master node and select the Add command.



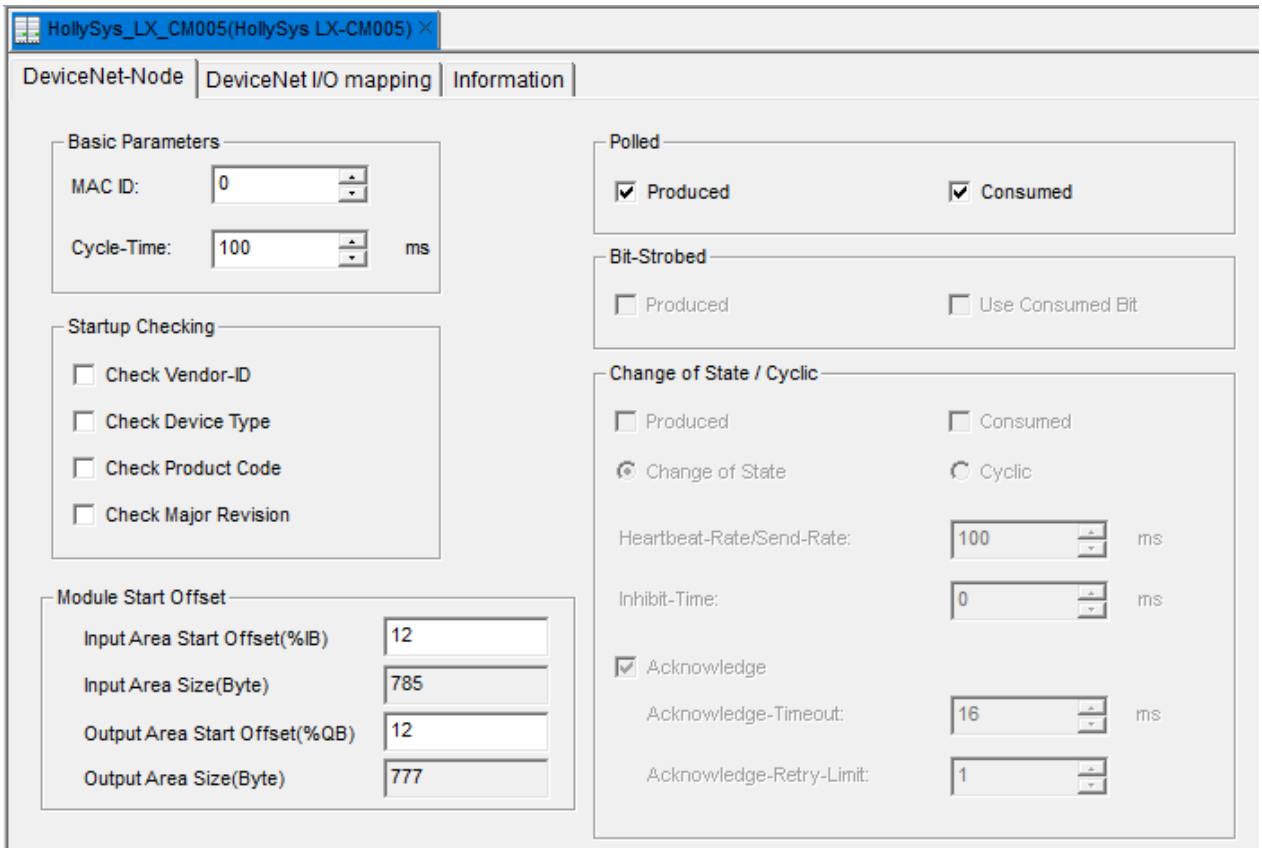
The Add Slave Station dialog box is displayed.



(2) Select the required slave module and click the Add button. If there are multiple slave stations, add multiple times and select the corresponding DeviceNet slave station.



(3) Double-click the added HollySys_LX_CM005 module tree node, and the configuration interface is displayed.



The configuration interface includes DeviceNet Node, DeviceNet I/O Mapping and Information.

- In the DeviceNet Node tab, the slave station MAC ID and Cycle Time can be modified.

Configuration Parameters	Function Description	Parameter Range	Configuration Description
MAC ID	Slave station ID identification	0~63	Each slave station is assigned a unique ID identification
Cycle Time	Polling cycle	1~65535	Poll service polling cycle
Startup check	Whether to check supplier ID, device type, product code, and version number	Multiple choice	Whether to check the supplier ID, device type, product code, and version number of the slave station. When these options are checked, if they do not match, the slave station cannot go online.
Polling	Production		Configure network model: producer
	Consumption		Configure network model: consumer
bit-strobed			Not support

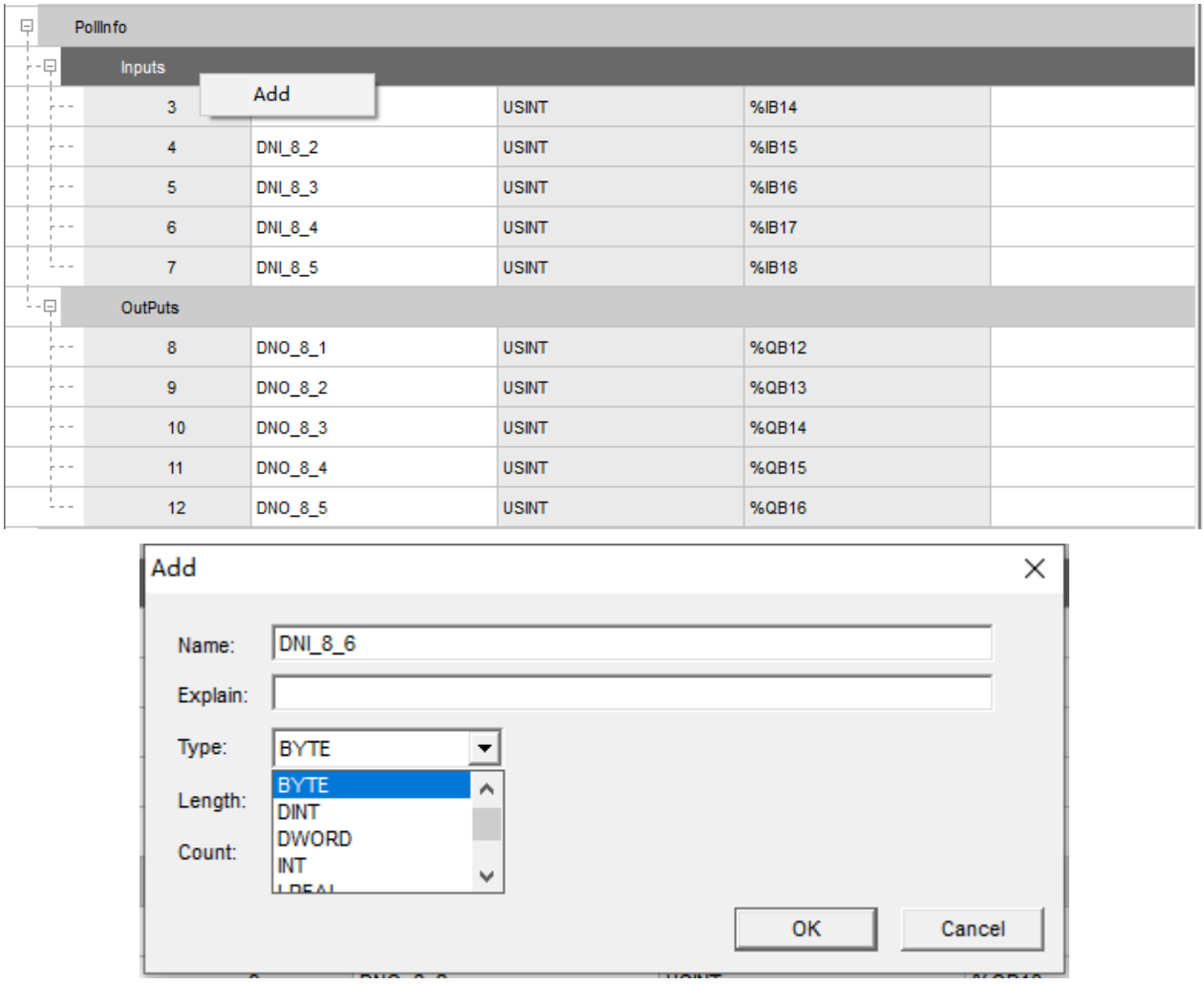
Status change/cycle change			Not support
Input Area Start Offset (%IB)	In offline mode, the starting address of the DeviceNet I/O mapping input channels can be modified.		
Input Area Size (Byte)	The number of bytes in the input area.		
Output Area Start Offset (%QB)	In offline mode, the starting address of the DeviceNet I/O mapping output channels can be modified.		
Output Area Size (Byte)	The number of bytes in the output area.		

- The DeviceNet I/O Mapping interface includes input/output data, polling information, bit-strobed information, status change information, and cycle information. Among them, the information in red (bit-strobed information, status change information and cycle information) is not supported yet.

DeviceNet-Node | DeviceNet I/O mapping | Information

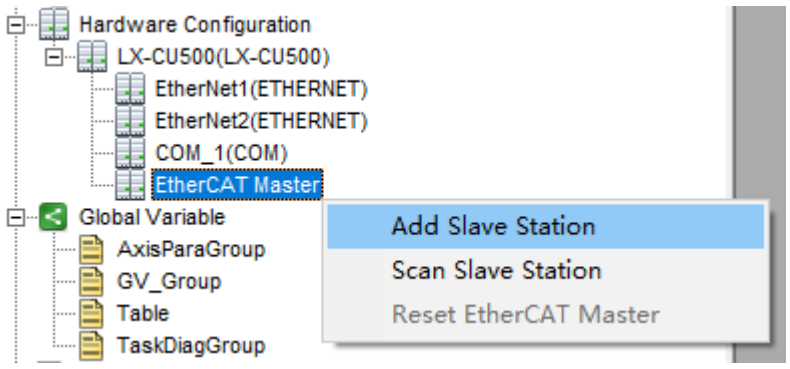
Channel Number	Channel Name	Channel Type	Channel Address	Channel Descrip
Inputs				
1	MacState_8	USINT	%IB12	
2	DiagFlag_8	USINT	%IB13	
OutPuts				
PollInfo				
Inputs				
3	DNI_8_1	USINT	%IB14	
4	DNI_8_2	USINT	%IB15	
5	DNI_8_3	USINT	%IB16	
6	DNI_8_4	USINT	%IB17	
7	DNI_8_5	USINT	%IB18	
OutPuts				
8	DNO_8_1	USINT	%QB12	
9	DNO_8_2	USINT	%QB13	
10	DNO_8_3	USINT	%QB14	
11	DNO_8_4	USINT	%QB15	
12	DNO_8_5	USINT	%QB16	
StrobeInfo				
COSInfo				
CyclicInfo				

Different data types can be added for each communication type (input or output), and the total length cannot exceed 255 bytes.

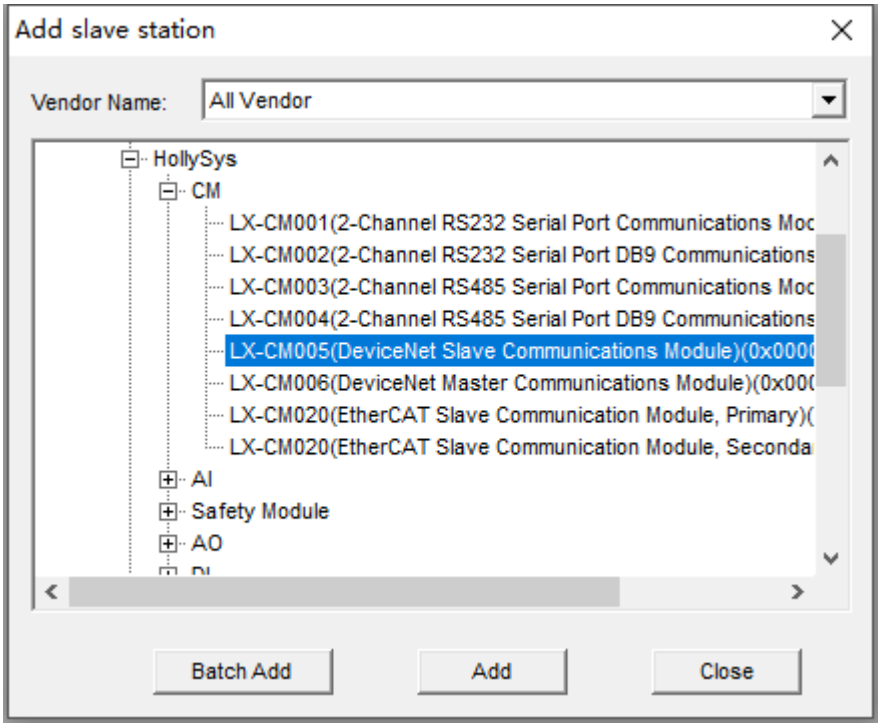


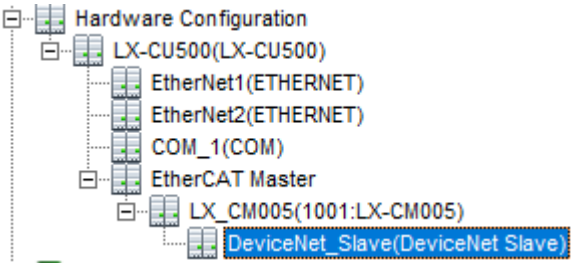
10.3.2 Configuring DeviceNet Slave Station

(1) Right-click the EtherCAT Master node and select Add Slave Station.

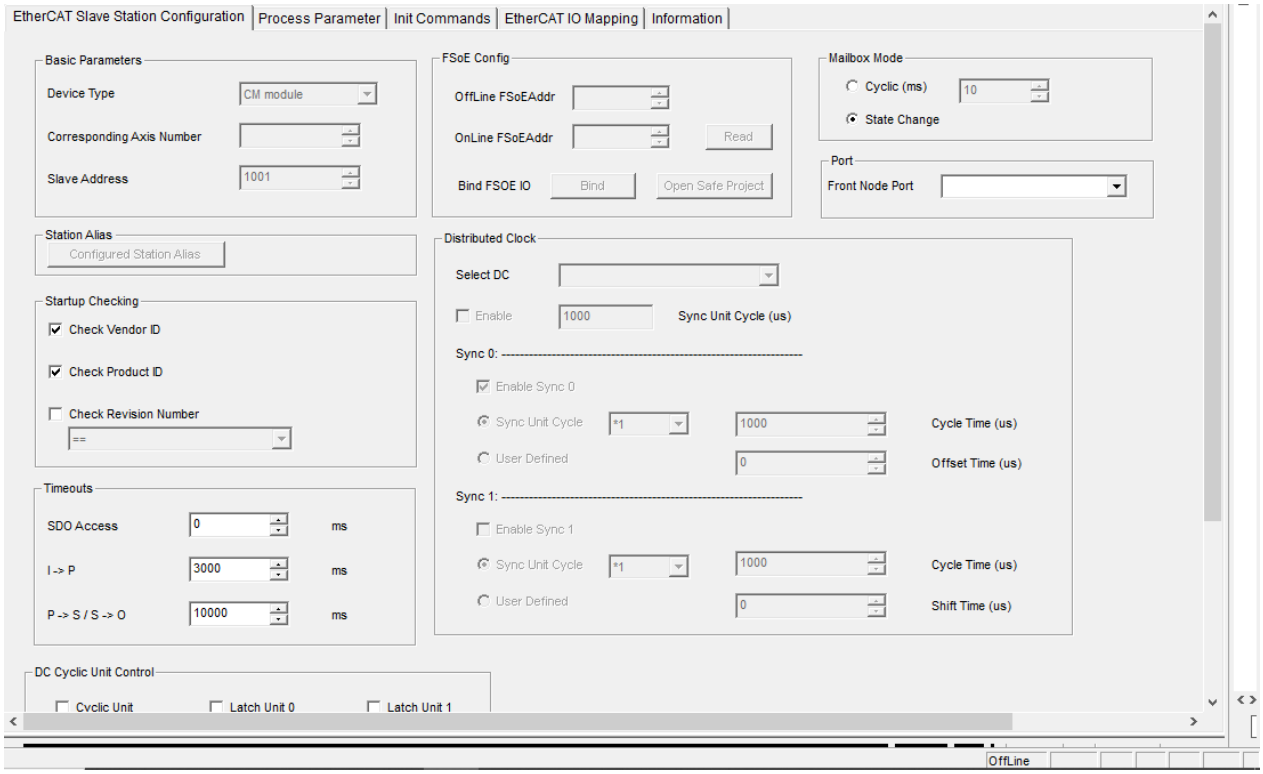


(2) Select LX-CM005 and click Add, the EtherCAT Master node displays modules and the DeviceNet slave station.

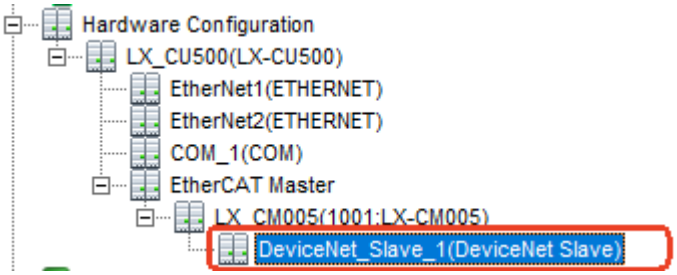




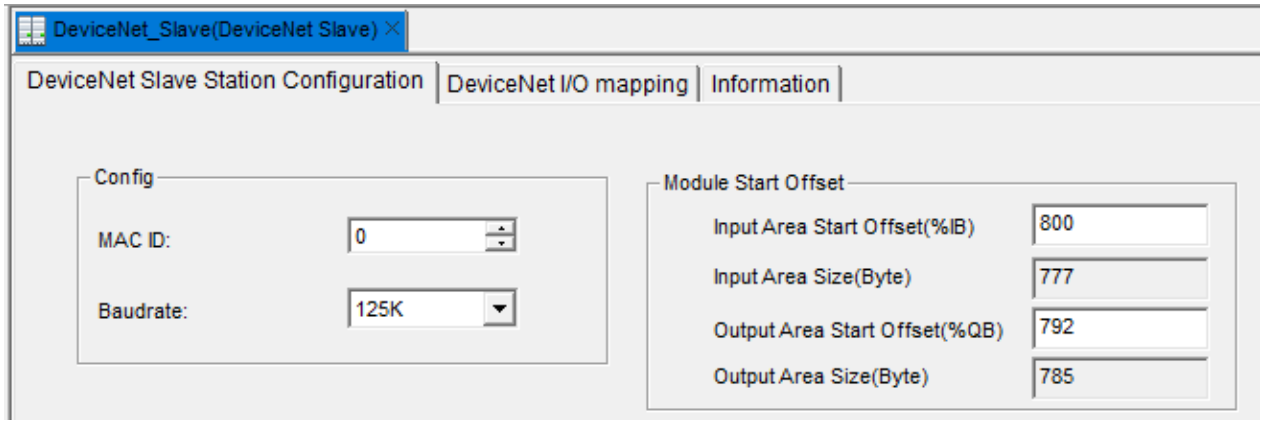
(3) Double-click the added LX-CM005 module node, and the DN slave station module configuration interface is displayed, of which the parameters can be kept as default.



(4) Double-click the added DeviceNet_Slave node:



The configuration interface is displayed as shown below:



The configuration interface includes DeviceNet Slave Station Configuration, DeviceNet I/O Mapping and Information.

■ Information description for the 'DeviceNet Slave Station Configuration tab:

□ Configuration: The master station MAC ID and Baud Rate can be modified.

Configuration Parameters	Function Description	Parameter Range	Configuration Description
MAC ID	Slave station ID identification	0~63	Each slave station is assigned a unique ID, that is, the ID cannot be repeated during the configuration
Baud Rate	Bus communication rate	125K 250K 500K	Configure baud rate based on end-to-end network distance 500 m, baud rate 125 Kbp/s 250 m, baud rate 250 Kbp/s 100 m, baud rate 500 Kbp/s

□ Module Start Offset

Input Area Start Offset (%IB): In offline mode, the starting address of the DeviceNet I/O mapping input channels can be modified.

Input Area Size (Byte): The number of bytes in the input area.

Output Area Start Offset (%QB): In offline mode, the starting address of the DeviceNet I/O mapping output channels can be modified.

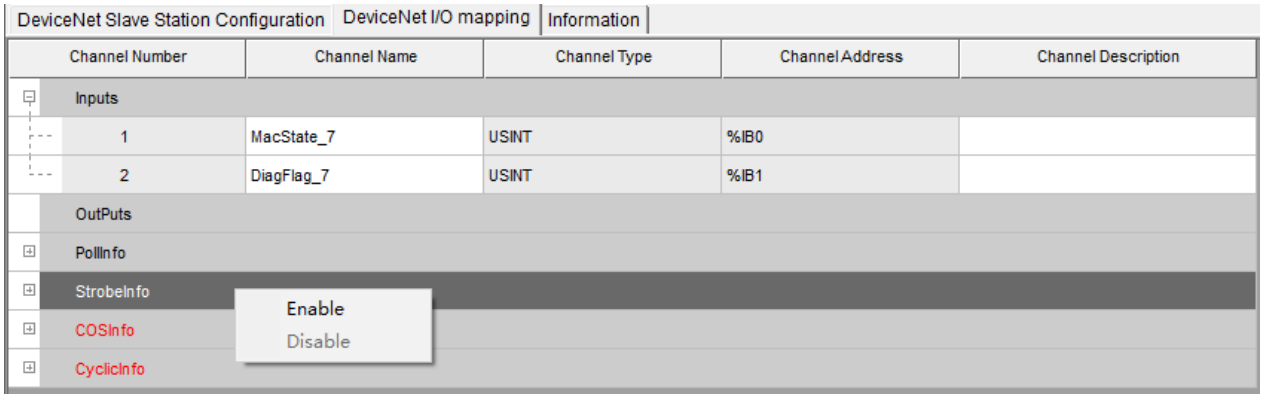
Output Area Size (Byte): The number of bytes in the output area.

■ The DeviceNet I/O Mapping interface includes input/output data, polling information, bit-strobed information, status change information, and cycle information.

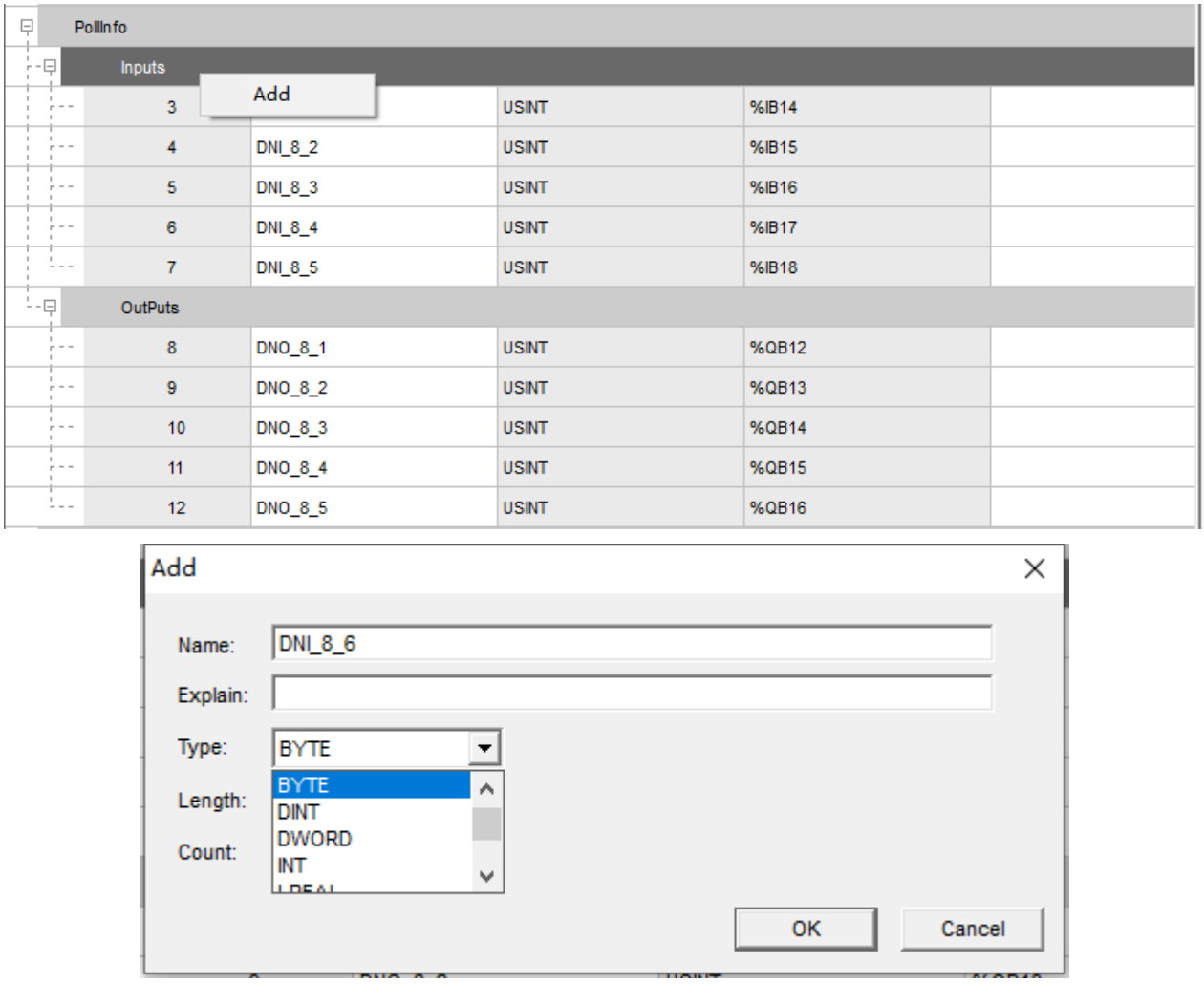
DeviceNet Slave Station Configuration | DeviceNet I/O mapping | Information

Channel Number	Channel Name	Channel Type	Channel Address	Channel Description
Inputs				
1	MacState_7	USINT	%IB0	
2	DiagFlag_7	USINT	%IB1	
OutPuts				
PollInfo				
Inputs				
3	DNI_7_1	USINT	%IB2	
4	DNI_7_2	USINT	%IB3	
5	DNI_7_3	USINT	%IB4	
6	DNI_7_4	USINT	%IB5	
7	DNI_7_5	USINT	%IB6	
8	DNI_7_6	USINT	%IB7	
9	DNI_7_7	USINT	%IB8	
10	DNI_7_8	USINT	%IB9	
OutPuts				
11	DNO_7_1	USINT	%QB0	
12	DNO_7_2	USINT	%QB1	
13	DNO_7_3	USINT	%QB2	
14	DNO_7_4	USINT	%QB3	
15	DNO_7_5	USINT	%QB4	
16	DNO_7_6	USINT	%QB5	
17	DNO_7_7	USINT	%QB6	

Among them, the information in red (bit-strobed information, status change information and cycle information) is not enabled by default. You can check this information and enable it from the right-click menu.



The user can customize each communication type (input or output) and add different data types. The total length of input or output shall not exceed 255 bytes.



10.4 DeviceNet Communication Example

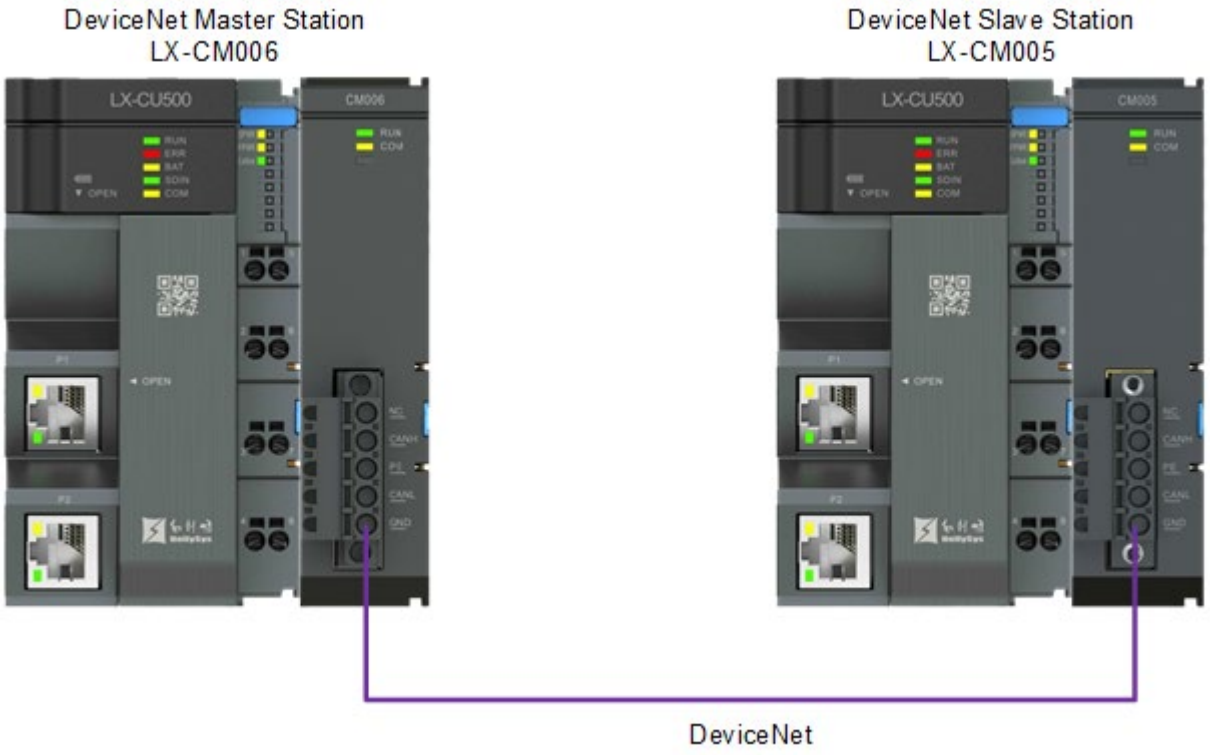
10.4.1 Example Function

This example uses the DeviceNet master station communication module LX-CM006 to read and write data from/to the DeviceNet slave station communication module LX-CM005.

10.4.2 System Composition

- DeviceNet master station: communication module LX-CM006
- DeviceNet slave station: communication module LX-CM005

10.4.3 Example Topology



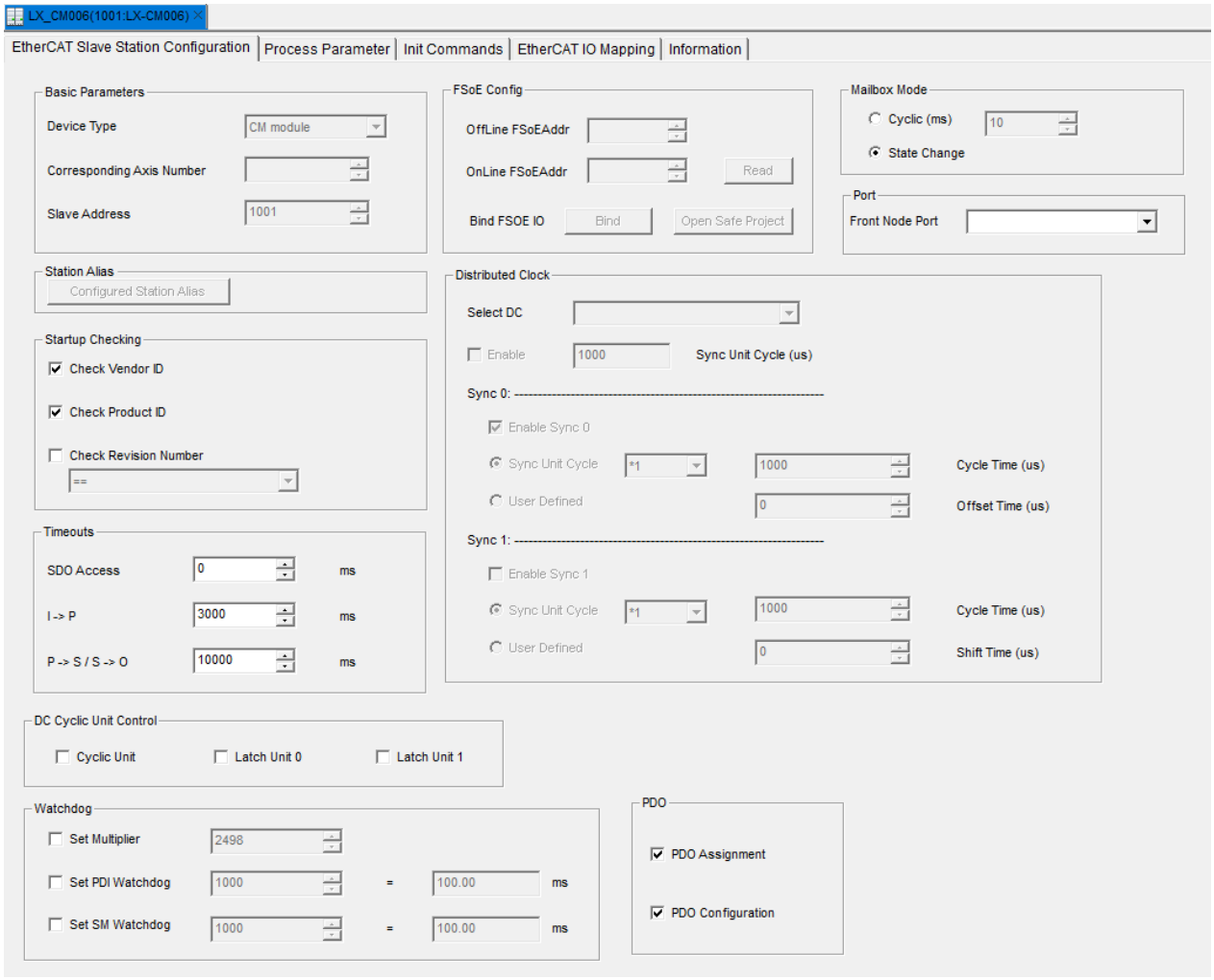
10.4.4 Programming

Create two projects (one master station project and one slave station project), and program and configure the two projects respectively.

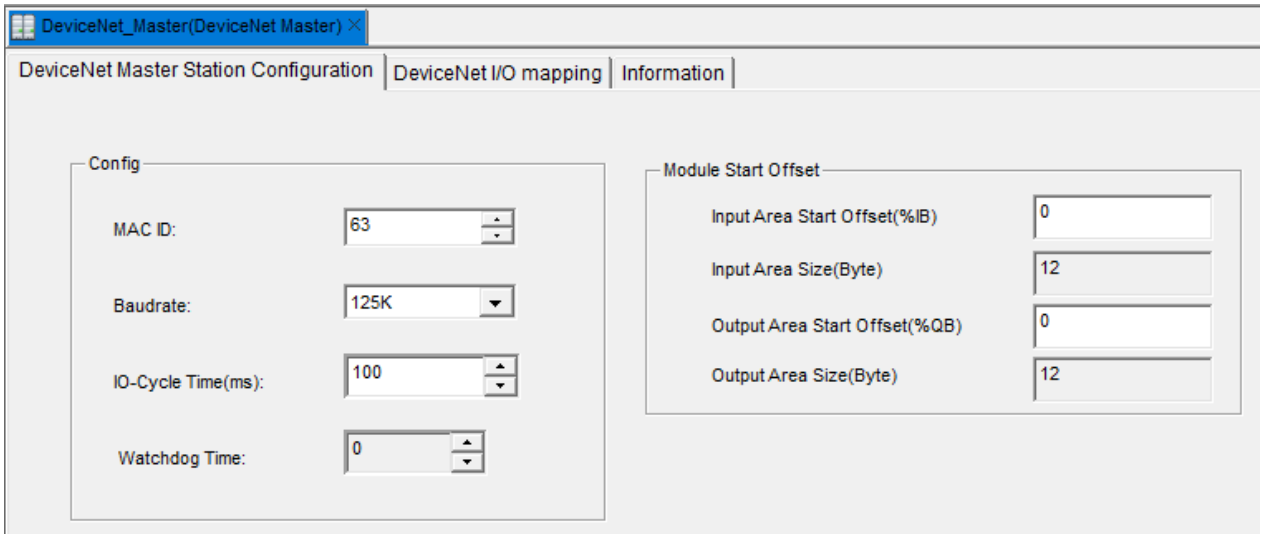
- For the configuration of module LX-CM006 in the DeviceNet master station communication program, refer to DeviceNet Configuration.

  The specific configuration in the master station program in this example is as follows:

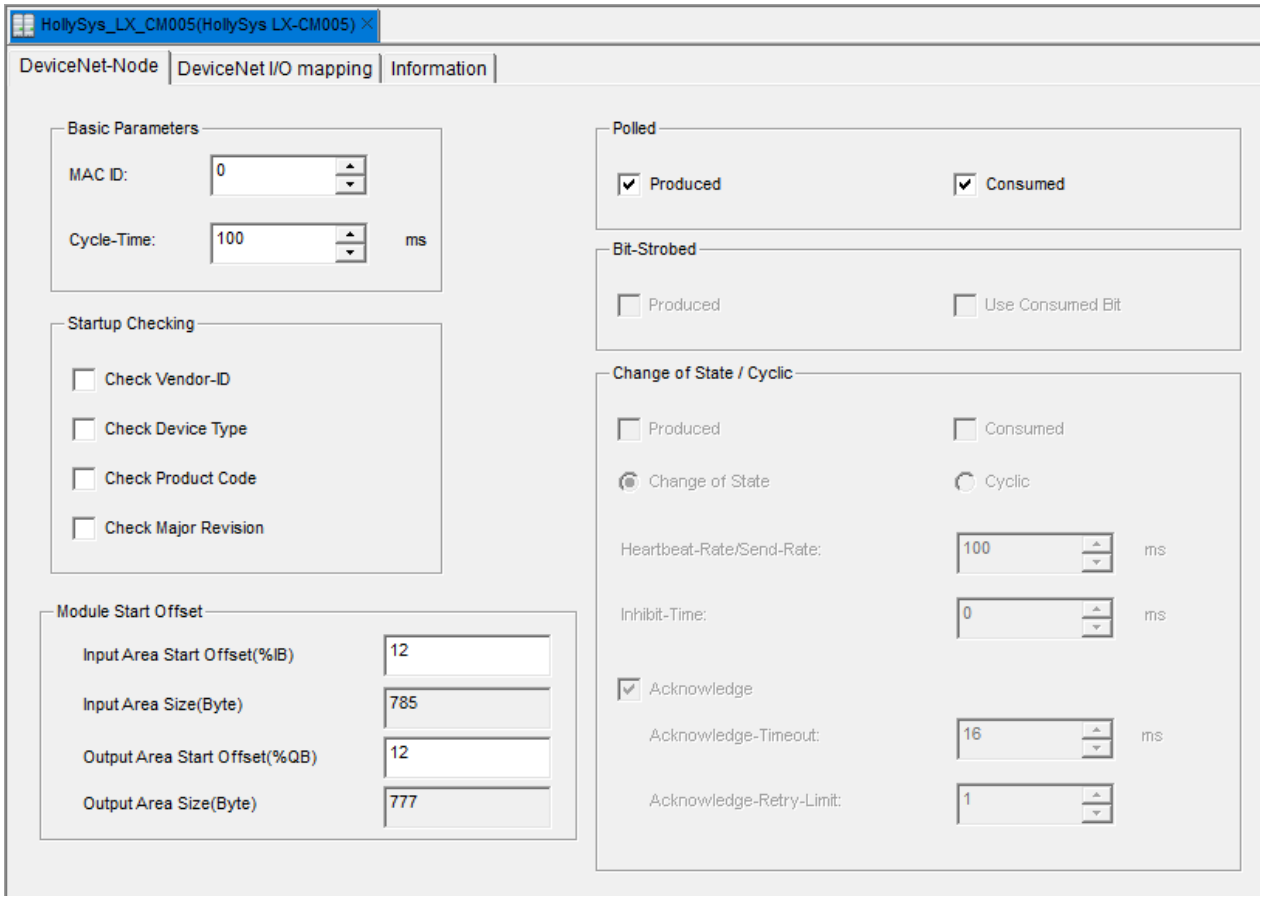
  - For the EtherCAT bus parameters, the default values are used:



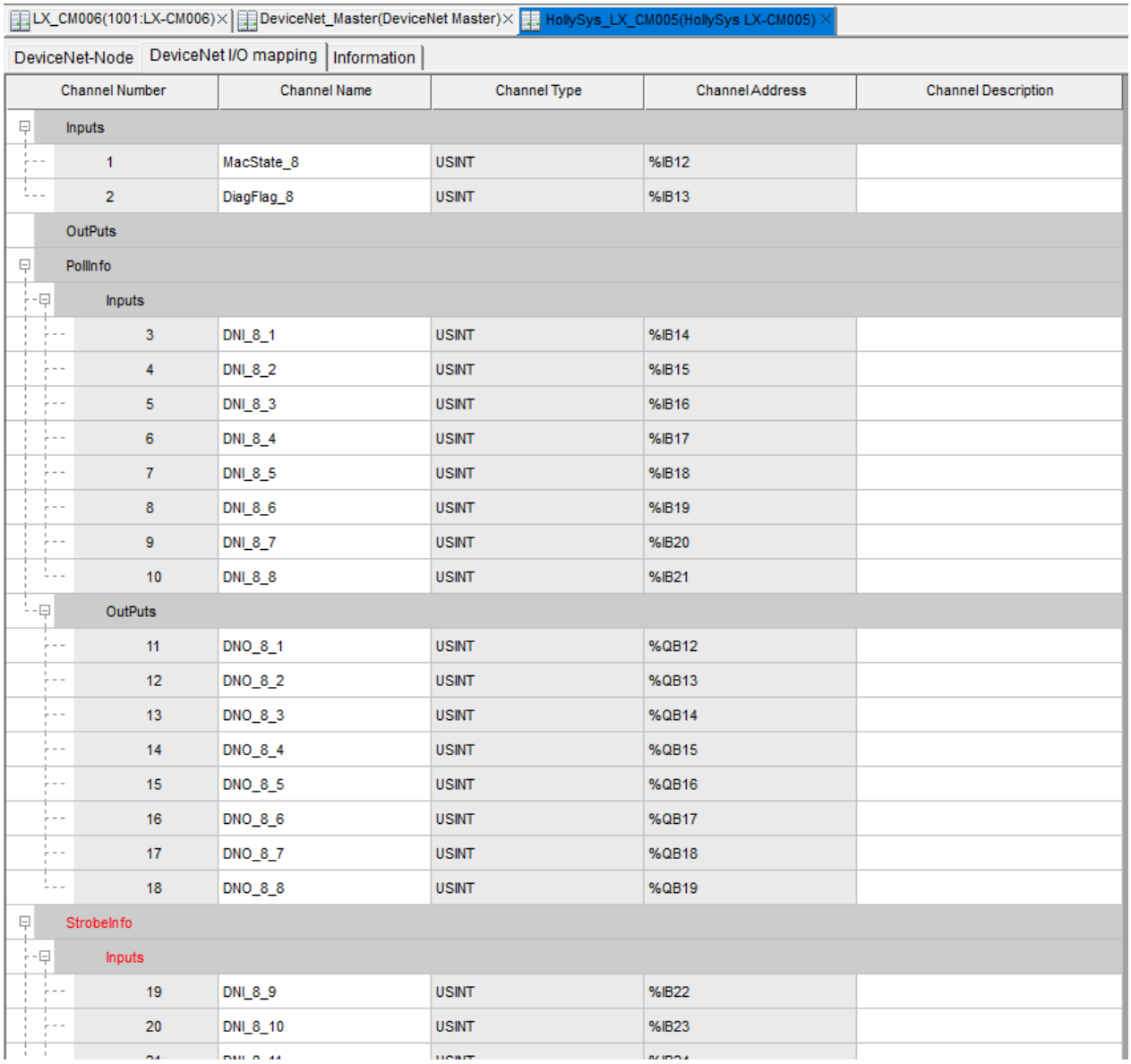
□ Parameters of DeviceNet master station:



□ Parameters of DeviceNet slave station:



□ IO variable mapping of DeviceNet slave station:



LX_CM006(1001:LX-CM006) | DeviceNet_Master(DeviceNet Master) | HollySys_LX_CM005(HollySys LX-CM005)

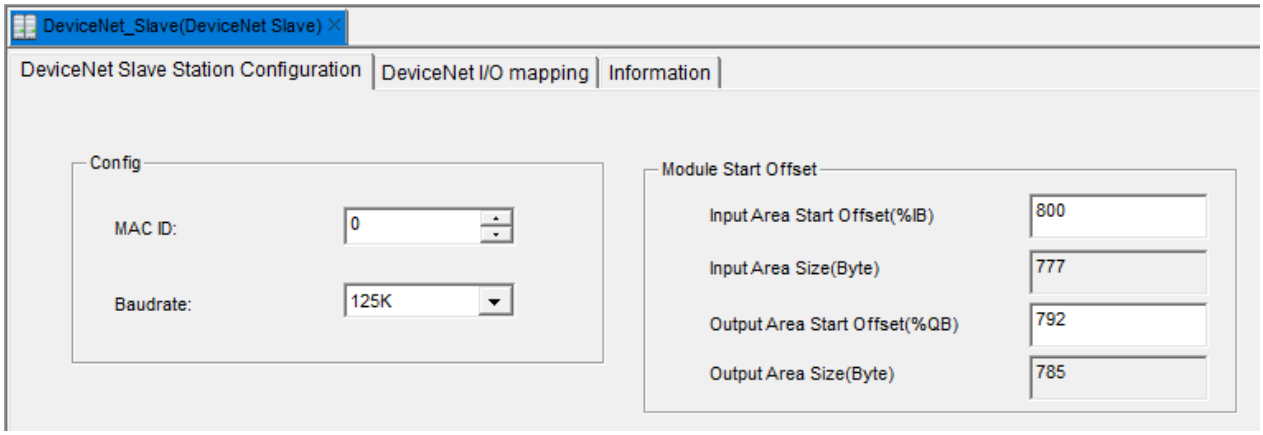
DeviceNet-Node | DeviceNet I/O mapping | Information

Channel Number	Channel Name	Channel Type	Channel Address	Channel Description
Inputs				
1	MacState_8	USINT	%IB12	
2	DiagFlag_8	USINT	%IB13	
OutPuts				
PollInfo				
Inputs				
3	DNI_8_1	USINT	%IB14	
4	DNI_8_2	USINT	%IB15	
5	DNI_8_3	USINT	%IB16	
6	DNI_8_4	USINT	%IB17	
7	DNI_8_5	USINT	%IB18	
8	DNI_8_6	USINT	%IB19	
9	DNI_8_7	USINT	%IB20	
10	DNI_8_8	USINT	%IB21	
OutPuts				
11	DNO_8_1	USINT	%QB12	
12	DNO_8_2	USINT	%QB13	
13	DNO_8_3	USINT	%QB14	
14	DNO_8_4	USINT	%QB15	
15	DNO_8_5	USINT	%QB16	
16	DNO_8_6	USINT	%QB17	
17	DNO_8_7	USINT	%QB18	
18	DNO_8_8	USINT	%QB19	
StrobeInfo				
Inputs				
19	DNI_8_9	USINT	%IB22	
20	DNI_8_10	USINT	%IB23	

- For the programming configuration of module LX-CM005 in the DeviceNet slave station communication program, refer to Configuring DeviceNet Slave Station.

  The specific configuration in the slave station program in this example is as follows:

  - Address and communication rate of DeviceNet slave station:



- IO variable mapping of DeviceNet slave station:

DeviceNet_Slave(DeviceNet Slave)

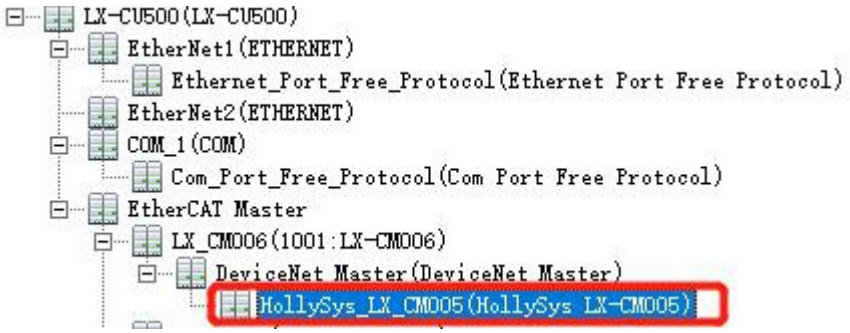
DeviceNet Slave Station Configuration | DeviceNet I/O mapping | Information

Channel Number	Channel Name	Channel Type	Channel Address	Channel Description
Inputs				
1	MacState_10	USINT	%IB800	
2	DiagFlag_10	USINT	%IB801	
OutPuts				
PollInfo				
Inputs				
3	DNI_10_1	USINT	%IB802	
4	DNI_10_2	USINT	%IB803	
5	DNI_10_3	USINT	%IB804	
6	DNI_10_4	USINT	%IB805	
7	DNI_10_5	USINT	%IB806	
8	DNI_10_6	USINT	%IB807	
9	DNI_10_7	USINT	%IB808	
10	DNI_10_8	USINT	%IB809	
OutPuts				
11	DNO_10_1	USINT	%QB792	
12	DNO_10_2	USINT	%QB793	
13	DNO_10_3	USINT	%QB794	
14	DNO_10_4	USINT	%QB795	
15	DNO_10_5	USINT	%QB796	
16	DNO_10_6	USINT	%QB797	
17	DNO_10_7	USINT	%QB798	
18	DNO_10_8	USINT	%QB799	
StrobeInfo				
Inputs				
OutPuts				
19	DNO_10_9	USINT	%QB800	

10.4.5 Online Debugging

After the AT program configuration is completed, install the program to the master and slave station PLCs respectively.

(1) In monitoring mode, double-click the slave station device under the DeviceNet master station to view and diagnose the device status.



HollySys_LX_CM005(HollySys LX-CM005)

DeviceNet-Node | DeviceNet I/O mapping | Information

Channel Number	Channel Name	GroupName	Channel Type	Channel Address	Online Value
Inputs					
1	MacState_12		USINT	%IB980	0
2	DiagFlag_12		USINT	%IB981	0
OutPuts					
PollInfo					
Inputs					
3	DNI_12_1		USINT	%IB982	0
4	DNI_12_2		USINT	%IB983	0
5	DNI_12_3		USINT	%IB984	0
6	DNI_12_4		USINT	%IB985	0
7	DNI_12_5		USINT	%IB986	0
8	DNI_12_6		USINT	%IB987	0
9	DNI_12_7		USINT	%IB988	0
10	DNI_12_8		USINT	%IB989	0
OutPuts					
11	DNO_12_1		USINT	%QB988	0
12	DNO_12_2		USINT	%QB989	0
13	DNO_12_3		USINT	%QB990	0
14	DNO_12_4		USINT	%QB991	0
15	DNO_12_5		USINT	%QB992	0
16	DNO_12_6		USINT	%QB993	0
17	DNO_12_7		USINT	%QB994	0
18	DNO_12_8		USINT	%QB995	0

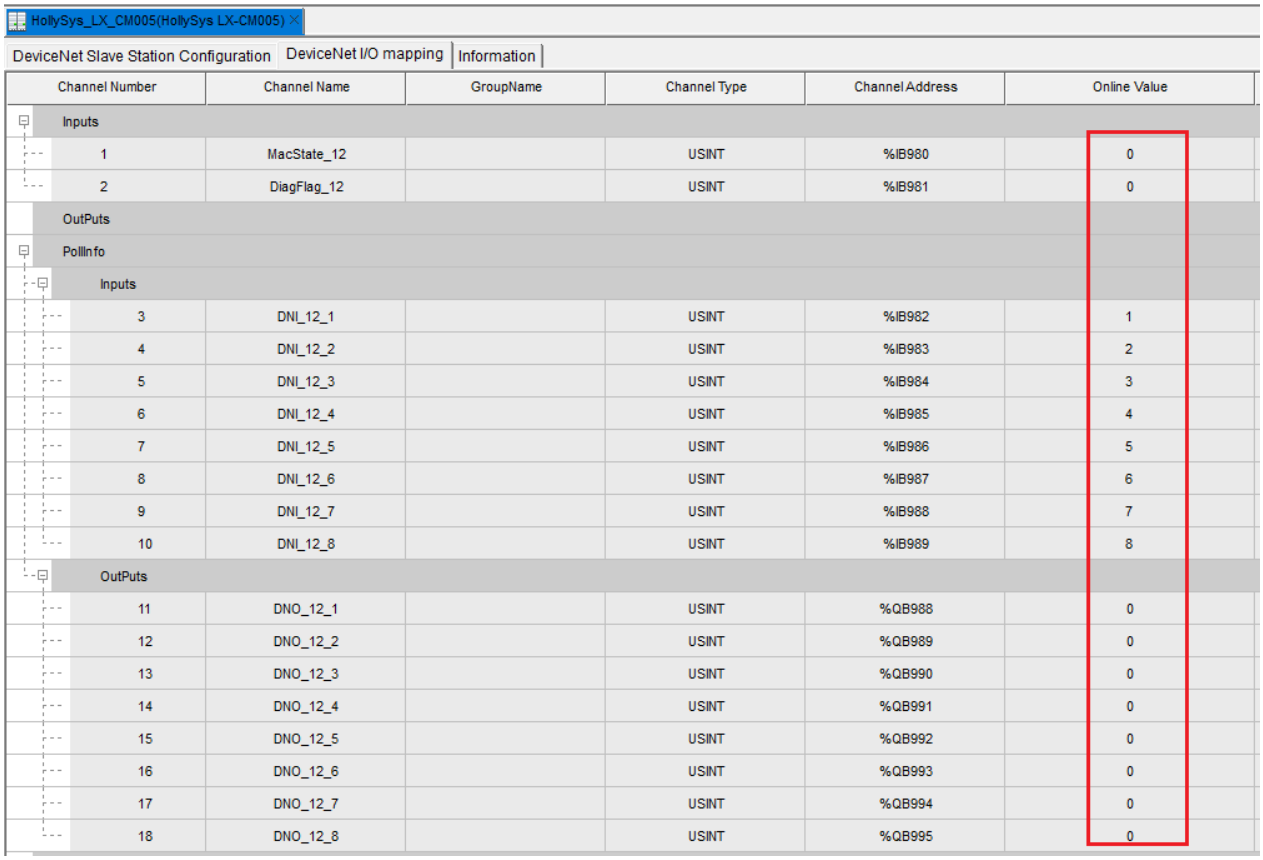
(2) Double-click the output variable to change the output variable value of the slave station module in the master station program.

HollySys_LX_CM005(HollySys LX-CM005)

DeviceNet-Node | DeviceNet I/O mapping | Information

Channel Number	Channel Name	GroupName	Channel Type	Channel Address	Online Value
Inputs					
1	MacState_12		USINT	%IB980	0
2	DiagFlag_12		USINT	%IB981	0
OutPuts					
PollInfo					
Inputs					
3	DNI_12_1		USINT	%IB982	0
4	DNI_12_2		USINT	%IB983	0
5	DNI_12_3		USINT	%IB984	0
6	DNI_12_4		USINT	%IB985	0
7	DNI_12_5		USINT	%IB986	0
8	DNI_12_6		USINT	%IB987	0
9	DNI_12_7		USINT	%IB988	0
10	DNI_12_8		USINT	%IB989	0
OutPuts					
11	DNO_12_1		USINT	%QB988	1
12	DNO_12_2		USINT	%QB989	2
13	DNO_12_3		USINT	%QB990	3
14	DNO_12_4		USINT	%QB991	4
15	DNO_12_5		USINT	%QB992	5
16	DNO_12_6		USINT	%QB993	6
17	DNO_12_7		USINT	%QB994	7
18	DNO_12_8		USINT	%QB995	8

From the input variables of the slave station module in the slave station program, it can be seen that the data changes are consistent.



HollySys_LX_CM005(HollySys LX-CM005)

DeviceNet Slave Station Configuration | DeviceNet I/O mapping | Information

Channel Number	Channel Name	GroupName	Channel Type	Channel Address	Online Value
Inputs					
1	MacState_12		USINT	%IB980	0
2	DiagFlag_12		USINT	%IB981	0
OutPuts					
PollInfo					
Inputs					
3	DNI_12_1		USINT	%IB982	1
4	DNI_12_2		USINT	%IB983	2
5	DNI_12_3		USINT	%IB984	3
6	DNI_12_4		USINT	%IB985	4
7	DNI_12_5		USINT	%IB986	5
8	DNI_12_6		USINT	%IB987	6
9	DNI_12_7		USINT	%IB988	7
10	DNI_12_8		USINT	%IB989	8
OutPuts					
11	DNO_12_1		USINT	%QB988	0
12	DNO_12_2		USINT	%QB989	0
13	DNO_12_3		USINT	%QB990	0
14	DNO_12_4		USINT	%QB991	0
15	DNO_12_5		USINT	%QB992	0
16	DNO_12_6		USINT	%QB993	0
17	DNO_12_7		USINT	%QB994	0
18	DNO_12_8		USINT	%QB995	0

(3) Similarly, double-click the output variable to change the output variable value of the slave station module in the slave program.

HollySys_LX_CM005(HollySys LX-CM005)

DeviceNet Slave Station Configuration | DeviceNet I/O mapping | Information

Channel Number	Channel Name	GroupName	Channel Type	Channel Address	Online Value
Inputs					
1	MacState_12		USINT	%IB980	0
2	DiagFlag_12		USINT	%IB981	0
OutPuts					
PollInfo					
Inputs					
3	DNI_12_1		USINT	%IB982	1
4	DNI_12_2		USINT	%IB983	2
5	DNI_12_3		USINT	%IB984	3
6	DNI_12_4		USINT	%IB985	4
7	DNI_12_5		USINT	%IB986	5
8	DNI_12_6		USINT	%IB987	6
9	DNI_12_7		USINT	%IB988	7
10	DNI_12_8		USINT	%IB989	8
OutPuts					
11	DNO_12_1		USINT	%QB988	11
12	DNO_12_2		USINT	%QB989	22
13	DNO_12_3		USINT	%QB990	33
14	DNO_12_4		USINT	%QB991	44
15	DNO_12_5		USINT	%QB992	55
16	DNO_12_6		USINT	%QB993	66
17	DNO_12_7		USINT	%QB994	77
18	DNO_12_8		USINT	%QB995	88

From the input variables of the slave station module in the master station program, it can be seen that the data changes are consistent.

HollySys_LX_CM005(HollySys LX-CM005)

DeviceNet-Node | DeviceNet I/O mapping | Information

Channel Number	Channel Name	GroupName	Channel Type	Channel Address	Online Value
Inputs					
1	MacState_12		USINT	%IB980	0
2	DiagFlag_12		USINT	%IB981	0
OutPuts					
PollInfo					
Inputs					
3	DNI_12_1		USINT	%IB982	11
4	DNI_12_2		USINT	%IB983	22
5	DNI_12_3		USINT	%IB984	33
6	DNI_12_4		USINT	%IB985	44
7	DNI_12_5		USINT	%IB986	55
8	DNI_12_6		USINT	%IB987	66
9	DNI_12_7		USINT	%IB988	77
10	DNI_12_8		USINT	%IB989	88
OutPuts					
11	DNO_12_1		USINT	%QB988	1
12	DNO_12_2		USINT	%QB989	2
13	DNO_12_3		USINT	%QB990	3
14	DNO_12_4		USINT	%QB991	4
15	DNO_12_5		USINT	%QB992	5
16	DNO_12_6		USINT	%QB993	6
17	DNO_12_7		USINT	%QB994	7
18	DNO_12_8		USINT	%QB995	8

10.5 Fault Diagnosis

10.5.1 Master Station Module Diagnosis

DeviceNet master module diagnosis is divided into periodic diagnosis and non-periodic diagnosis.

DeviceNet_Master(DeviceNet Master)

DeviceNet Master Station Configuration | DeviceNet I/O mapping | Information

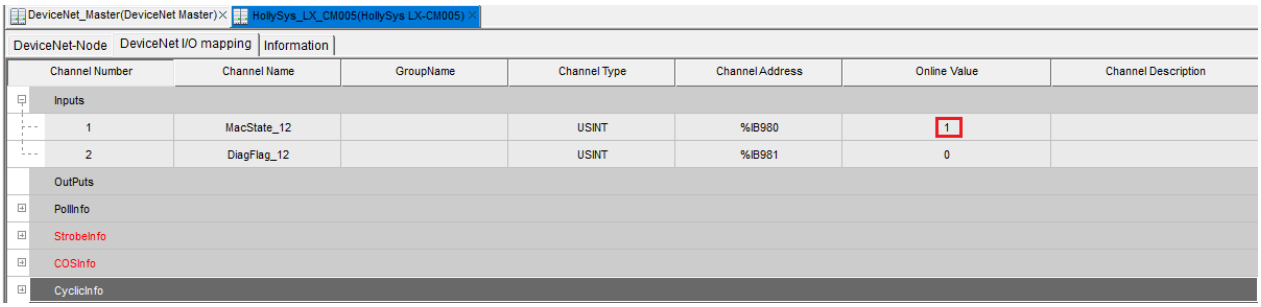
Channel Number	Channel Name	GroupName	Channel Type	Channel Address	Channel Description
Inputs					
1	Error_7		USINT	%IB0	
2	DiagFlag_7		BOOL	%IX1.0	
3	NetPowerFault_7		BOOL	%IX1.1	
4	CommStatusNode0_15_7		UINT	%IW2	
5	CommStatusNode16_31_7		UINT	%IW4	
6	CommStatusNode32_47_7		UINT	%IW6	
7	CommStatusNode48_63_7		UINT	%IW8	
Outputs					

Diagnosis Information description

Diagnosis	Diagnostic Description
Error	Number of slave stations that are not online, indicating the number of slave stations that have been configured but are not online. If it is 0, it means that all slave stations are online.

DiagFlag	Diagnosis flag
NetPowerFault	Is the network power supply faulty: • TRUE: Network power supply is faulty • FALSE: Network power supply is normal Note: LX-CM006 does not have this diagnostic function
CommStatusNode0_15	Representation of Communication Status for Slave MacID 0 to 15 Bit0: 0: Slave 0 IO communication is active or not configured 1: Slave 0 IO communication has stopped ... Bit15: 0: Slave 15 IO communication is active or not configured 1: Slave 15 IO communication has stopped
CommStatusNode16_31	Representation of Communication Status for Slave MacID 16 to 31 Bit16: 0: Slave 16 IO communication is active or not configured 1: Slave 16 IO communication has stopped ... Bit31: 0: Slave 31 IO communication is active or not configured 1: Slave 31 IO communication has stopped
CommStatusNode32_47	Representation of Communication Status for Slave MacID 32 to 47 Bit32: 0: Slave 32 IO communication is active or not configured 1: Slave 32 IO communication has stopped ... Bit47: 0: Slave 47 IO communication is active or not configured 1: Slave 47 IO communication has stopped
CommStatusNode48_63	Representation of Communication Status for Slave MacID 48 to 63 Bit48: 0: Slave 48 IO communication is active or not configured 1: Slave 48 IO communication has stopped ... Bit63: 0: Slave 63 IO communication is active or not configured 1: Slave 63 IO communication has stopped

The diagnostic MacState in the slave station is the slave status.



The diagnostic information of the slave station is as follows:

Diagnostic Status	Diagnostic Description	Diagnostic Online Value
NO_ERROR	The slave station is normal	0
STATION_DEACTIVATED	The slave station is not online	1
VENDOR_ID_ERR	Vendor ID does not match	42
DEVICE_TYPE_ERR	Device Type does not match	43
PRODUCT_CODE_ERR	Product Code does not match	44
REVISION_ERR	Revision does not match	45

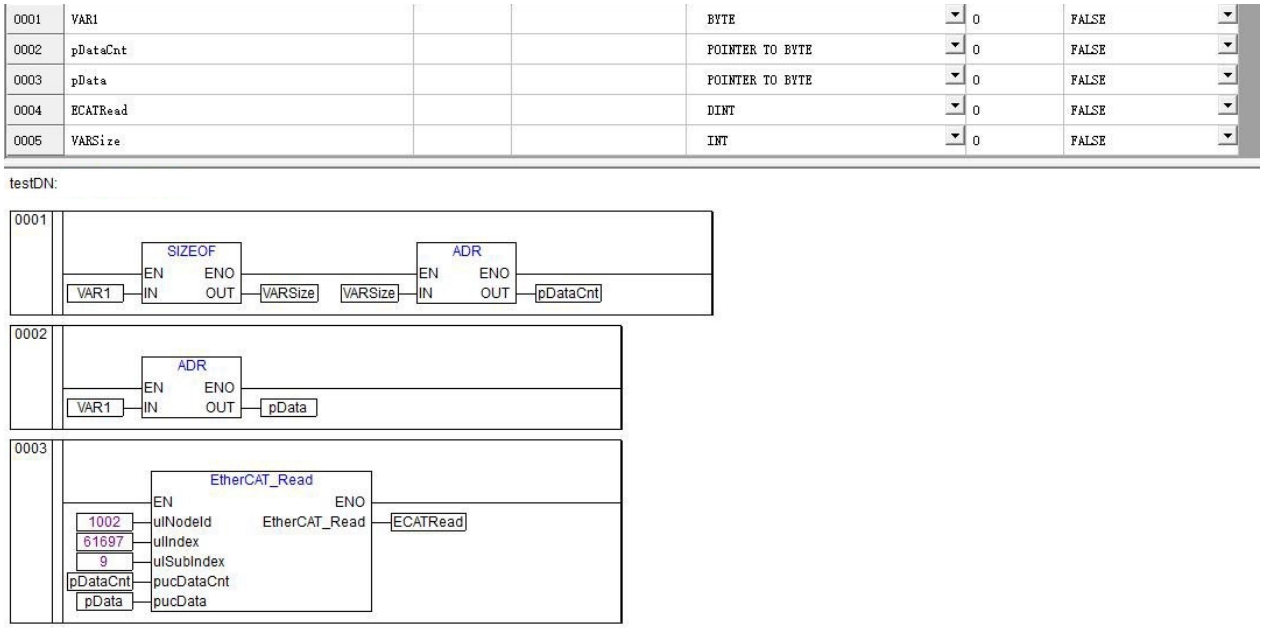
PRODUCED_IO_SIZE_ERR	The production data does not match	47
CONSUMED_IO_SIZE_ERR	The consumption data does not match	48

The non-periodic diagnosis mainly covers module status, which needs to be read using the Ether CAT_Read functional block. The specific description is as follows:

Index	SubIndex	Diagnostic Description	Diagnostic Value
0xF100 (61696)	01	Number of slave stations that are not online	Number of slave stations that are not online
0xF101 (61697)	01	RUN MODE	0
		IDLE MODE	1
		MACID CHECK FAILED	2
	09	CAN BUS-OFF	1
		CAN NO ERROR	0

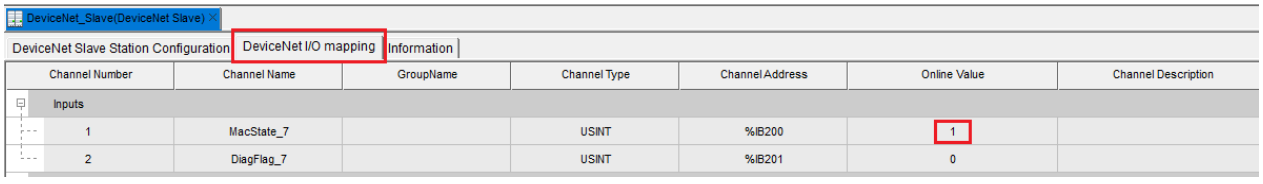
Call Library Manager → Fieldbus Library → ETHERCAT Protocol Related Instruction Library → EtherCAT_Read.

Diagnostic program example:



10.5.2 Slave Station Module Diagnosis

The LX-CM005 module has periodic diagnosis and non-periodic diagnosis. The periodic diagnosis is the MacState of the module, which mainly includes protocol stack diagnosis and device parameter diagnosis.



The periodic diagnostic information is shown in the following table:

Diagnosis	Diagnostic Description	Diagnostic Value

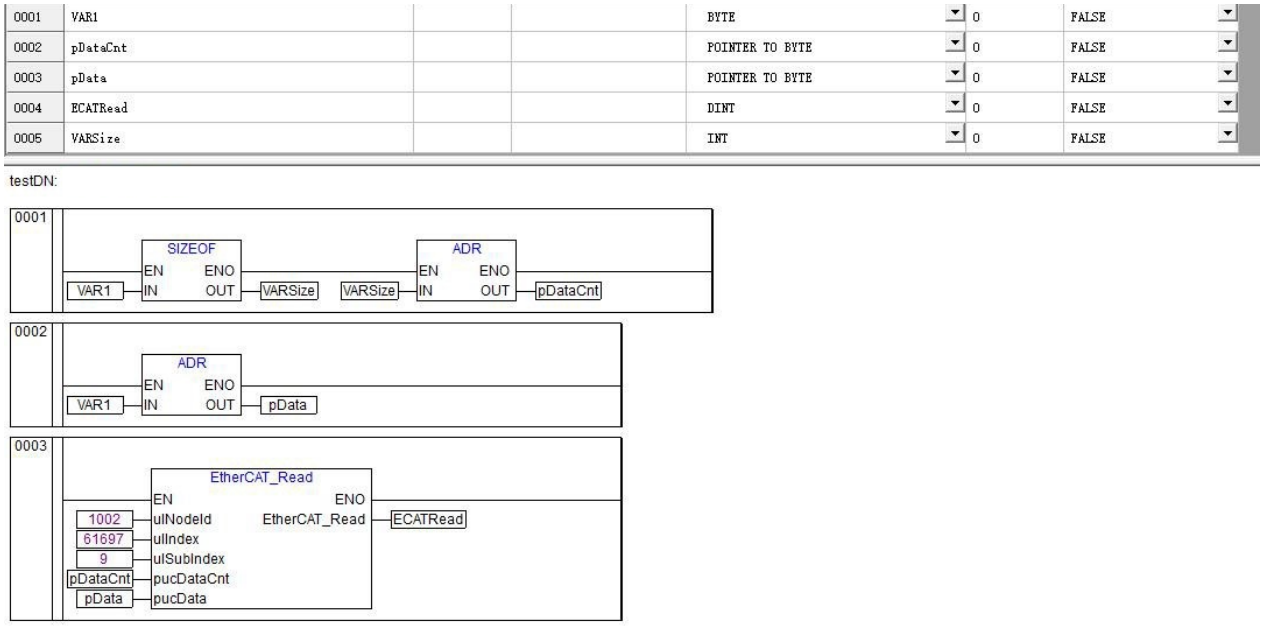
NO_ERROR	No error occurred in the protocol stack	0
STATION_DEACTIVATED	Errors occurred in the protocol stack	1
STATION_READY	Set after the protocol stack initialization is completed and reset after normal operation	18
VENDOR_ID_ERR	Vendor ID does not match	42
DEVICE_TYPE_ERR	Device Type does not match	43
PRODUCT_CODE_ERR	Product Code does not match	44
REVISION_ERR	Revision does not match	45
PRODUCED_IO_SIZE_ERR	The output data length exceeds 255	47
CONSUMED_IO_SIZE_ERR	The input data length exceeds 255	48
CAN_CONNECT_ERR	Disconnected from the master station	50

The non-periodic diagnosis mainly covers module status, which needs to be read using the Ether CAT_Read functional block. The specific description is as follows:

Index	SubIndex	Diagnostic Description	Diagnostic Value
0xF101 (61697)	01	RUN MODE	0
		IDLE MODE	1
		MACID CHECK FAILED	2
	09	CAN BUS-OFF	1
		CAN NO ERROR	0

Call Library Manager → Fieldbus Library → ETHERCAT Protocol Related Instruction Library → EtherCAT_Read.

Diagnostic program example:



Chapter 11 PROFINET Communication

11.1 Network Overview

11.1.1 Introduction

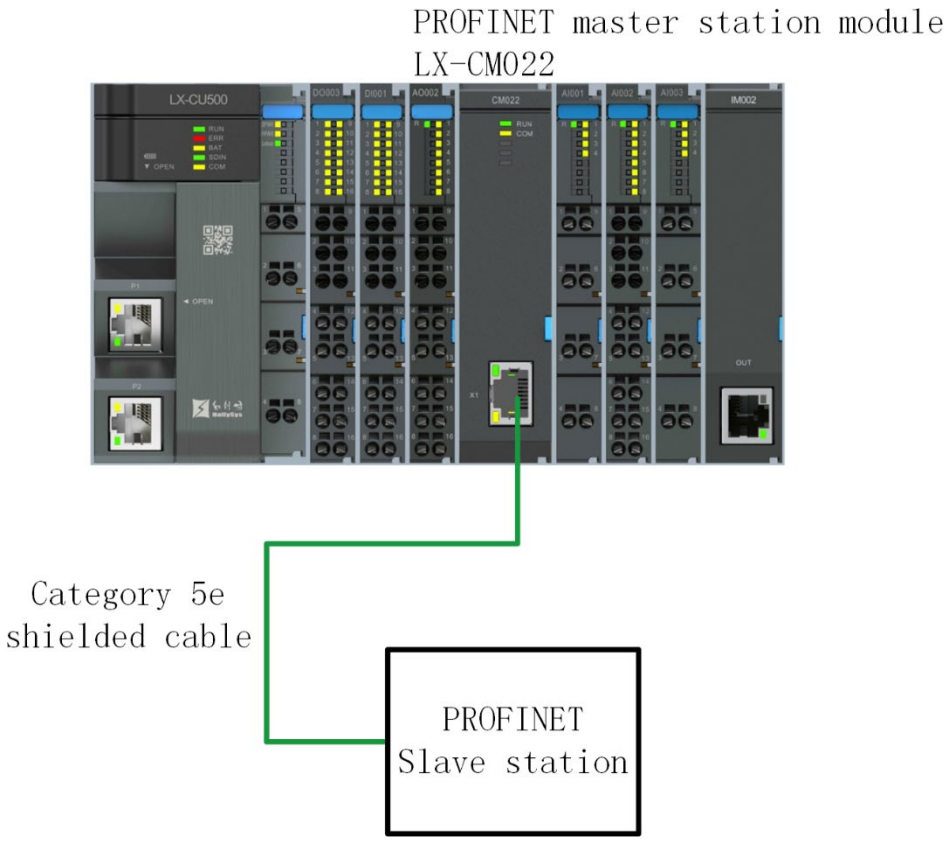
The LX-CM022 module supports the PROFINET RT protocol, enabling communication conversion from an EtherCAT slave station to a PROFINET master station. It allows monitoring the operational status of PROFINET slave devices on the PN side through the LX-CU500 CPU module or an EtherCAT master controller.

The LX-CM022 module can be embedded into the LX series PLC system, receiving power via the bus interface and functioning as an EtherCAT slave module within the LX series PLC. The EtherCAT master supports up to 3 LX-CM022 modules. The module features only one external RJ45 port for connecting PN slave devices, enabling their monitoring and control.

PROFINET slave devices must be configured with a device name and IP address using third-party software (such as TIA Portal, PRONETA, or TwinCAT).

11.2 Network Topology

The following figure shows the network structure of PROFINET.



11.3 Configuration

11.3.1 Configuration of PROFINET Protocol

11.3.1.1 Adding the LX-CM022 Module

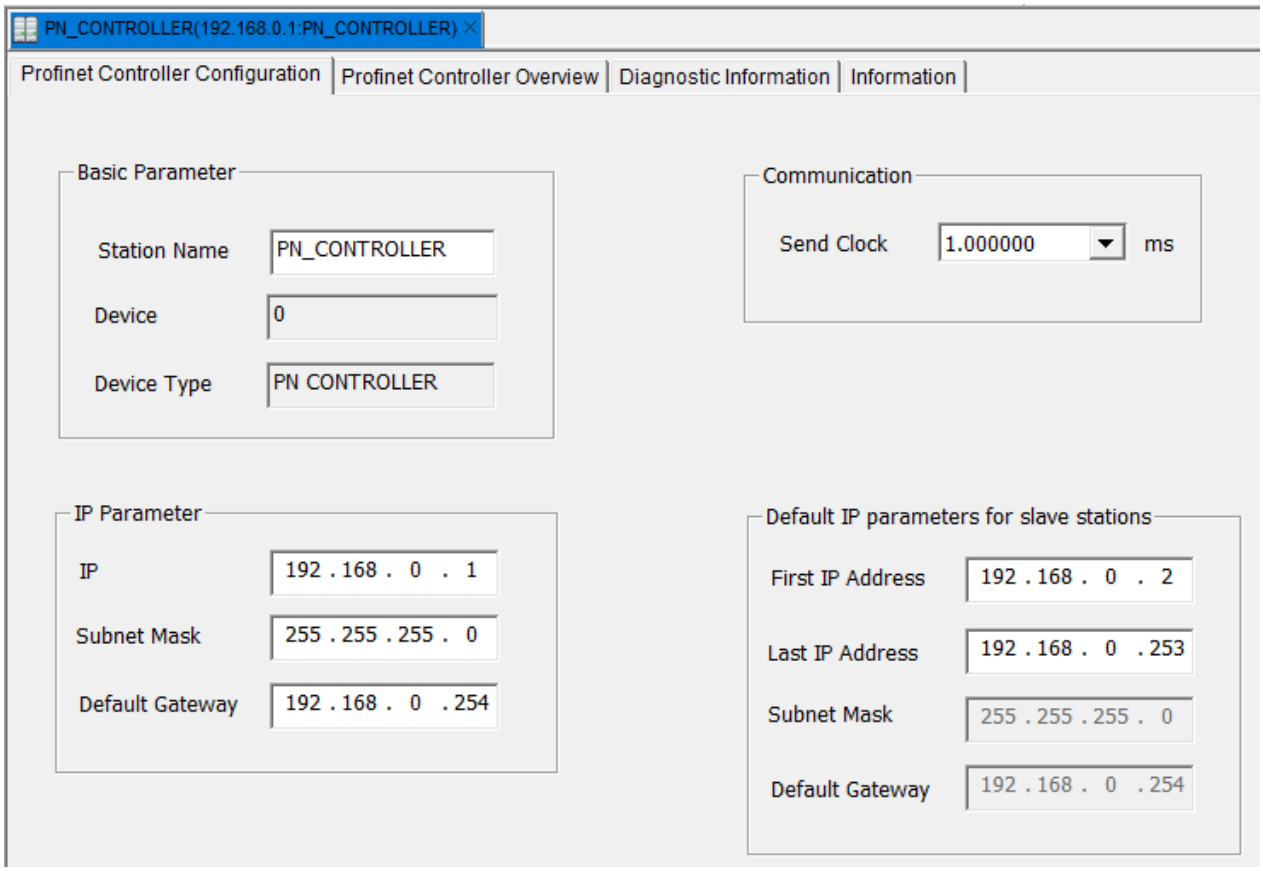
The method for adding the LX-CM022 module can be referred to in the LX-CM006 module.

- An EtherCAT master station supports the addition of up to 3 LX-CM022 modules. Each LX-CM022 module only supports one PROFINET nodes, and these nodes cannot be deleted. When copying and pasting configurations on a node, no new nodes are created; only the node configuration and sub-nodes are pasted.

11.3.1.2 Configuring the PROFINET Master Station

Double-click the added "PN_CONTROLLER" node to configure the master station parameters:

The configuration interface includes the following sections: "Profinet Controller Configuration", " Profinet Controller Overview", "Diagnostics Information", "Information". As shown in the figure below:



- Explanation of Information on the "PROFINET Master Configuration" Tab:
  - Basic Parameter

    Station Name: Default is generated as "PN_CONTROLLER", but can be modified. Changing this site name does not synchronize with the management tree node name or the device name in the overview. Similarly, modifying the management tree node name or the device name in the overview does not affect the site name in the configuration interface. However, changes to the management tree node name and the device name in the overview are synchronized with each other. When modifying the site name, a duplicate check must be performed to ensure it does not conflict with any existing node device names under the current module's PROFINET master station.

    Device: Not modifiable.

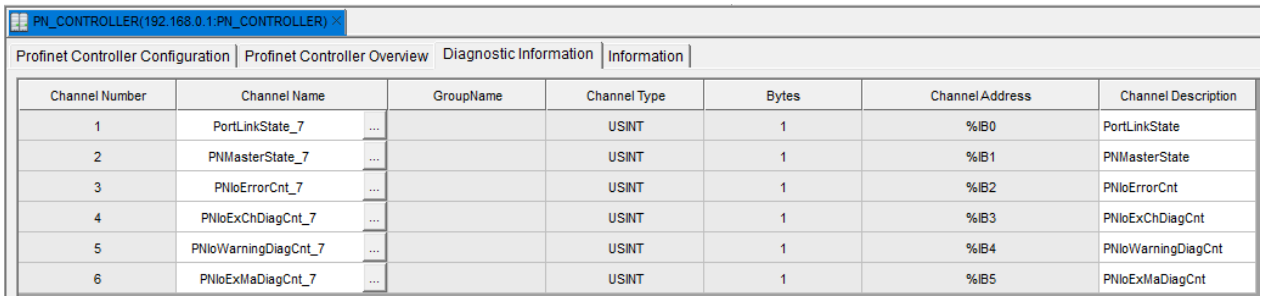
    Device Type: Not modifiable.
  - Communication

    Send Clock: Configure the transmission clock. Currently, the master station transmission clock only supports 1.000ms..
  - IP Parameter: Modifiable; configure IP address information based on actual conditions.
  - Default IP parameters for slave stations: Modifiable; configure the IP address range for PROFINET slave devices.
- Explanation of information on the "PROFINET Master Overview" tab: The "Profinet Controller Overview" tab displays the PROFINET slave devices added under the "PN_CONTROLLER" node.

It mainly shows the following information about PROFINET slave devices: Device Name ,IP Address, RT Claass, Software Version, Hardware Version, Order Number.

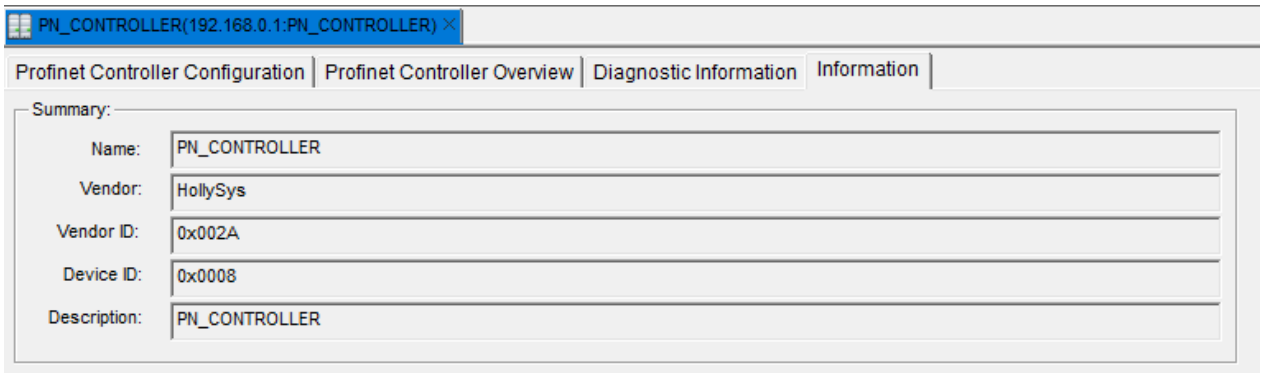
Columns such as Sequence Number, Device, Software Version, Hardware Version, and Order Number cannot be modified. All other columns can be edited, and changes must be synchronized to the corresponding slave devices.

- Explanation of Diagnostic Information:



For detailed instructions, refer to the periodic diagnostic description of the PROFINET master module.

- The "Information" page displays relevant information about the PROFINET master protocol.

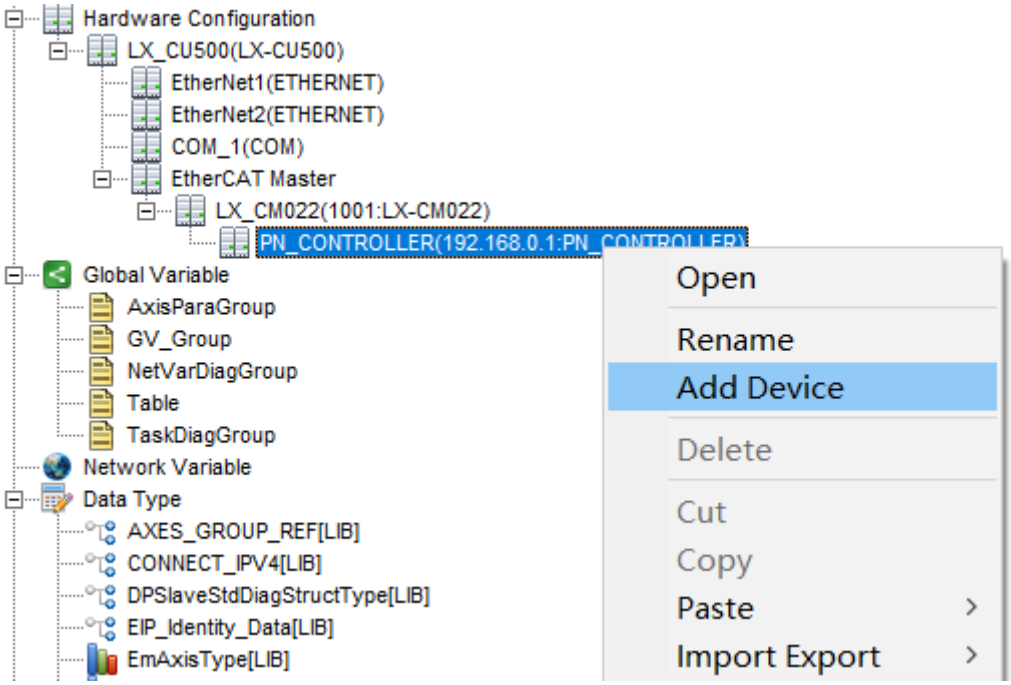


11.3.2 Configure PROFINET Slave Device

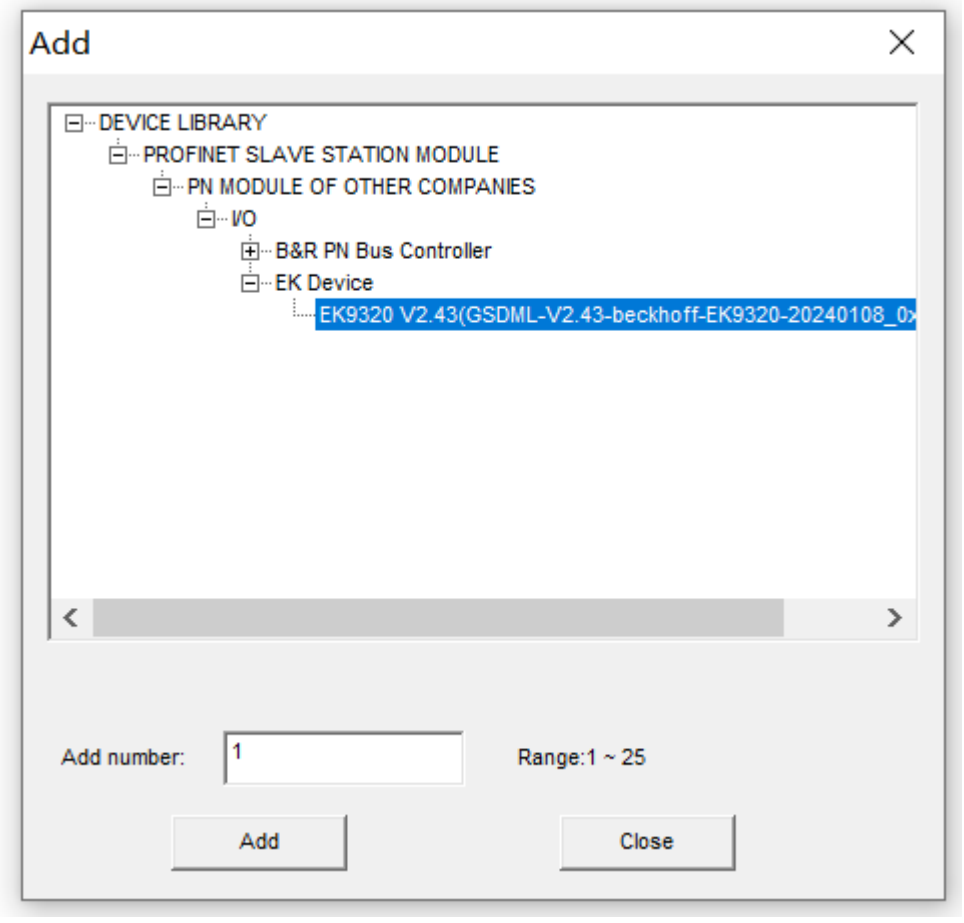
11.3.2.1 Adding PROFINET Slave

(1) Import PROFINET: From the menu bar, go to "Project" - "Import Device Description File" to import the PROFINET device file.

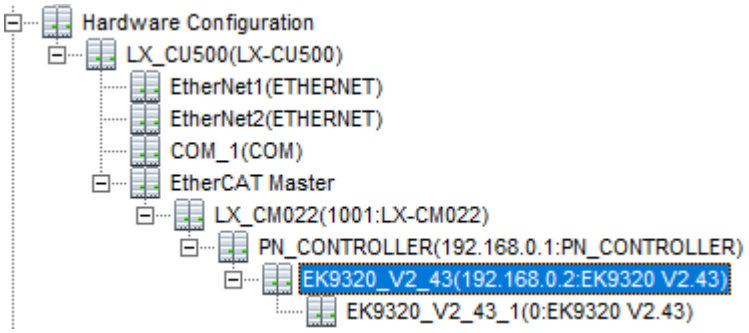
(2) Right-click the "PN_CONTROLLER" node and select the "Add" command, as shown in the figure below.



(3) In the pop-up "Add" dialog box, select the device imported in step (1) and click the "Add" button.



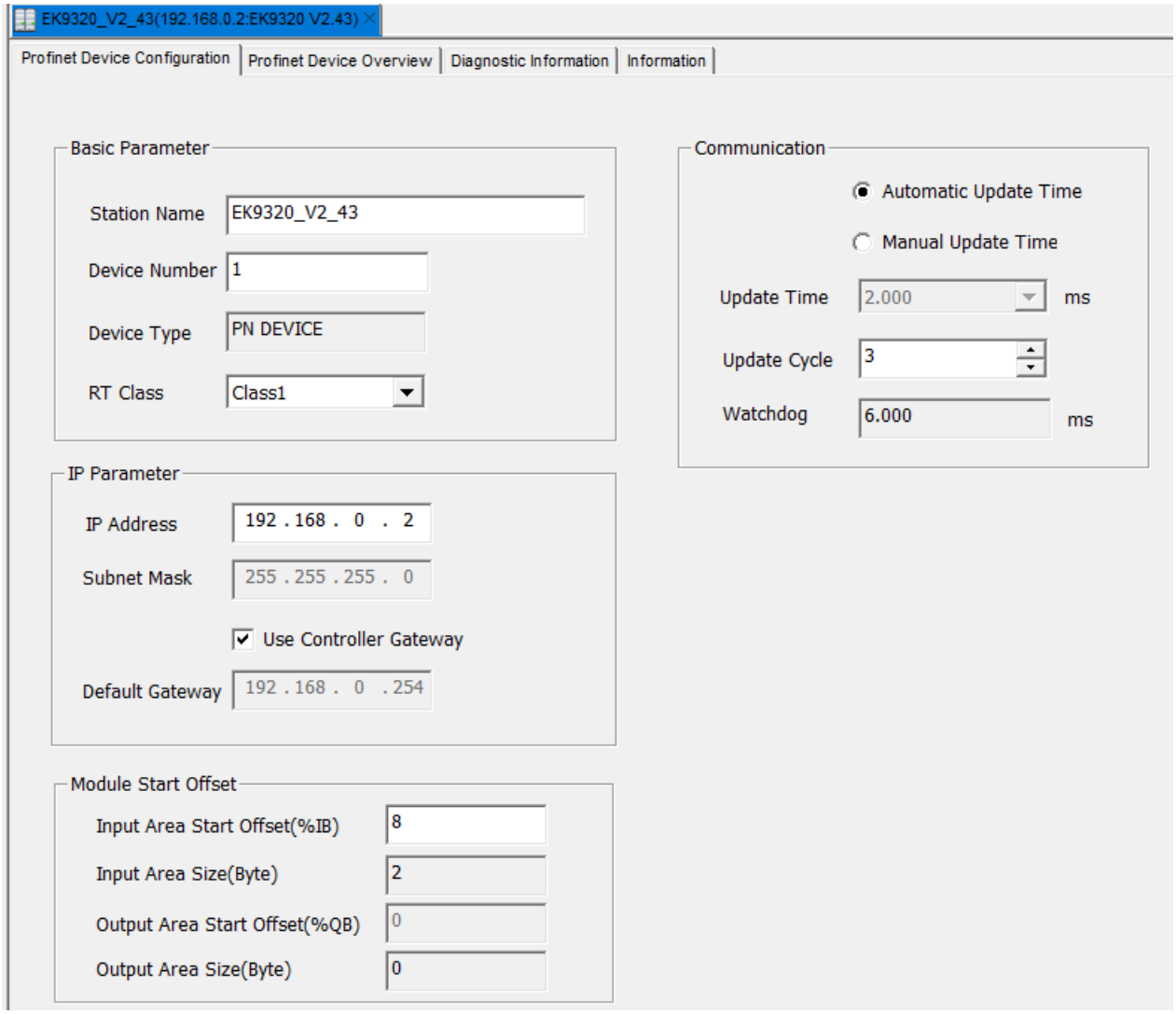
(4) The PROFINET slave and its modules will be added under the "PN_CONTROLLER" node.



PROFINET modules can be added manually by right-clicking the PROFINET slave, selecting the "Add Device" command, and then choosing the device to add in the pop-up "Add" dialog box. This completes the addition of the PROFINET slave module.

11.3.2.2 Configuration of PROFINET Slave

Double-click the PROFINET slave node to configure the slave parameters: "Profinet Device Configuration", " Profinet Device Overview", "Diagnostics Information", "Information". As shown in the figure below:



■ PROFINET Device Configuration

□ Basic Parameters

Site Name: Generated by default, users can modify it. Changes to the site name do not synchronize with the management tree node name or the device name in the overview. Conversely, changes to the management tree node name or the device name in the overview do not affect the site name in the configuration interface. However, the management tree node name and the device name in the overview are synchronized. When modifying the site name, a duplicate check is required to ensure it does not conflict with any other node device names under the same PROFINET master station.

Device Number: Generated by default, unique, and modifiable by users within the range of 1~2047.

Device Type: Non-modifiable.

RT Class: RT type.

□ Communication

Update Time: The update time for data exchange between the master and slave stations. When "

Automatic Update Time " is selected, this field cannot be modified—the update time will be automatically calculated based on the configuration, and the result will be displayed in the box. When " Manual Update Time " is selected, the range can be set to [1, 512].

Update Cycle: When no data update is detected after exceeding the set number of cycles, the slave station may be offline. The update cycle range is [3, 255].

Watchdog: If no data update is detected after the slave station goes offline and exceeds the set watchdog time, the slave station will be judged as having a watchdog timeout. The watchdog time for a PN slave station cannot be manually changed, as its calculation depends on the update time multiplied by the update cycle.

- IP Parameter: Modifiable based on actual requirements for configuring IP address information.
- Use Controller Gateway: When checked, the default gateway cannot be modified and will be consistent with the master station's gateway. When unchecked, the default gateway can be modified.
- Module Start Offset

Input Area Start Offset (%IB): In offline mode, the start address of the PROFINET slave input channel can be modified.

Input Area Size (Byte): Number of bytes in the input area.

Output Area Start Offset (%QB): In offline mode, the start address of the PROFINET slave output channel can be modified.

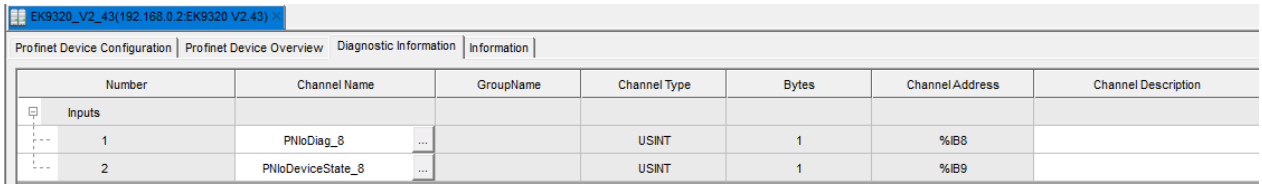
Output Area Size (Byte): Number of bytes in the output area.

- For a PROFINET slave station to establish communication with a PROFINET master station, it is necessary to configure the device name and IP address of the PROFINET slave station. The configuration of the device name and IP address for the PROFINET slave station requires the use of third-party configuration software.

■ PROFINET Device Overview: Displays information about slots, channels, modules, and submodules under the PROFINET slave. This interface information is read-only. If module or submodule information changes, this interface must be updated accordingly.

Show I/O Channels: When checked, all channel information is displayed.When unchecked, only module information is displayed, without channel details.

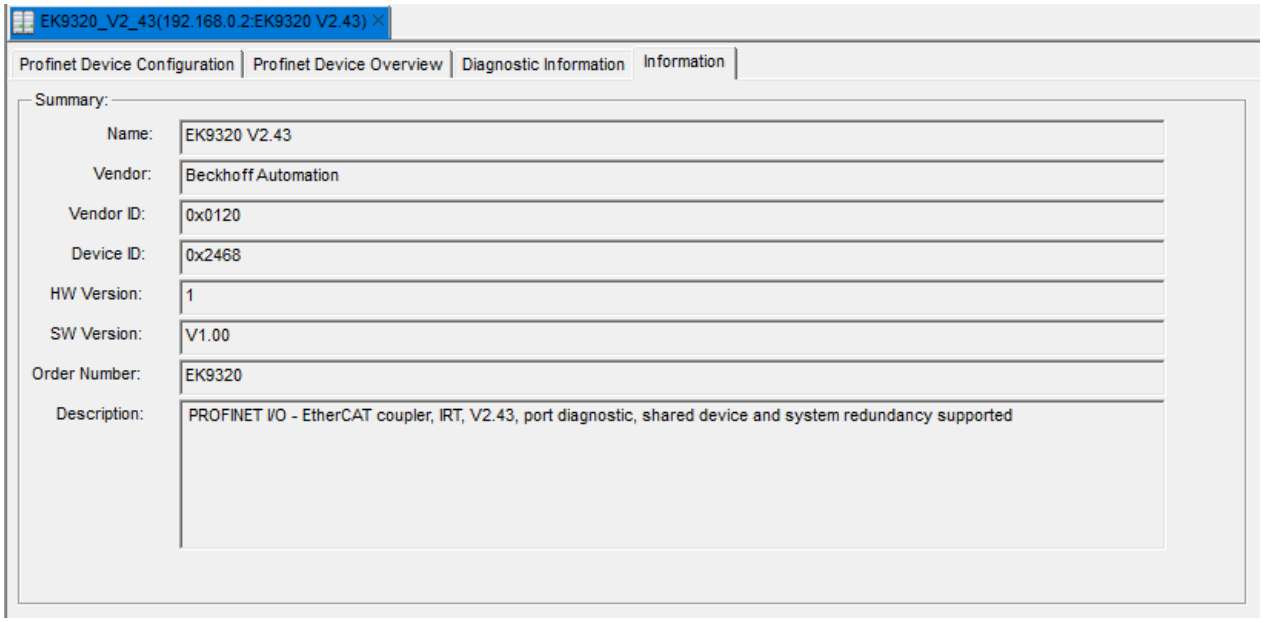
■ Diagnostic Information



For detailed diagnostic information, refer to the periodic diagnostic instructions for the PROFINET slave.

■ Information

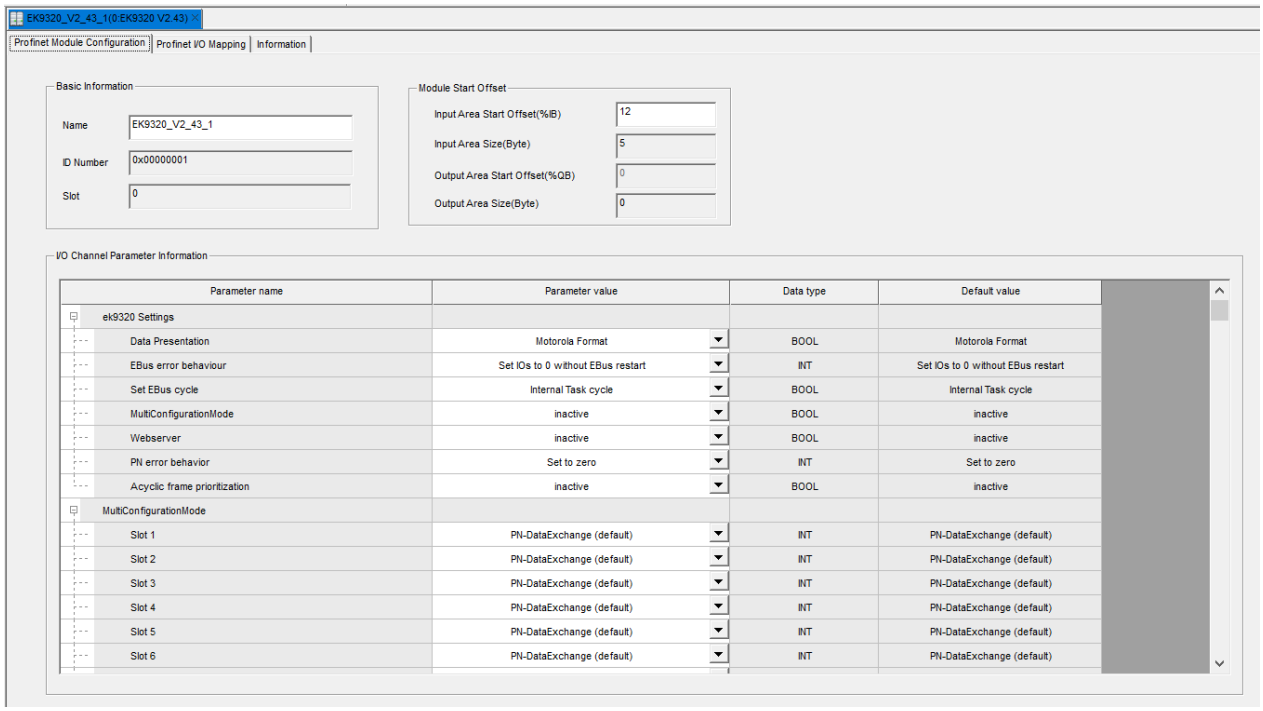
Relevant information about the PROFINET slave in the configuration file.



11.3.2.3 Configuring the PROFINET Slave Module

Double-click the PROFINET slave module node to configure the slave parameters:

The configuration interface includes "Profinet Module Configuration", " Profinet I/O Mapping", and "Information". As shown in the figure below:



- Profinet Module Configuration: Display Basic Information and Channel Parameter Information
  - Basic Information

Name: Generated by default, but can be modified by the user and must be unique. Changing this name does not synchronize with the management tree node name or the device name in the slave overview. Similarly, modifying the management tree node name does not synchronize with the station name in the configuration interface. However, changes to the management tree node name and the device name in the slave overview are synchronized. When modifying the name, a duplicate check is required to ensure it does not conflict with any other node device names under the same PROFINET master.

ID Number: Not modifiable.

Slot: Not modifiable; it is a fixed value determined by the insertion position.

- Channel Parameter Information: Data Source from Configuration File
- Module Start Byte Offset

Input Area Start Offset (%IB): In offline mode, the start address of the PROFINET module input channel can be modified.

Input Area Size (Byte): The number of bytes in the input area.

Output Area Start Offset (%QB): In offline mode, the start address of the PROFINET module output channel can be modified.

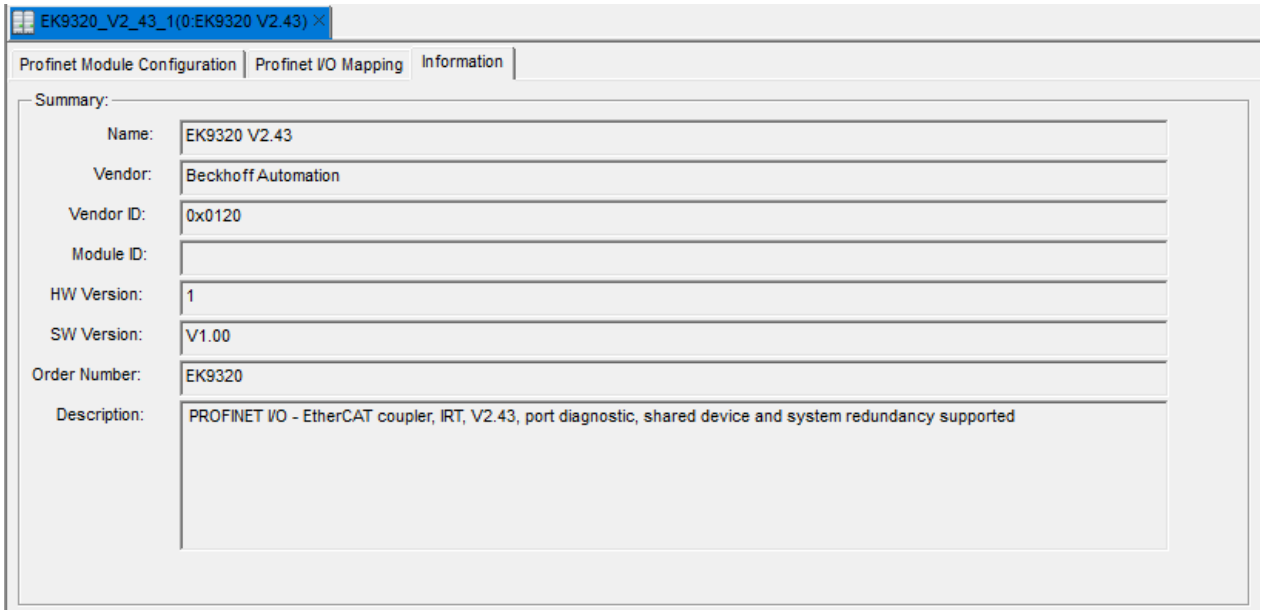
Output Area Size (Byte): The number of bytes in the output area.

- PROFINET I/O Mapping: Displays Channel Information, this section includes channel name, type, byte count, address, and description. The channel name and channel description can be modified.

If input channels exist, the last row is fixed to display the validity of the input data, named DataState, with the type Byte. 0 indicates invalid input data, and 1 indicates valid input data.

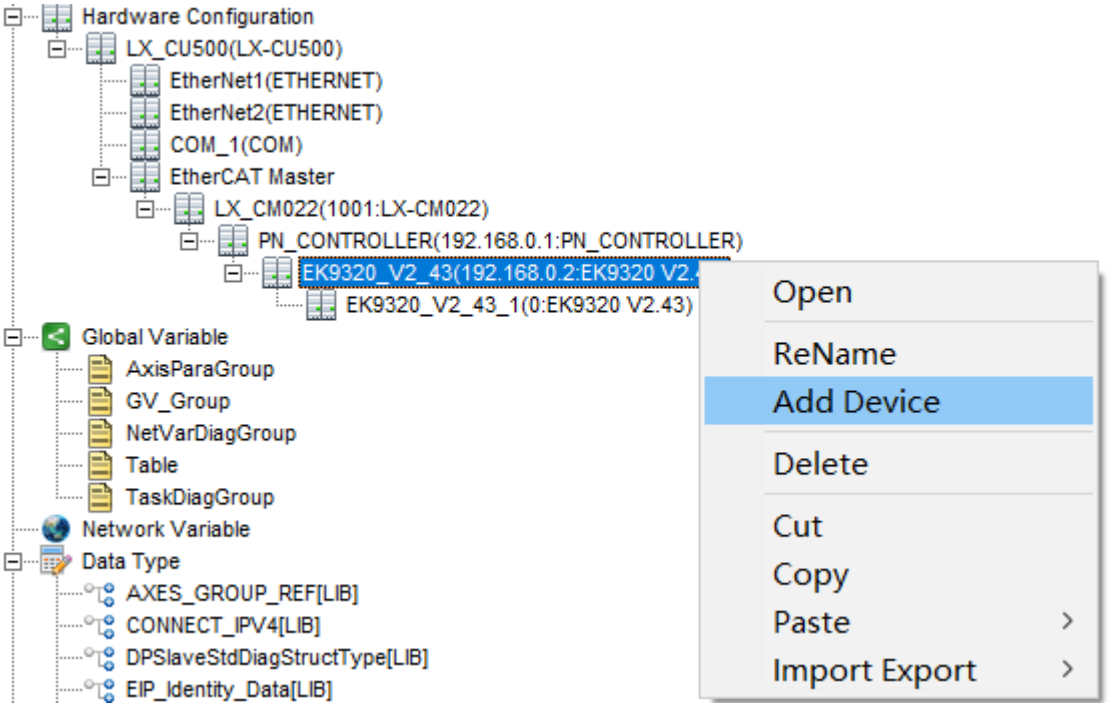
- Information

Displays relevant information about the PROFINET slave module from the configuration file.

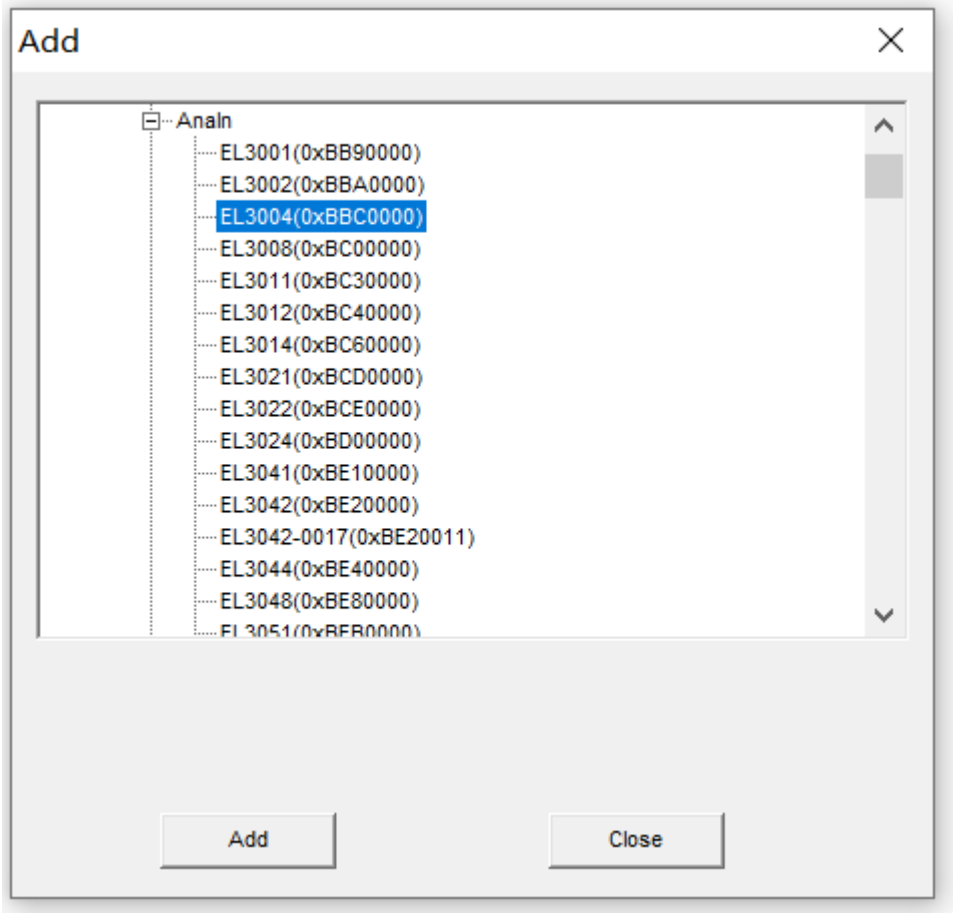


11.3.2.4 Adding PROFINET Slave Submodule

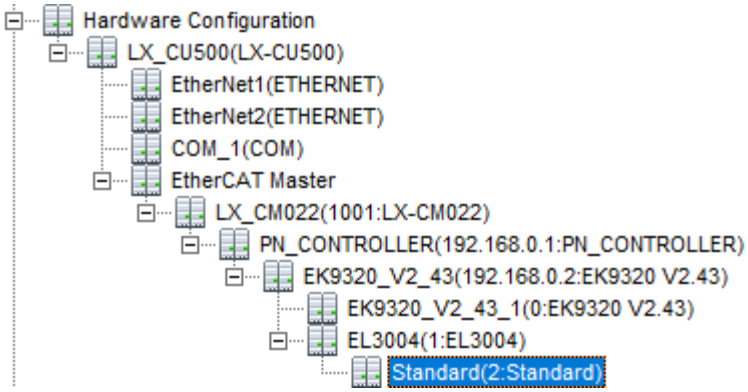
(1) Right-click the PROFINET slave tree node and select the "Add" command, as shown in the figure below.



(2) In the pop-up "Add" dialog box, select the device you want to add and click the "Add" button.



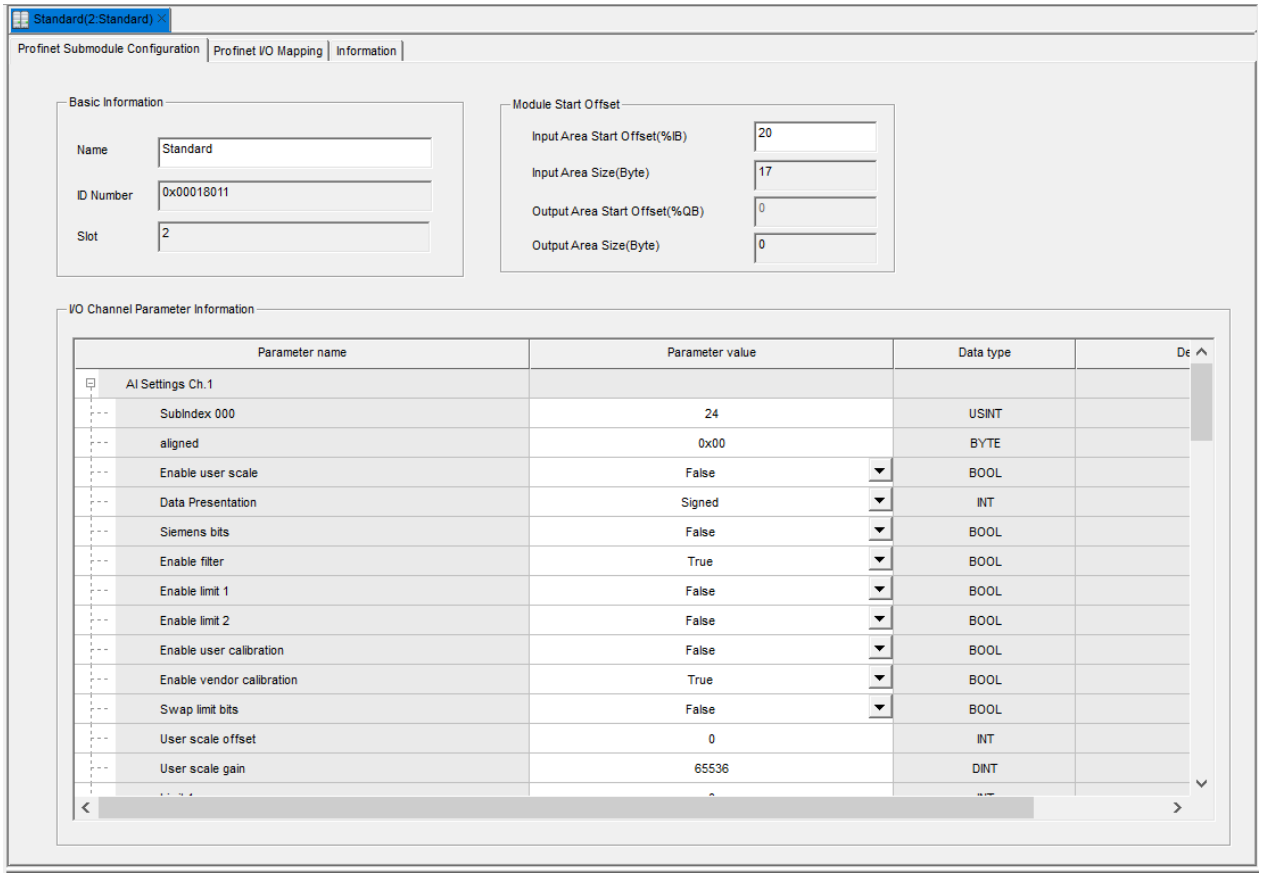
(3) The PROFINET slave submodule has been added under the "PN_CONTROLLER" node.



11.3.2.5 Configure The PROFINET Slave Submodule

Double-click the PROFINET slave submodule node to configure the slave parameters:

The configuration interface includes "Profinet Submodule Configuration", " Profinet I/O Mapping", and "Information", as shown in the figure below:



- PROFINET Submodule Configuration: Display Basic Information and Channel Parameter Information for the PROFINET Slave Submodule

  - Basic Information

    Name: Generated by default, but can be modified by the user and must be unique. Changing this name does not synchronize with the management tree node name or the device name in the slave overview. Similarly, modifying the management tree node name does not synchronize with the station name in the configuration interface. However, changes to the management tree node name and the device name in the slave overview are synchronized. When modifying the name, a duplicate check is required to ensure it does not conflict with any other node device names under the same PROFINET master.

    ID Number: Not modifiable.

    Slot: Not modifiable; it is a fixed value determined by the insertion position.

    Channel Parameter Information: Data Source from Configuration File

  - Module Start Offset

    Input Area Start Offset (%IB): In offline mode, the start address of the PROFINET slave submodule input channel can be modified.

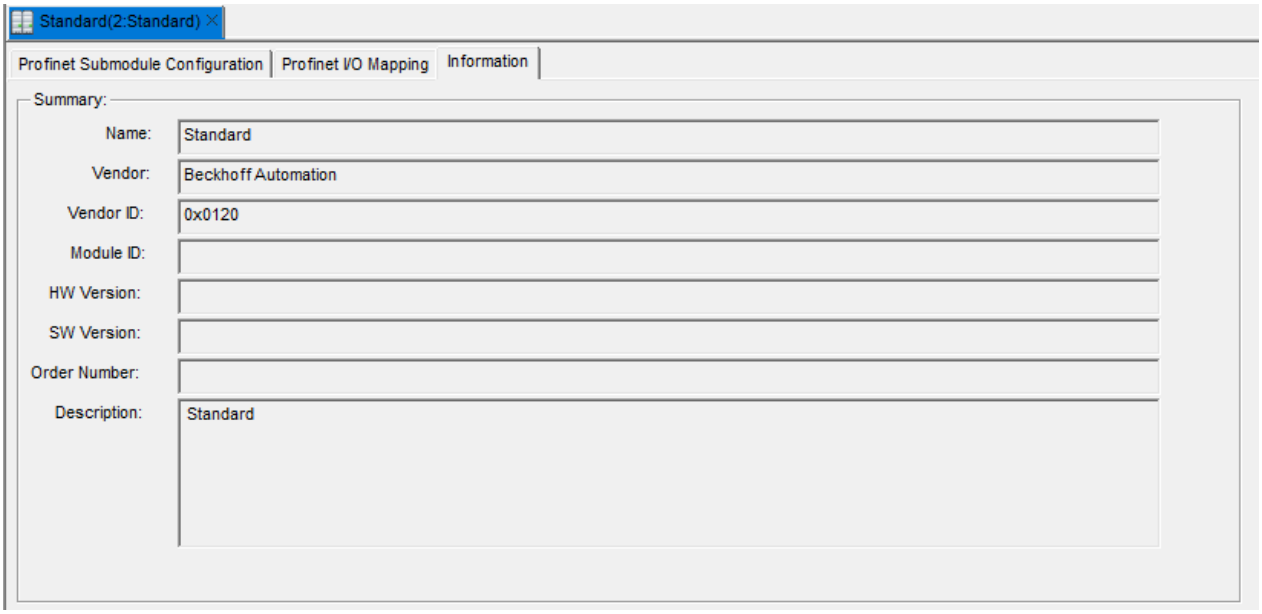
    Input Area Size (Byte): The number of bytes in the input area.

    Output Area Start Offset (%QB): In offline mode, the start address of the PROFINET slave submodule output channel can be modified.

Output Area Size (Byte): The number of bytes in the output area.

- Profinet I/O Mapping: Displays Channel Information: this section includes channel name, type, byte count, address, and description. The channel name and channel description can be modified.
- Information

Displays relevant information about the PROFINET slave submodule from the configuration file.



11.4 PROFINET Communication Example

11.4.1 Example Functionality

This example demonstrates data read/write between a PROFINET master communication module (LX-CM022) and a PROFINET slave LX-EC007 module.

11.4.2 System Composition

- Controller: LX-CU500, Firmware Version LX-CU500-B02.
- PROFINET Master: Communication Module LX-CM022.
- PROFINET Slaves: LX Series Compact Controller LX-CU311 (Firmware Version LX-CU311-A01)
- Expansion Card LX-EC007
- Configuration Software:
  - AutoThink V3.2.3SP2
  - Intelligent AutoThink V5.0.0A5

- PRONETA 3.8.0.0

11.4.3 Programming

Establish a PROFINET project, following the configuration steps outlined in the setup instructions.

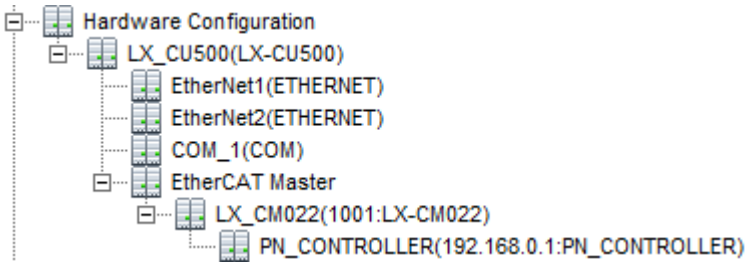
11.4.4 Online Debugging

11.4.4.1 Slave Configuration

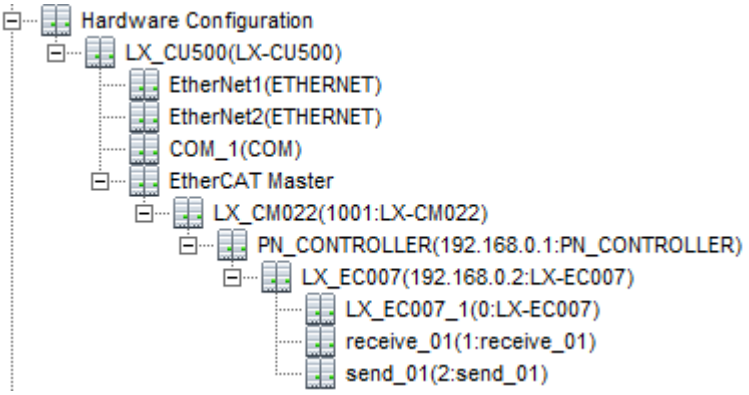
In Intelligent AutoThink, after completing the configuration of the PROFINET slave LX-EC007 parameters, compile the project and download it to the controller to establish communication between LX-CU311 and LX-EC007. Finally, export the device description file of LX-EC007.

11.4.4.2 Master Configuration

(1) In AutoThink, create an LX-CU500 project. Right-click the "EtherCAT" node and select "Scan for Slaves" to add the LX-CM022 module under the EtherCAT node.



(2) Go to "Project" → "Import Device Description File", then import the GSD file of Slave Configuration to add LX-EC007 under LX-CM022.



For detailed instructions on adding master/slave stations in AutoThink, refer to the PROFINET Configuration.

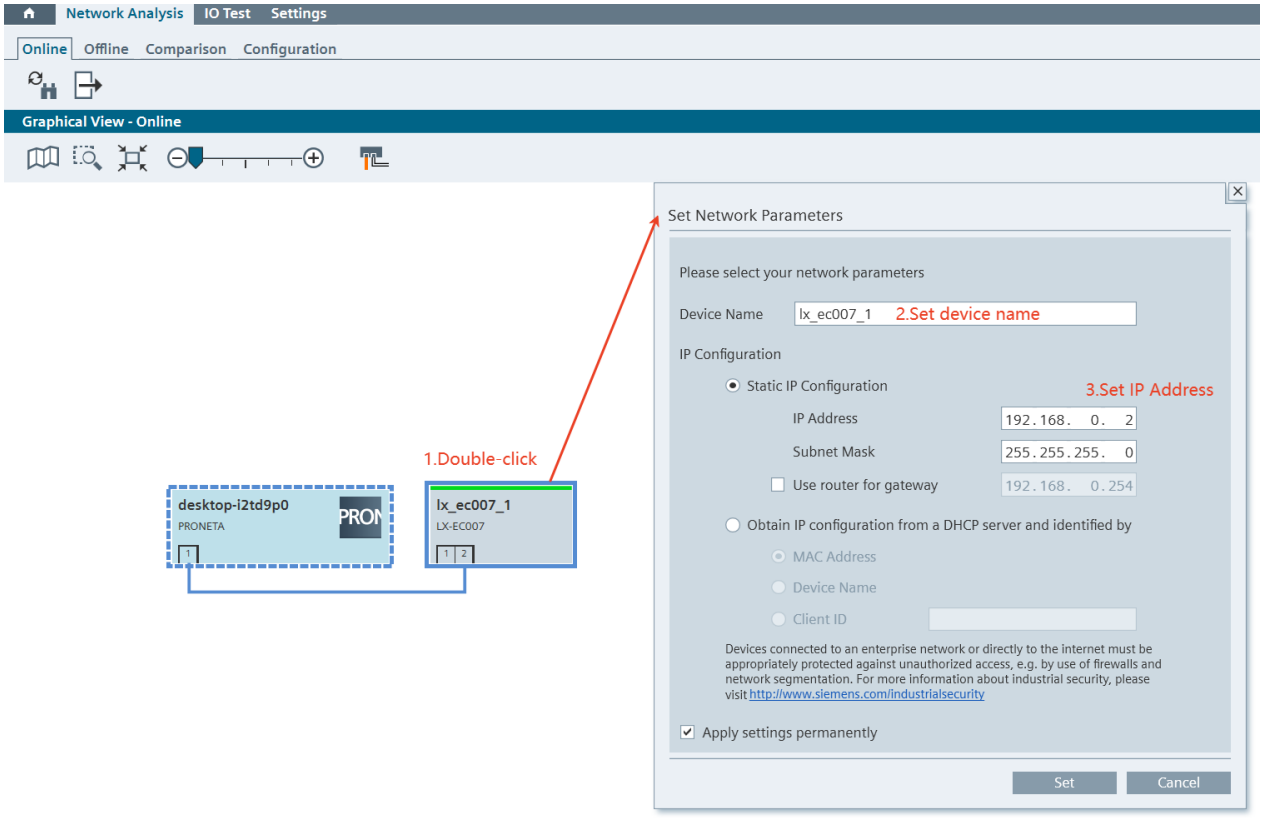
11.4.4.3 Assign Device Name and IP Address

In PRONETA, assign the device name and IP address to ensure they match the slave station configuration in AutoThink.

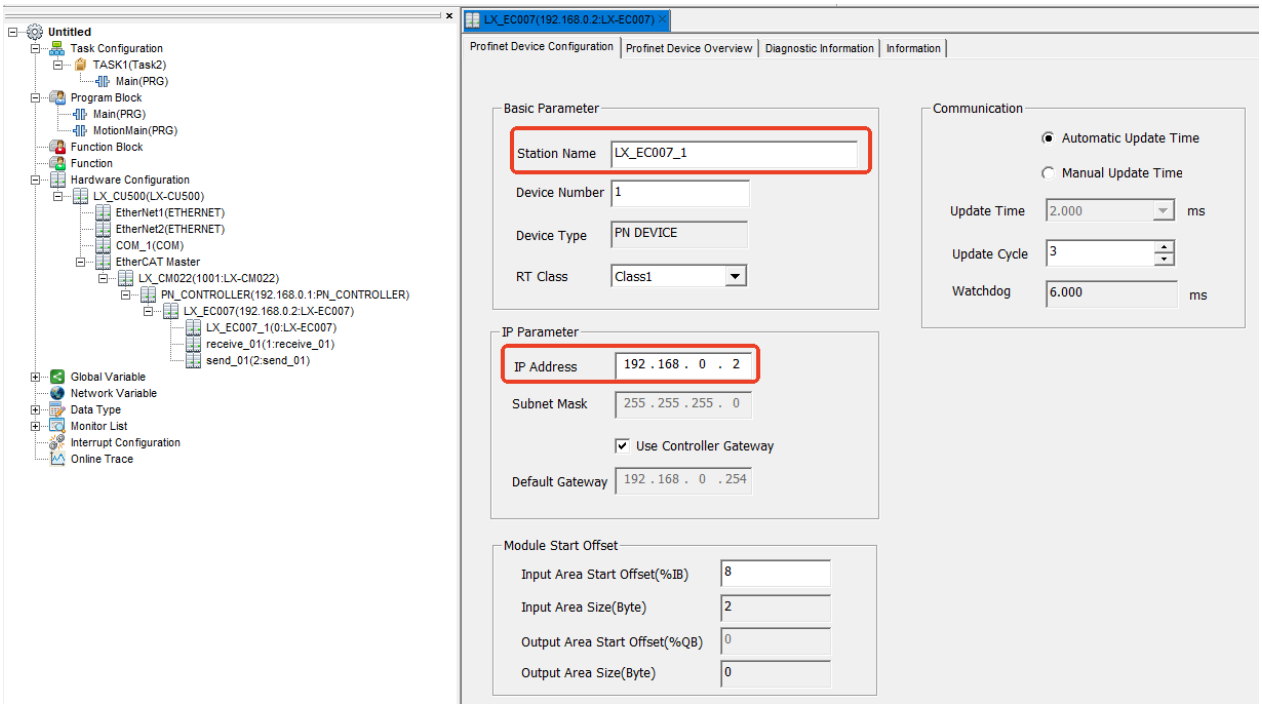
(1) Navigate to Network Analysis → Online, then click the button  to establish communication between PRONETA and LX-EC007.

- If the connection fails, verify that the correct network adapter is selected in PRONETA.

(2) Double-click LX-EC007 to open the "Network Parameters" dialog. Set the Device Name to: lx_ec007_1 (Note: PRONETA treats device names as case-insensitive). Set the IP Address to: 192.168.0.2, then click "Apply" to confirm the configuration.



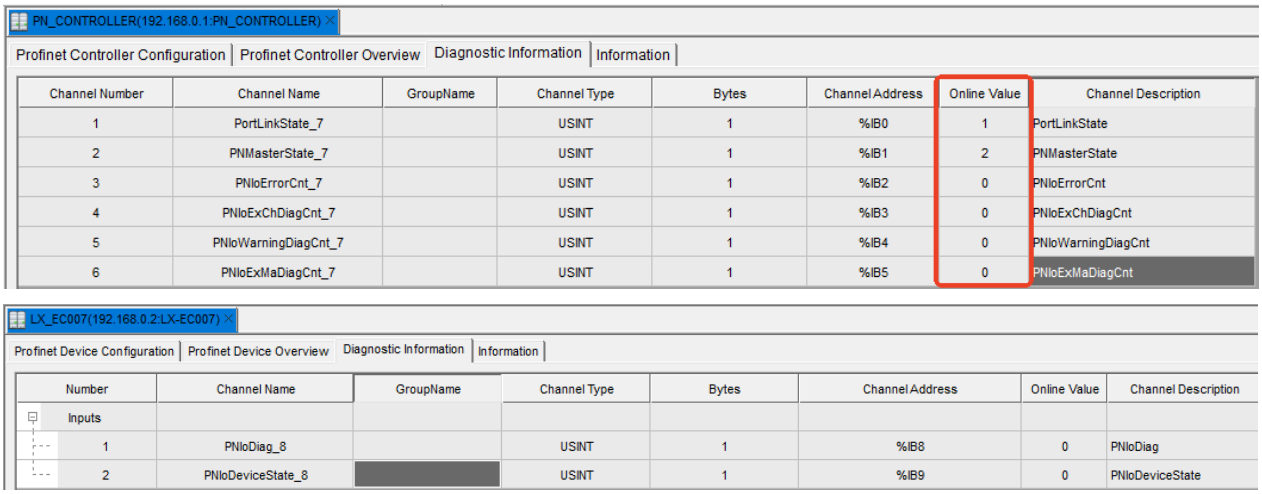
(3) The configuration settings for LX-EC007 in AutoThink are shown in the figure below.



11.4.5 Online Debugging

After the project is created, download the programs to the master and slave PLCs separately.

In monitor mode, double-click the PROFINET master or slave station to view and diagnose using online values.



11.5 Fault Diagnosis

11.5.1 Master Station Module Diagnosis

The periodic diagnostics of the PROFINET master station include network port link status, encountered abnormal diagnostics, channel diagnostics, warning diagnostics, and module mismatch diagnostics and status of the PROFINET master protocol stack. The specific definitions of the periodic diagnostics for the PROFINET master module are shown in the table below.

Periodic Diagnostic Description for the PROFINET Master Module

Diagnosis	Diagnostic Description	Diagnostic Value
PNIoErrorCnt	Number of PN slave modules with abnormal diagnostics	If PNIoDiag Bit 0 ≠ 0 or the value in Table 14-6 of PNIoDeviceState ≠ 0, increment the PNIoErrorCnt value by 1.
PNIoExChDiagCnt	Number of PN slave modules with channel diagnostics	If PNIoDiag Bit 1 ≠ 0, increment the PNIoExChDiagCnt value by 1.
PNIoExMaDiagCnt	Number of PN slave modules with module mismatch diagnostics	If PNIoDiag Bit 3 ≠ 0, increment the PNIoExMaDiagCnt value by 1.
PNIoWarningDiagCnt	Number of PN slave modules with warning diagnostics	If PNIoDiag Bit 2 ≠ 0, increment the PNIoWarningDiagCnt value by 1.
PortLinkState	Network port link status	Network port link status definitions: 0: Link down 1: Link up
PNMasterState	Status of the PROFINET master protocol stack	PROFINET master configuration and protocol stack status: 0: No master cnfiguration profile 1: IP address or MAC error 2: Protocol stack running normally 3: Protocol stack stopped 5: Protocol stack has experienced an operational abnormality

Aperiodic diagnostics include diagnostics related to the PROFINET master network port. The specific definitions of the diagnostics are shown in the table below.

PROFINET Master Network Port Diagnostic Definitions

Diagnosis	Diagnostic Description	Diagnostic Value
TxFrameCnt	Total number of transmitted packets	Report based on diagnostic values
RxFrameCnt	Total number of received packets	Report based on diagnostic values
RxErrorCnt	Number of erroneous received packets	Report based on diagnostic values
PhyPayload	Current network load value	Report based on diagnostic values

11.5.2 PROFINET Slave Diagnostics

Each PN slave periodic diagnostic includes connection status, diagnostic flags, and exception diagnostics. The specific definitions of the diagnostics are shown in the table below.

Periodic Diagnostic Instructions for PROFINET Slave

Diagnosis	Diagnostic Description	Diagnostic Value
PNIoDiag	Bit0: Connection status between PN slave and PN master Bit1: Indicates whether channel diagnostics exist for the PN slave Bit2: Indicates whether alarm diagnostics exist for the PN slave Bit3: Indicates whether a module mismatch diagnosis has occurred for the PN slave	Bit0: 0: Connection normal 1: No connection Bit1: 0: PN slave is operating normally, no channel diagnostic information generated 1: PN slave has channel diagnostic information Bit2: 0: PN slave is operating normally, no alarm diagnostic information generated 1: PN slave has alarm diagnostic information Bit3: 0: PN slave is operating normally, no module mismatch diagnostic information generated 1: PN slave has module mismatch diagnostic information
PNIoDeviceState	0: No error 1: IP address conflict 2: Station name not unique 8: Unable to set IP 9: Device not found 11: Watchdog timeout 13: Communication interrupted 16: Incorrect VendorID / DeviceID	-

Explanation of PROFINET Slave Abnormal Diagnostics

PNIoDeviceState Diagnostic Description	Diagnostic Value
No error	0
IP address conflict	1
Non-unique station name	2
Unable to set IP	8
Device not found	9
Watchdog timeout	11
Incorrect VendorID/DeviceID	16

Explanation of Slave Module Alarm Diagnostics

Diagnostic Description	Diagnostic Value
No alarm	0
Module removed	3

Module inserted	4
Module insertion error	10

Explanation of PROFINET Slave Module Mismatch Diagnostics

Diagnostic Description	Diagnostic Value
Device number	If a slave station encounters a module mismatch, it reports based on the diagnostic value
Number of mismatched modules	If a slave station encounters a module mismatch, it reports based on the diagnostic value
Slot number of mismatched module	If a slave station encounters a module mismatch, it reports based on the diagnostic value

- Note: To obtain channel diagnostics, warning diagnostics, and module mismatch information, it is necessary to call function blocks for reading. Channel diagnostics and module mismatch information are obtained on a per-module basis, while warning diagnostics retrieve all warning information from all slave stations.

PROFINET slave stations can also perform diagnostics on channels, warnings, and slots using the function blocks PN_Get_Channel_Diag, PN_Get_Warning_Diag, and PN_Get_Mismatch_Diag. Refer to the corresponding documentation for diagnostic methods and explanations.

Chapter 12 EtherCAT Communication

12.1 Network Overview

12.1.1 Introduction

Based on the standard Ethernet physical layer, EtherCAT is an ultra-high-speed Ethernet fieldbus protocol that makes full use of the full-duplex characteristics of Ethernet and changes the original CSMA/CD media access control mechanism. EtherCAT uses a technology called "processing on the fly" to improve bandwidth utilization, so that one data frame per cycle is usually enough to achieve data communication for all nodes. The unique way EtherCAT processes frames makes it the fastest industrial Ethernet technology.

12.1.2 Network Characteristics

- EtherCAT uses a standard Ethernet interface to perform automatic address allocation, thus no IP address settings are required.
- EtherCAT is an industrial Ethernet technology with high performance, low cost, simple application, and flexible topology.
- The EtherCAT slave station device has at least two Ethernet interfaces to forward data from one network port to another.
- It can perform accurate calibration of distributed clocks, so that the "master station clock" can simply and accurately determine the operating compensation for each "slave station clock", so as to enable precise synchronization of the servo axes when executing simultaneous linkage tasks.
- The EtherCAT transmission mechanism is called "On The Fly". After the slave station processes its own data, it forwards the data to the next slave station until the processing completion message is returned to the master station. Generally speaking, only one data frame is enough for all nodes on the network to send and receive data.
- The EtherCAT network supports the Safety over EtherCAT protocol, which meets the requirements of IEC61508 and can reach the functional safety level SIL3. Both standard information and safety information can be transmitted simultaneously on the same communication bus.

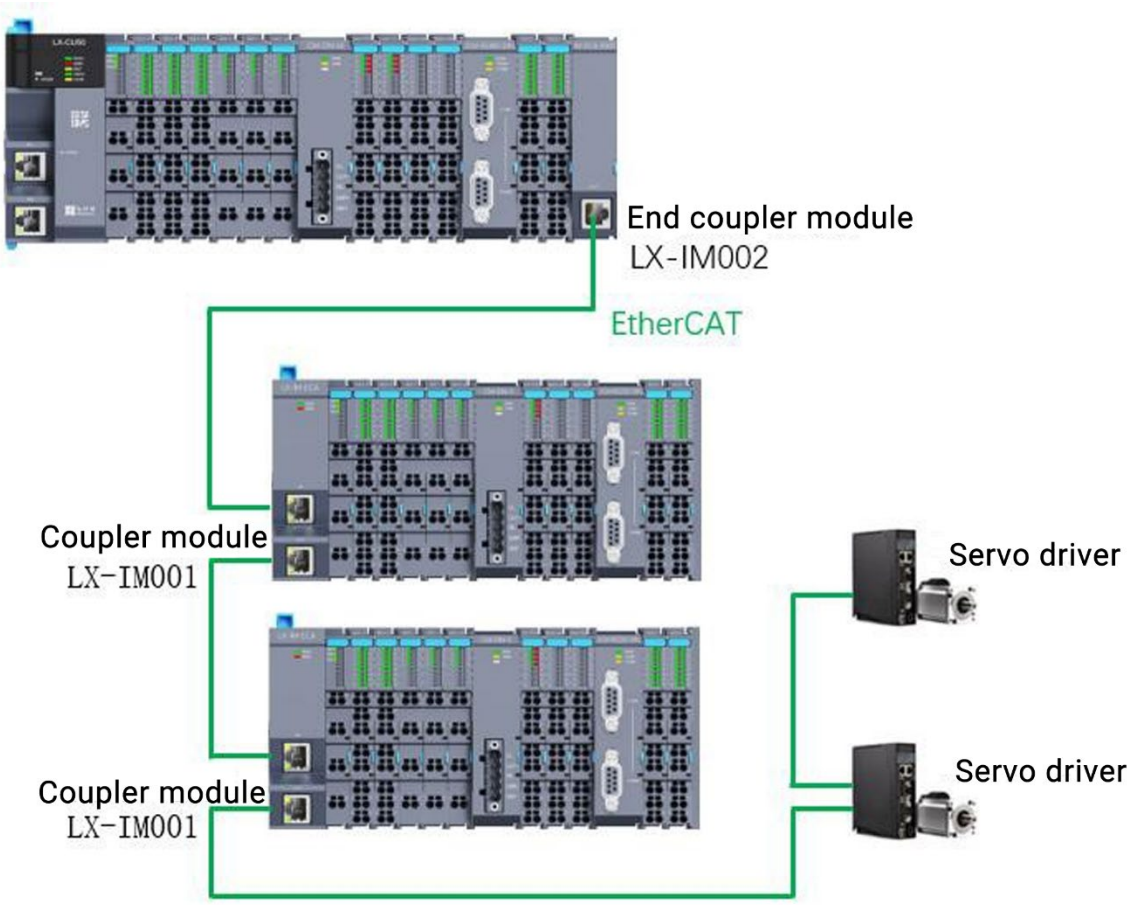
12.2 Network Topology

The EtherCAT network of the LX series CPUs can support device connection topologies such as daisy chain and star, and the physical medium can be 100Base-TX standard Ethernet cables or optical cables. When the 100Base-TX cable is used, the distance between stations can reach 100 m.

The total number of EtherCAT slave stations supported by LX-CU500 is 300, of which the maximum number of drive-type slave stations is 64.

12.2.1 Daisy Chain Network Topology

Since dual Ethernet ports are the standard configuration of EtherCAT slave station devices, a daisy chain network topology can be formed through slave station communication modules and servo drivers. as shown below:

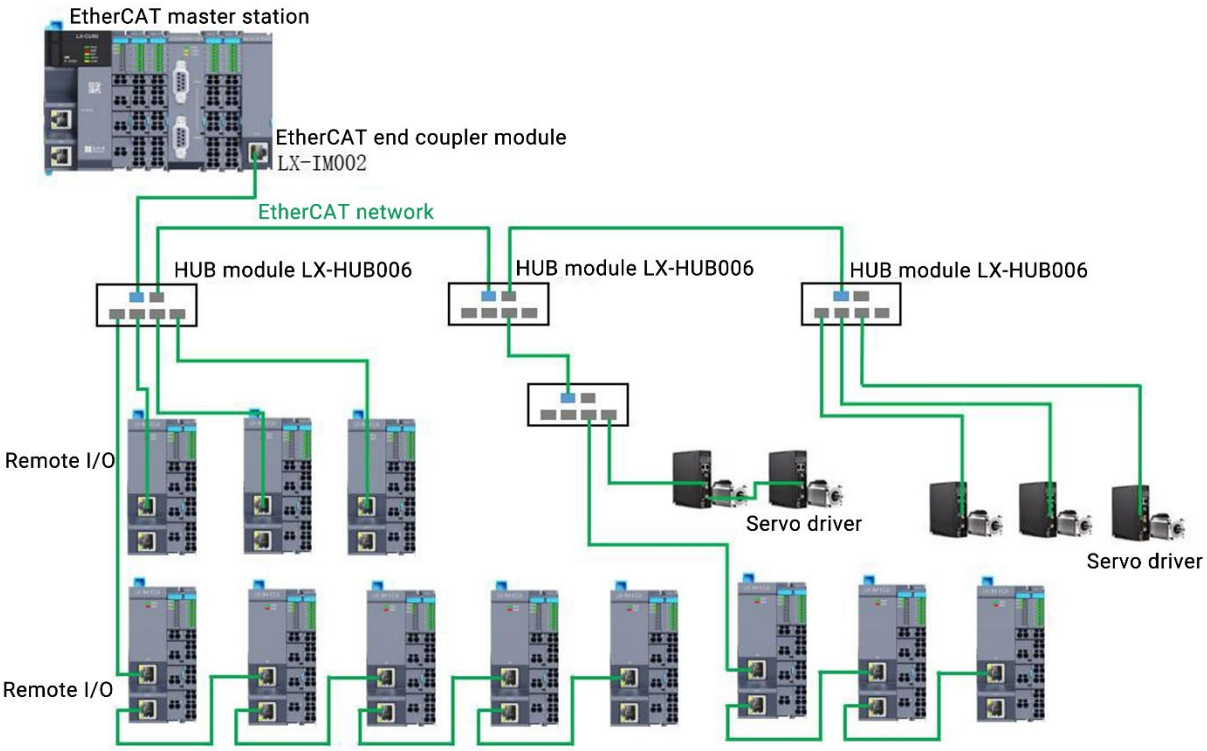


Requirements for communication cables:

Cable Model	Specifications	Communication Distance
Shielded twisted-pair cable	CAT5e	100 m

12.2.2 Branch Network Topology

EtherCAT can be branch connected through the LX-HUB-006 module (6-channel EtherCAT HUB module). The combination of daisy chain and star connection makes field installation and wiring more flexible and easier to manage and maintain. When using LX-HUB-006 for cascading, please note that during hardware configuration, the number of levels of parent nodes and child nodes shall not exceed 12 levels. The network topology diagram is as follows:



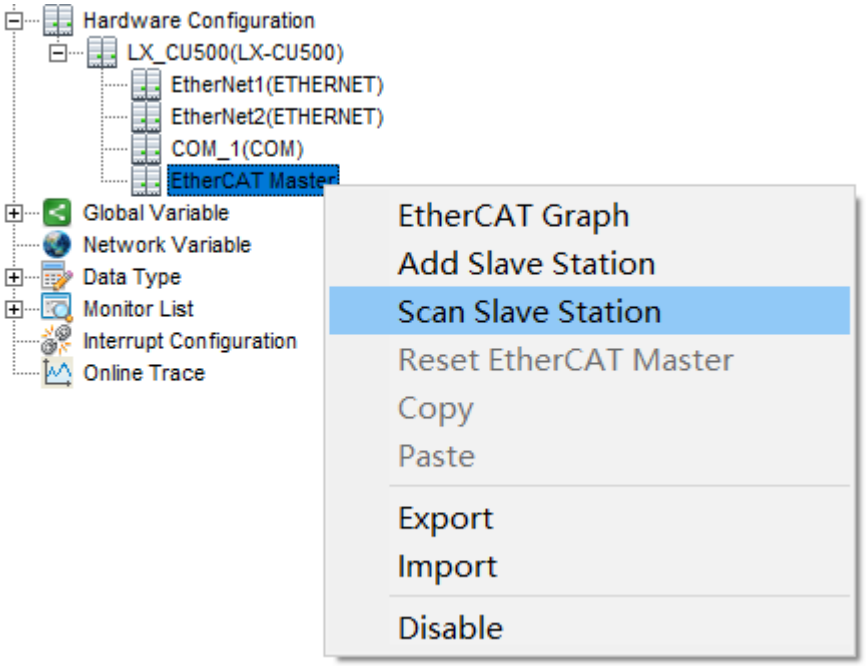
Requirements for communication cables:

Cable Model	Specifications	Communication Distance
Shielded twisted-pair cable	CAT5e	Maximum 100 meters

12.3 Configuration

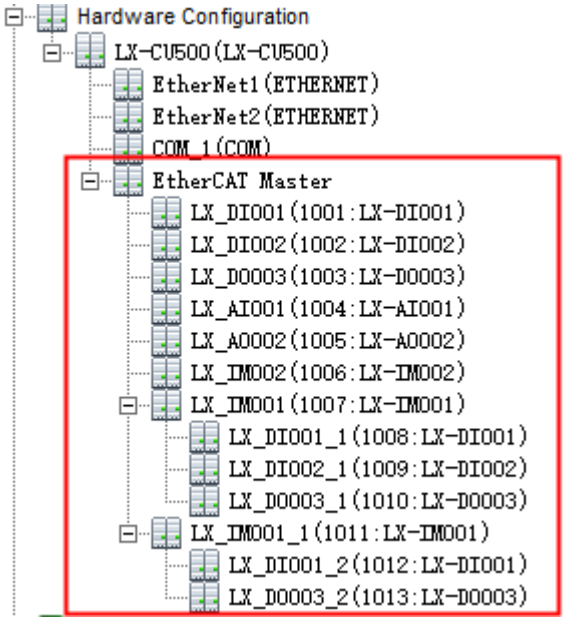
12.3.1 Automatic Scanning Device

Right-click the EtherCAT Master node, select Scan Slave Station, and the Scan Slave Station dialog box is displayed, as shown in the figure:


Hardware Configuration
LX_CU500(LX-CU500)
EtherNet1(ETHERNET)
EtherNet2(ETHERNET)
COM_1(COM)
EtherCAT Master
Global Variable
Network Variable
Data Type
Monitor List
Interrupt Configuration
Online Trace
EtherCAT Graph
Add Slave Station
Scan Slave Station
Reset EtherCAT Master
Copy
Paste
Export
Import
Disable

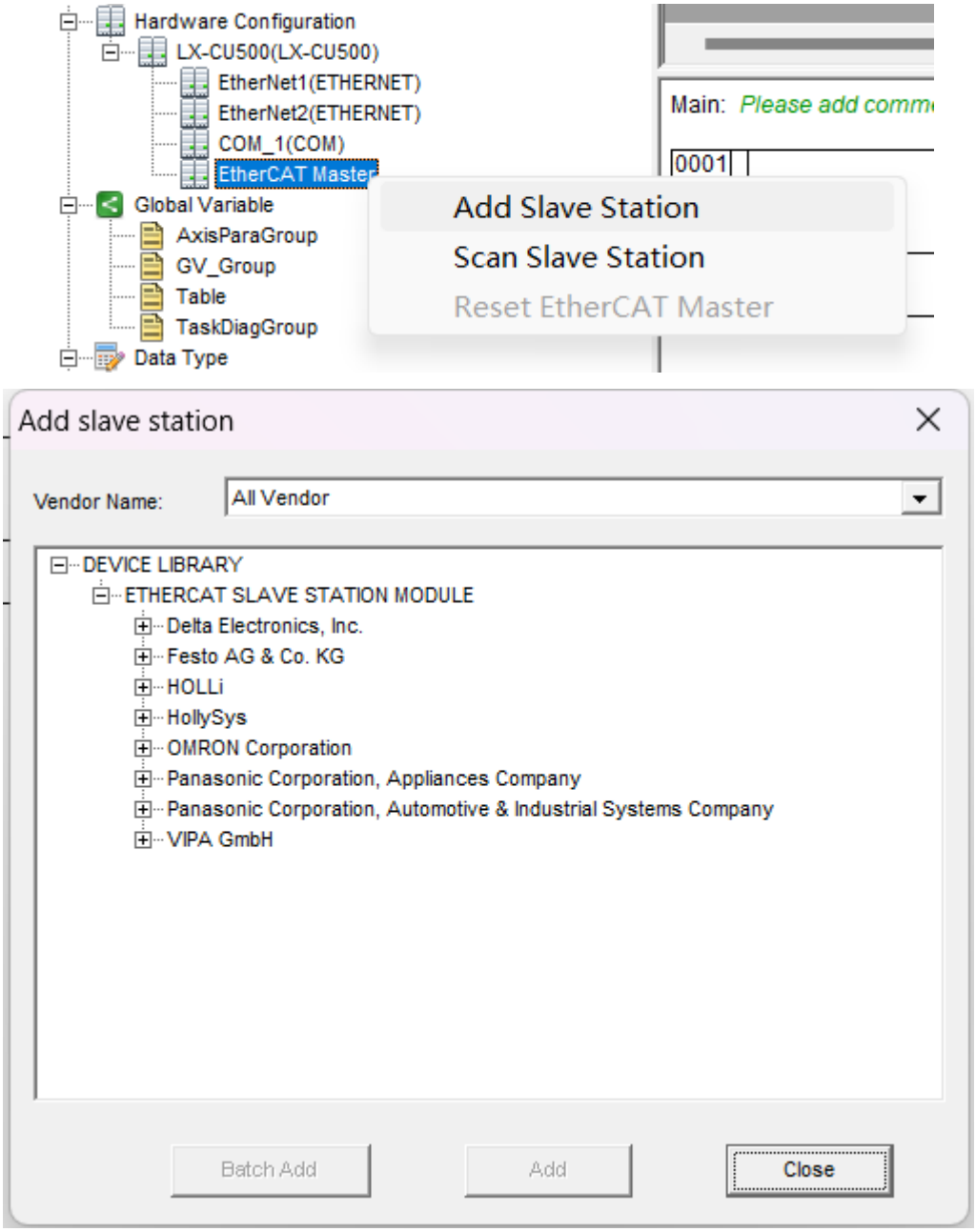
Select Start Scan. After the scan is completed, the EtherCAT Master Topology dialog box is displayed. Click the Add button, and all scanned modules are automatically added.

The configuration example is as follows:

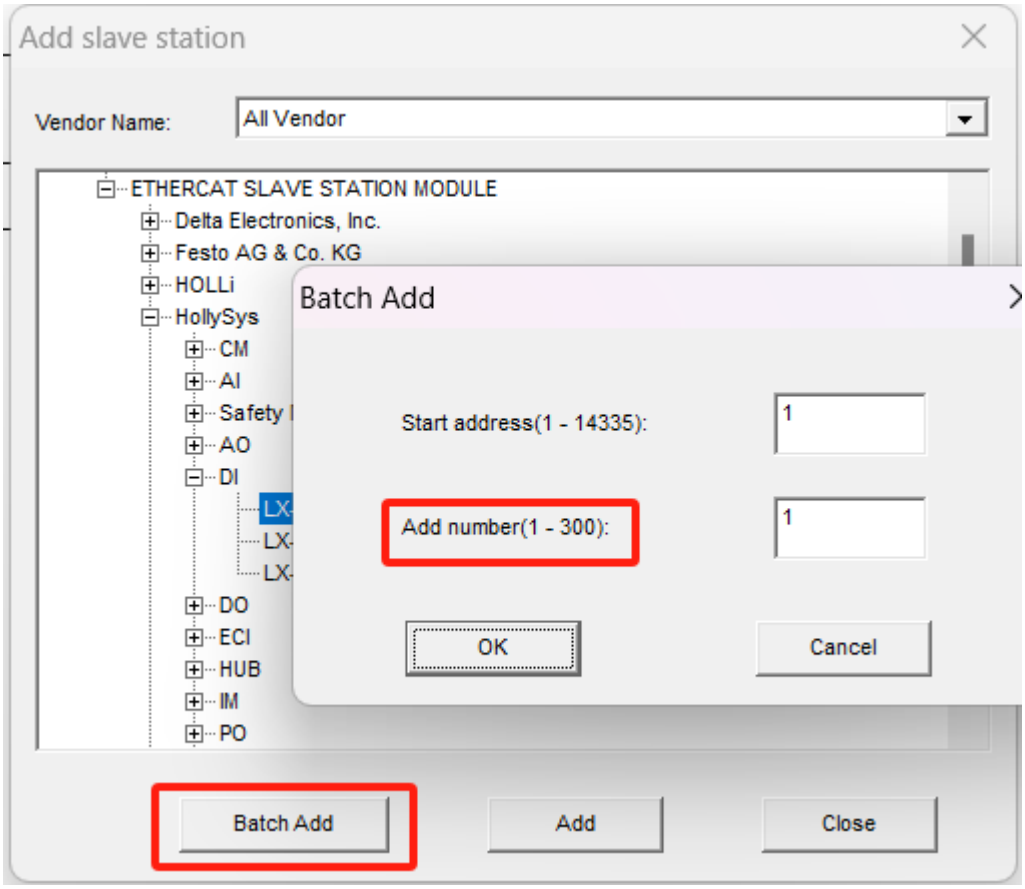


12.3.2 Adding Device Manually

Right-click the EtherCAT Master node, select Add Slave Station, and the Add Slave Station dialog box is displayed, as shown in the figure. Select the module model to be added and select Add. Repeat this operation until the hardware addition is completed.



For consecutively arranged identical modules, you can select the module models to be added and add them in batches. As shown below:



12.3.3 EtherCAT Configuration

12.3.3.1 Configuring EtherCAT master station

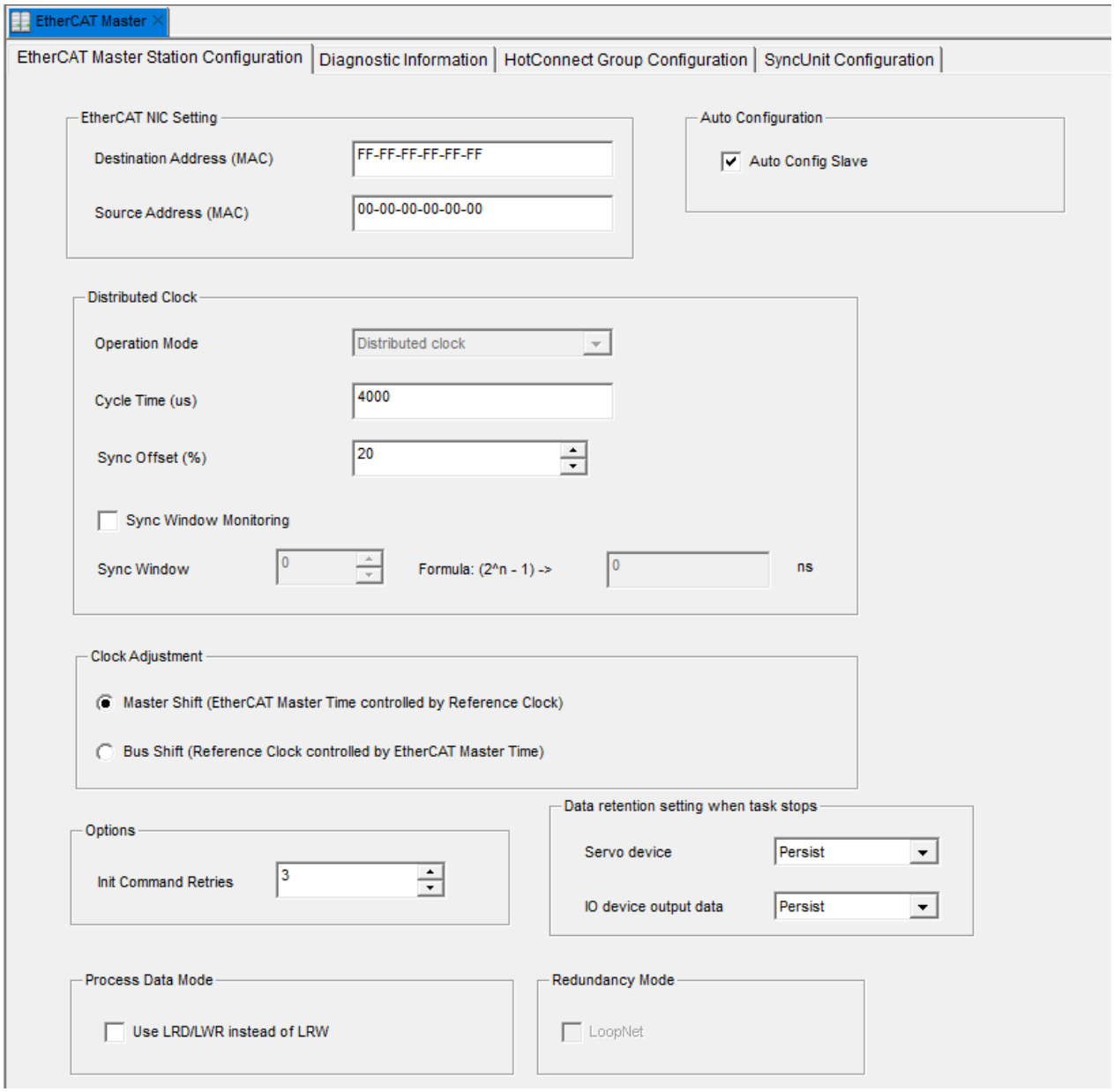
12.3.3.1.1 EtherCAT Master Station Configuration

Double-click the EtherCAT Master master station node, and the master configuration interface is displayed in the right area, as shown in the figure. This interface includes four tabs: EtherCAT Master Station Configuration, Diagnostic Information, HotConnect Group Configuraiton and SyncUnit Configuration.

Note:

- The master station parameters usually use the default settings and do not need to be modified.
- The LX controller performs EtherCAT master initialization, which causes the slave output to be reset first and then output according to the operation logic. The operations related to EtherCAT master station initialization include: controller power on, full download, incremental download (EtherCAT related configuration changes), reset, cold reset, reset EtherCAT master station, and BusReset instruction.
- If some slave stations are removed and then reinserted into their original positions after the project is downloaded—either the same stations or replacements with identical models/versions where identity

verification is disabled (i.e., neither the removed nor the replacement slaves have identity verification enabled)—the EtherCAT master must be reinitialized before they can go online.



- EtherCAT NIC settings
  - Destination address (MAC): MAC address of the EtherCAT network member that shall accept the message, FF-FF-FF-FF- FF-FF (default).
  - Source address (MAC): MAC address of PLC, 00-00-00-00-00-00 (default).
- Auto Configuration

  Enable automatic slave address assignment. If disabled, the slave address can be manually configured; otherwise, it will be automatically assigned by the system.
- Distributed clock

- Run mode: Distributed clock (default).
- Cycle time: The time interval at which the master periodically sends data frames, in microseconds (μs). In each cycle, the master sends frames, collects input data from slaves, executes the control algorithm, and writes output data. This parameter must be configured to be consistent with the execution cycle of the bound IEC task.
- Synchronization offset: This value allows correction of the offset of the synchronization interrupt within the PLC cycle period of the EtherCAT slave station. Normally, the PLC task cycle starts 20% later than the slave station synchronization interrupt. This means that the PLC task will lag by 80% of the cycle period with no data loss.
- Synchronization window monitoring: If this option is activated, it allows to monitor the synchronization status of the slave station.
- Synchronization window: Synchronization window monitoring time, which is 10 by default.

■ Clock adjustment

- Master station offset: The reference clock controls the EtherCAT master time.
- Bus offset: The EtherCAT master station time controls the reference time.

■ Options

- Number of initialization command retries: Number of retries to handle transmission errors.

■ Data retention settings when task stops

- Servo device: When an IEC task stops, the EtherCAT bus output and Q area status of the servo device can be configured to either remain held or disable the axis enable.
- IO device output data: When an IEC task stops, the EtherCAT bus output and Q area status of the IO device can be configured to either remain held or be cleared.

In versions that do not support this configuration, the default behavior is to retain (hold) the data. The configuration takes effect after download.

Scenarios that trigger task stop include: clicking the stop button in AutoThink, clicking the emergency stop button in AutoThink, or stopping via the dip switch. When a task stop is triggered, forced EtherCAT Q area variables are not cleared.

When AutoThink triggers task runtime (e.g., clicking the run task button, toggling the switch to run, full download, incremental download, reset, cold reset, controller power cycle, etc.), the EtherCAT bus output executes according to the actual value of the Q area.

The following firmware versions support this parameter configuration: LX-CU430 V1.1.0 and above, LX-CU433 V1.1.0 and above, LX-CU500 V1.4.0 and above, LX-CU501 V1.2.0 and above, LX-CU510 V1.1.0 and above, LX-CU511 V1.0.1 and above.

■ Process data mode

- Use LRD/LWR instead of LRW: Determine whether to use the LRD/LWR command or LRW command to access process data. LRD/LWR is required for cable redundancy and LRW for slave replication.

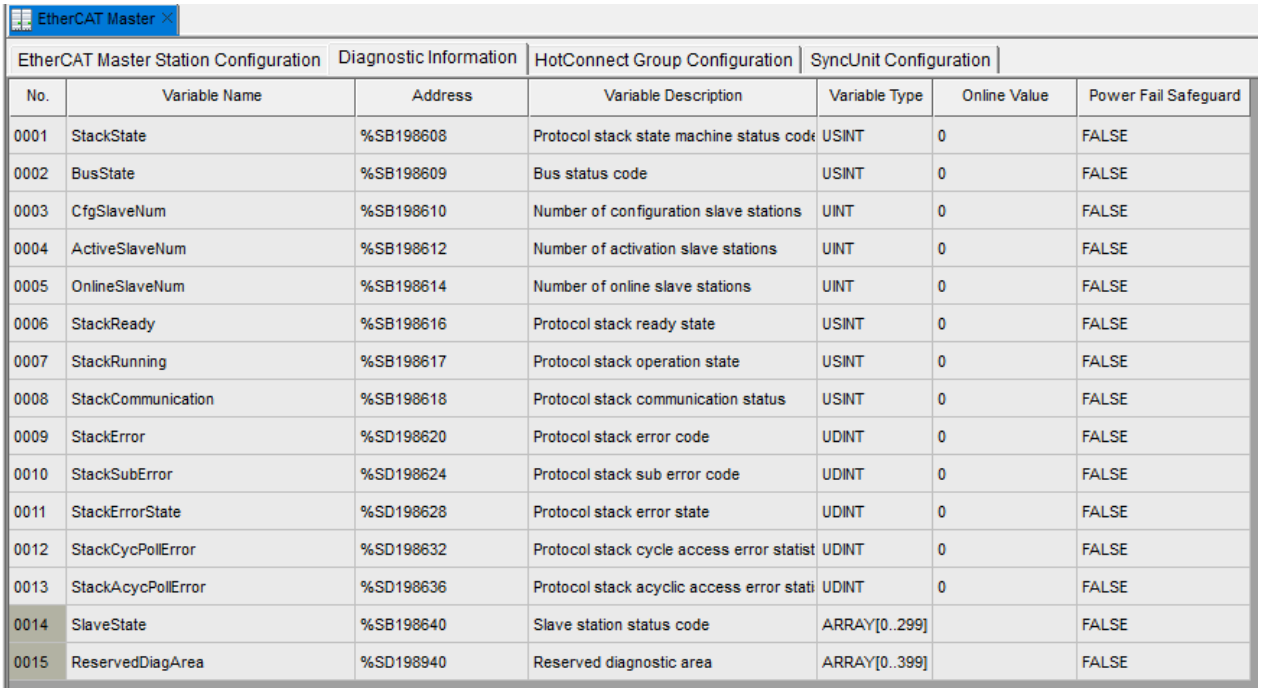
- Redundancy Mode

  Controllers supporting ring topology (e.g., LX-CU501, LX-CU511) are configurable with this option enabled by default. If disabled, the ring network functionality will be unavailable.

12.3.3.1.2 Diagnostic Information

Double-click on the EtherCAT Master node in the project management tree and open the master configuration interface in the right-hand area. Click on the Diagnostic Information tab to view, as shown in the following figure.

Detailed instructions can be found in the EtherCAT Fault Diagnosis.



No.	Variable Name	Address	Variable Description	Variable Type	Online Value	Power Fail Safeguard
0001	StackState	%SB198608	Protocol stack state machine status code	USINT	0	FALSE
0002	BusState	%SB198609	Bus status code	USINT	0	FALSE
0003	CfgSlaveNum	%SB198610	Number of configuration slave stations	UINT	0	FALSE
0004	ActiveSlaveNum	%SB198612	Number of activation slave stations	UINT	0	FALSE
0005	OnlineSlaveNum	%SB198614	Number of online slave stations	UINT	0	FALSE
0006	StackReady	%SB198616	Protocol stack ready state	USINT	0	FALSE
0007	StackRunning	%SB198617	Protocol stack operation state	USINT	0	FALSE
0008	StackCommunication	%SB198618	Protocol stack communication status	USINT	0	FALSE
0009	StackError	%SD198620	Protocol stack error code	UDINT	0	FALSE
0010	StackSubError	%SD198624	Protocol stack sub error code	UDINT	0	FALSE
0011	StackErrorState	%SD198628	Protocol stack error state	UDINT	0	FALSE
0012	StackCycPollError	%SD198632	Protocol stack cycle access error statist	UDINT	0	FALSE
0013	StackAcycPollError	%SD198636	Protocol stack acyclic access error stati	UDINT	0	FALSE
0014	SlaveState	%SB198640	Slave station status code	ARRAY[0..299]		FALSE
0015	ReservedDiagArea	%SD198940	Reserved diagnostic area	ARRAY[0..399]		FALSE

12.3.3.1.3 HotConnect Group Configuraiton

The transmission mechanism of EtherCAT is linear transmission, and even for star branches, the internal logic has a sequential linear structure. Therefore, in a non ring network PLC system, if a slave station goes offline or loses power due to loose network cables or other reasons, it will cause other slave stations behind the slave station to be unable to operate normally. At this point, a branch led out by a splitter (such as HUB) can be set as a hot connection group by configuring a hot connection group, so that after the branch is powered off, it will not affect the normal operation of other slave stations. At the same time, when the branch is powered on again, there are no restrictions on the inserted network port (in the scenario of a ring network PLC system, reference can be made to 8.3.3.1.4 SyncUnit Configuration).

Constraint：

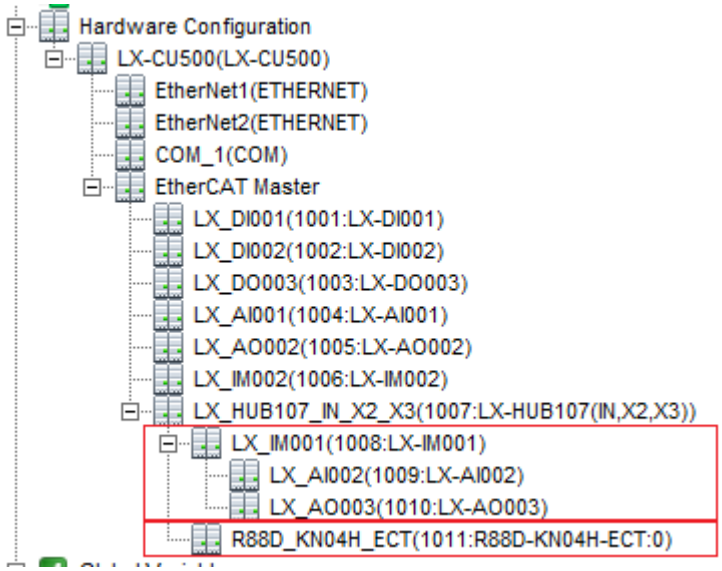
1. The hot connection group needs to be used in conjunction with LX-HUB107, LX-IM001, LX-IM002 and LX-IM003. Based on the actual physical structure, a branch from one port of the HUB can be configured

as a hot connection group. It does not support multiple hot connection groups on the same port of the HUB, nor does it support configuring HUB modules as hot connection groups.

2. When the controller is LX-CU500 or LX-CU501, the functional safety module does not support configuring hot connection groups. There is no such restriction when the controller is LX-CU510 or LX-CU511.
3. Multiple hot connection groups do not support simultaneous power on. Only when all modules in a hot connection group have entered the OP state can the next hot connection group be powered on.
4. Inserting a slave station between two adjacent hot connection groups is not supported. If insertion is required, the hot connection group must be deleted first.
5. When deleting a slave station that has already been configured with a hot connection group, the hot connection group must be deleted first.
6. The rings network linear and the main rings of star topology do not support hot connection.
7. When the controller is LX-CU510 or LX-CU511, the 500us bus cycle does not support hot connection. When the master station has only one slave station, the slave station does not support setting as a hot connection.
8. When the module configuration within a hot-connect group in the topology does not match the software configuration, that hot-connect group and subsequent modules cannot be guaranteed to go online properly, but the modules before this hot-connect group can still go online.
9. If the loss or error of packets during hot-plugging of the connection group causes the connected device (e.g., servo) to become inactive, the device can be reactivated by clearing the axis error via the command EtherCAT_AxisFaultReset in AT (Axis Technology) according to the axis number.

Operation：

Here is an example to illustrate how to configure a hot connection group. The hardware configuration shown in the figure below is to configure the two branches from the X2 and X3 ports of HUB107 into two hot connection groups. Namely: Stations 1008, 1009, and 1010 are configured as a hot connection group; The 1011 slave station is configured as a hot connection group.



Configure slave alias：

Due to the hot connection group being identified as the slave alias of the first module within the group, it is necessary to configure the slave alias of the first module within the group before configuring the hot connection group.

- Note: Slave aliases must be globally unique and cannot be duplicated with other slaves within the group or outside the group.

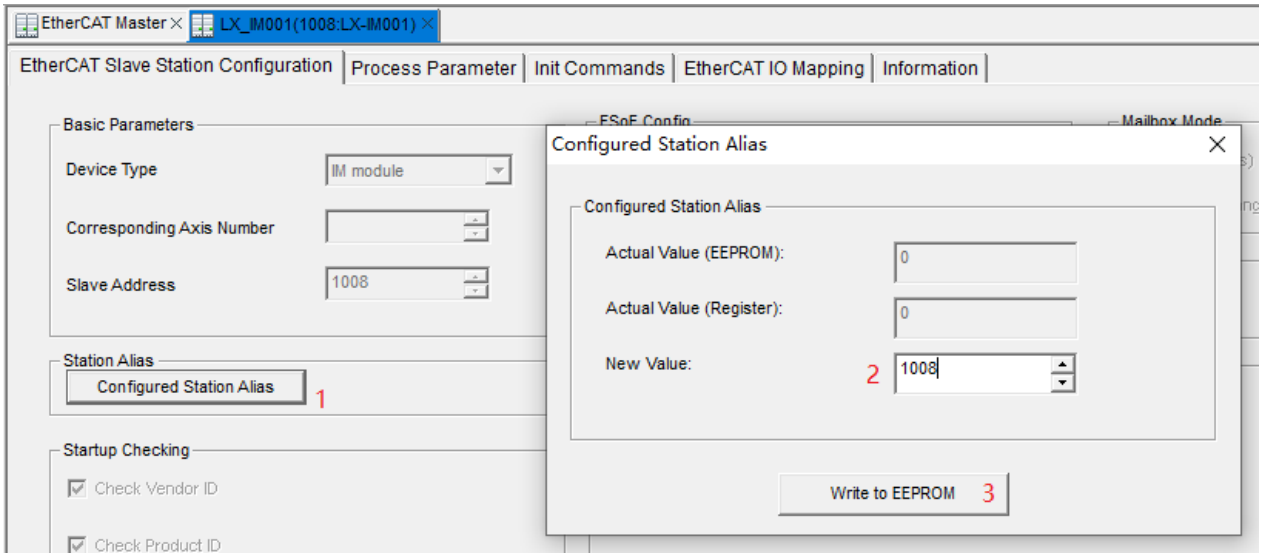
(1) Module without dip switch (such as station 1008 in the above picture):

(a) Double click on the slave to open the "EtherCAT Slave Station Configuration" tab.

(b) In the pop-up dialog box, configure station alias, such as 1008.

(c) Click "Configured Slave Alias" (this button is enabled in monitoring mode).

(d) Click on 'Write to EEPROM'.



(e) After EEPROM writing is completed, perform power-off and power on operations on the module.

(f) Repeat steps (a) to (b) above to confirm that the slave alias has been loaded into the register, i.e. the parameter value of the Actual Value (Register) has been modified to the value set in step (c)

(2) Module with dip switch (such as station 1011 in the above picture):

(a) Set the alias address of the slave station by toggling the dip switch.

(b) Perform power-off and power on operations on the module.

(c) Double click on the slave to open the "EtherCAT Slave Station Configuration" tab.

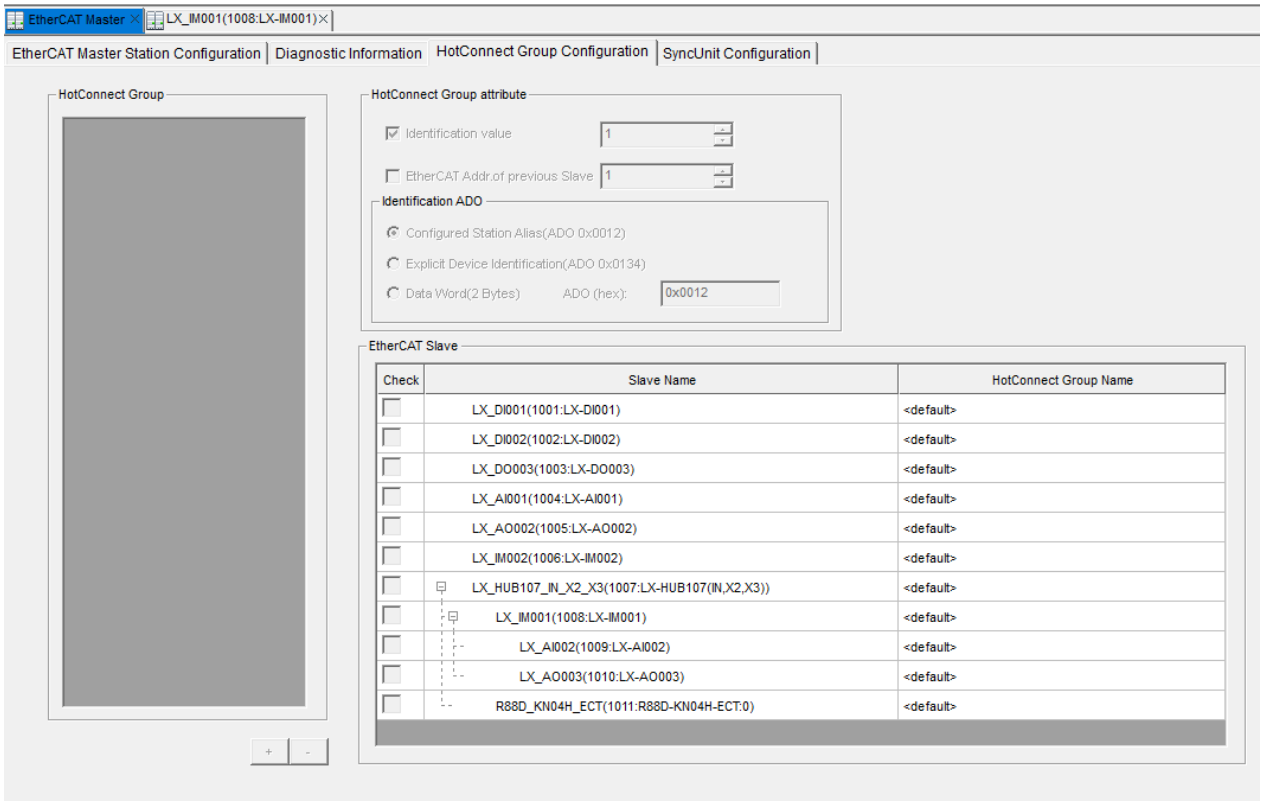
(d) Click on 'Configured Station Alias' (this button is enabled in monitoring mode).

(e) In the pop-up dialog box, confirm that the slave alias has been loaded into the register, that is, the parameter value of the Actual Value (Register) is the value set by the dip switch (there is no need to pay attention to the actual value (EEPROM)).

Configure hot connection group

(1) Double-click on the "EtherCAT Master" master node in the project management tree to open the master configuration interface.

(2) Click on the HotConnect Group Configuraiton tab, as shown in the following figure. The interface parameters are described in the table below.



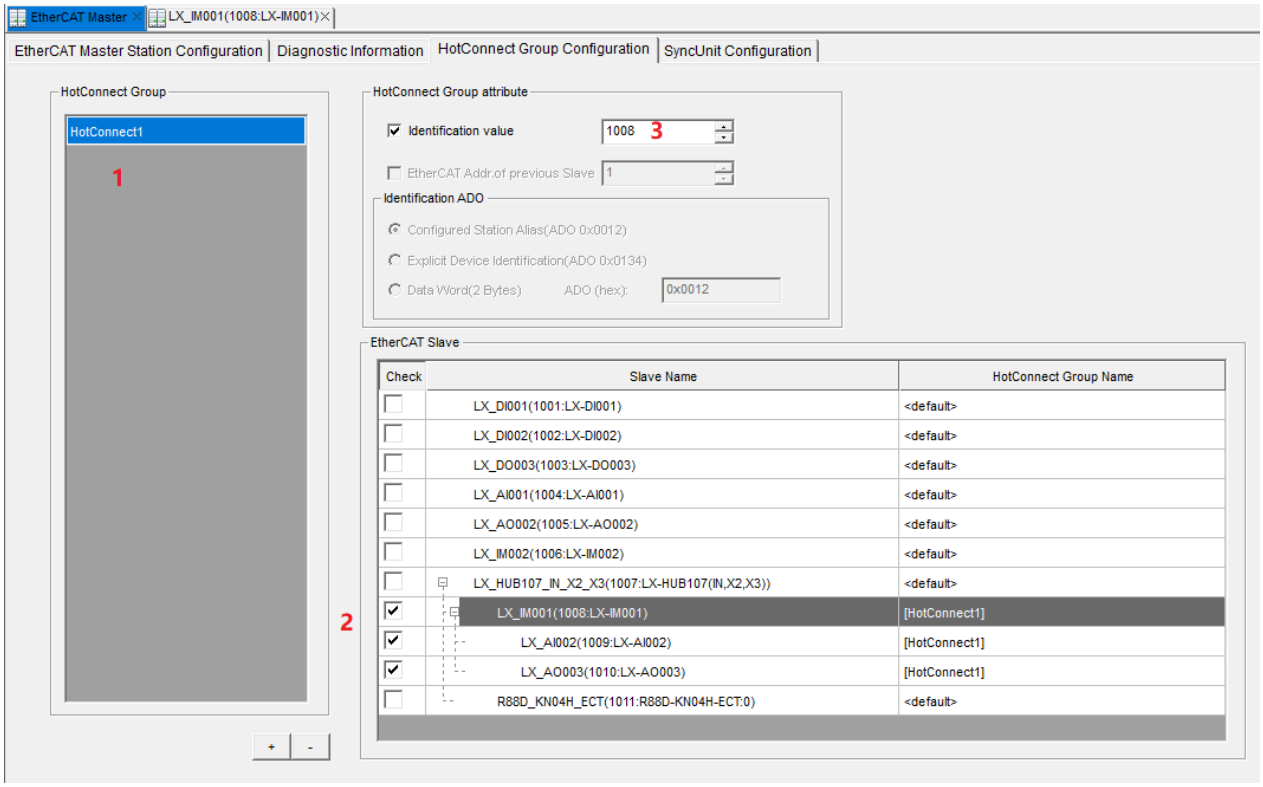
Hot connection group parameter description:

Parameter		Description
HotConnect Group		Hot connection group list.
HotConnect Group attribute	Identification value	The slave alias of the first module within the group.
	EtherCAT Addr. Of previous Slave	The EtherCAT address of the previous slave station.
	Identification ADO	The configuration method for 'HotConnec Group' currently only supports 'Configured Station Alias'.
EtherCAT Slave		Slave list.

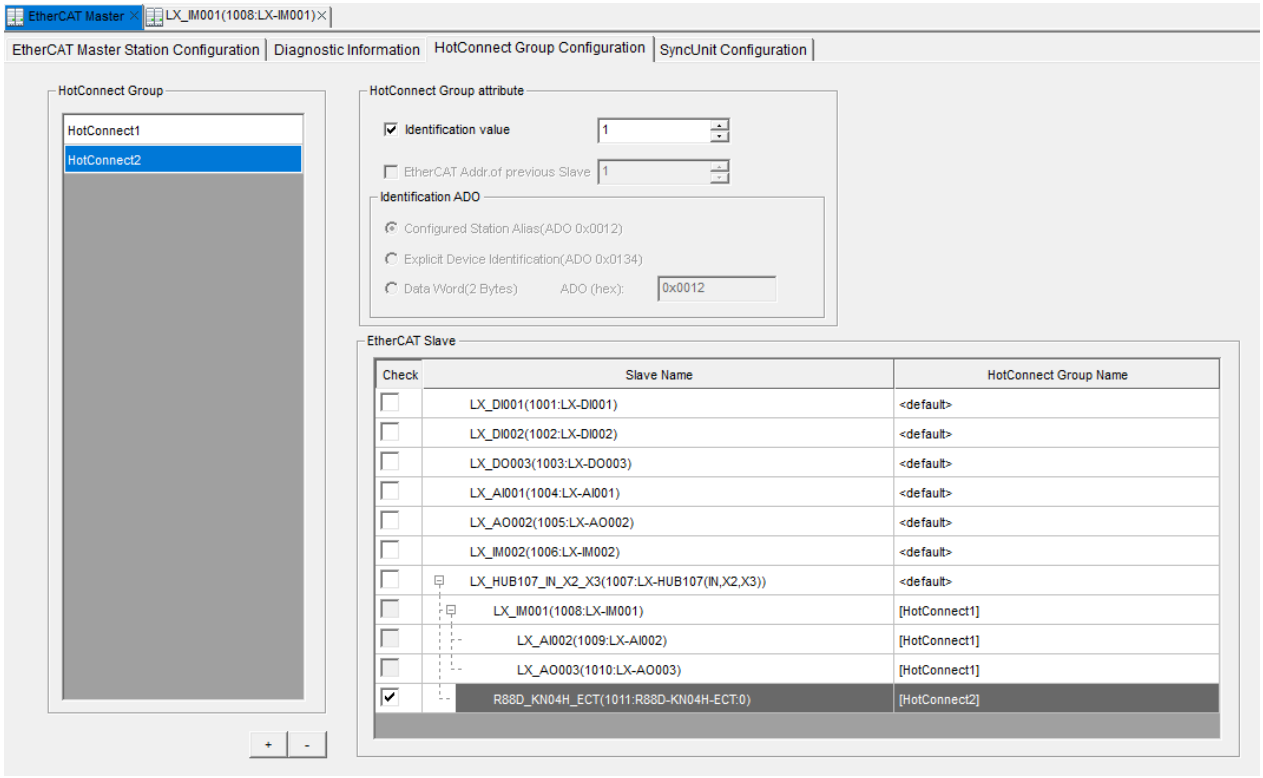
(3) Click + to add a hot connection group.

(4) Select the newly added hot connection group HotConnect1, in the EtherCAT slave area on the right, check the slave stations 1008, 1009, and 1010, and add these three slave stations to the hot connection group HotConnect1.

(5) Configure the 'Identification value' as the slave alias of the first module in the group, such as 1008.



(6) Repeat steps (3) to (5) above to add a second hot connection group, HotConnect2, and add slave number 1011 to the group. Set the 'Identification value' to the slave alias of 1011.



Related Operations:

(1) After inserting a new module into a module that has already been configured with a HotConnect Group, the new module is automatically added to the hot connection group.

(2) When you need to delete a module from a HotConnect Group, you can select the HotConnect Group and uncheck the checkbox before that module.

(3) When you need to delete a hot connection group, select the medium HotConnect Group and click －. After deletion, all modules in the group have no hot connection group configuration information.

12.3.3.1.4 SyncUnit Configuration

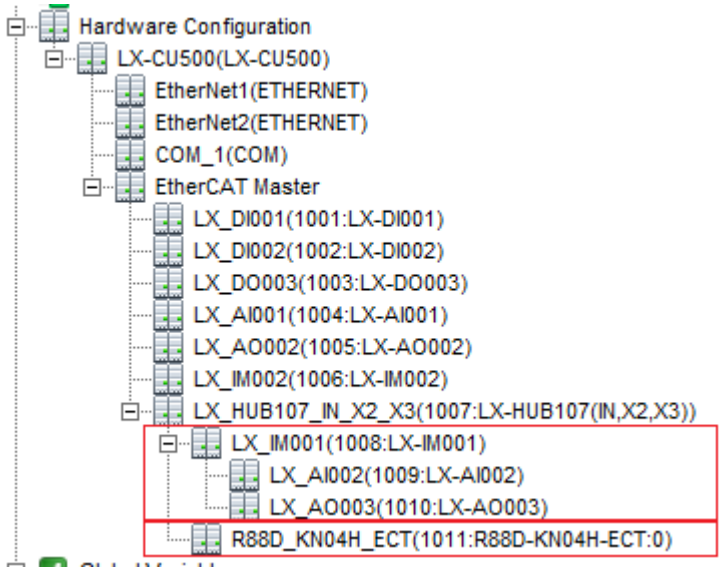
Similar to a HotConnect Group Configuraiton, in a ring network PLC system, a branch led out by a splitter (such as HUB) is set as a synchronization unit to ensure that the normal operation of other slave stations is not affected after the branch is powered off. At the same time, when the branch is powered on again, there are no restrictions on the inserted network port (in non ring network PLC system scenarios, refer to 8.3.3.1.3 HotConnect Group Configuraiton).

Constraint：

1. The SyncUnit needs to be used in conjunction with LX-HUB107, LX-IM001, and LX-IM002. Based on the actual physical structure, a branch from one port of the HUB can be configured as a SyncUnit. It does not support multiple SyncUnit hand in hand on the same port of the HUB, nor does it support configuring HUB modules as a SyncUnit.
2. When the controller is LX-CU500 or LX-CU501, the functional safety module does not support configuring SyncUnit. There is no such restriction when the controller is LX-CU510 or LX-CU511.
3. Multiple SyncUnits do not support simultaneous power on. Only when all modules in a SyncUnit have entered the OP state can the next SyncUnit be powered on.
4. Inserting a slave station between two adjacent SyncUnits is not supported. If insertion is required, the SyncUnit must be deleted first.
5. When deleting a slave station that has already been configured with a SyncUnit, the SyncUnit must be deleted first.
6. When the controller is LX-CU510 or LX-CU511, the 500us bus cycle does not support SyncUnit.

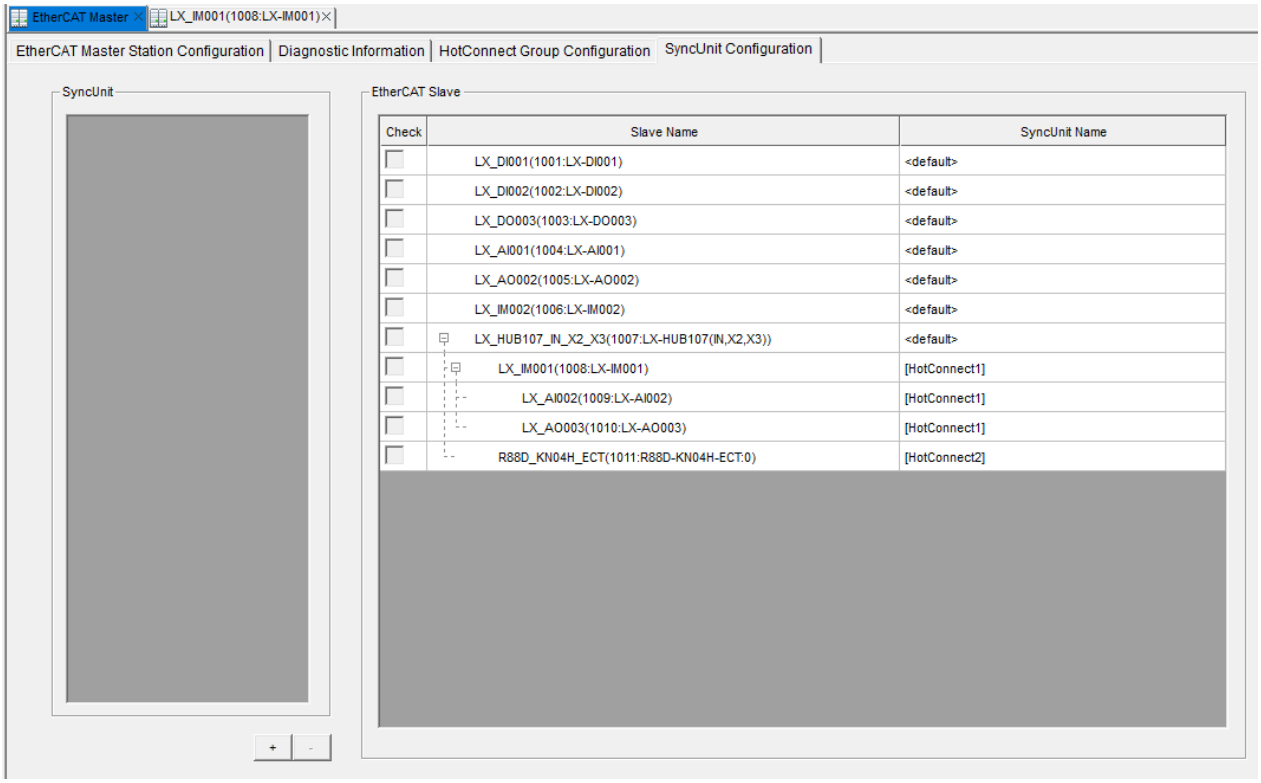
Operation：

Here is an example to illustrate how to configure a SyncUnit. The hardware configuration shown in the figure below is to configure the two branches from the X2 and X3 ports of HUB107 into two SyncUnits. Namely: Stations 1008, 1009, and 1010 are configured as a SyncUnit; The 1011 slave station is configured as a SyncUnit.



(1) Double-click on the "EtherCAT Master" master node in the project management tree to open the master configuration interface.

(2) Click on the SyncUnit Configuraiton tab, as shown in the following figure. The interface parameters are described in the table below.

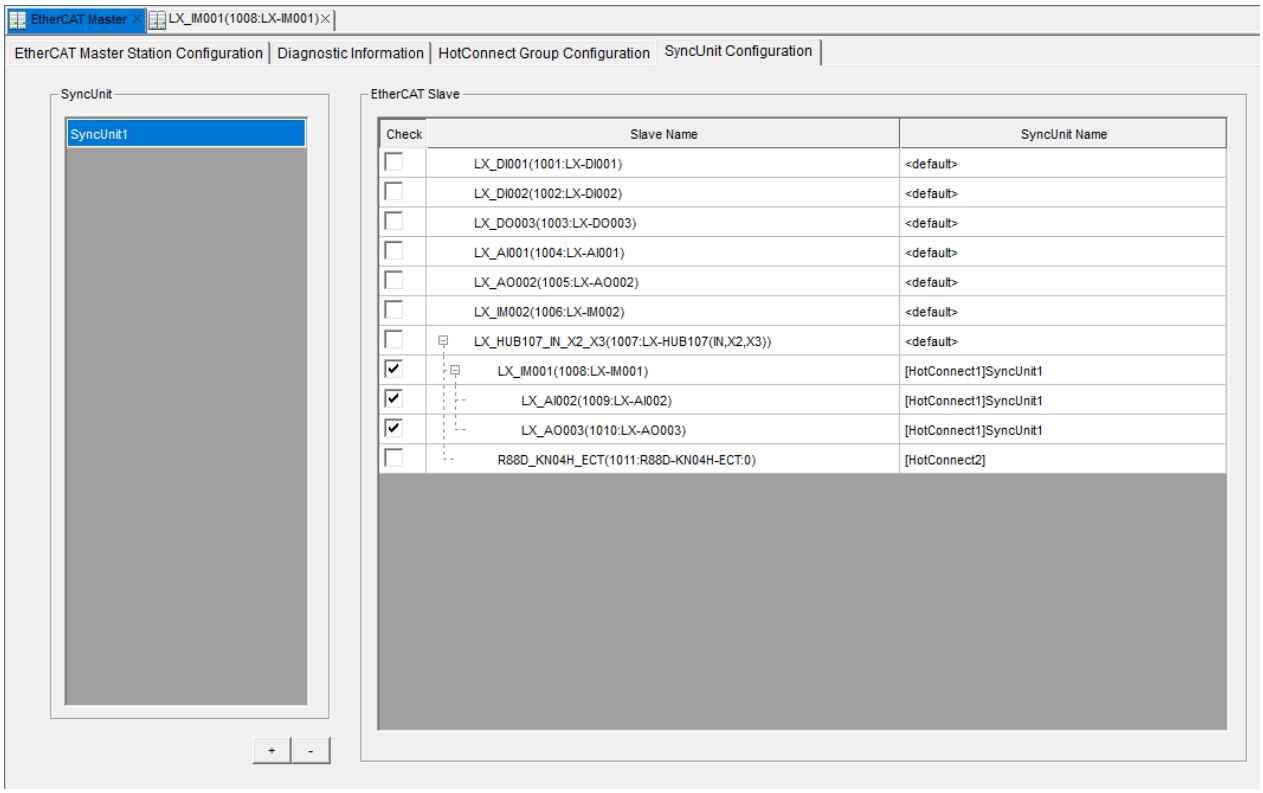


SyncUnit parameter description:

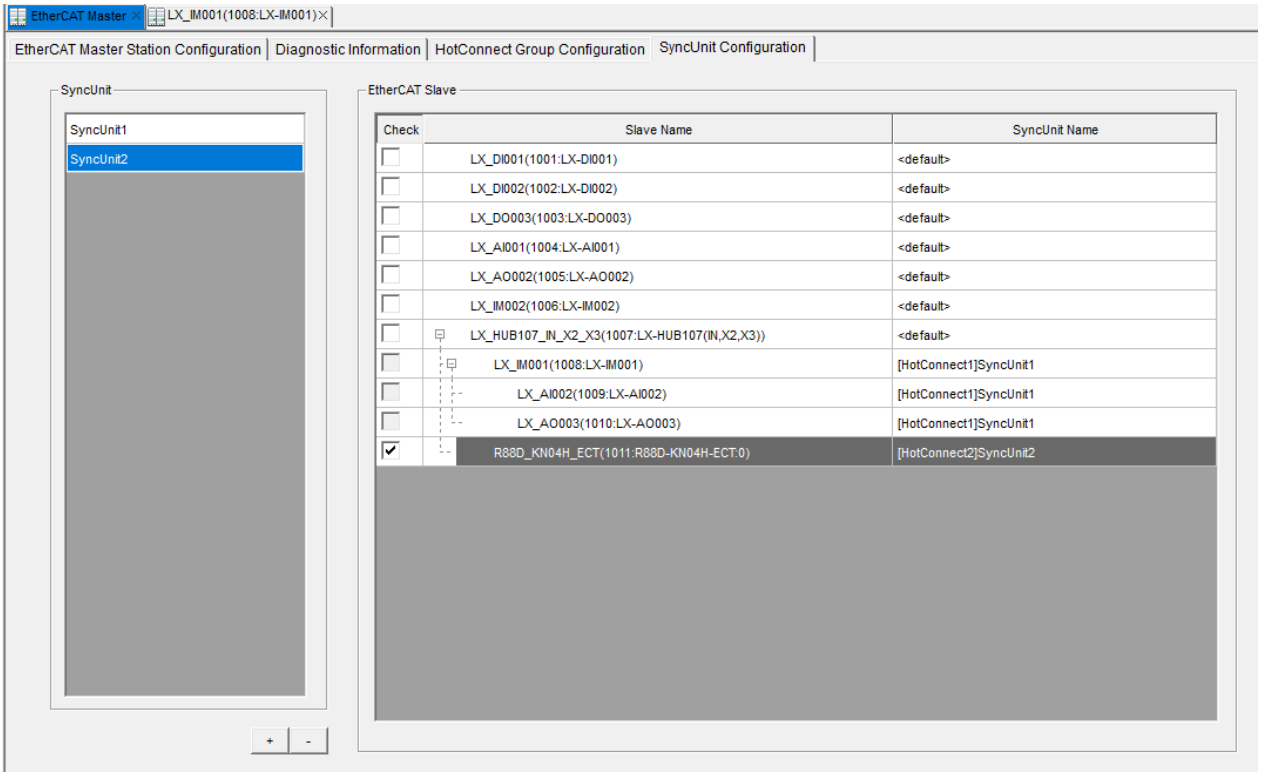
Parameter	Description
SyncUnit	SyncUnit list.
EtherCAT Slave	Slave list.

(3) Click + to add a SyncUnit.

(4) Select the newly added SyncUnit, in the EtherCAT slave area on the right, check the slave stations 1008, 1009, and 1010, and add these three slave stations to the SyncUnit.



(5) Repeat steps (3) to (4) above to add a second SyncUnit, and add slave number 1011 to the group.



Related Operations:

(1) After inserting a new module into a module that has already been configured with a SyncUnit, the new module is automatically added to the SyncUnit.

(2) When you need to delete a module from a SyncUnit, you can select the SyncUnit and uncheck the checkbox before that module.

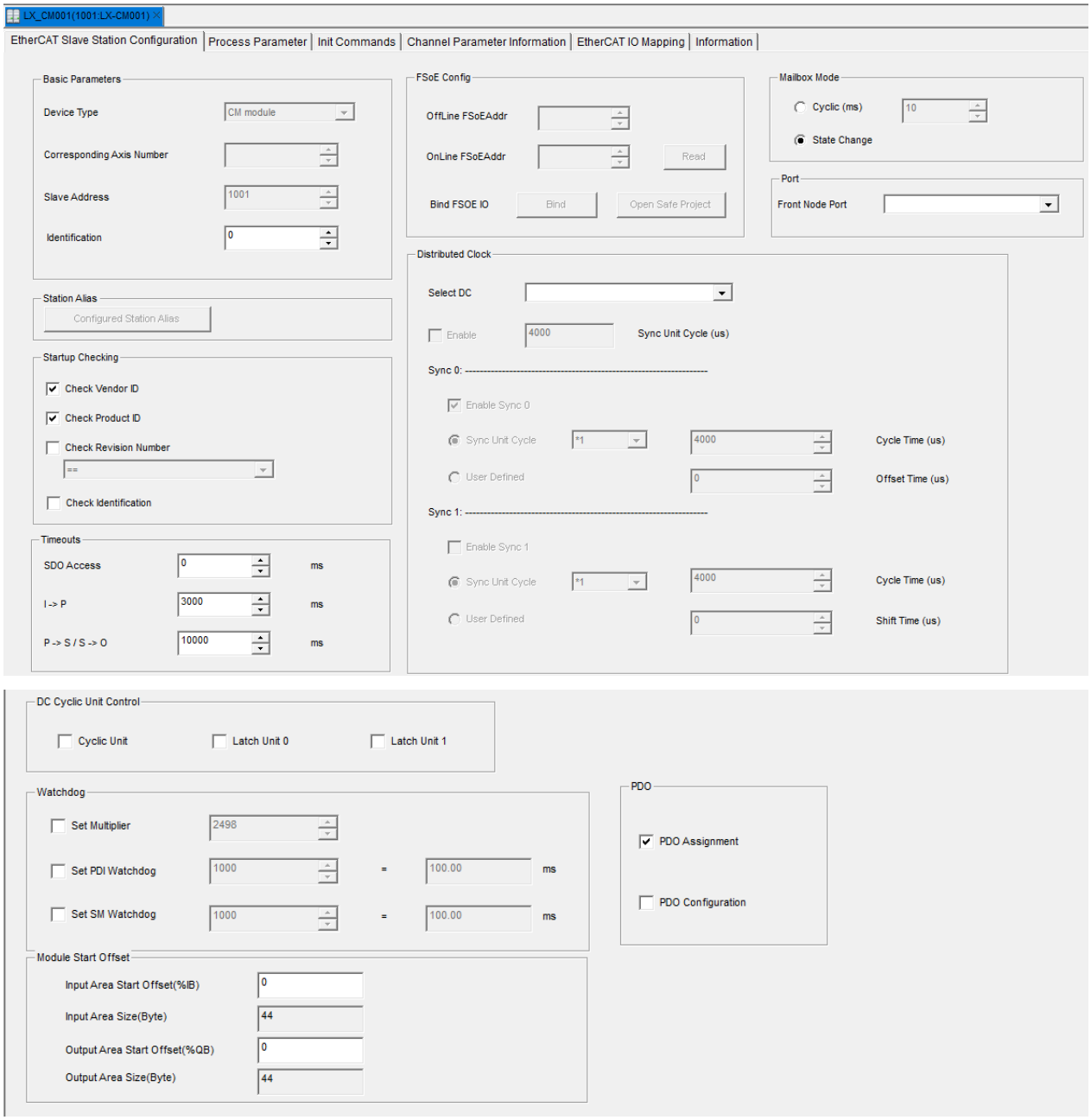
(3) When you need to delete a SyncUnit, select the medium SyncUnit and click . After deletion, all modules in the group have no SyncUnit configuration information.

12.3.3.2 Configuring EtherCAT slave station

12.3.3.2.1 EtherCAT slave station configuration

Double-click the slave station under the EtherCAT Master node, and the slave station parameter configuration interface is displayed in the right area, as shown in the figure.

This interface includes six tabs: EtherCAT Slave Station Configuration, Process Parameters, Init Commands, Channel Parameter Information, EtherCAT IO Mapping, and Information. Related parameters can be set through each tab.



- Basic parameters
  - Device type: Servo driver and general IO supported.
  - Axis number: When the device type is a servo driver, the axis number needs to be set with a range of 0 to 63 and the default value of 0. When the device type is an IO module, this option is invalid.
  - Slave address: The default slave station address is start from 1001, and the user can set the slave address here. Address range: 1~14355.
  - Identification: Set the slave identity ID number. If identity verification is enabled, it must match the written slave alias.
- Mailbox mode

- Cycle: Interval (milliseconds) for reading the input mailbox (polling mode), setting range: 0~99999.
- State Change: The input mailbox will only be read when the status bit is set.

■ Port

- When the device type is a branch module (e.g., servo drive), this parameter identifies the port of the parent node connected to the current node.

■ Station Alias

This function is primarily used to write the configured slave alias into the slave's EEPROM and is only available in online mode. After configuring the "New Value," click "Write to EEPROM" to immediately update the actual value (EEPROM). However, the actual value (register) will only take effect after power cycling the physical module.

■ Startup Checking

By default, when the system starts, it automatically checks the vendor ID or product ID based on the current configuration settings. If a mismatch is detected, the bus stops and no further action is taken. This setting is to avoid downloading the wrong configuration.

After enabling the "Verify Identity" function, the system will check whether the slave device's identity matches the preset alias. If they do not match, the slave device will be prohibited from going online.

■ Timeouts

By default, the following operations are not defined as timeouts. However, if you want to see if they exceed a specific time, you can set the time (in microseconds) here:

- SDO access: When the system starts, it sends the SDO list, setting range: 0~100000.
- I->P: Convert from 'initialization' to 'pre-run' mode, setting range: 0~100000.
- P->S/S->O: Convert from 'pre-run' to 'safe running' mode or from 'safe running' to 'run' mode, setting range: 0~100000.

■ Distributed clock

- Select DC: The drop-down menu provides settings for distributed clocks provided by the device description file. The default is DC-Synchronous.
- Enable: If the Distributed Clock function is activated, the data exchange cycle time displayed in the Sync Unit Cycle (us) area will be determined by the master station cycle time. The master station clock is therefore able to synchronize data exchange within the network.
- Sync 0:
- Enable sync 0: If this option is activated, the sync unit of sync 0 is used. A synchronization unit describes a series of synchronously exchanged process data.

  Synchronization unit cycle: If this option is activated, the result of the master station cycle time multiplied by the selected factor will be used as the synchronization cycle time of the slave station. The Cycle Time (us) field displays the currently set cycle time. Setting range: 0~4294967.

  User-defined: If this option is activated, the desired time in microseconds can be entered in the Cycle Time (us) field. Offset time setting range: 0~4294967.

- Sync 1:

  Enable sync 1: If this option is activated, the sync unit of sync 1 is used. A synchronization unit describes a series of synchronously exchanged process data.

  Synchronization unit cycle: If this option is activated, the result of the master station cycle time multiplied by the selected factor will be used as the synchronization cycle time of the slave station. The Cycle Time (us) field displays the currently set cycle time. Setting range: 0~4294967.

  User-defined: If this option is activated, the desired time in microseconds can be entered in the Cycle Time (us) field. Offset time setting range: 0~4294967.

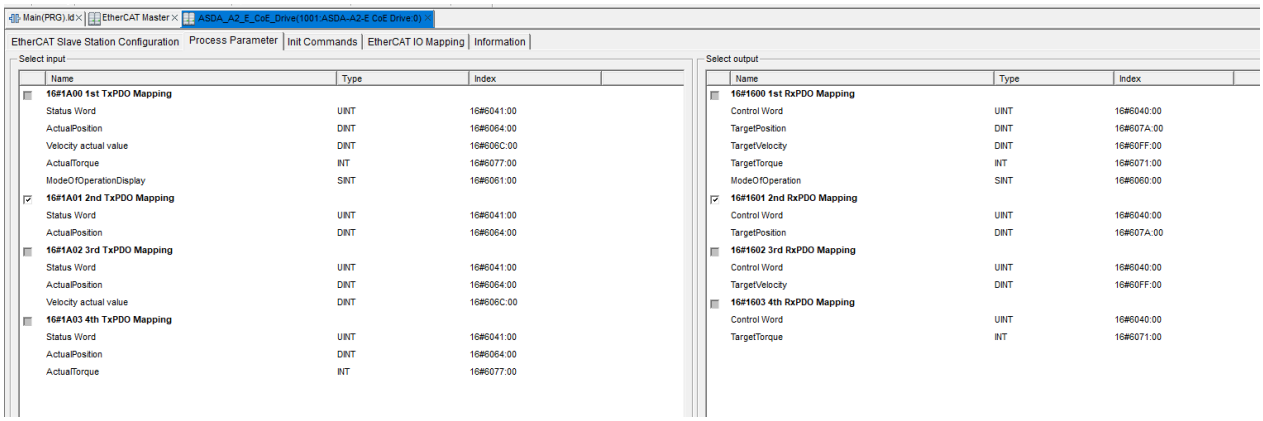
- DC cycle unit control

  This is an option to select the function definition of distributed clocks, which is allocated to the local microprocessor. The control function has been completed in the register of the slave station. Possible settings: cycle unit, latch unit 0, latch unit 1.

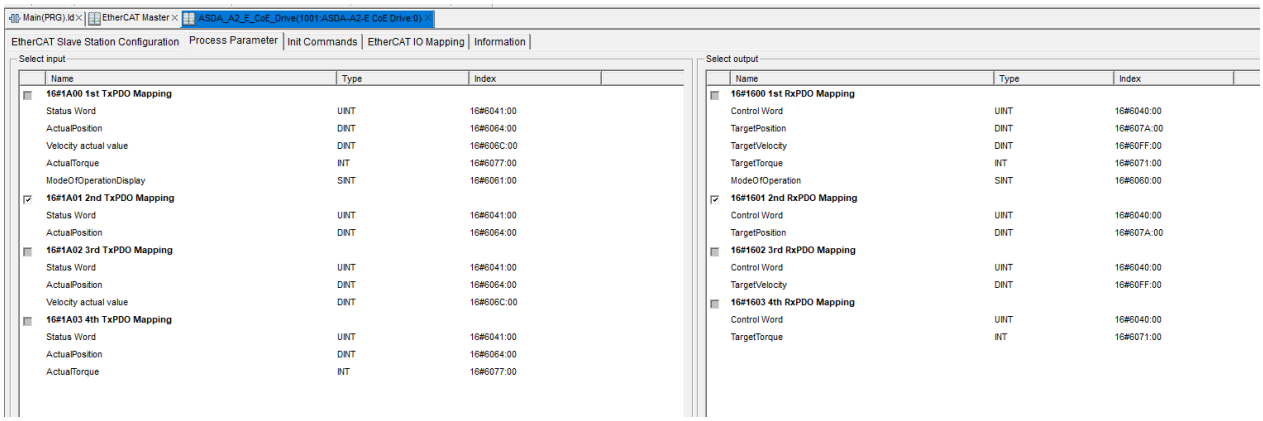
- Watchdog
  - Setting multiplier: The watchdogs PDI and SM (Synchronization Manager) get their cycles from the local terminal, and then the watchdog receives the cycles.
  - Setting PDI watchdog: If the PDI (Process Data Interface) communication time with the EtherCAT slave station exceeds the set and activated PDI watchdog time, the watchdog will be triggered.
  - Setting SM watchdog: If the EtherCAT process data communication cycle is longer than the set and activated SM (Synchronization Manager) watchdog time, the watchdog will be triggered.
- PDO Assignment
  - PDO allocation: Check this parameter, and the request command with index 0x1cxx is added to Init Command, and then downloaded to the EtherCAT slave station when the system starts. No settings are required, just keep the default configuration.
  - PDO configuration: Check this parameter to add the corresponding PDO configuration to the Initialization Command and it will be downloaded to the EtherCAT slave station when the system starts. No settings are required, just keep the default configuration.
- Module Start Offset
  - Input Area Start Offset(%IB): In offline mode, the starting address of the input channel in the EtherCAT IO mapping for the module can be modified.
  - Input Area Size (Byte): The number of bytes in the input area.
  - Output Area Starti Offset(%QB): In offline mode, the starting address of the output channel in the EtherCAT IO mapping for the module can be modified.
  - Output Area Size (Byte): The number of bytes in the output area.

12.3.3.2.2 Process parameters

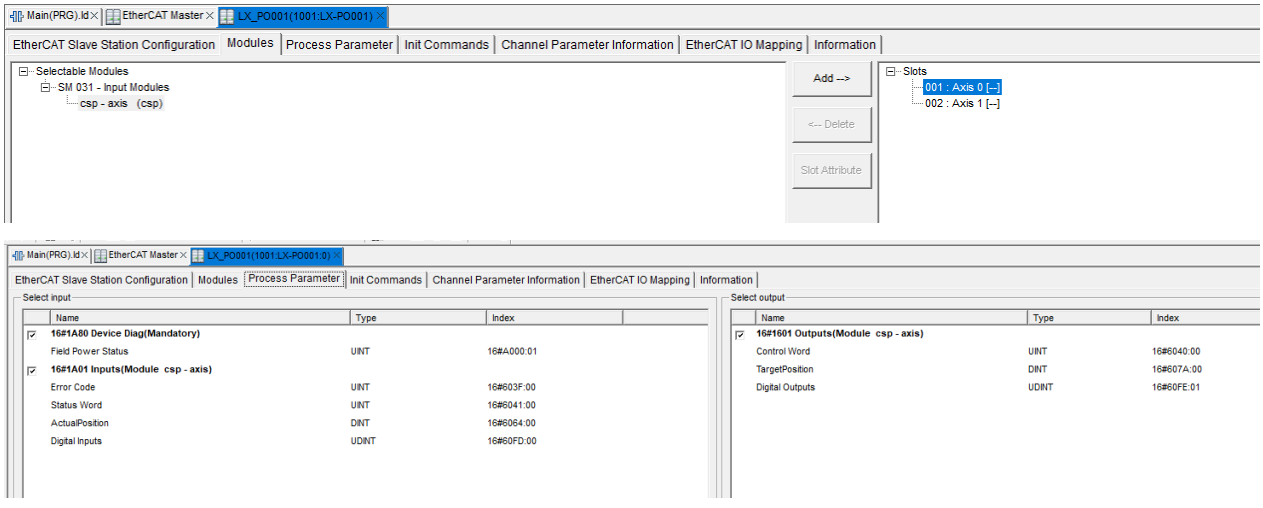
When the device type is a servo driver, the PDO type can be selected.



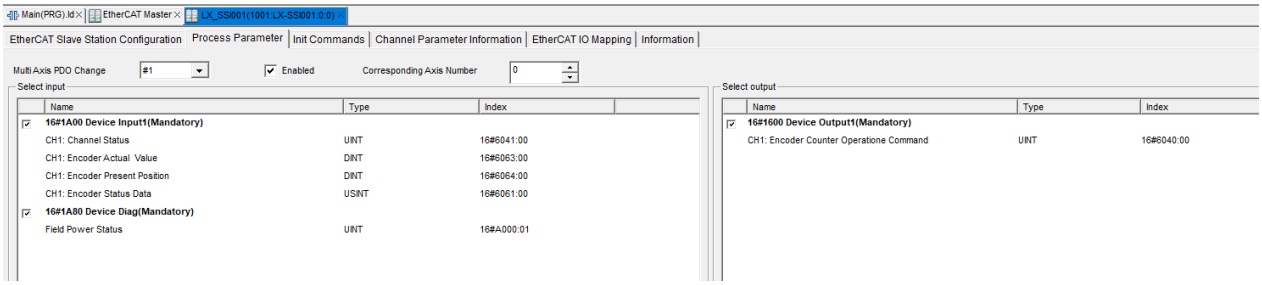
When the device type is an IO module, the IO channel can be selected.



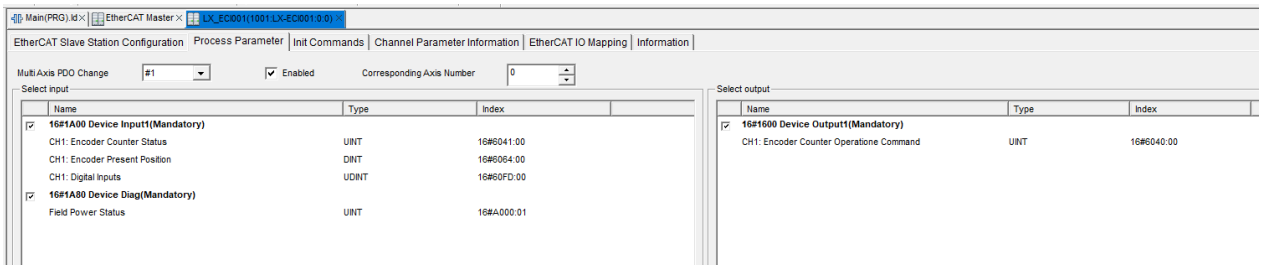
When the device type is a PO module, in the Modules tab, add the csp control mode to Axis0 and Axis1, and then the PDO parameters corresponding to Axis0 and Axis1 are displayed in the Process Parameters tab.



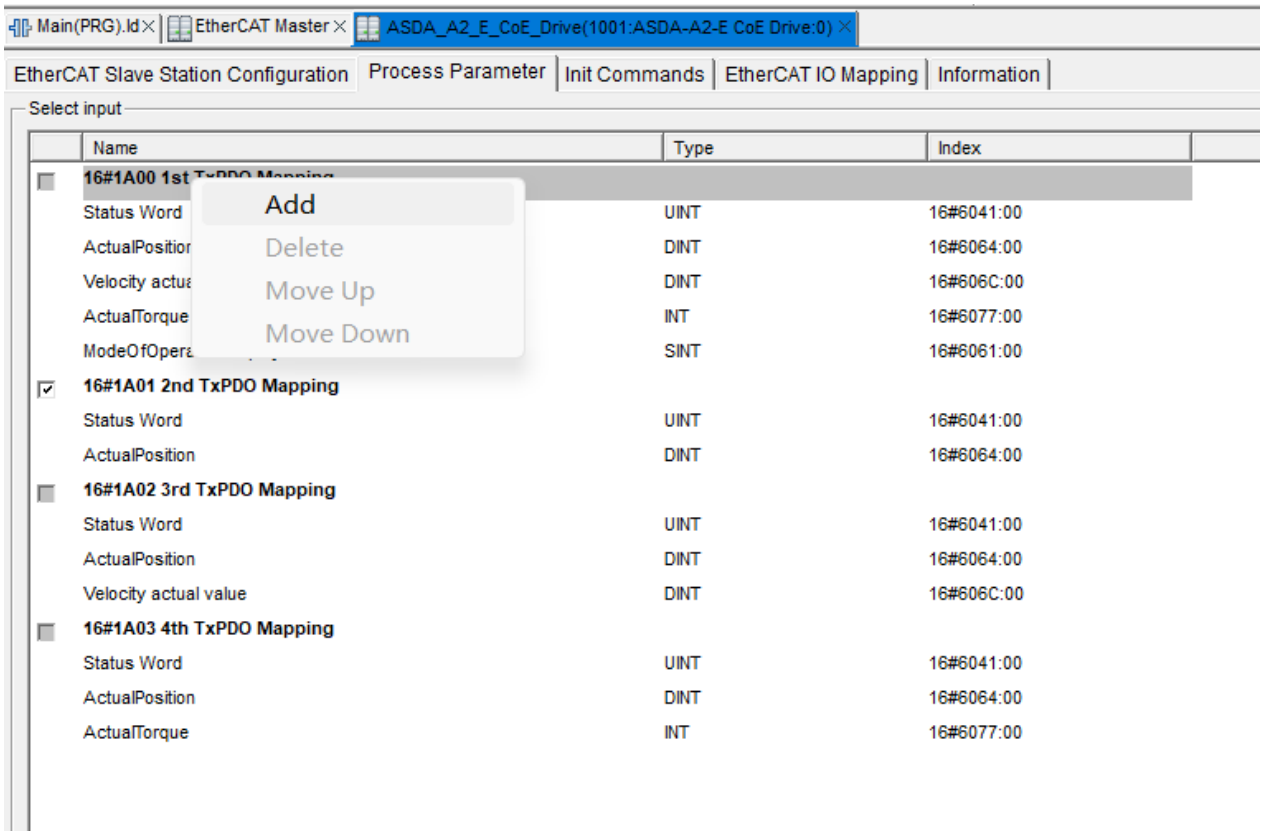
When the device type is an SSI module, the PDO parameters can be selected in the multi-axis process parameters.

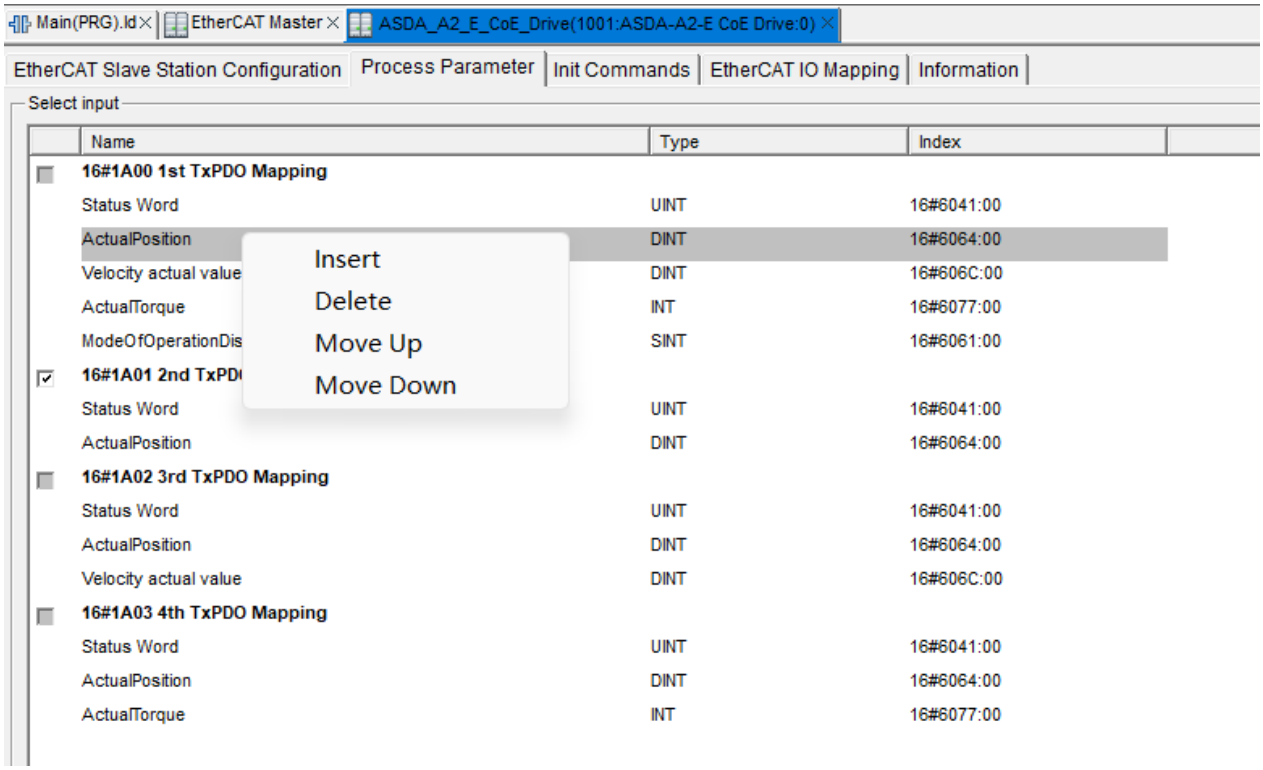


When the device type is an ECI module, the PDO parameters can be selected in the multi-axis process parameters.

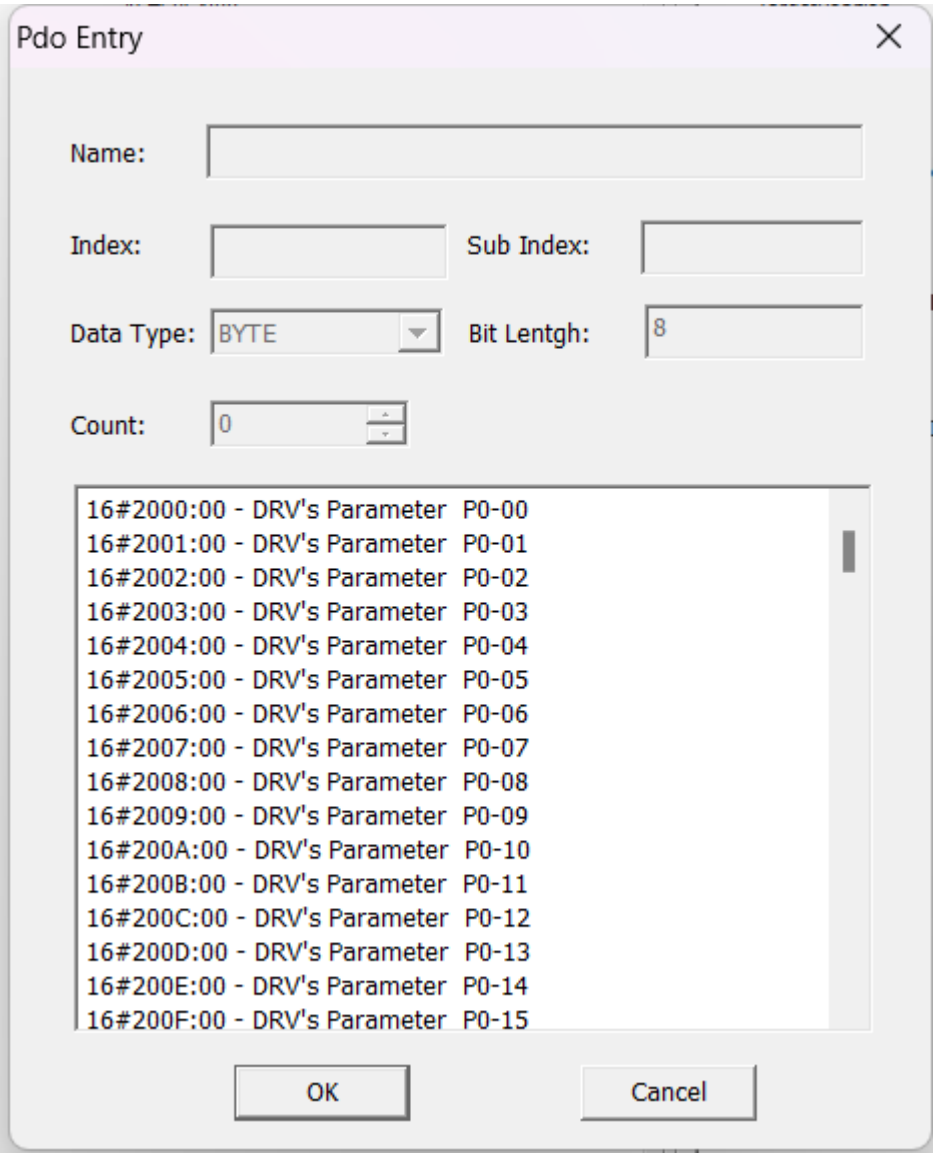


Custom PDO parameters: When the device type is a servo driver, you can add or delete system PDO parameters in the process parameters. The operation method is as follows:





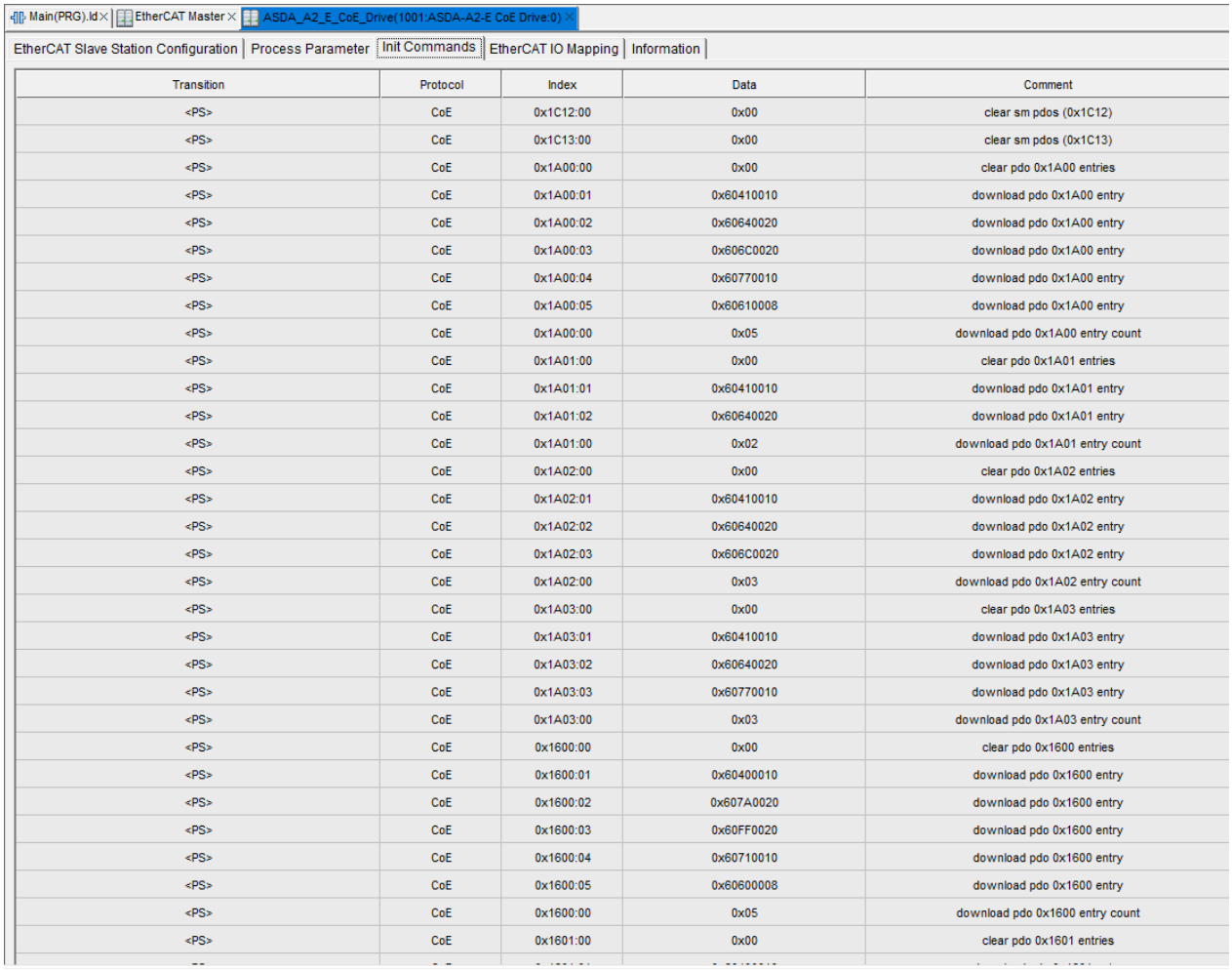
The PDO Parameters dialog box is displayed, as shown in the figure, and then select and add the parameters in the list box.



- Note that the same PDO parameter cannot be added to the same process parameter repeatedly.

12.3.3.2.3 Initialization command

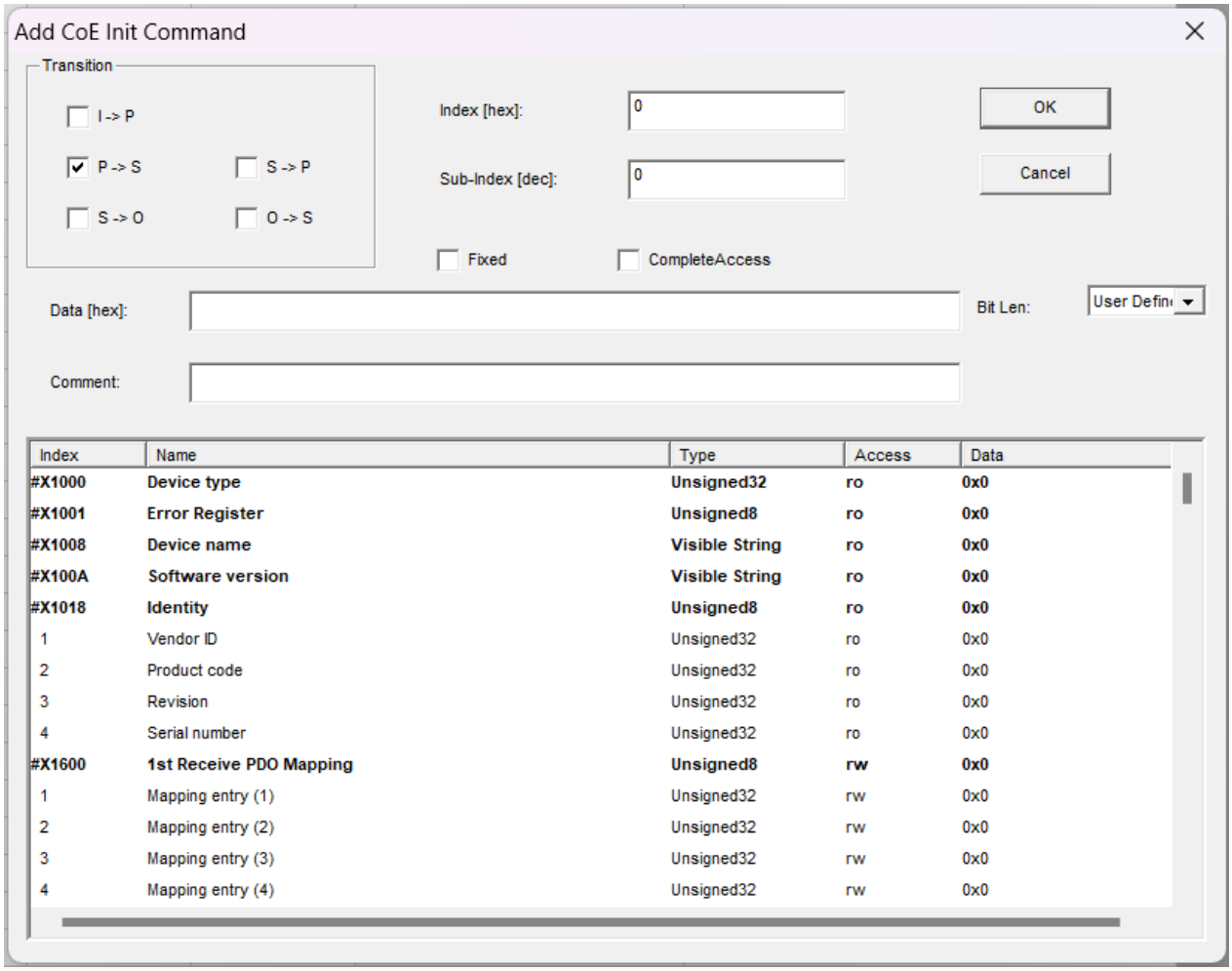
The initialization command is sent to the slave station when the system starts to initialize the slave station.



Transition	Protocol	Index	Data	Comment
<PS>	CoE	0x1C12:00	0x00	clear sm pdos (0x1C12)
<PS>	CoE	0x1C13:00	0x00	clear sm pdos (0x1C13)
<PS>	CoE	0x1A00:00	0x00	clear pdo 0x1A00 entries
<PS>	CoE	0x1A00:01	0x60410010	download pdo 0x1A00 entry
<PS>	CoE	0x1A00:02	0x60640020	download pdo 0x1A00 entry
<PS>	CoE	0x1A00:03	0x606C0020	download pdo 0x1A00 entry
<PS>	CoE	0x1A00:04	0x60770010	download pdo 0x1A00 entry
<PS>	CoE	0x1A00:05	0x60610008	download pdo 0x1A00 entry
<PS>	CoE	0x1A00:00	0x05	download pdo 0x1A00 entry count
<PS>	CoE	0x1A01:00	0x00	clear pdo 0x1A01 entries
<PS>	CoE	0x1A01:01	0x60410010	download pdo 0x1A01 entry
<PS>	CoE	0x1A01:02	0x60640020	download pdo 0x1A01 entry
<PS>	CoE	0x1A01:00	0x02	download pdo 0x1A01 entry count
<PS>	CoE	0x1A02:00	0x00	clear pdo 0x1A02 entries
<PS>	CoE	0x1A02:01	0x60410010	download pdo 0x1A02 entry
<PS>	CoE	0x1A02:02	0x60640020	download pdo 0x1A02 entry
<PS>	CoE	0x1A02:03	0x606C0020	download pdo 0x1A02 entry
<PS>	CoE	0x1A02:00	0x03	download pdo 0x1A02 entry count
<PS>	CoE	0x1A03:00	0x00	clear pdo 0x1A03 entries
<PS>	CoE	0x1A03:01	0x60410010	download pdo 0x1A03 entry
<PS>	CoE	0x1A03:02	0x60640020	download pdo 0x1A03 entry
<PS>	CoE	0x1A03:03	0x60770010	download pdo 0x1A03 entry
<PS>	CoE	0x1A03:00	0x03	download pdo 0x1A03 entry count
<PS>	CoE	0x1600:00	0x00	clear pdo 0x1600 entries
<PS>	CoE	0x1600:01	0x60400010	download pdo 0x1600 entry
<PS>	CoE	0x1600:02	0x607A0020	download pdo 0x1600 entry
<PS>	CoE	0x1600:03	0x60FF0020	download pdo 0x1600 entry
<PS>	CoE	0x1600:04	0x60710010	download pdo 0x1600 entry
<PS>	CoE	0x1600:05	0x60600008	download pdo 0x1600 entry
<PS>	CoE	0x1600:00	0x05	download pdo 0x1600 entry count
<PS>	CoE	0x1601:00	0x00	clear pdo 0x1601 entries

Parameter	Description
Convert	The conversion command can be: I->P: From "initialization" to "pre-run" mode P->S: From "pre-run" to "safe running" mode S->P: From "safe running" to "pre-run" mode S->O: From "safe running" to "run" mode O->S: From "run" to "safe running" mode If converted to <PS> , the conversion command is fixed and cannot be modified or deleted.
Protocol	CANopen over EtherCAT (CoE) protocol
Index	Index of object
Value	Data values downloaded to objects
Description	Description of the initialization command sent to the slave station

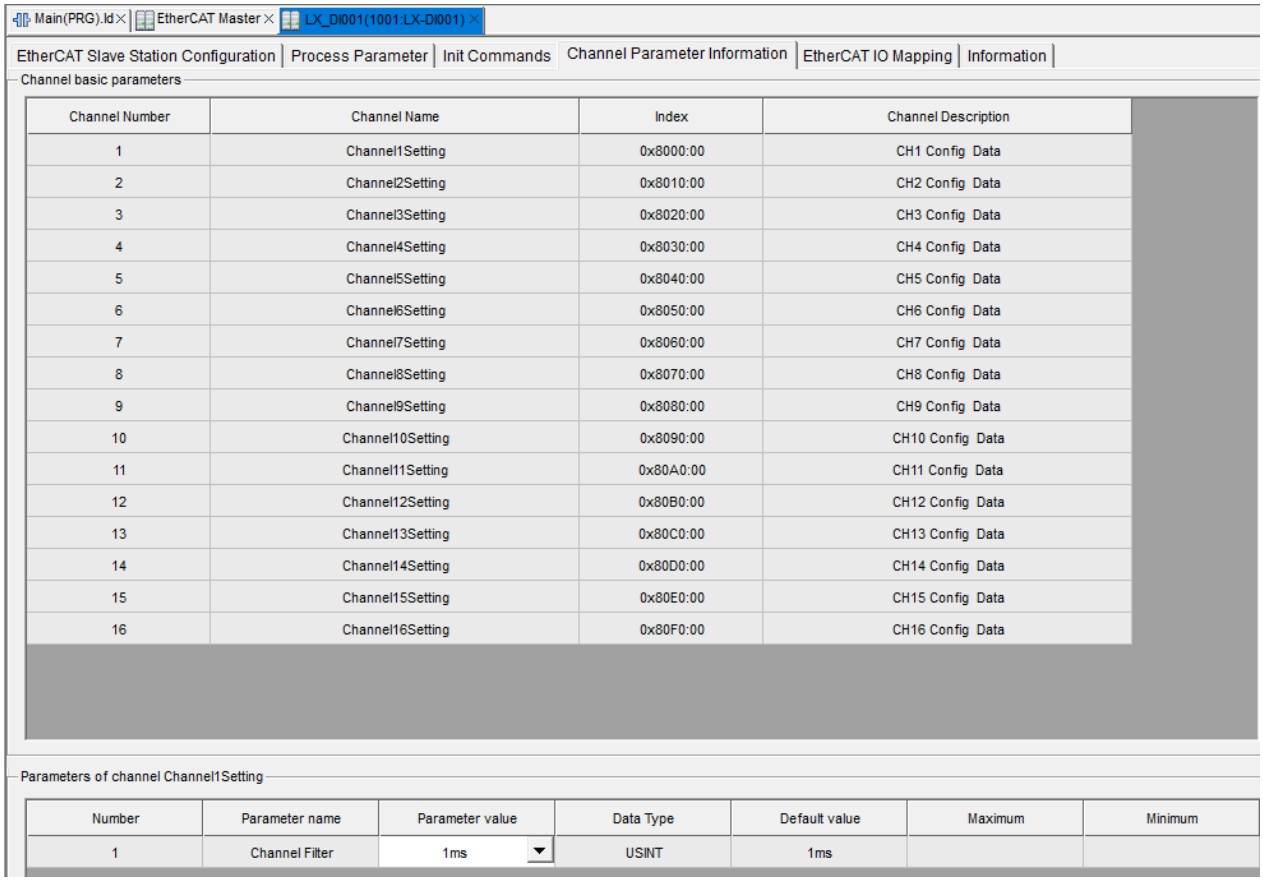
You can add an initialization command by right-clicking the Add command.



After selecting the parameters, check the Convert command and set the index and parameter values. The specific setting rules are different for each slave station. Please refer to the relevant instructions of each slave station product.

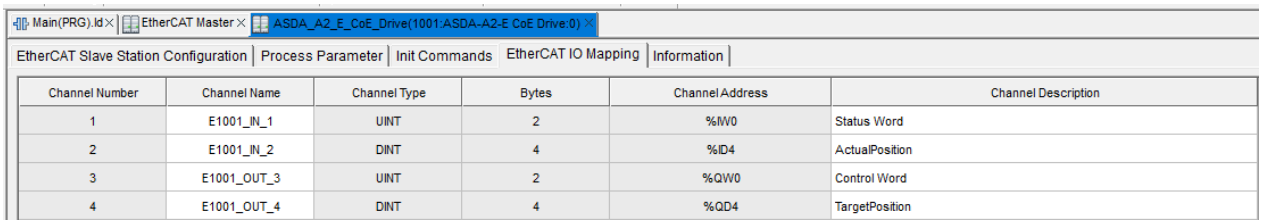
12.3.3.2.4 Channel parameter information

When the device type is an IO module, the channel parameter information displays all channel data of the module, and relevant channel parameters need to be configured on this page. The configurable channel parameters are related to the device type. Please configure according to the actual configuration module.



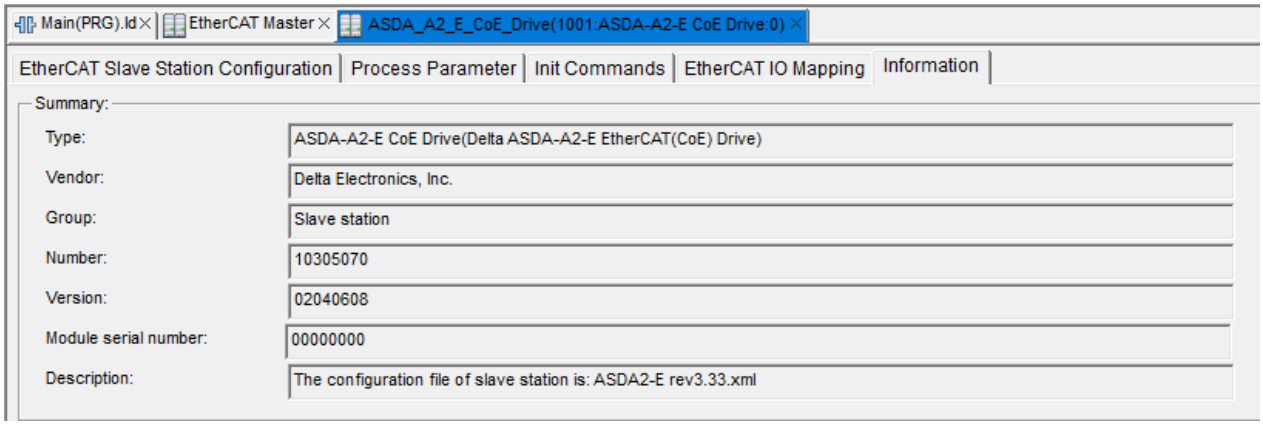
12.3.3.2.5 EtherCAT IO mapping

This tab displays the mapping relationship between the selected Process Parameters and areas I and O If the configuration of Process Parameters is changed, the EtherCAT IO mapping will be redistributed.



12.3.3.2.6 Information

The Information tab displays the device name, device model, manufacturer, version, configuration file, and other information.



12.4 EtherCAT Communication Example

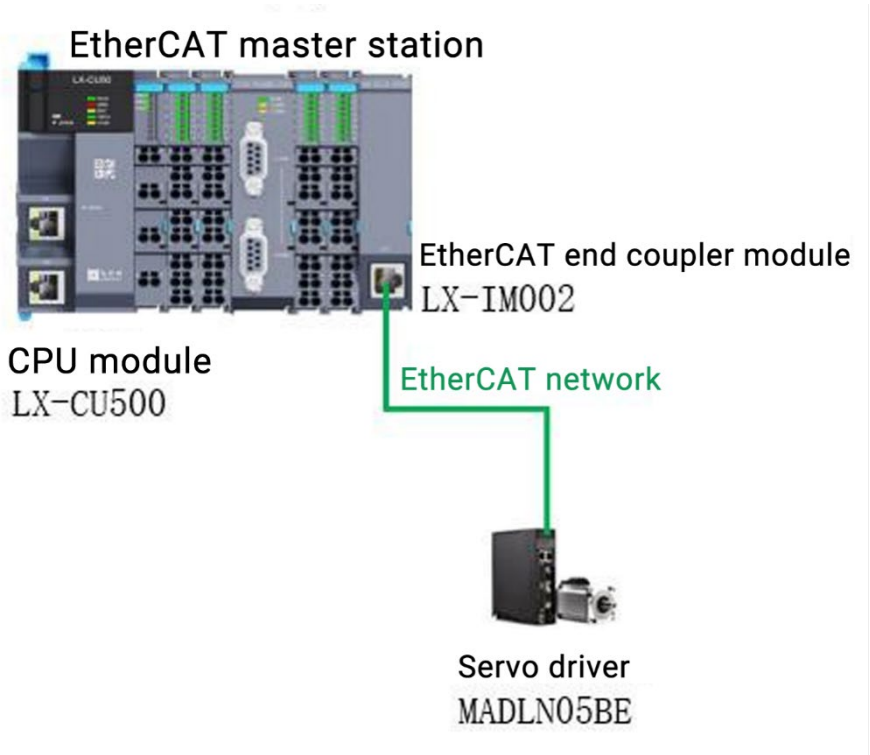
12.4.1 Example Function

This document uses the Panasonic MADLN05BE EtherCAT bus servo driver as an example.

12.4.2 System Composition

Name	Model	Quantity
LX-CPU	LX-CU500	1
End coupler	LX-IM002	1
Panasonic servo driver	MADLN05BE	1

12.4.3 Example Topology

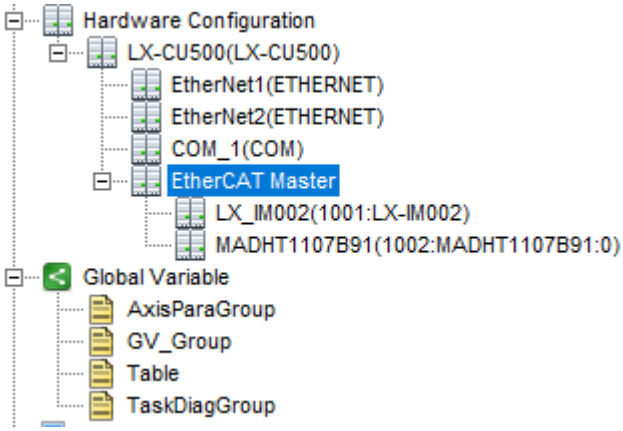


12.4.4 Programming

1. Adding servo driver

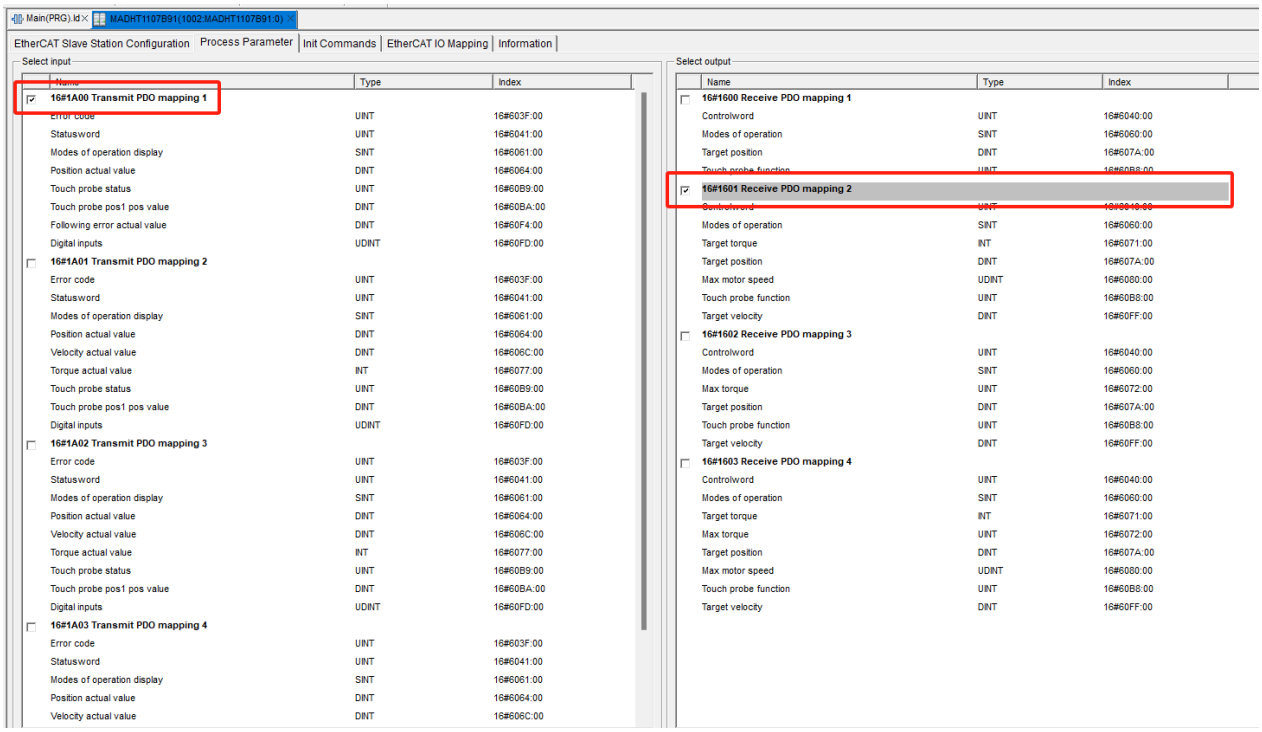
(1) Click Import Device Description File to add the servo driver XML file, and then on the AutoThink main interface menu, click Project → Import Device Description File, and select the device description file provided by the corresponding servo driver manufacturer.

(2) To add a servo driver slave station, right-click the EtherCAT Master node in the hardware configuration and select Add Slave Station or Scan Slave Station, as shown in the figure.



2. Configuring servo driver

(1) Double-click to open the servo driver page, and set the Axis Number of the servo driver in EtherCAT Slave Station Configuration, as shown in the figure. This Axis Number will be applied to the corresponding motion control instruction.



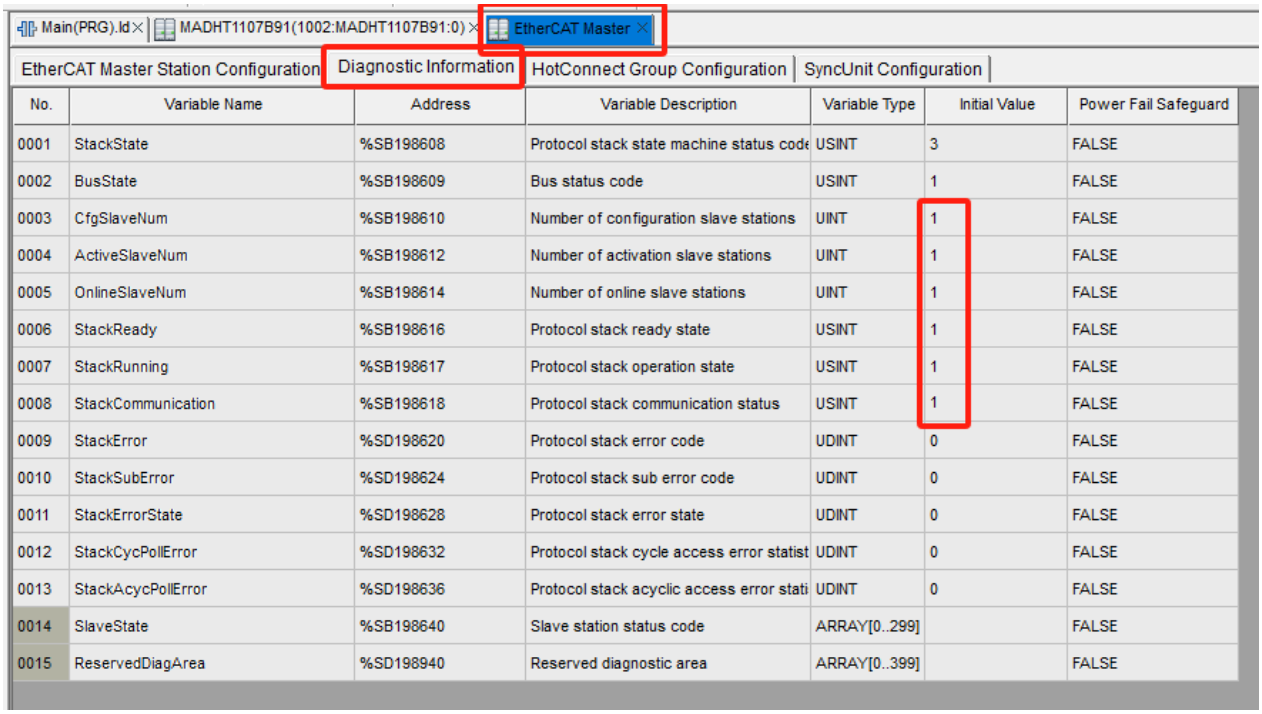
Process Parameters shall be set according to the requirements of the device and field applications, as shown in the figure:

12.4.5 Online Debugging

1. Downloading configuration

Download after compilation is correct. If the model and sequence of the connected slave stations are the same as the AutoThink configuration, then after the downloading and configuration are completed,

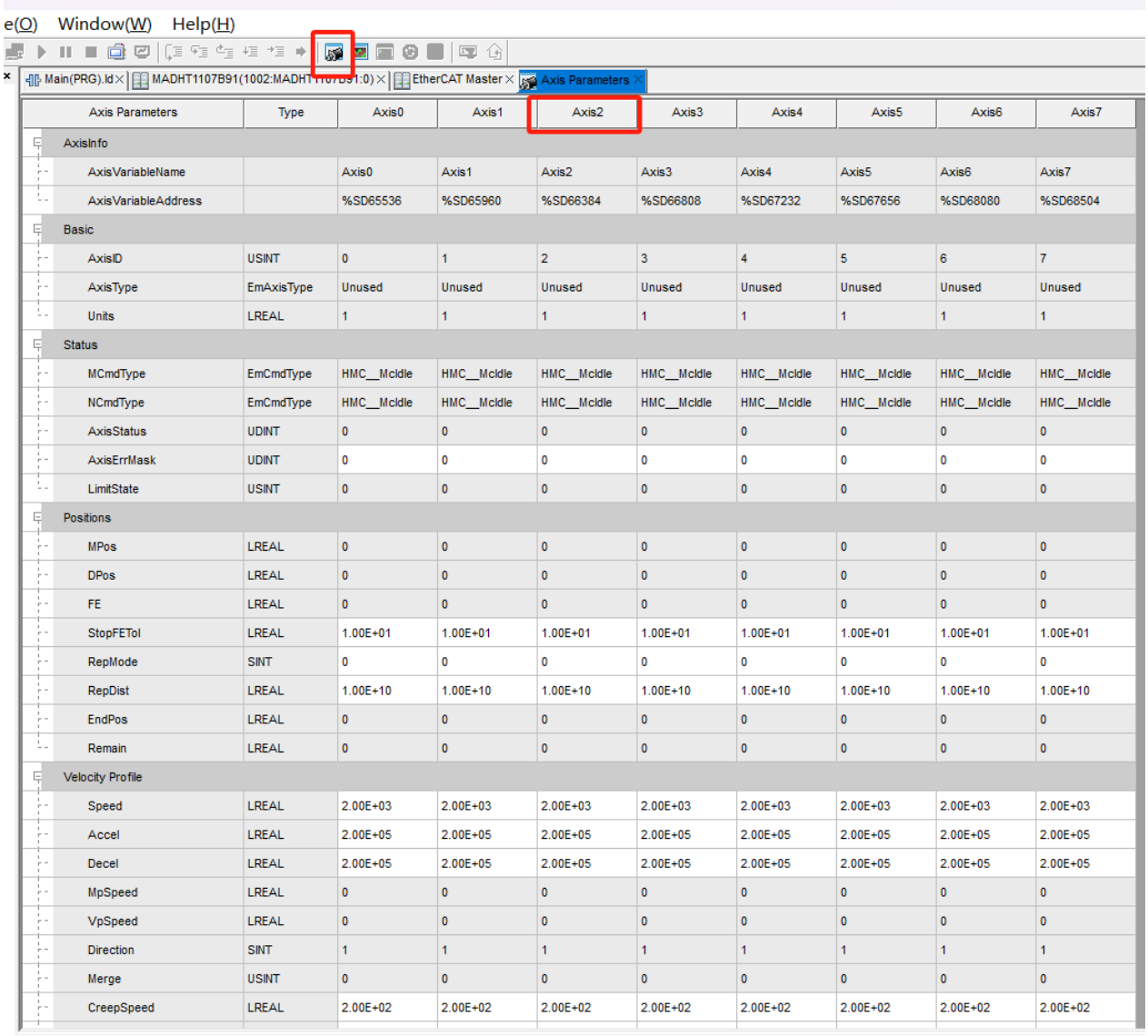
the EtherCAT bus servo driver is online. For online monitoring, you can check whether the servo driver is online normally in EtherCAT Master → Diagnostic Information. As shown in the figure, the number of activated and online slave stations is 1.



No.	Variable Name	Address	Variable Description	Variable Type	Initial Value	Power Fail Safeguard
0001	StackState	%SB198608	Protocol stack state machine status code	USINT	3	FALSE
0002	BusState	%SB198609	Bus status code	USINT	1	FALSE
0003	CfgSlaveNum	%SB198610	Number of configuration slave stations	UINT	1	FALSE
0004	ActiveSlaveNum	%SB198612	Number of activation slave stations	UINT	1	FALSE
0005	OnlineSlaveNum	%SB198614	Number of online slave stations	UINT	1	FALSE
0006	StackReady	%SB198616	Protocol stack ready state	USINT	1	FALSE
0007	StackRunning	%SB198617	Protocol stack operation state	USINT	1	FALSE
0008	StackCommunication	%SB198618	Protocol stack communication status	USINT	1	FALSE
0009	StackError	%SD198620	Protocol stack error code	UDINT	0	FALSE
0010	StackSubError	%SD198624	Protocol stack sub error code	UDINT	0	FALSE
0011	StackErrorState	%SD198628	Protocol stack error state	UDINT	0	FALSE
0012	StackCycPollError	%SD198632	Protocol stack cycle access error statist	UDINT	0	FALSE
0013	StackAcycPollError	%SD198636	Protocol stack acyclic access error stati	UDINT	0	FALSE
0014	SlaveState	%SB198640	Slave station status code	ARRAY[0..299]		FALSE
0015	ReservedDiagArea	%SD198940	Reserved diagnostic area	ARRAY[0..399]		FALSE

2. Setting axis type

The HMC_SetAxisType instruction can be used to set the axis type. Various axis parameters can be viewed on the View Axis Parameters page of the AutoThink online status interface. Some of the parameters can be modified online on this page:



Axis Parameters	Type	Axis0	Axis1	Axis2	Axis3	Axis4	Axis5	Axis6	Axis7
AxisInfo									
AxisVariableName		Axis0	Axis1	Axis2	Axis3	Axis4	Axis5	Axis6	Axis7
AxisVariableAddress		%SD65536	%SD65960	%SD66384	%SD66808	%SD67232	%SD67656	%SD68080	%SD68504
Basic									
AxisID	USINT	0	1	2	3	4	5	6	7
AxisType	EmAxisType	Unused	Unused	Unused	Unused	Unused	Unused	Unused	Unused
Units	LREAL	1	1	1	1	1	1	1	1
Status									
MCmdType	EmCmdType	HMC__McIdle	HMC__McIdle	HMC__McIdle	HMC__McIdle	HMC__McIdle	HMC__McIdle	HMC__McIdle	HMC__McIdle
NCmdType	EmCmdType	HMC__McIdle	HMC__McIdle	HMC__McIdle	HMC__McIdle	HMC__McIdle	HMC__McIdle	HMC__McIdle	HMC__McIdle
AxisStatus	UDINT	0	0	0	0	0	0	0	0
AxisErrMask	UDINT	0	0	0	0	0	0	0	0
LimitState	USINT	0	0	0	0	0	0	0	0
Positions									
MPos	LREAL	0	0	0	0	0	0	0	0
DPos	LREAL	0	0	0	0	0	0	0	0
FE	LREAL	0	0	0	0	0	0	0	0
StopFETol	LREAL	1.00E+01	1.00E+01	1.00E+01	1.00E+01	1.00E+01	1.00E+01	1.00E+01	1.00E+01
RepMode	SINT	0	0	0	0	0	0	0	0
RepDist	LREAL	1.00E+10	1.00E+10	1.00E+10	1.00E+10	1.00E+10	1.00E+10	1.00E+10	1.00E+10
EndPos	LREAL	0	0	0	0	0	0	0	0
Remain	LREAL	0	0	0	0	0	0	0	0
Velocity Profile									
Speed	LREAL	2.00E+03	2.00E+03	2.00E+03	2.00E+03	2.00E+03	2.00E+03	2.00E+03	2.00E+03
Accel	LREAL	2.00E+05	2.00E+05	2.00E+05	2.00E+05	2.00E+05	2.00E+05	2.00E+05	2.00E+05
Decel	LREAL	2.00E+05	2.00E+05	2.00E+05	2.00E+05	2.00E+05	2.00E+05	2.00E+05	2.00E+05
MpSpeed	LREAL	0	0	0	0	0	0	0	0
VpSpeed	LREAL	0	0	0	0	0	0	0	0
Direction	SINT	1	1	1	1	1	1	1	1
Merge	USINT	0	0	0	0	0	0	0	0
CreepSpeed	LREAL	2.00E+02	2.00E+02	2.00E+02	2.00E+02	2.00E+02	2.00E+02	2.00E+02	2.00E+02

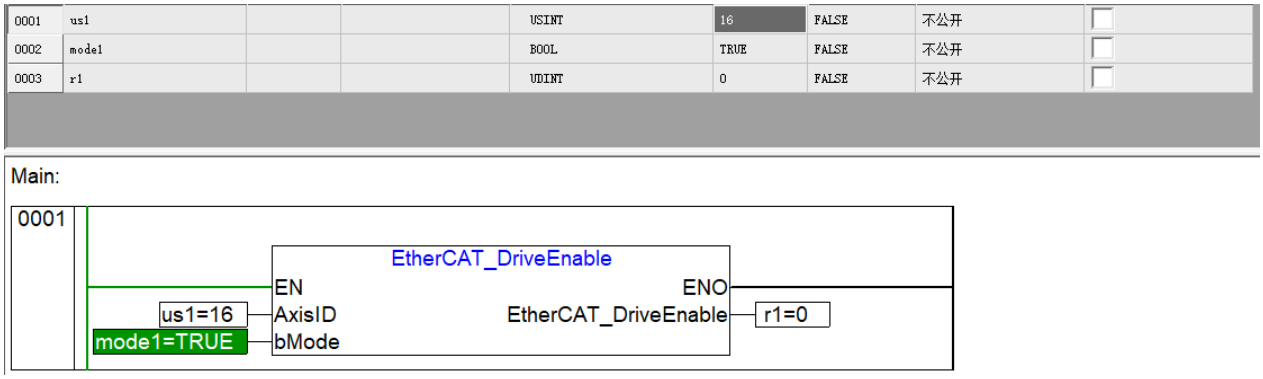
3. Servo enable

When the project is online normally, click the Enable button to enable the servo, as shown in the figure:

You can also call the HMC_DriveEnable instruction to turn on or off the servo enable.

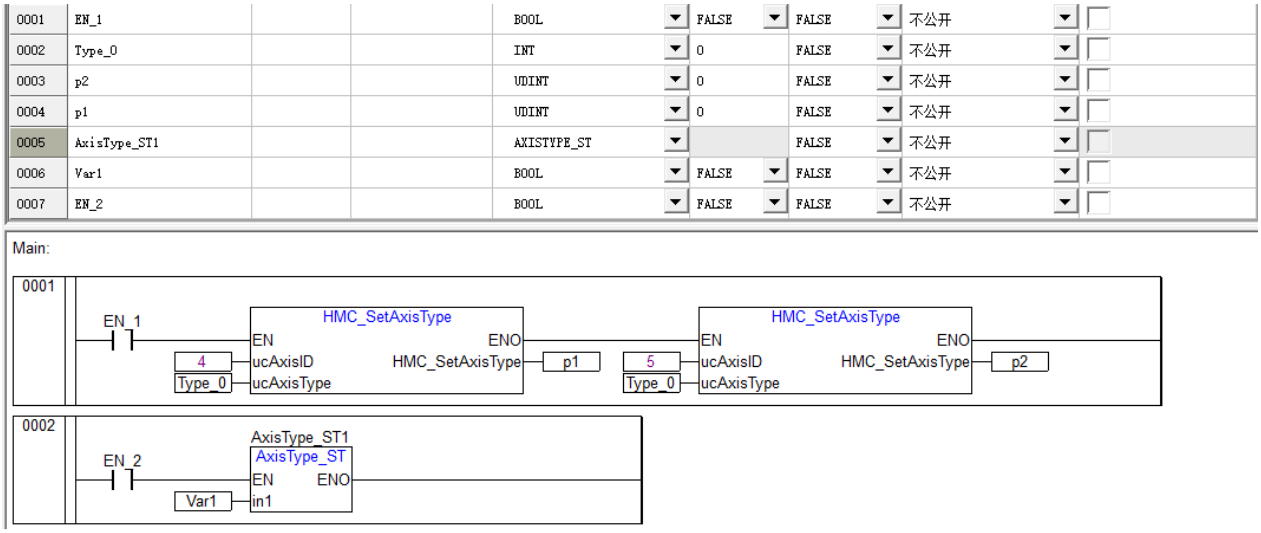
Use the EtherCAT_DriveEnable instruction to turn on/off the servo enable switch, and you can independently control the enable status of each slave station axis.

Instruction example:

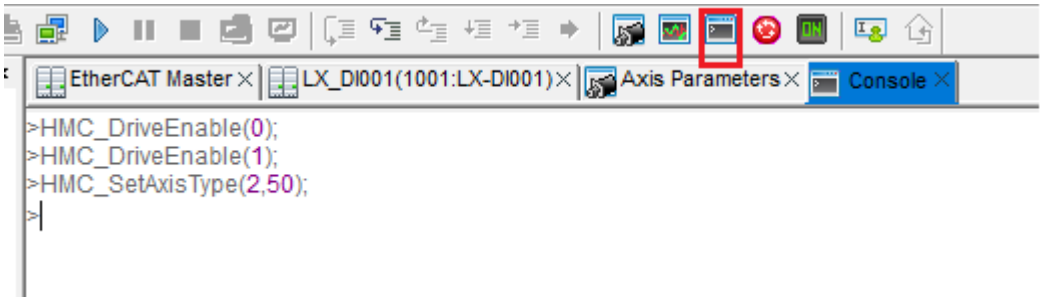


4. Instruction debugging

Instruction debugging can be done in two ways. One is to write the instructions in the program and execute them through enable or subroutine calls, as shown in the figure:

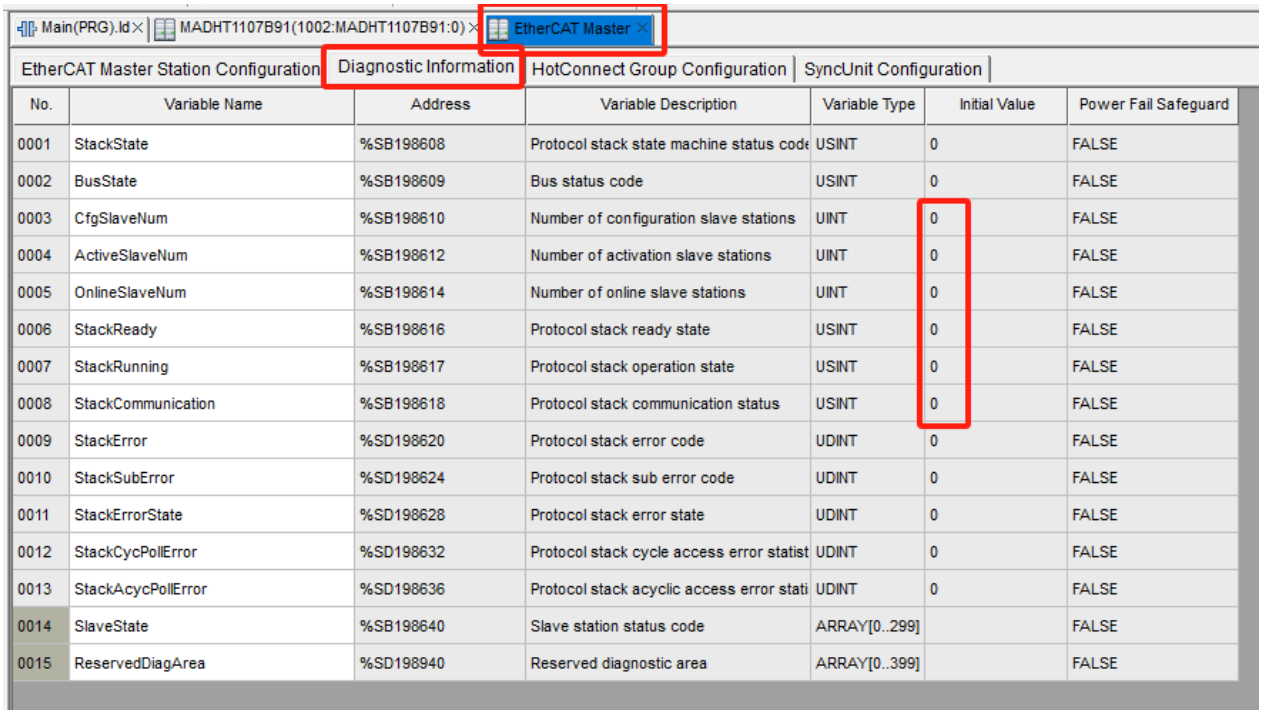


Another way is to enter the instruction in real time through the online Terminal tool of Auto Think:



12.5 Fault Diagnosis

Double-click the EtherCAT Master node, and the device configuration information page is displayed, then select the Diagnostic Information tab to view the diagnostic information online.



No.	Variable Name	Address	Variable Description	Variable Type	Initial Value	Power Fail Safeguard
0001	StackState	%SB198608	Protocol stack state machine status code	USINT	0	FALSE
0002	BusState	%SB198609	Bus status code	USINT	0	FALSE
0003	CfgSlaveNum	%SB198610	Number of configuration slave stations	UINT	0	FALSE
0004	ActiveSlaveNum	%SB198612	Number of activation slave stations	UINT	0	FALSE
0005	OnlineSlaveNum	%SB198614	Number of online slave stations	UINT	0	FALSE
0006	StackReady	%SB198616	Protocol stack ready state	USINT	0	FALSE
0007	StackRunning	%SB198617	Protocol stack operation state	USINT	0	FALSE
0008	StackCommunication	%SB198618	Protocol stack communication status	USINT	0	FALSE
0009	StackError	%SD198620	Protocol stack error code	UDINT	0	FALSE
0010	StackSubError	%SD198624	Protocol stack sub error code	UDINT	0	FALSE
0011	StackErrorState	%SD198628	Protocol stack error state	UDINT	0	FALSE
0012	StackCycPollError	%SD198632	Protocol stack cycle access error statist	UDINT	0	FALSE
0013	StackAcycPollError	%SD198636	Protocol stack acyclic access error stati	UDINT	0	FALSE
0014	SlaveState	%SB198640	Slave station status code	ARRAY[0..299]		FALSE
0015	ReservedDiagArea	%SD198940	Reserved diagnostic area	ARRAY[0..399]		FALSE

Diagnostic Information Table:

Classification	Variable Name	Value Range	Description
Master station diagnosis	StackState	0~4	Protocol stack status machine status code 0: Not initialized 1: Initialization successful 2: Configured successfully 3: Normal communication 4: Error status
	BusState	0,1	Bus status code 0: Bus not available 1: Bus available
	CfgSlaveNum	0~300	Configure the number of slave stations
	ActiveSlaveNum	0~300	Number of activated slave stations
	OnlineSlaveNum	0~300	Number of online slave stations
comX status information	StackReady	0,1	Protocol stack readiness status 0: Protocol stack not started 1: Protocol stack started
	StackRunning	0,1	Protocol stack running status 0: Protocol stack not running normally 1: Protocol stack running normally
	StackCommunication	0,1	Protocol stack communication status 0: Protocol stack communication fault 1: Protocol stack communicating normally
Protocol stack error message	StackError		Protocol stack error code: 0: No error Others: Refer to the EtherCATDiagGroup diagnostic information correspondence
	StackSubError		Protocol stack sub-error code
	StackErrorState		Protocol stack error status. For specific error codes, please refer to EtherCAT diagnostic error codes
	StackCycPollError	0~2^{32}-1	Protocol stack cycle access error statistics
	StackAcycPollError	0~2^{32}-1	Protocol stack non-cycle access error statistics

The diagnostic information of the slave station is as follows:	SlaveState[0-299]	-	EtherCAT can mount up to 300 slaves, and each member in the array corresponds to the slave in the AT hardware configuration from top to bottom. For example, SlaveState [0] corresponds to the first slave in the AT, and SlaveState [1] corresponds to the second slave in the AT. This parameter represents the status information of each slave station, where: The bits 0~3 of each member in the array indicate whether the slave is online: 0000: slave offline 0001: slave online 0010: slave fault The bits 4-7 of each member in the array indicate whether there are external links to the four ports of the ESC of the slave station: bit x is 0: corresponding port has no external connection bit x is 1: corresponding port has external connection
Reserve diagnostic area	ReservedDiagArea[0-399]	-	Reserve diagnostic area

EtherCATDiagGroup diagnostic information correspondence

Stack State	Status Description	BusState	CfgSlaveNum	Active Slave Num	Stack Error	Stack SubError	Error Code and Sub-error Code Description
0	Protocol stack in uninitialized status	0	0	0	0	0	Protocol stack initialization failed
1	Protocol stack in initialization success status	0	N	0	2	X	The protocol stack configuration failed, and starting from the slave station with a slave station number X, the configured number or model of slave stations is inconsistent with the actual connection
		0	N	0	6	0	The protocol stack failed to load configuration file
		0	0	0	7	0	The protocol stack failed to perform configuration and set the diagnostic information address
		0	0	0	8	0	The protocol stack failed to perform configuration and set the master station information
		0	0	0	9	0	The protocol stack failed to perform configuration and set the slave station ID information
		0	0	0	10	0	The protocol stack failed to perform configuration, resulting in that the slave station is not configured
2	Protocol stack in configuration success status						This is the suspended status and there is no error message
3	Protocol stack in normal communication status	1	N	N	0	0	The protocol stack is in normal communication and no slave station is offline
		1	N	Smaller than N	5	X	The protocol stack communication is normal, and the slave station with slave station number X is faulty
4	Protocol stack in error status	0	N	0	4	0	The protocol stack communication is faulty, and the controller side is disconnected

EtherCAT diagnostic error codes

Error Code (Hex)	Error Code (Dec)	Description
EtherCAT Diagnostic Error Codes		
0xC0B00001	3232759809	Failed to initialize the slave via the protocol mailbox.
0xC02B4242	3224060482	The EtherCAT configuration file generated during configuration is too large.
0xC0640001	3227779073	No response from the slave to the initialization command.
0xC0650005	3227844613	Hot link group identification failed.
0xC065000B	3227844619	The bus cycle time is set too small, making it impossible to exchange data with all connected slaves within one communication cycle.
0xC0650024	3227844644	The slave does not support CoE.
0xC065002A	3227844650	During EtherCAT SDO transfer, the master or slave detected that the object dictionary length is invalid or exceeds the range supported by the slave. For example, the SDO object dictionary length requested by the master is greater than the maximum length allowed by the slave firmware, or an invalid length such as 0.
0xC065002D	3227844653	The slave rejected the SDO request because there is insufficient memory inside the slave to process the SDO operation.
0xC065002F	3227844655	Read operation performed on a write-only object.
0xC0650030	3227844656	Write operation performed on a read-only object.
0xC0650037	3227844663	Data type or length mismatch for the parameter.
0xC065003A	3227844666	The index or sub-index of the slave's object dictionary being read does not exist.
0xC065003F	3227844671	Internal error occurred during SDO read/write to the slave.
0xC0CD0020	3234660384	Loss of connection (communication interruption) with a slave detected in the EtherCAT network.
0xC0CD0034	3234660404	A slave not belonging to a hot link group is offline.
0xC0CD0043	3234660419	A ring disconnection was detected when initializing the network card's physical layer. Communication can still use the right-side network port when this error code is reported. Note: This error code is displayed at reserved diagnostic area index 85.
0xC0CF0003	3234791427	Station address invalid.
0xC0CF000A	3234791434	The master sent an SDO command (e.g., read/write object dictionary, modify parameter, read status) to the slave. However, upon receiving it, the slave determined there was a problem with the command information, directly refused to execute it, and returned a 'Malformed Message' response, ultimately causing the master-slave SDO interaction to fail.
0xC0CF8002	3234824194	Acyclic communication request timed out.
0xC0CF8017	3234824215	When the master wrote a parameter to the slave via the CoE SDO service, the slave checked and found that the parameter value to be written fell outside the allowed range defined for that object dictionary entry (less than the minimum or greater than the maximum). The slave rejected the write and returned an SDO Abort Code, which the master encapsulated and reported as error code 0xC0CF8017.
0xC0CF801A	3234824218	The slave rejected the SDO request primarily because the device's current operating mode or state does not allow the requested operation.
0xCFF90000	3489202176	Inconsistency between the configuration and the actual physical connection within the hot link group.
0xCFFA0000	3489267712	Error setting the hot link identifier.
0xCFFB0000	3489333248	Slave device ID check failed, meaning the master detected a mismatch between the slave ID and the configuration.
0xCFFC0000	3489398784	Slave device ID and hot link identifier are the same.
0xCFFE0000	3489529856	Error parsing the ENI file, usually indicating the master configuration file has an abnormal format or content.
0xCFFFFFF9	3489660921	Hot link group is offline.
0xCFFFFFFB	3489660923	The entire EtherCAT network has no link; communication between the master and all slaves is interrupted.

EtherCAT Slave AL Status Code Error Codes		
0xC0D50001	3235184641	Unspecified error - The slave encountered an unclassified serious error.
0xC0D50002	3235184642	Insufficient memory on the slave.
0xC0D50003	3235184643	Slave device configuration invalid.
0xC0D50011	3235184657	Requested state transition is invalid.
0xC0D50012	3235184658	The state requested by the master is unknown or not supported.
0xC0D50013	3235184659	Slave does not support bootloader mode.
0xC0D50014	3235184660	Slave firmware is invalid or corrupted.
0xC0D50015	3235184661	BOOT mailbox configuration invalid.
0xC0D50016	3235184662	PRE-OP mailbox configuration invalid.
0xC0D50017	3235184663	Sync Manager configuration invalid.
0xC0D50018	3235184664	Slave has no valid input.
0xC0D50019	3235184665	Slave has no valid output.
0xC0D5001A	3235184666	Synchronization error.
0xC0D5001B	3235184667	Sync Manager watchdog timeout.
0xC0D5001C	3235184668	Sync Manager type configuration invalid.
0xC0D5001D	3235184669	Output configuration invalid.
0xC0D5001E	3235184670	Input configuration invalid.
0xC0D5001F	3235184671	Watchdog configuration invalid.
0xC0D50020	3235184672	Slave requires a cold start.
0xC0D50021	3235184673	Slave requires INIT state.
0xC0D50022	3235184674	Slave requires PRE-OP state.
0xC0D50023	3235184675	Slave requires SAFE-OP state.
0xC0D50024	3235184676	Invalid input mapping.
0xC0D50025	3235184677	Invalid output mapping.
0xC0D50026	3235184678	Configuration inconsistent, needs reconfiguration.
0xC0D50027	3235184679	Free-run mode not supported.
0xC0D50028	3235184680	Synchronous mode not supported.
0xC0D50029	3235184681	Free-run mode requires triple buffering.
0xC0D5002B	3235184683	Invalid output and input.
0xC0D5002D	3235184685	SYNC loss error.
0xC0D50030	3235184688	Invalid DC synchronization configuration.
0xC0D50031	3235184689	Invalid DC Latch configuration.
0xC0D50032	3235184690	PLL (Phase-Locked Loop) error.
0xC0D50033	3235184691	DC synchronization I/O error.
0xC0D50034	3235184692	DC synchronization timeout error.
0xC0D50035	3235184693	DC synchronization cycle time invalid.
0xC0D50036	3235184694	Sync0 cycle time error.
0xC0D50037	3235184695	Sync1 cycle time error.
0xC0D50041	3235184705	AoE (ADS over EtherCAT) mailbox fault.
0xC0D50042	3235184706	EoE (Ethernet over EtherCAT) mailbox fault.
0xC0D50043	3235184707	CoE (CANopen over EtherCAT) mailbox fault.
0xC0D50044	3235184708	FoE (File over EtherCAT) mailbox fault.
0xC0D50045	3235184709	SoE (Service over EtherCAT) mailbox fault.
0xC0D5004F	3235184719	VoE (Vendor over EtherCAT) mailbox fault.
0xC0D50050	3235184720	EEPROM has no PDI access rights.

0xC0D50051	3235184721	EEPROM error.
0xC0D50060	3235184736	Slave local restart (informational diagnostic).
0xC0D50061	3235184737	Device identification value updated (informational diagnostic).
0xC0D50070	3235184752	Mismatch between the list of detected modules (0xF030) and the configured modules (0xF050).
0xC0D50080	3235184768	Device voltage or temperature is too low.
0xC0D50081	3235184769	Device voltage or temperature is too high.

Chapter 13 Modbus TCP Network Communication

13.1 Network Overview

13.1.1 Introduction

Modbus TCP is a version of the Modbus protocol applied to Ethernet TCP/IP. It uses a standard Ethernet interface and has been widely used in Ethernet communications in today's industrial control field. It is commonly used for communication between SCADA software (or HMI) and LX controllers, and communication between LX controllers and other controllers (or devices).

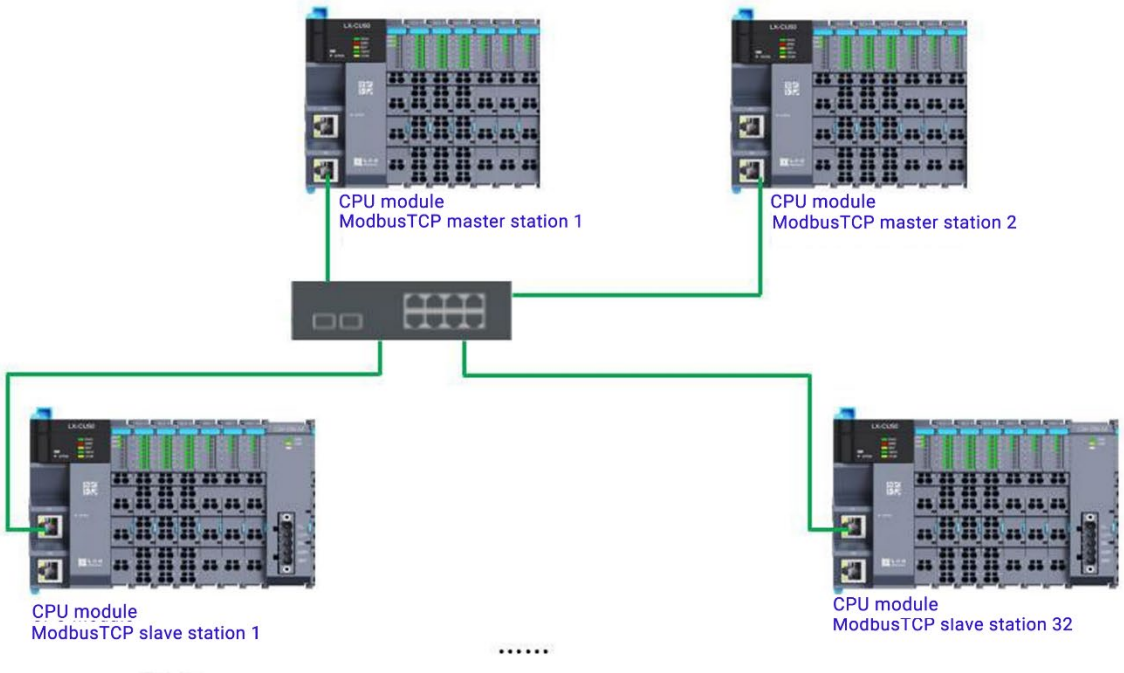
13.1.2 Network Characteristics

- Based on MODBUS protocol and standard Ethernet TCP/IP protocol, it is widely used.
- It supports multiple master stations.
- The network structure is flexible.

13.2 Network Topology

Modbus TCP supports various standard Ethernet connection architectures and can form a variety of network topologies such as ring, star, and tree topology through the use of switches.

The LX series CPU integrates two independent Ethernet interfaces, which can be used for interface expansion during network connection. When the LX series CPU serves as the Modbus TCP master station, it can establish connections with up to 32 slave stations.



List of devices required:

Module model	Number
LX-CU500	1

Requirements for communication cables:

Cable Model	Specifications	Communication Distance
Shielded twisted-pair cable	CAT5e	100 m

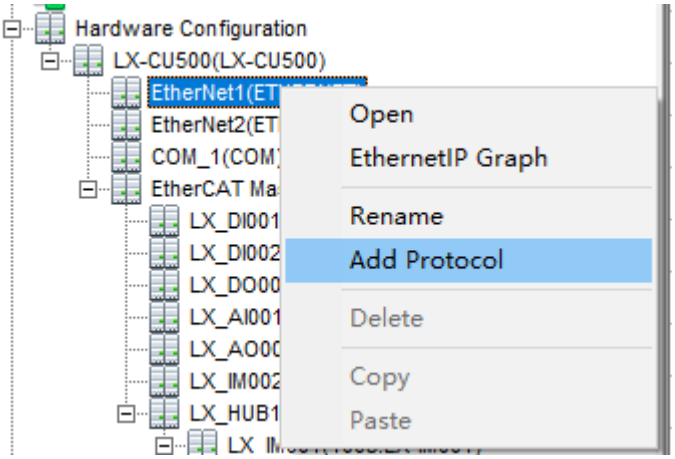
13.3 Configuration

13.3.1 Configuring Modbus TCP Master Station

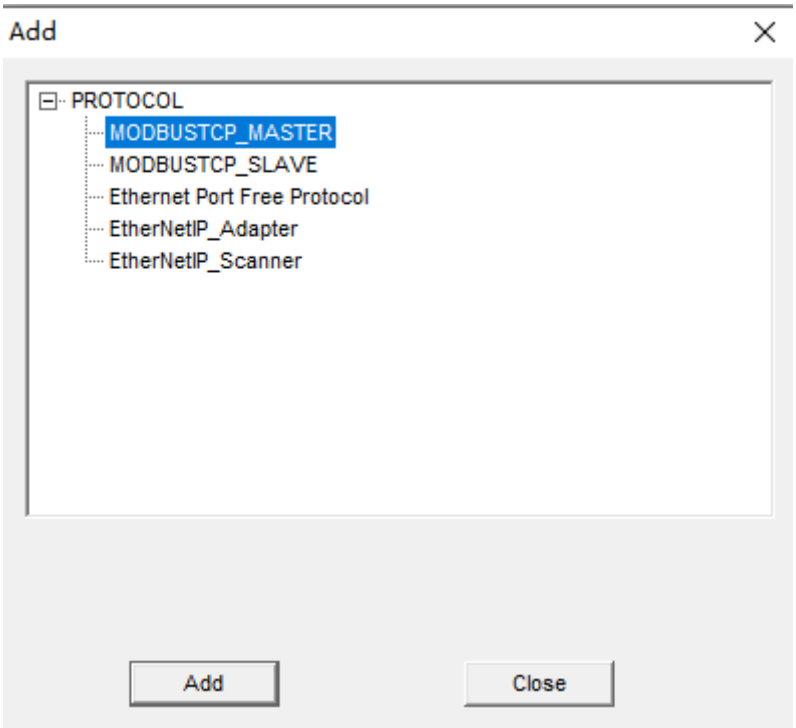
When the controller serves as the MODBUS TCP master station, its configuration process is as follows:

13.3.1.1 Adding protocol

(1) Select the Add Protocol command in the right-click menu of the EtherNet1 node, and the Add dialog box is displayed.



(2) Select MODBUSTCP_MASTER master station protocol.



13.3.1.2 Configuring master station protocol

■ Configuring master station parameters

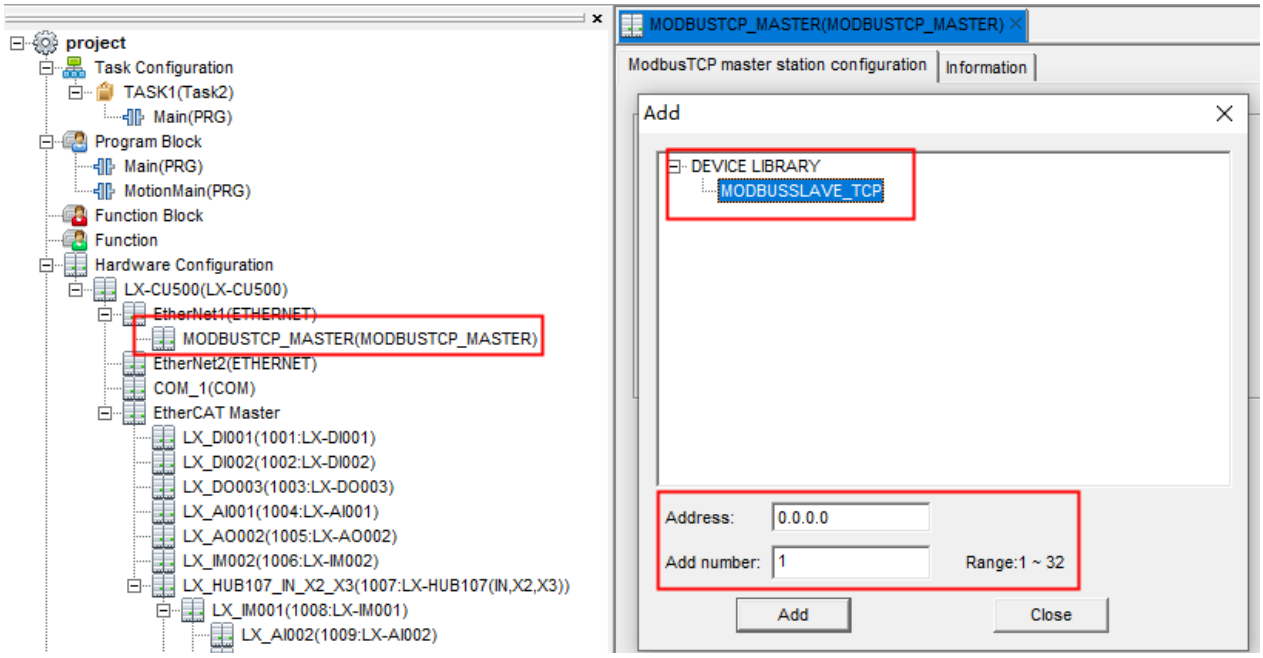
Double-click the MODBUSTCP_MASTER node or select the Open command in the right-click menu, and the master station configuration window is displayed, as shown in the figure.

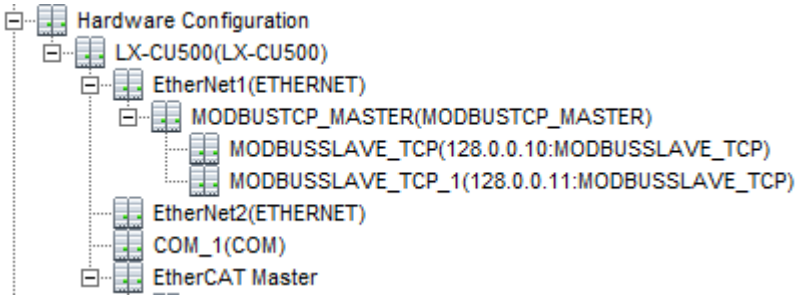
Master Station Parameters	Parameter Value	Default Value	Description
Response timeout (ms)	10~2,147,483,000	1000	Allowed slave station delay response time after the master station sends the request frame
Socket timeout (ms)	10~2,147,483,000	100	Timeout of TCP/IP connecting Socket
Number of retries	0~10	0	Number of times the master station resends the request after an abnormal response

- Note: In most cases, the default parameters can be used normally.

■ Adding a slave station under the master station protocol

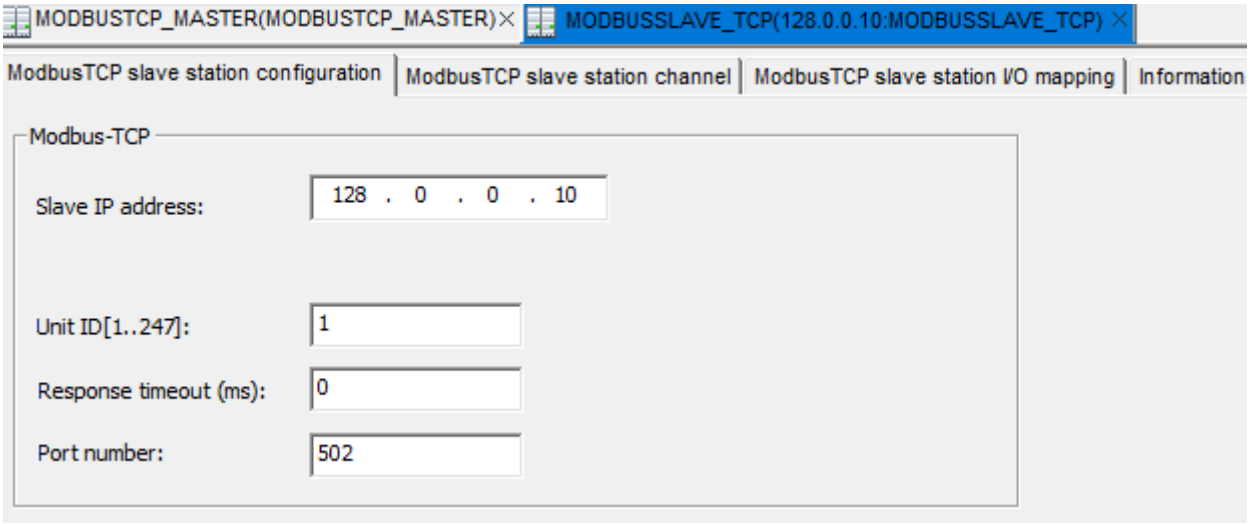
Select the Add Device command in the right-click menu of the MODBUSTCP_MASTER node, and the Add dialog box is displayed. Select MODBUSSLAVE_TCP, and the default IP address of the slave station 0.0.0.0 is displayed in the address box, which can be modified. Enter the number to add, click Add, and the slave station is added to the master station node.





■ MODBUS TCP slave station configuration

Double-click the MODBUSSLAVE_TCP node or select the Open command in the right-click menu, and the slave station configuration window is displayed, as shown in the figure.

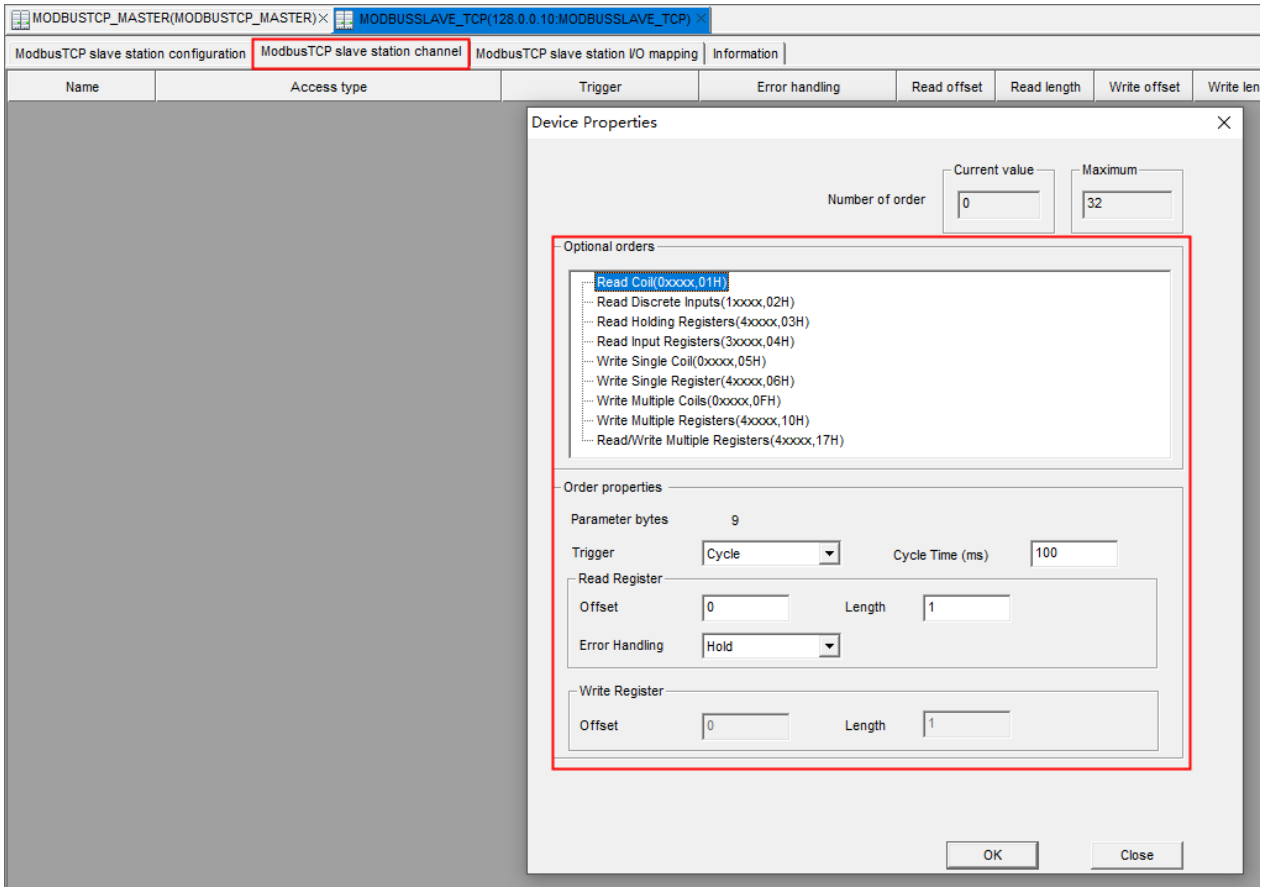


The description of slave station parameters is shown in the following table:

Parameter	Parameter Value	Default Value	Description	Remarks
Slave station IP address	Actual demand	0.0.0.0	Slave station IP address requested by the master station	
Unit ID	1~247	1	Modbus TCP protocol unit ID	
Response timeout (ms)	0~2,147,483,000	0	Default situation: The default timeout of the slave station is zero. At this time, the Response Timeout configured by the master station shall prevail. The user can individually configure the response timeout of a slave station. If the parameter configuration is greater than zero, the timeout of the slave station is subject to the current slave station configuration.	
Port number	1~65,535	502	Modbus TCP protocol port	Not recommended to modify

■ MODBUS TCP slave station channel

In the ModbusTCP Slave Station Channel tab, you can add operations to the slave station through the Add command in the right-click menu, as shown in the figure.

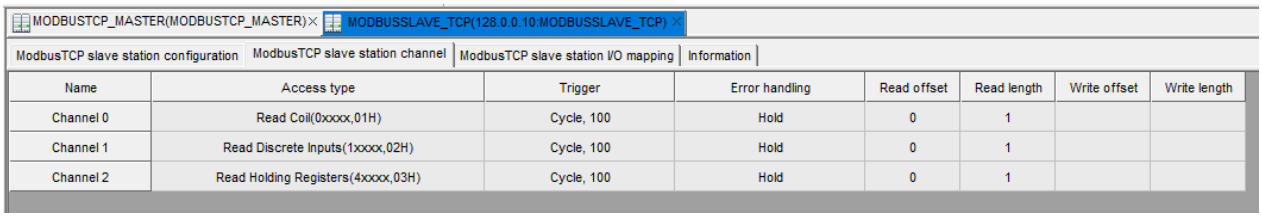


Parameter		Description
Trigger	Cycle	The communication mode between the master station and the slave station is polling
	Rising edge	When the trigger variable is rising edge, the master station performs a slave station operation
	High level	When the trigger variable is high level, the master station polls the slave station
Cycle time (ms)	100	Polling cycle when the trigger variable is cycle
Read register	Offset	Range 0~65535; the set value is the starting address of the instruction
	Length	Read coil/read discrete input: 1~2000 Read holding register/read input register: 1~125 Read/write multiple registers: 1~118
	Error handling	Keep: Keep the current data after an abnormal response Clear: Clear the current data after an abnormal response
Write register	Offset	Range 0~65535; the set value is the starting address of the instruction
	Length	Write multiple coils: 1~1968 Write multiple registers: 1~123 Read/write multiple registers: 1~118

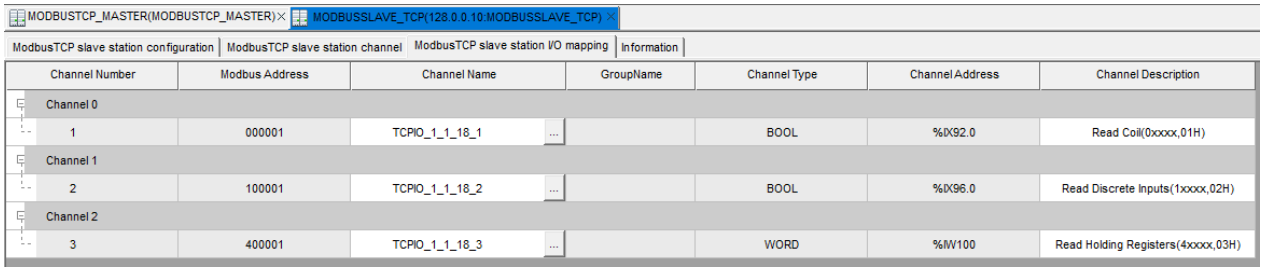
■ Slave station I/O mapping (online data and Modbus address)

After the instruction is configured, the corresponding I/O channel will be mapped in the ModbusTCP Slave Station I/O Mapping tab.

□ Configured instructions:



□ IO mapping corresponding to the configured instructions:



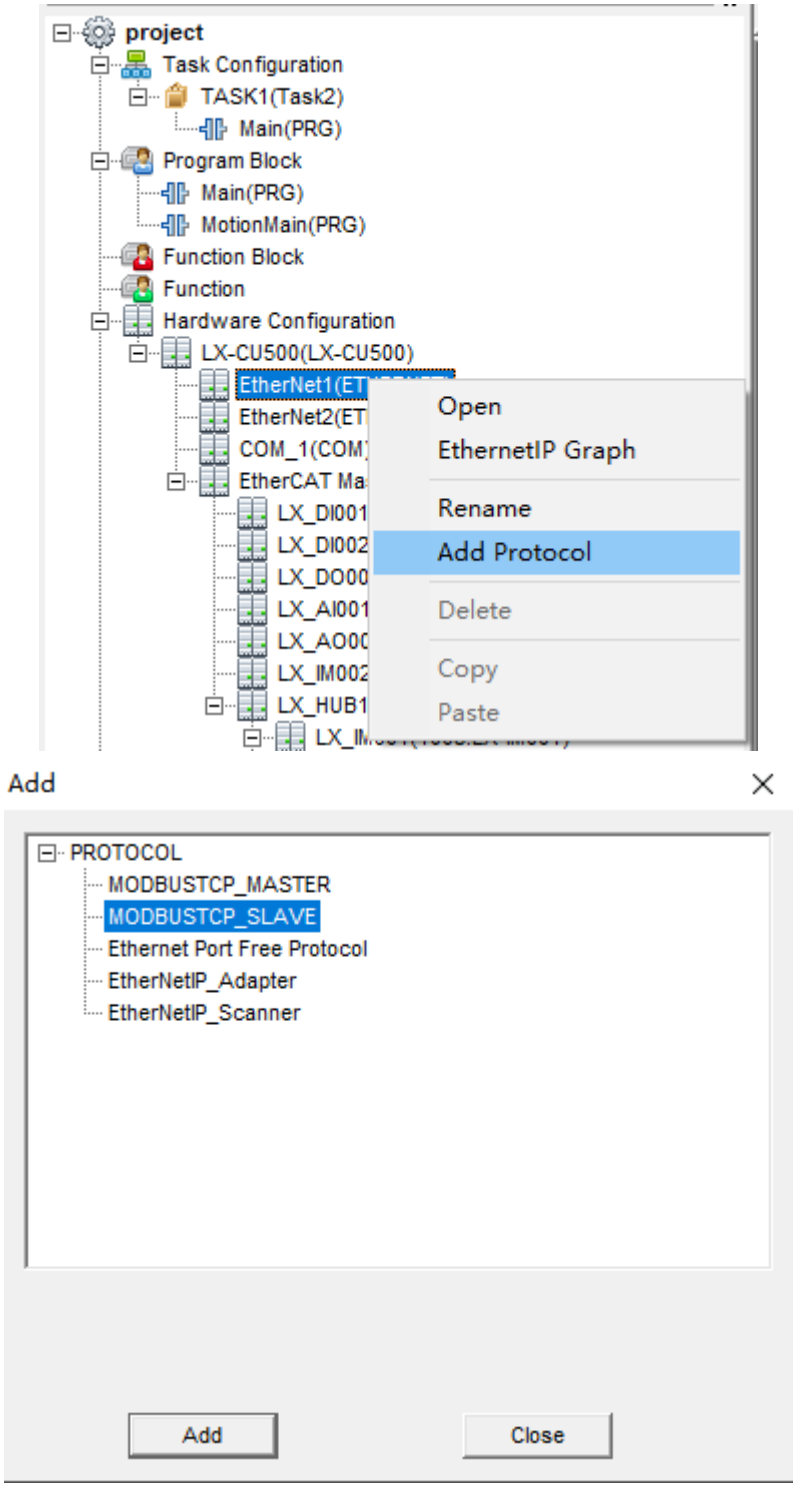
Parameter	Description
Channel Number	Channel 0~n: Corresponding to the instruction name (Name in ModbusTCP Slave Station Channel)
	Serial number 1 or 1~10 under Channel x, corresponding to the length of the instruction (If the length of Channel 0 is 1, there is only 1 row; if the length of Channel 3 is 5, there are 5 rows, namely 5~9)
	Channel 1 is a rising edge trigger, and Channel 3 is a high level trigger, both of which have a Trigger Variable
Modbus Address	Actual Modbus address of the instruction

13.3.2 Configuring Modbus TCP Slave Station

When the controller serves as the MODBUS TCP slave station, its configuration process is as follows:

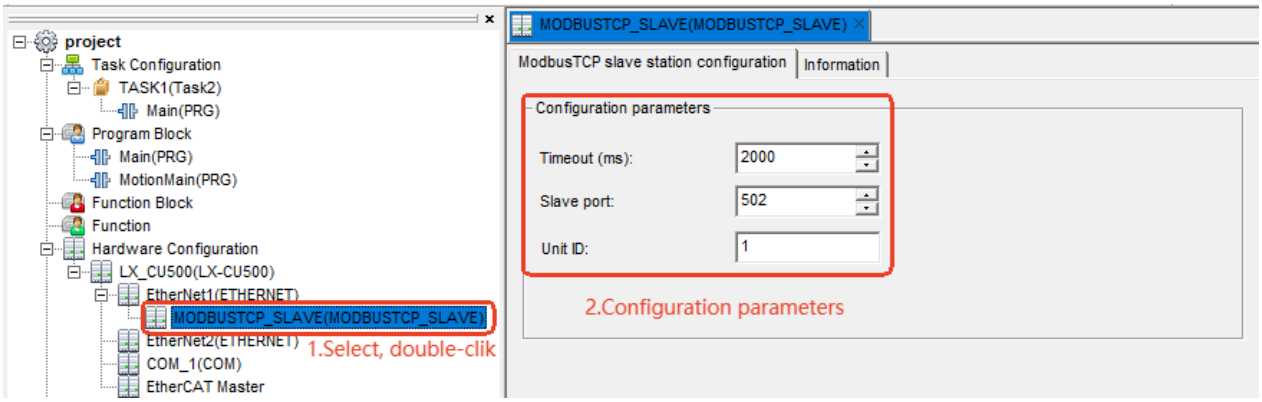
(1) Adding slave station protocol

Select the Add Protocol command in the right-click menu of the EtherNet1 node, and the Add dialog box is displayed, as shown in the figure, and select the MODBUSTCP_SLAVE slave station protocol.



(2) Configuring slave station protocol

Double-click the added MODBUSTCP_SLAVE slave station protocol or select the Open command in the right-click menu, and the MODBUS TCP slave station configuration window is displayed, as shown in the figure.



The description of slave station parameters is shown in the following table:

Parameter	Parameter Value	Default Value	Description
Timeout (ms)	1~2,147,483,000	2000	The master station will disconnect if it does not poll within the timeout period.
Slave Station Port	1~65,535	502	Modbus TCP protocol port
Unit ID	1~247	1	Modbus TCP protocol unit ID

13.3.3 Data mapping relationship

The data areas that the MODBUS TCP slave station protocol can access through BOOL or WORD data include: input area (area I), output area (area Q), and intermediate area (area M).

The mapping relationship between the data area and the Modbus TCP slave station communication protocol is shown in the following table.

Data Area		Type	Address Range	Modbus Address	Mapping Relationship	X (Register Type) Selection
Area I	%IX	BOOL	%IX0.0, …, %IX0.7 %IX1.0, …, %IX1.7 …, %IX374.7	X00000, …, X00007 X00008, …, X00015…, X02999	IXm.n: m*8+n	Read only, select 1 for X
	%IW	WORD	%IW0, %IW2, …, %IW5998	X00000, X00001, …, X02999	IWm: m/2	Read only, select 3 for X
Area Q	%QX	BOOL	%QX0.0, …, %QX0.7 %QX1.0, …, %QX1.7 …, %QX374.7	X00000, …, X00007 X00008, …, X00015…, X02999	QXm.n: m*8+n	Read/write, select 0 for X
	%QW	WORD	%QW0, %QW2, …, %QW5998	X00000, X00001, …, X02999	QWm: m/2	Read/write, select 4 for X
Area M	%MX	BOOL	%MX0.0, …, %MX0.7 %MX1.0, …, %MX1.7 …, %MX7816.7	X03000, …, X03007 X03008, …, X03015, …, X65535	MXm.n: m*8+n+3000	Read only, select 1 for X Read/write, select 0 for X
	%MW	WORD	%MW0, %MW2, …, %MW125070	X03000, X03001, …, X65535	MWm: m/2+3000	Read only, select 3 for X Read/write, select 4 for X

Mapping relationship description:

- Reading area I (or writing area Q) switch data: For example, when reading a section of data starting from %IX2.6 (or writing %QX2.6), you need to refer to the mapping formula of BOOL type data in area I (or area Q), and the starting address of the read (or write) slave station filled in the master station shall be: 2*8+6=22;

- Reading area I (or writing area Q) analog data: For example, when reading a section of data starting from %IW8 (or writing %QW8), you need to refer to the mapping formula of WORD type data in area I (or area Q), and the starting address of the read (or write) slave station filled in the master station shall be: 8/2=4;
- Reading/writing switch data in area M: For example, when reading/writing a section of data starting from %MX2.6, you need to refer to the mapping formula of BOOL type data in area M and the starting address of the read/write slave station filled in the master station shall be 2* 8+6+3000=3022;
- Reading/writing analog data in area M: For example, when reading/writing a section of data starting from %MW1000, you need to refer to the mapping formula of WORD type data in area M and the starting address of the read/write slave station filled in the master station shall be 1000/2+3000=3500.

13.4 Modbus TCP Communication Example

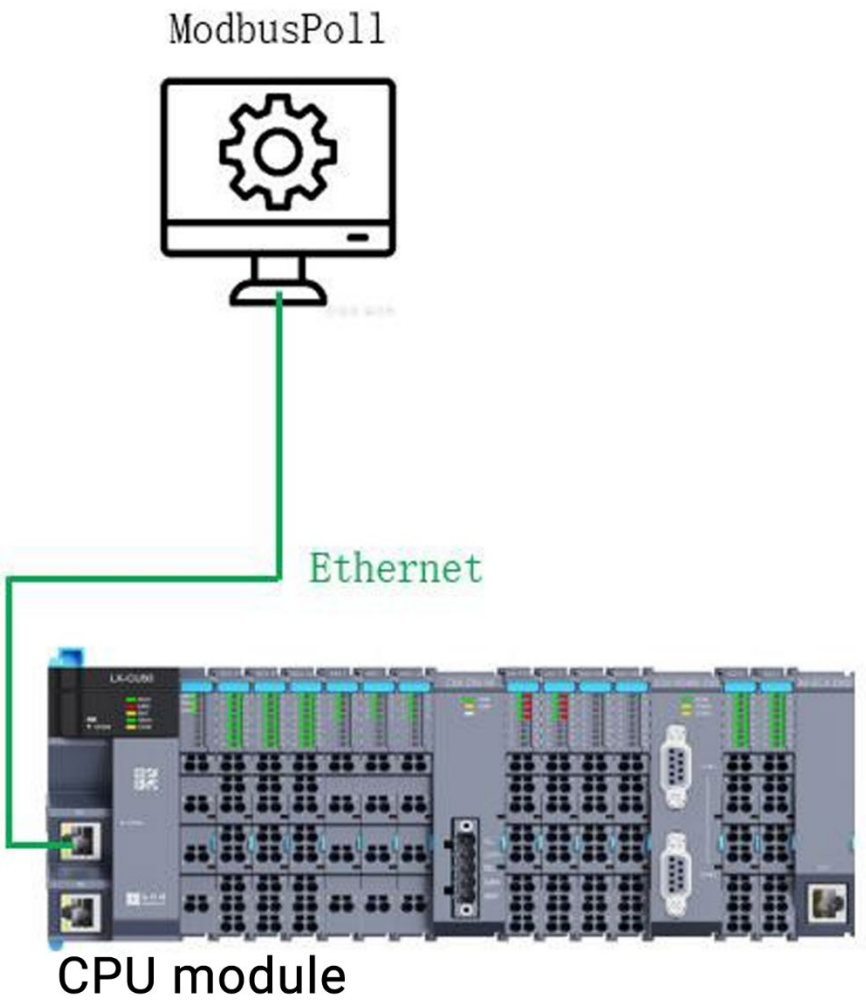
13.4.1 Example Function

LX-CU500 serves as a Modbus TCP slave station to communicate with the ModbusPoll analog master station. The master station reads and writes data from the operation controllers %MX2.6 and %MW1000.

13.4.2 System Composition

- Simulate the master station host.
- CPU LX-CU500 serves as a Modbus TCP slave station.

13.4.3 Example Topology



13.4.4 Programming

(1) Step 1: Add the slave station protocol MODBUSTCP_SLAVE, with default parameters.

(2) Step 2: Perform engineering configuration of 2 arrays, starting from the addresses %MX2.0 and %MW1000.

序号	变量名 △	直接地址	变量说明	变量类型	初始值	掉电保护
0001	p1	%MX2.0		ARRAY[0..10] OF BOOL		FALSE
0002	p2	%MW1000		ARRAY[0..10] OF WORD		FALSE

(3) ModbusPoll simulates the master station, and the configuration functions are as follows:

Function code 1: read %MX2.6, Modbus address: 3022;

Function code 3: read %MX2.6, Modbus address: 3022;

Function code 5: write %MX1000, Modbus address: 3500;

Function code 16: write %MX1000, Modbus address: 3500.

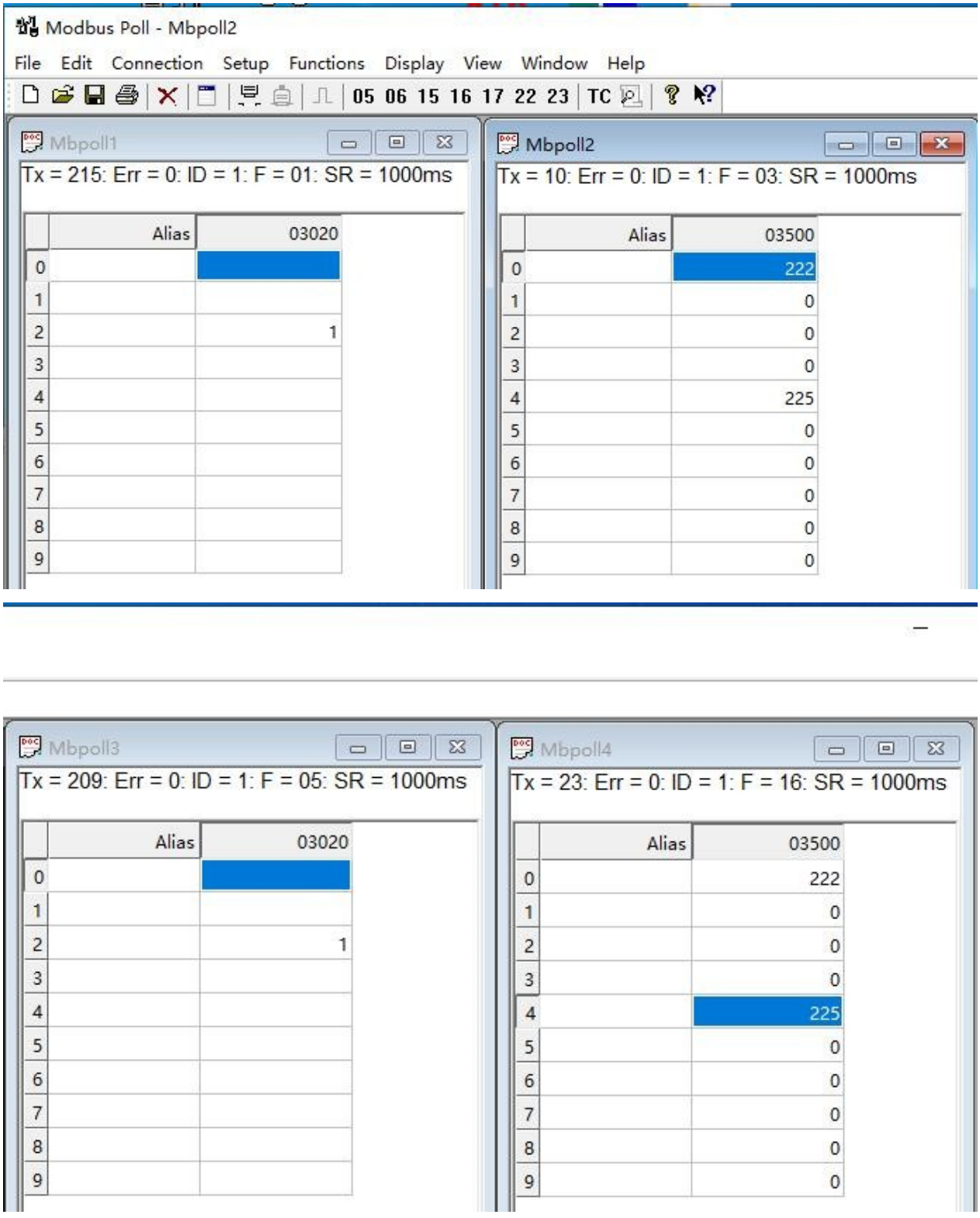
13.4.5 Online Debugging

(1) After configuring the slave station parameters in ModbusPoll, configure the following 4 types of slave station operations.

- Function code 1: read %MX2.6, Modbus address: 3022;
- Function code 3: read %MX2.6, Modbus address: 3022;
- Function code 5: write %MX1000, Modbus address: 3500;
- Function code 16: write %MX1000, Modbus address: 3500.

(2) Click Connect, and then you can view normal communication through the ModbusPoll software interface.

The ModbusPoll software data monitoring interface is as follows:



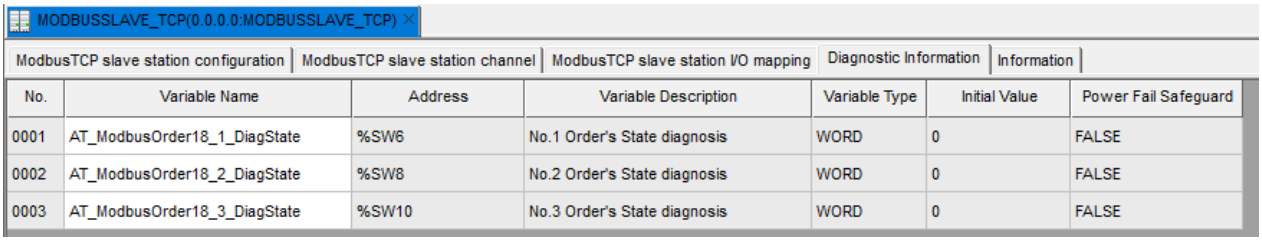
The AutoThink variable monitoring table is as shown in the figure below:

0001	p1[0]	%MX2.0		BOOL	FALSE	FALSE
0002	p1[1]	%MX2.1		BOOL	FALSE	FALSE
0003	p1[2]	%MX2.2		BOOL	FALSE	FALSE
0004	p1[3]	%MX2.3		BOOL	FALSE	FALSE
0005	p1[4]	%MX2.4		BOOL	FALSE	FALSE
0006	p1[5]	%MX2.5		BOOL	FALSE	FALSE
0007	p1[6]	%MX2.6		BOOL	TRUE	FALSE
0008	p1[7]	%MX2.7		BOOL	FALSE	FALSE
0009	p1[8]	%MX3.0		BOOL	FALSE	FALSE
0010	p1[9]	%MX3.1		BOOL	FALSE	FALSE
0011	p1[10]	%MX3.2		BOOL	FALSE	FALSE

0001	p2[0]	%MW1000		WORD	222	FALSE
0002	p2[1]	%MW1002		WORD	0	FALSE
0003	p2[2]	%MW1004		WORD	0	FALSE
0004	p2[3]	%MW1006		WORD	0	FALSE
0005	p2[4]	%MW1008		WORD	225	FALSE
0006	p2[5]	%MW1010		WORD	0	FALSE
0007	p2[6]	%MW1012		WORD	0	FALSE
0008	p2[7]	%MW1014		WORD	0	FALSE
0009	p2[8]	%MW1016		WORD	0	FALSE
0010	p2[9]	%MW1018		WORD	0	FALSE
0011	p2[10]	%MW1020		WORD	0	FALSE

13.5 Fault Diagnosis

When the LX controller serves as the Modbus TCP master station, after the ModbusTCP slave station channel is configured and compiled, the Diagnostic Information tab appears in the slave station configuration window.



When online, the status of each instruction can be diagnosed through the value of the diagnostic status:

Diagnostic Value	Meaning	Possible Causes or Solutions
0	No error	-

1	TCP connection not established or disconnected	Check if the network connection, IP address/port number, slave configuration, etc. are normal: 1. Configuration check: (1) If the slave device is a HollySys LX series controller module, ensure that the ModbusTCP_Slave protocol is added under the EtherNet node corresponding to the Modbus TCP communication IP segment in the slave module's configuration project. (2) Adjust the master's socket timeout. 2. Check network connectivity: Use the ping command to test if the slave IP is reachable, check network cable connections and LED status. 3. Check IP configuration: Ensure master and slave IP addresses are on the same subnet with no address conflicts, and gateway settings are correct. 4. Check TCP port: Ensure the master and slave Modbus TCP port is configured correctly (default 502), test using telnet or a port scanning tool. 5. Check firewall: Temporarily disable firewalls (including system and network firewalls) to test if they are blocking the connection. 6. Check slave configuration: Ensure the slave allows connections, with no blacklists, connection limits, etc. 7. Comparative testing: Test with a third-party standard Modbus TCP master, or use a network packet capture tool (e.g., Wireshark) to analyze the connection process. 8. Contact the device manufacturer: If the above steps fail, contact the slave device manufacturer.
2	Slave response timeout	1. Configuration check: Adjust the response timeout on the master or the slave. 2. Check the master's command polling frequency, adjust it based on different types (periodic, high-level, rising edge) to reduce the slave load. 3. Check network latency and stability: Use the ping command to test network latency and packet loss; check the load on network devices (switches, routers). 4. If the slave device is a HollySys LX series controller module, ensure the master's command request interval is less than the timeout configured on the slave side. 5. Contact the device manufacturer: If the above steps fail, contact the slave device manufacturer.
4	Function code in the slave response does not match the master request	1. Check network latency and stability: Use the ping command to test network latency and packet loss; check the load on network devices (switches, routers). 2. Adjust timeout: Increase the master timeout appropriately to avoid sequence number confusion caused by retransmissions due to latency. 3. Check the master's command polling frequency, adjust it based on different types (periodic, high-level, rising edge) to reduce the slave load. 4. Packet capture analysis: Use tools like Wireshark to capture Modbus TCP packets, analyze the request-response correlation, and find the cause of sequence number confusion. 5. One-to-one connection test to eliminate interference from other devices. 6. Comparative testing: Test the same slave with a third-party standard Modbus master, or test the master with a standard slave. 7. Contact the device manufacturer: If the above steps fail, contact the slave device manufacturer.
8	Data length in the slave response does not match the master's expected length	1. Check request message: Ensure the data length requested by the master (number of registers/coils) is within the slave's allowable range (many slaves have a maximum length limit, e.g., 125 registers). 2. Check network latency and stability: Use the ping command to test network latency and packet loss; check the load on network devices (switches, routers). 3. Adjust timeout: Increase the master timeout appropriately to avoid sequence number confusion caused by retransmissions due to latency. 4. Check the master's command polling frequency, adjust it based on different types (periodic, high-level, rising edge) to reduce the slave load. 5. Check response message: Monitor messages to see if the slave returns an exception response (shorter length); if so, handle according to the exception code. 6. Packet capture analysis: Use tools like Wireshark to capture Modbus TCP packets, analyze the request-response correlation, and find the cause of sequence number confusion.

		7. One-to-one connection test to eliminate interference from other devices. 8. Comparative testing: Test the same slave with a third-party standard Modbus master, or test the master with a standard slave. 9. Contact the device manufacturer: If the above steps fail, contact the slave device manufacturer.
16	TCP connection established, but command not sent	The trigger condition for the command has not been met.
64	Sequence number of the response does not match the request	1. Check network latency and stability: Use the ping command to test network latency and packet loss; check the load on network devices (switches, routers). 2. Adjust timeout: Increase the master timeout appropriately to avoid sequence number confusion caused by retransmissions due to latency. 3. Check the master's command polling frequency, adjust it based on different types (periodic, high-level, rising edge) to reduce the slave load. 4. Packet capture analysis: Use tools like Wireshark to capture Modbus TCP packets, analyze the request-response correlation, and find the cause of sequence number confusion. 5. One-to-one connection test to eliminate interference from other devices. 6. Comparative testing: Test the same slave with a third-party standard Modbus master, or test the master with a standard slave. 7. Contact the device manufacturer: If the above steps fail, contact the slave device manufacturer.
128	The Unit ID in the request message does not match the Unit ID configured in the ModbusTCP_Slave protocol	1. Modify the Unit ID on either the master or the slave to ensure consistency on both sides. 2. Use a third-party standard Modbus master to test the same slave and observe the Unit ID returned by the slave.

Chapter 14 TCP/IP Network Communication

14.1 Network Overview

14.1.1 Introduction

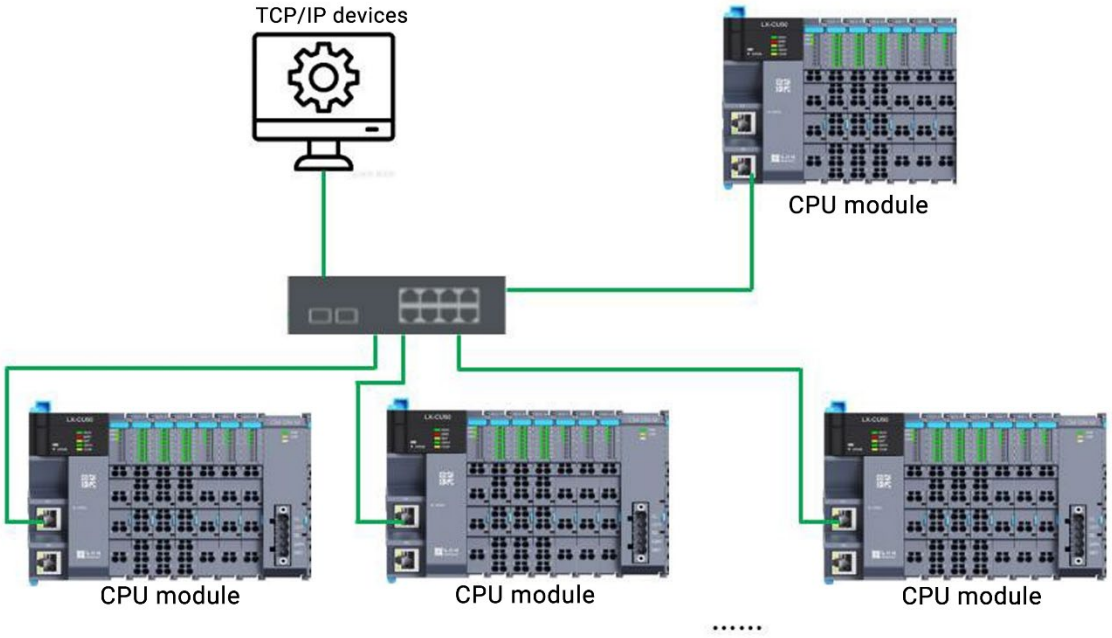
The LX controller can communicate with other devices in TCP/UDP custom formats via TCP/IP.

14.1.2 Network Characteristics

- Based on standard Ethernet TCP/IP protocol.
- Support custom data transmission under TCP/UDP protocol.
- The network structure is flexible.
- The maximum number of connections is 16.

14.2 Network Topology

It supports various standard TCP/IP connection architectures and can be used to form ring, star, tree, and other network topologies through the use of switches, with flexible configurations.



List of devices required:

Module Model	Number
LX-CU500	1

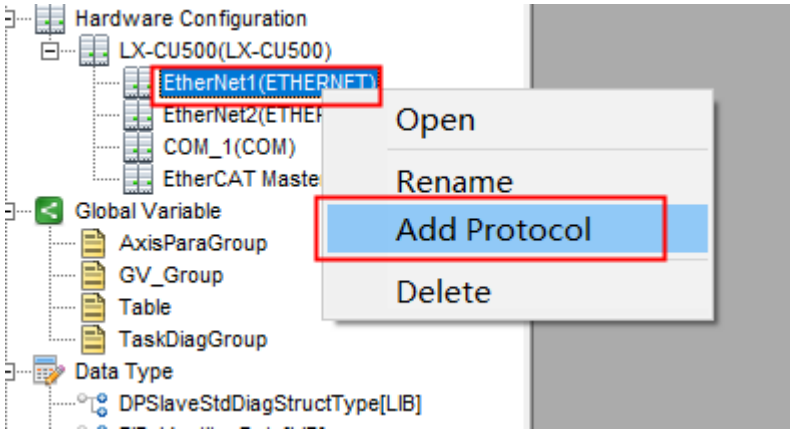
Requirements for communication cables:

Cable Model	Specifications	Communication Distance
Shielded twisted-pair cable	CAT5e	100 m

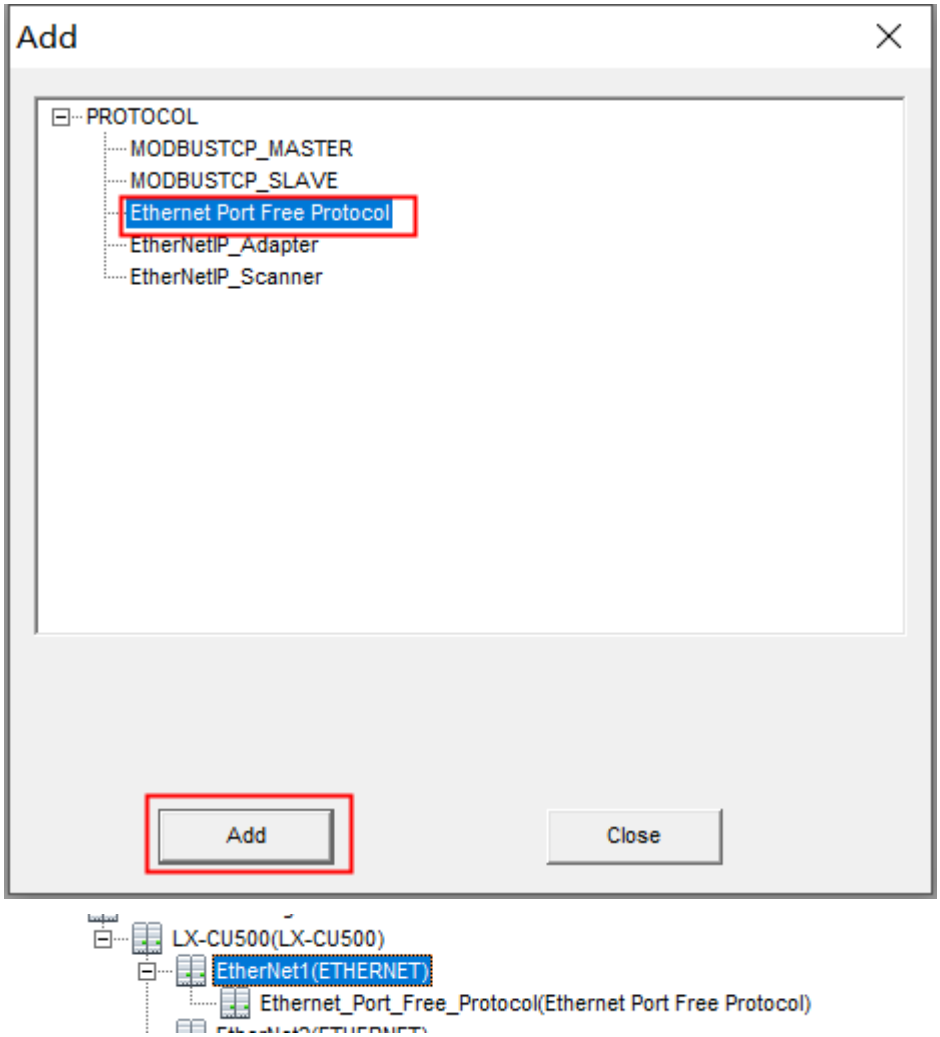
14.3 Configuration

14.3.1 Adding protocol

(1) Select the Add Protocol command in the right-click menu of the EtherNet1 node, and the Add dialog box is displayed.



(2) Select Ethernet Port Free Protocol and click Add to complete the hardware configuration.



14.4 TCP/IP Communication Example

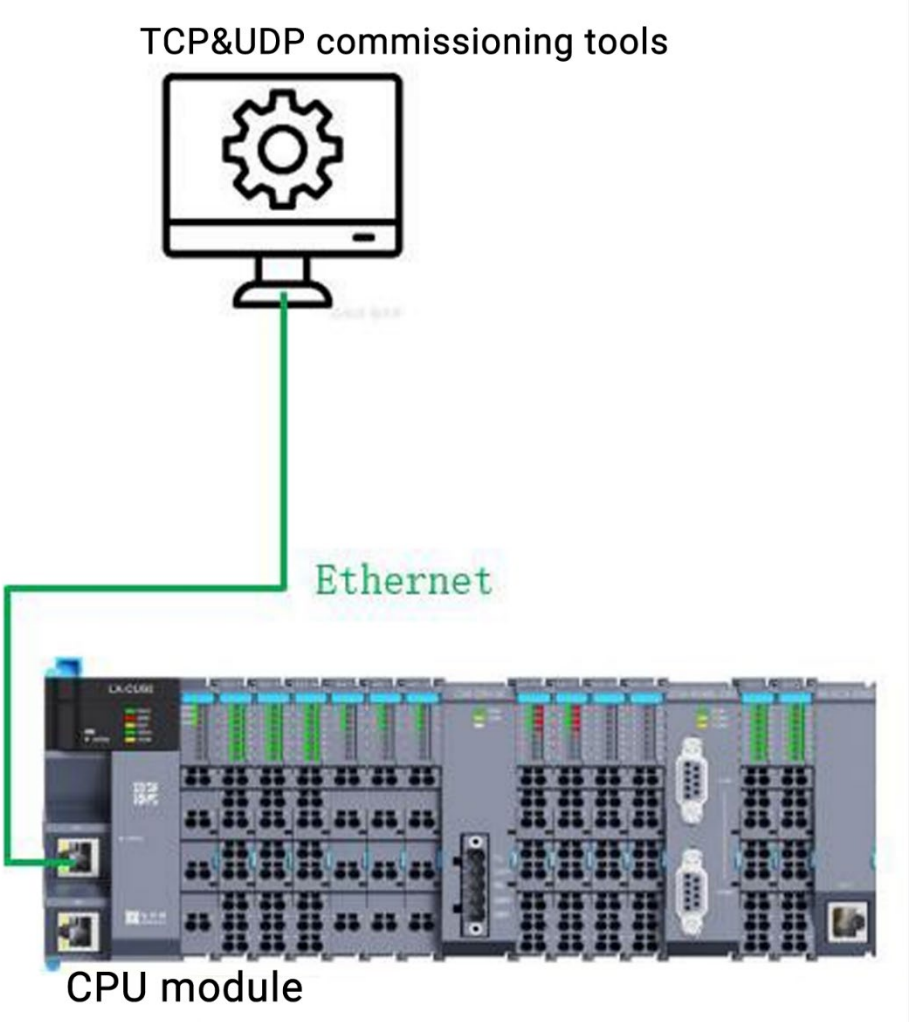
14.4.1 Example Function

It uses the TCP&UDP debugging tool and LX-CU500 to conduct Ethernet free communication test. LX-CU500 serves as a TCP server to receive the connection request from the client and realize the bidirectional data Rx/Tx function.

14.4.2 System Composition

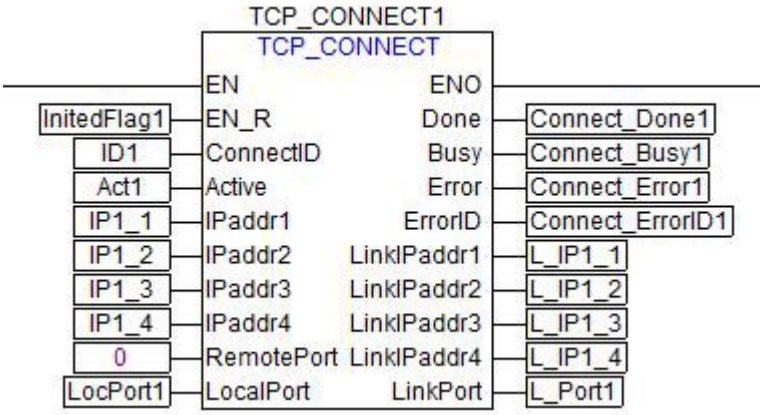
- CPU module LX-CU500.
- Communication test host + TCP&UDP debugging tool software.

14.4.3 Example Topology



14.4.4 Programming

- TCP instruction configuration
  - AT configures the TCP_CONNECT functional block

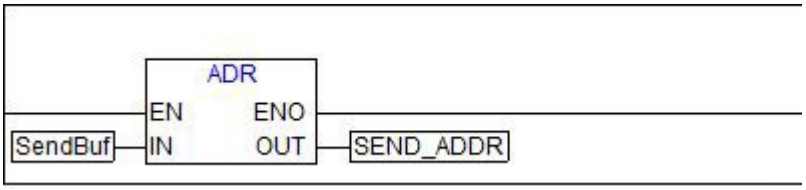


Parameter Name	Description of Parameters
EN_R	It is used to enable the TCP connection operation. It shall be triggered via a rising edge instruction so that the operation is initiated only once. When the output parameter Busy is true, the program will ignore the EN_R input.
ConnectID	It is the connection identifier. When a connection is created via TCP_CONNECT, ConnectID is created which can be chosen to be any value in the range 1 to 16. Each connection must have a unique ConnectID. The program uses ConnectID to specify the connection required for subsequent send, receive, and disconnect operations.
Active	When Active=TRUE creates an active connection, the local CPU initiates a communication connection request to the remote device. When Active=FALSE creates a passive connection, the local CPU waits for the remote device to initiate a communication connection request.
IPaddr1~IPaddr4	It is the IP address of the remote device to be connected. IPaddr1 is the most significant byte of the IP address, and IPaddr4 is the least significant byte of the IP address. For example, for the IP address 192.168.0.101, the following values are set: IPaddr1=192 IPaddr2=168 IPaddr3=0 IPaddr4=101 The IP address cannot be the following values: 0.0.0.0 (for active connections) Any broadcast IP address (for example, 255.255.255.255) Any multicast address IP address of local CPU The IP address 0.0.0.0 can be used for passive connections. By selecting the IP address 0.0.0.0, the CPU accepts connections from any remote IP address. If a non-zero IP address is selected for a passive connection, the CPU will only accept connections from the specified address.
RemotePort	Port number on the remote device. The port number can be used for connections with TCP and UDP protocols. The rules for remote port numbers are as follows: Valid port number range: 1 to 49151 The recommended port number range is 2000 to 49151. For passive connections, the CPU ignores the remote port number (it can be set to zero).
LocalPort	It is the port number on the local CPU. The port number can be used for connections with TCP and UDP protocols. For all passive connections, the local port number must be unique. The rules for local port numbers are as follows: Valid port number range: 1 to 49151 Port numbers 20, 21, 25, 80, 102, 135, 161, 162, 443, 1200, and 34962 to 34964 cannot be used. These ports are used for specific purposes. The recommended port number range is 2000 to 49151. For passive connections, the local port number must be unique (not duplicated).
Done	When the operation is completed without errors, the OUC instruction sets the Done output. If the instruction sets the Done output, the Busy, Error, and ErrorID outputs are zero. Other outputs (e.g. number of bytes received) are valid only when the Done output is set.
Busy	The Busy output indicates that an operation is in progress. When an operation is initiated by

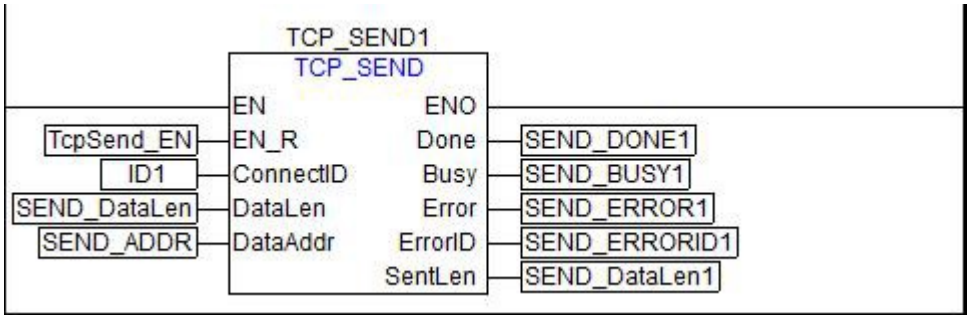
	setting EN_R to TRUE, the OUC instruction sets the Busy output. For all subsequent calls to the instruction, the Busy output remains set until the operation is completed.
Error	The Error output indicates that the operation is completed with errors. If the OUC instruction sets the Error output, the Done and Busy outputs will be set to FALSE. If the instruction sets the Error output, the ErrorID output will display the error cause. If the Error output is set, all other outputs are invalid.
ErrorID	See Fault Diagnosis
LinkIPaddr1 ~ LinkIPaddr4	It is the IP address of the remote device that has been connected. LinkIPaddr1 is the most significant byte of the IP address, and LinkIPaddr4 is the least significant byte of the IP address.
LinkPort	Port number of connected remote device

□ ending TCP communication data

Define data sending buffer:



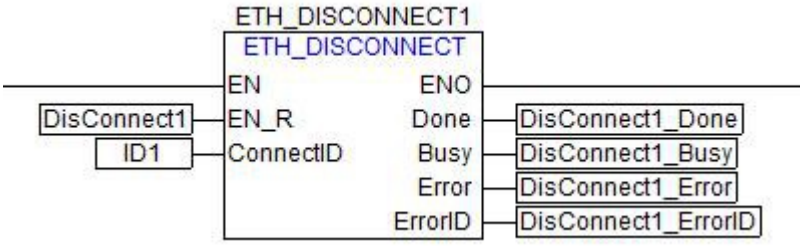
AT configures the TCP_SEND functional block:



Parameter Name	Description of Parameters
EN_R	Rising edge: The CPU starts sending operation.
ConnectID	The connection ID (ConnectID) is the number of the connection used by this send operation. Use the ConnectID you selected for the TCP_CONNECT operation.
DataLen	Data length requested to send, (1~32*1024)Byte
DataAddr	POINTER TO BYTE type, and DataAddr is a pointer to the storage location of the sent data
Done	When the send operation is completed without errors, the instruction sets the Done output.
Busy	When the send operation is in progress, the instruction sets the Busy output
Error	When the sending operation is completed but an error occurs, the instruction sets the Error output
ErrorID	See Fault Diagnosis
SentLen	SentLen is the actual number of bytes sent. Only valid when Done or Error is output.

□ Closing TCP communication

AT configures the ETH_DISCONNECT functional block to close the Ethernet connection.



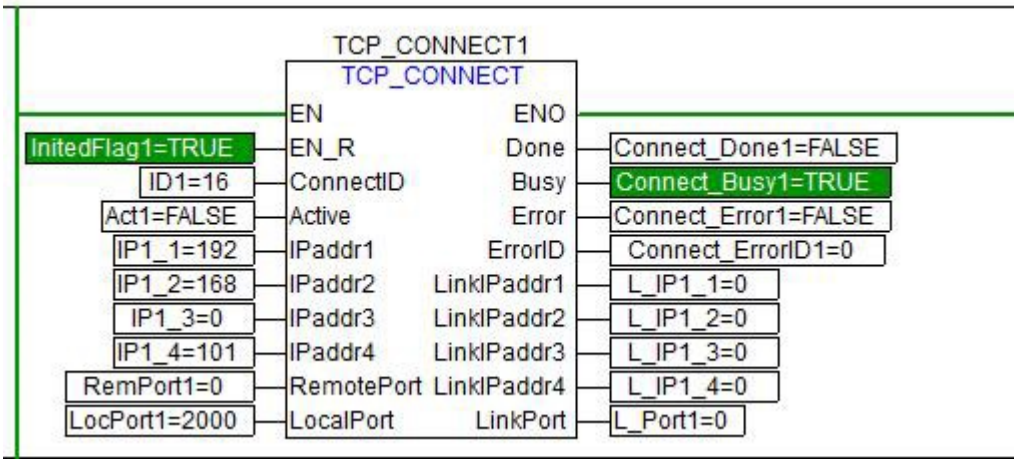
Parameter Name	Description of Parameters
EN_R	Rising edge trigger closing connection
ConnectID	The CPU uses the connection ID (ConnectID) to identify the connection to be terminated (defined during the connection process)
Done	When the disconnection operation is completed without errors, the instruction sets the Done output
Busy	When the disconnection operation is in progress, the instruction sets the Busy output
Error	When the disconnection operation is completed but an error occurs, the instruction sets the Error output
ErrorID	See Fault Diagnosis

14.4.5 Online Debugging

- Establishing a connection

(1) Set the Active input variable of the TCP_CONNECT instruction to FALSE, adopt the passive connection mode, and set the controller as the TCP server.

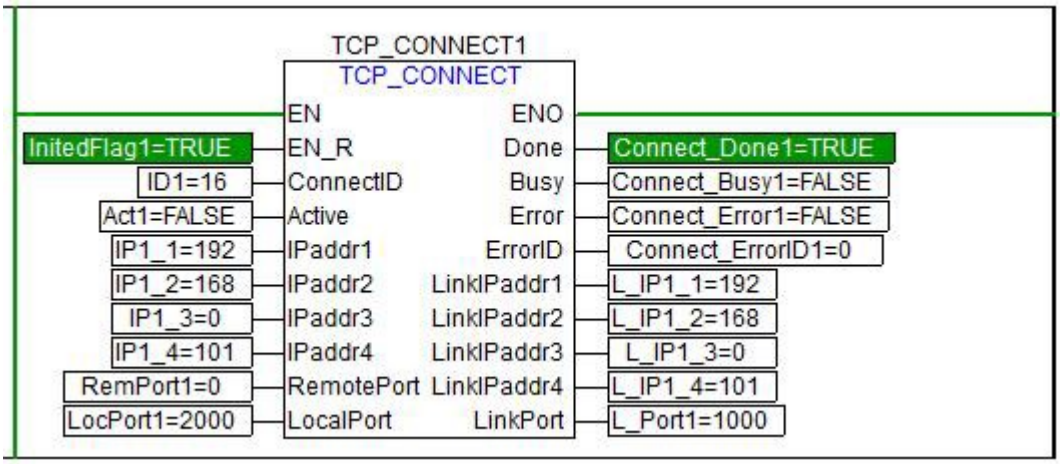
(2) Enable the functional block EN_R pin variable (rising edge trigger) and wait for the client to initiate a connection.



(3) Select a TCP communication debugging tool as the client device.

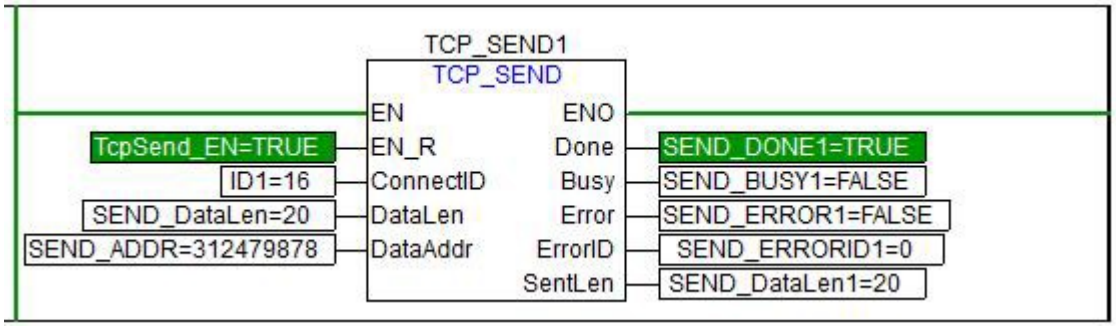
(4) Initiate a TCP connection to make the connection successful.

If the connection between the controller and the debugging tool is successful, the output parameter Done is TRUE, and if not, the cause needs to be found based on the fault code.



■ Sending data

Set the EN_R pin variable of the TCP_SEND functional block (rising edge trigger) and send the data in the SendBuf variable array to the client.



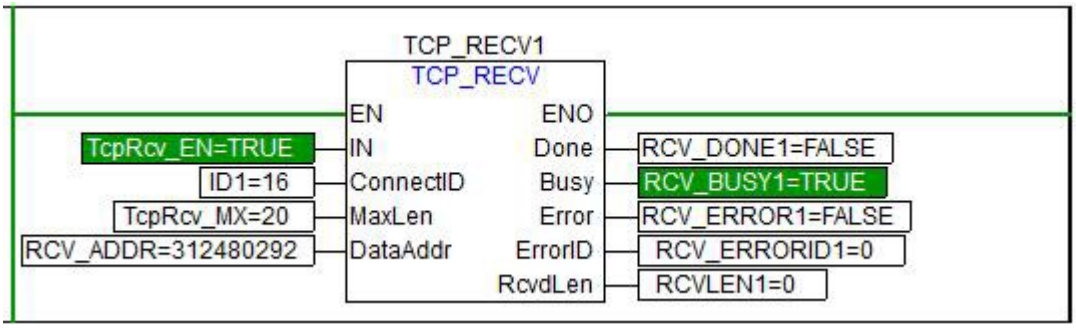
Main.SendBuf

No.	Variable Name	Address	Variable Description	Variable Type	Online Value
0001	SendBuf[0]			WORD	1
0002	SendBuf[1]			WORD	2
0003	SendBuf[2]			WORD	3
0004	SendBuf[3]			WORD	4
0005	SendBuf[4]			WORD	5
0006	SendBuf[5]			WORD	6
0007	SendBuf[6]			WORD	7
0008	SendBuf[7]			WORD	0
0009	SendBuf[8]			WORD	0
0010	SendBuf[9]			WORD	0

Make sure the data received by the debugging tool are the same as the data sent by controller.

■ Receiving data

Set the EN_R pin variable of the TCP_RCV functional block and wait to receive data sent by the client.



Send the data of a certain length by using the debugging tool.

The TCP_RECV functional block continues to receive data. If the reception is successful, the output parameter Done is TRUE, RcvdLen is the actual received data length, and the received data can be seen in the recvBuf array variable configured by AT.

Make sure the data received by controller are the same as the data sent by debugging tool.

No.	Variable Name	Type	Online Value	Variable Description
0001	MAIN.RcvBuf[0]	WORD	WORD#16#11	
0002	MAIN.RcvBuf[1]	WORD	WORD#16#22	
0003	MAIN.RcvBuf[2]	WORD	WORD#16#33	
0004	MAIN.RcvBuf[3]	WORD	WORD#16#44	
0005	MAIN.RcvBuf[4]	WORD	WORD#16#55	
0006	MAIN.RcvBuf[5]	WORD	WORD#16#66	
0007	MAIN.RcvBuf[6]	WORD	WORD#16#77	
0008	MAIN.RcvBuf[7]	WORD	WORD#16#88	
0009	MAIN.RcvBuf[8]	WORD	WORD#16#99	
0010	MAIN.RcvBuf[9]	WORD	WORD#16#1111	

14.5 Fault Diagnosis

Error code ErrorID description of Ethernet free protocol communication instructions:

Error Code	Error Message Description
0	No error
1	Wrong input ConnectID
2	Invalid input IP address
3	Invalid input remote port number
4	The input local port number is invalid
5	The input parameters changed while trying to connect using the TCP_CONNECT instruction
6	The input information for this active connection is duplicated with other connection information while trying to use the TCP_CONNECT instruction to create an active connection

7	Trying to use the TCP_CONNECT instruction to create a passive connection, but the input information for this connection is duplicated with other connection information
8	Connection closed
9	Non-TCP type connection
10	Failed to establish connection
11	The connection object refused the connection
12	Received a connection request from a non-configured IP address
13	Ethernet protocol link is not configured with the Ethernet free protocol function
14	Incorrect data length or address information while sending or receiving data using the TCP or UDP free protocol
15	TCP_SEND data sending failed
16	TCP_RECV data receiving failed
17	Non-UDP type connection
18	ETH_DISCONNECT instruction is being used to close the closed link
19	UDP_CONNECT connection input information is duplicated with other connection information
20	Trying to connect using the UDP_CONNECT instruction, but the input parameters changed
21	An error occurred during TCP_CONNECT setting socket options
22	TCP_CONNECT port binding process failed
23	TCP_CONNECT link listening process failed
24	An error occurred during UDP_CONNECT setting socket options
25	UDP_CONNECT port binding process failed
26	Multiple tasks are performing the same connection operation at the same time

Chapter 15 Serial Port Communication

15.1 Network Overview

15.1.1 Introduction

It is used to communicate with devices with RS-232/RS-485 interfaces. The communication supports Modbus RTU master/slave station protocol and COM port free protocol.

15.1.2 Network Characteristics

The communication network of serial port has the following characteristics:

Serial port is a very common interface for communication between devices, and is also widely used for communication between devices and instruments;

Common serial ports include RS-232, half-duplex RS-485 and full-duplex RS-422;

Strong versatility and easy networking;

RS-232 can only achieve point-to-point communication, with the communication distance not exceeding 15 meters;

The RS-485 interface has the advantages of strong anti-interference ability, long transmission distance and multi-station capability, making it the preferred serial interface.

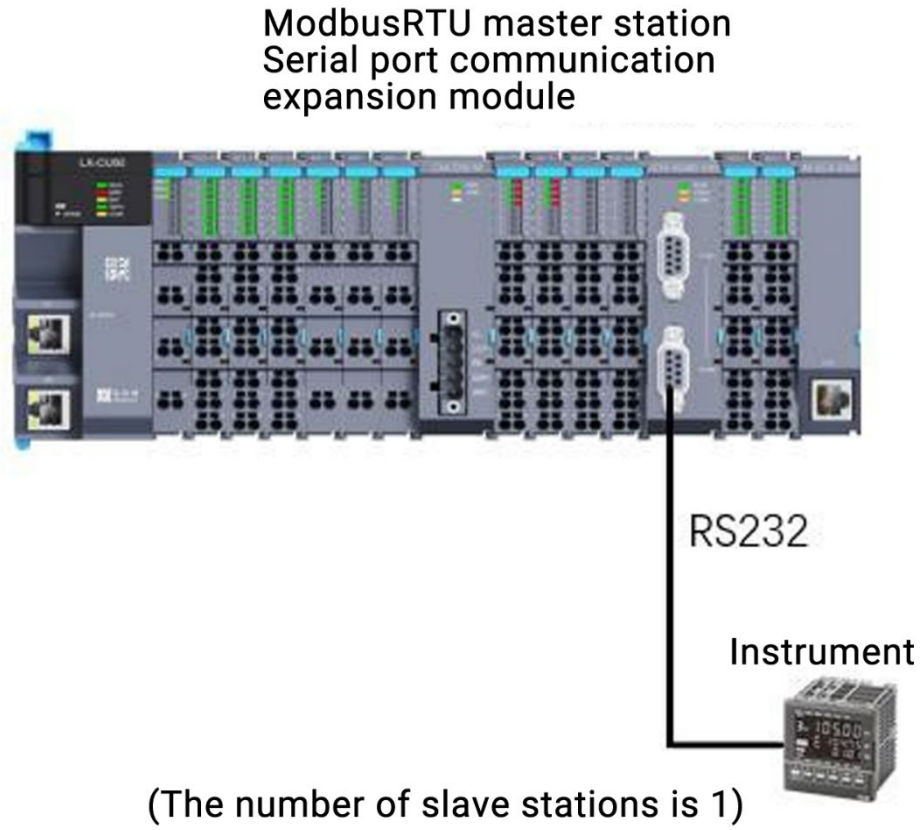
15.2 Network Topology

The LX series PLC serial ports are divided into two types: RS-232 and RS-485.

The CPU module integrates an RS-485 terminal interface for RS485 expansion connection. There are also 4 types of serial port modules that support DB9 pin type and terminal type expansion connections of RS-232 and RS-485 interfaces respectively.

15.2.1 RS-232 Network Topology

The RS-232 interface is limited to point-to-point communication between two devices, with a maximum transmission distance of 15 meters.



List of devices required:

Module Model	Number
LX-CM001	1
LX-CM002	1

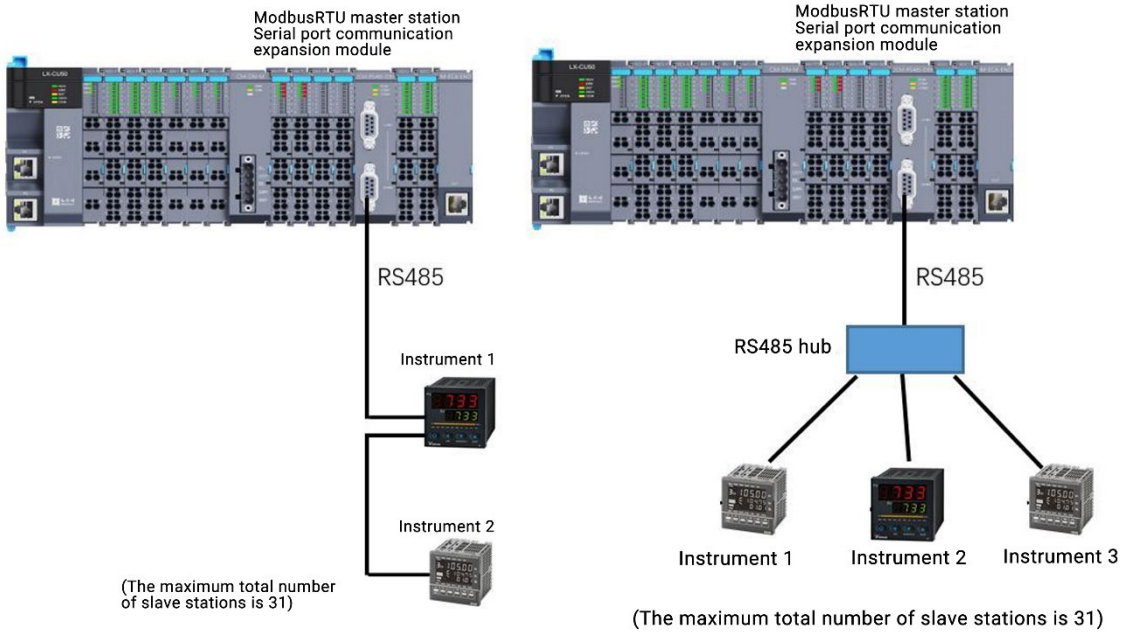
Requirements for communication cables:

Cable Model	Specifications	Communication Distance
RS-232 cable	Shielded cable D-SUB 9-pin serial cable	15 m

15.2.2 RS-485 Network Topology

RS-485 can connect up to 32 nodes on the same bus, which can realize true multi-point communication. However, it generally adopts the master-slave communication mode, and there can only be one master station.

The RS-485 bus has daisy chain topology and star topology.



List of devices required:

Module model	Number
LX-CM-003	1
LX-CM-004	1

Requirements for communication cables:

Cable Model	Specifications	Communication Distance
3105A shielded twisted-pair	1*2*22 AWG, 120 Ω	1200 m

15.3 Configuration

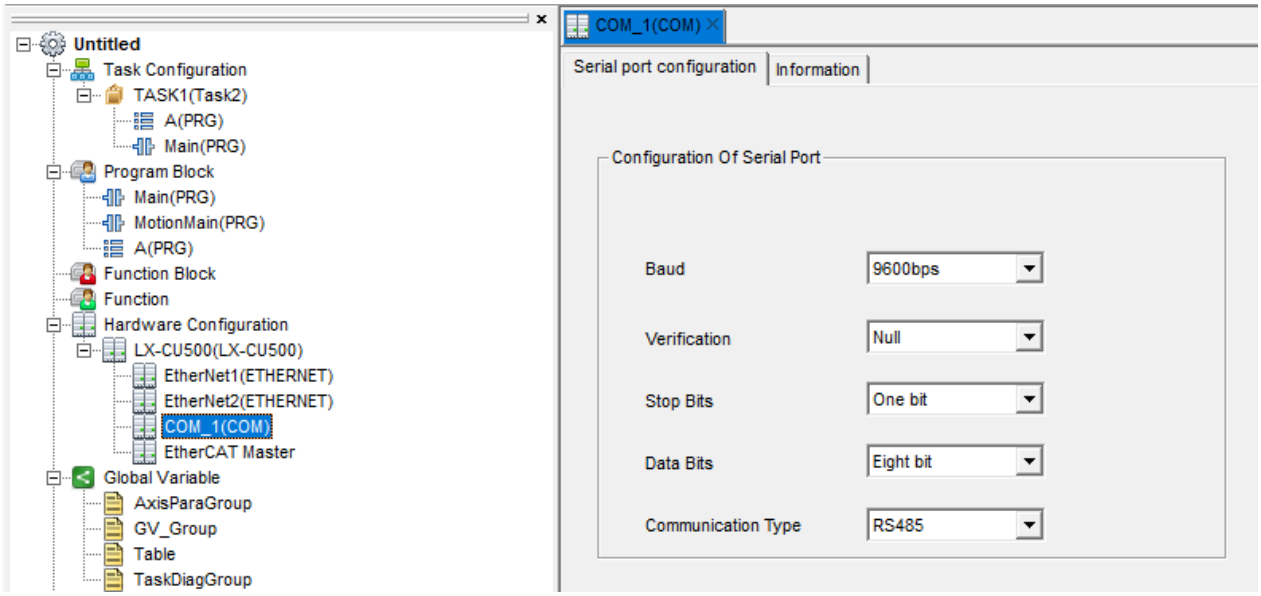
15.3.1 Configuring Modbus RTU Master Station

15.3.1.1 Configuration process using the CPU integrated RS-485 interface

When the controller serves as the MODBUS RTU master station, its configuration process is as follows:

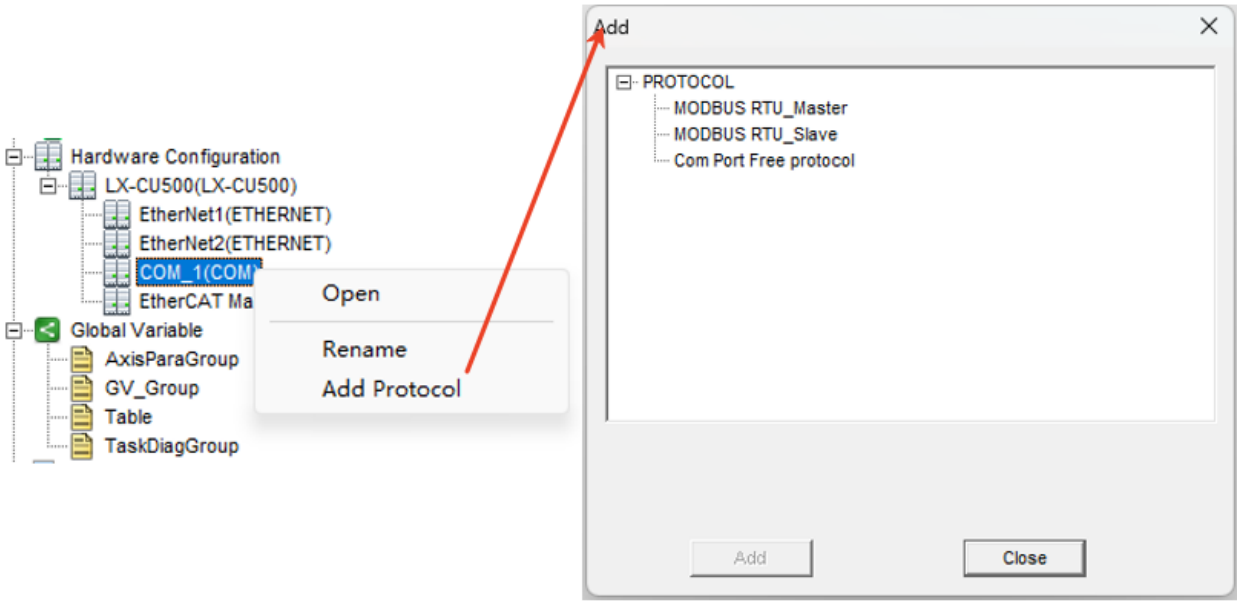
- COM parameter configuration

  Double-click on the COM_1 node, and the serial port configuration window is displayed, then modify the serial port parameters.



■ Adding protocol

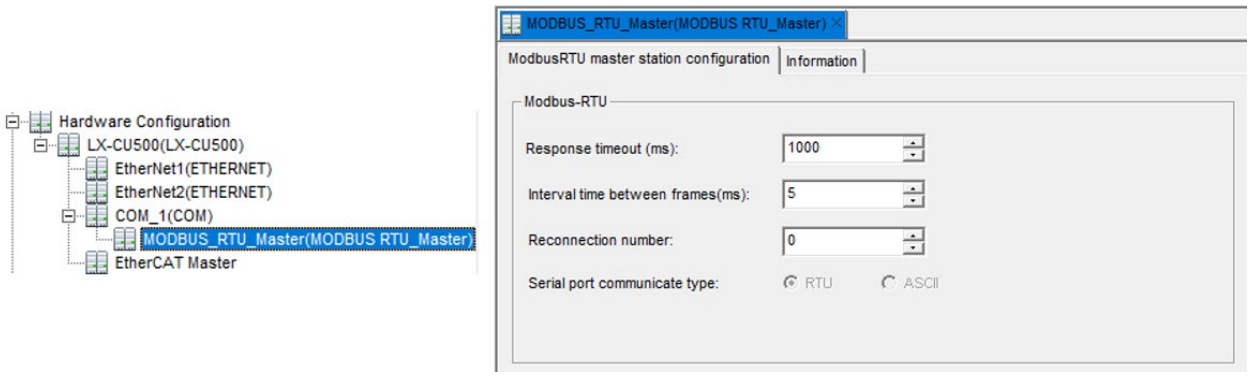
Select the Add Protocol command in the right-click menu of the COM_1 node, and the Add dialog box is displayed.



Select MODBUS RTU_MASTER master station protocol. Click the Add button.

■ Configuring master station protocol

Double-click the MODBUS_RTU_MASTER node or select the Open command in the right-click menu, and the master station configuration window is displayed, as shown in the figure.

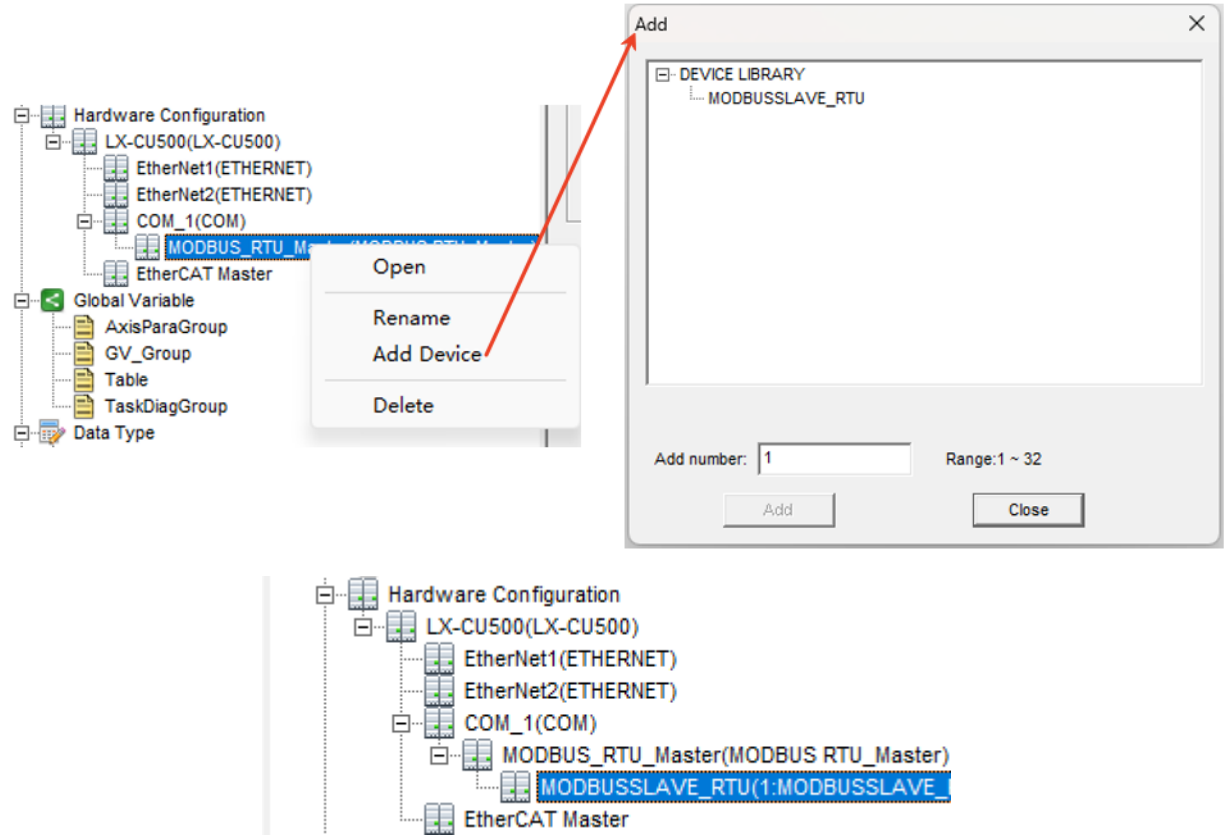


The master station parameters are described as follows:

Parameter	Parameter Value	Default Value	Description
Response timeout (ms)	10~65535	1000	Allowed slave station delay response time after the master station sends the request frame
Frame to frame interval time (ms)	2~65535	5	After the serial port detects that the time interval between two frames of data is greater than the set value, it is considered that one frame has been received.
Number of retries	0~10	0	Number of times the master station resends the request after an abnormal response

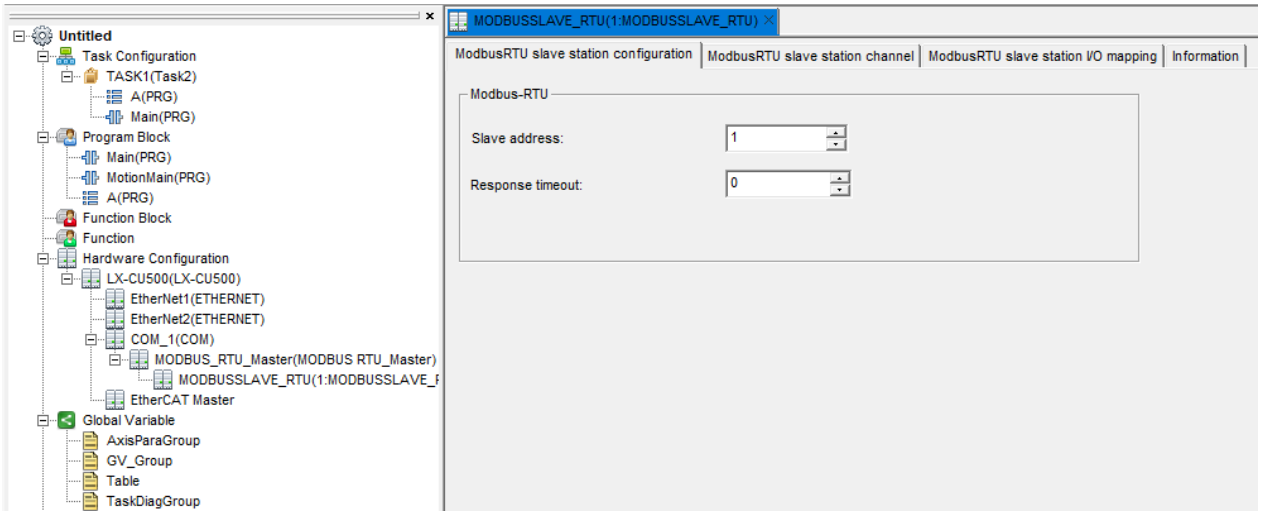
■ Adding a slave station under the master station protocol

Select the Add Device command in the right-click menu of the MODBUS_RTU _MASTER node, and the Add dialog box is displayed, then select MODBUSSLAVE_RTU, enter the number to add, click Add, and the slave station is added to the master station node. Up to 32 slave stations can be added.



■ MODBUS RTU slave station configuration

Double-click the MODBUSSLAVE_RTU node or select the Open command in the right-click menu and the slave station configuration window is displayed, as shown in the figure.



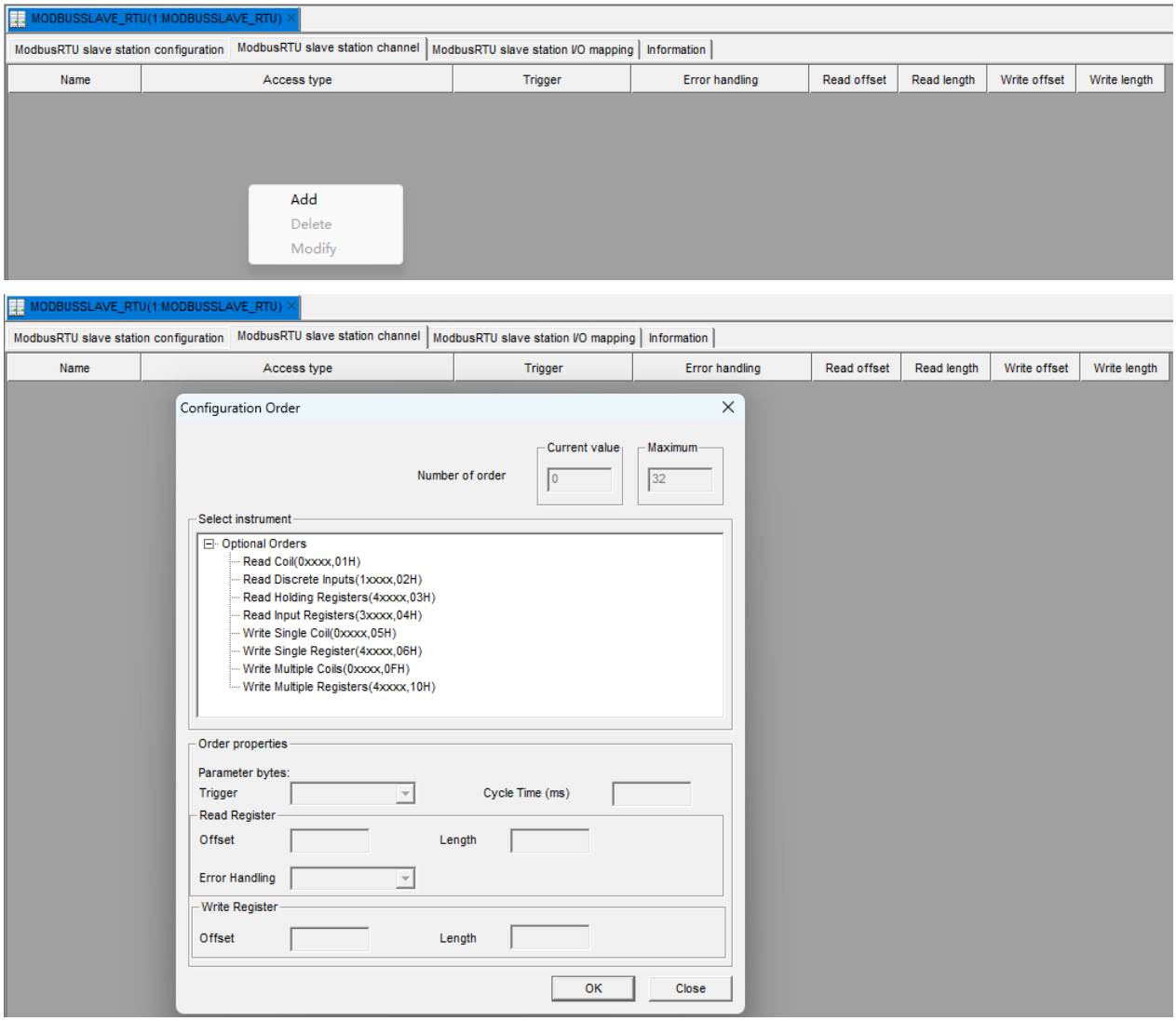
The slave station parameters are described as follows:

Parameter	Parameter Value	Default Value	Description
Slave station address	1~247	1	Modbus RTU slave station device ID
Response timeout (ms)	0~65535	0	Default situation: The default timeout of the slave station is zero. At this time, the Response Timeout configured by the master station shall prevail. The user can

			individually configure the response timeout of a slave station. If the parameter configuration is greater than zero, the timeout of the slave station is subject to the current slave station configuration.

■ MODBUS RTU slave station channel

In the ModbusRTU Slave Station Channel tab, you can add operations to the slave station through the Add command in the right-click menu, as shown in the figure.



The description of Modbus RTU slave station channel parameters is as follows:

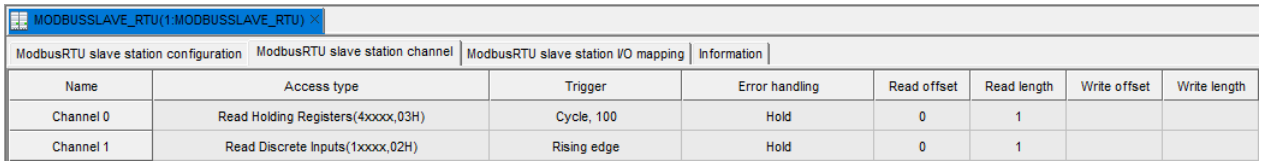
Parameter		Description
Trigger	Cycle	The communication mode between the master station and the slave station is polling
	Rising edge	When the trigger variable is rising edge, the master station polls slave stations
	High level	When the trigger variable is high level, the master station performs a slave station operation
Cycle time (ms)	100	Polling cycle when the trigger variable is cycle
Read register	Offset	Range 0~65535; the set value is the starting address of the instruction
	Length	Read coil/read discrete input: 1~2000 Read holding register/read input register: 1~125 Read/write multiple registers: 1~118

	Error handling	Keep: Keep the current data after an abnormal response Clear: Clear the current data after an abnormal response
Write register	Offset	Range 0~65535; the set value is the starting address of the instruction
	Length	Write multiple coils: 1~1968 Write multiple registers: 1~123 Read/write multiple registers: 1~118

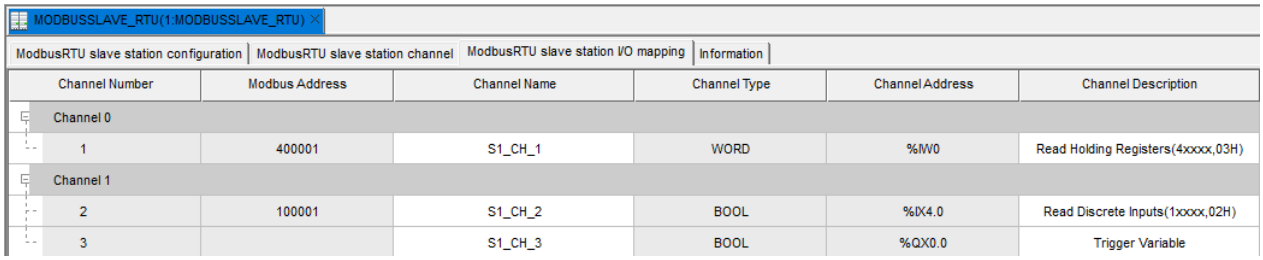
■ Slave station I/O mapping (online data and Modbus address)

After the instruction is configured, the corresponding I/O channel will be mapped in the ModbusRTU Slave Station I/O Mapping tab.

Configured instructions:



I/O mapping corresponding to the configured instructions:



Parameter	Description
Channel Number	Channel 0~n: Corresponding to the instruction name (Name in ModbusRTU Slave Station Channel)
	Serial number 1 or 1~10 under Channel x, corresponding to the length of the instruction (If the length of Channel 0 is 1, there is only 1 row; if the length of Channel 3 is 5, there are 5 rows, namely 5~9)
	Channel 1 is a rising edge trigger, and Channel 3 is a high level trigger, both of which have a Trigger Variable
Modbus Address	Actual Modbus address of the instruction

■ Diagnostic Information

When the LX controller serves as the Modbus RTU master station, after the ModbusRTU Slave Station Channel（ModbusRTU） is configured and compiled, the Diagnostic Information tab appears in the slave station configuration window.

For detailed information, please refer to the Modbus RTU Protocol Fault Diagnosis.

MODBUSSLAVE_RTU(1:MODBUSSLAVE_RTU)

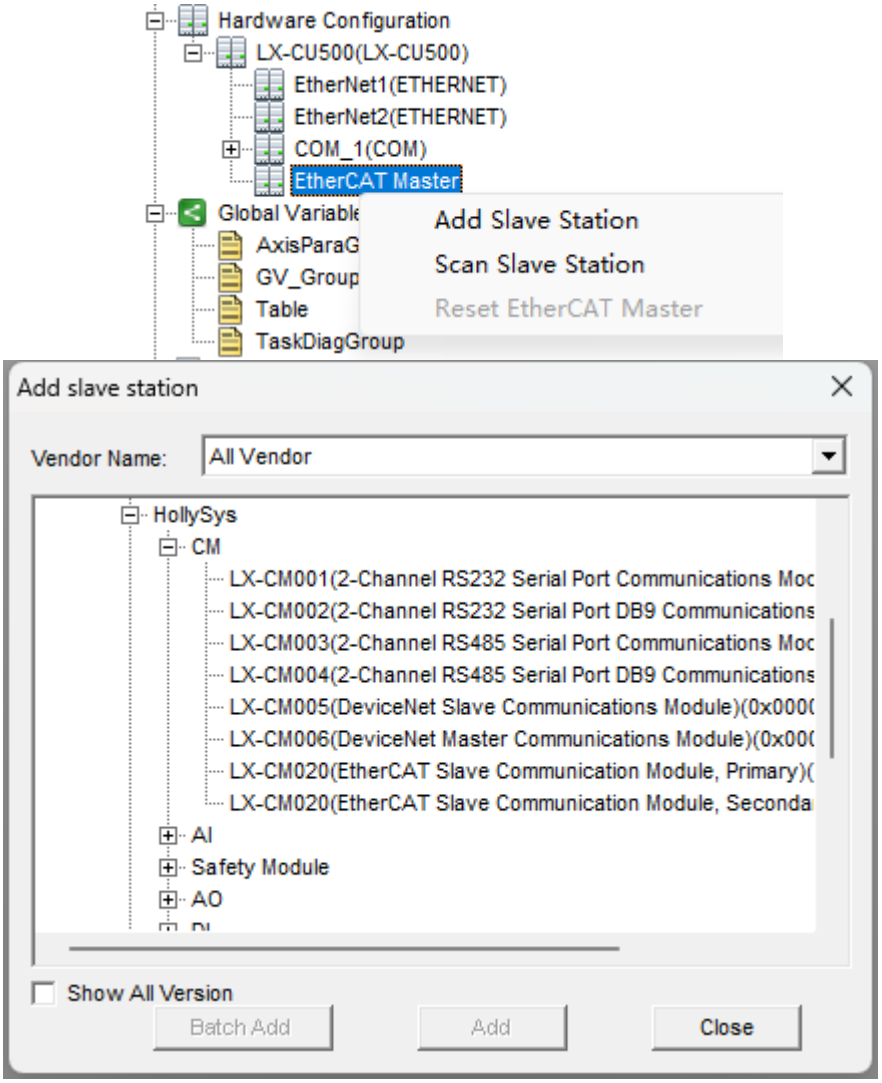
ModbusRTU slave station configuration | ModbusRTU slave station channel | ModbusRTU slave station I/O mapping | Diagnostic Information | Information

No.	Variable Name	Address	Variable Description	Variable Type	Online Value	Power Fail Safeguard
0001	AT_ModbusRTUOrder1_1_DiagState	%SW0	No.1 Order's State diagnosis	WORD	0	FALSE
0002	AT_ModbusRTUOrder1_2_DiagState	%SW2	No.2 Order's State diagnosis	WORD	0	FALSE

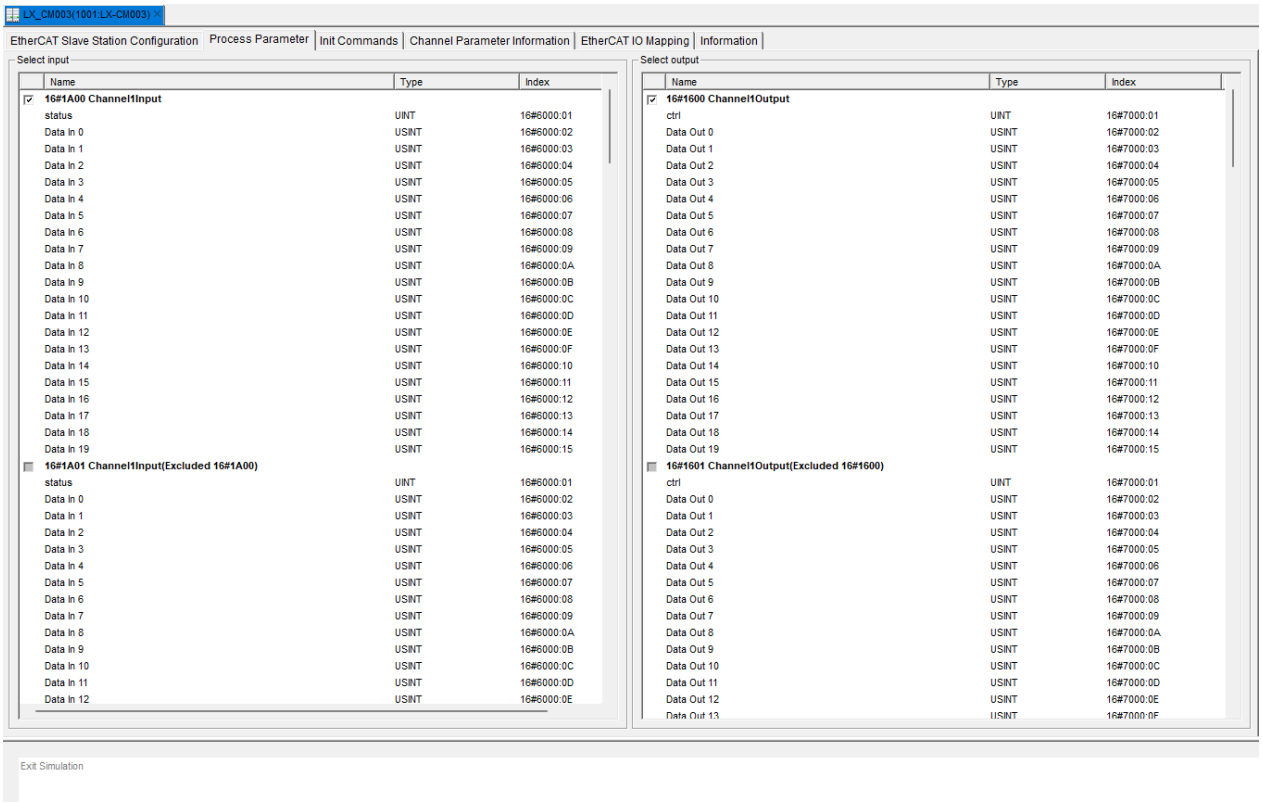
15.3.1.2 Configuration process using the serial port module LX-CM00X

- LX-CM00X module hardware configuration

  Right-click the EtherCAT Master node, select Add Slave Station, and the Add Slave Station dialog box is displayed, as shown in the figure, then select the required COM communication module, click the Add button, and the module is displayed under the EtherCAT Master node.



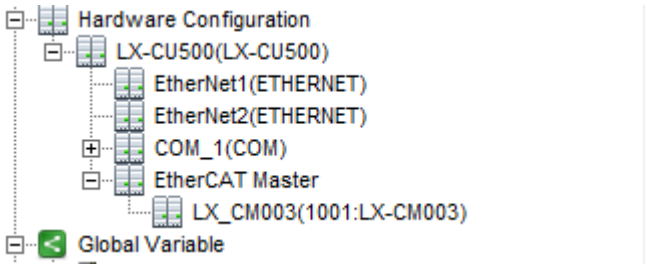
- Process parameters



- Note: The input and output byte options must be consistent. For example, if you select 20 bytes for input, you must also select 20 bytes for output. The same goes for 40/80 bytes.

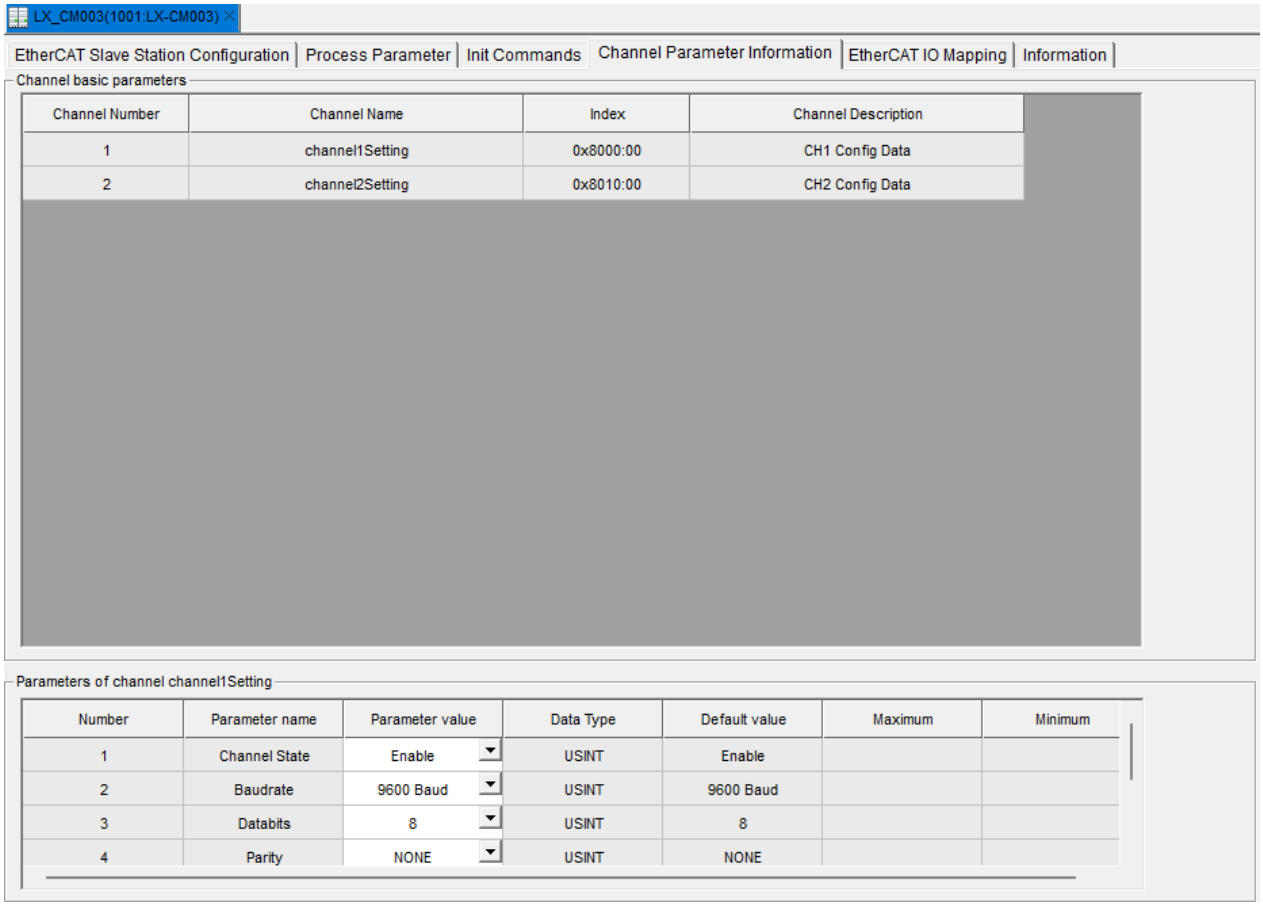
■ Configuring channel parameters

Double-click the LX_CM00X node or select the Open command in the right-click menu and the slave station configuration window is displayed, as shown in the figure.



Select the Channel Parameter Information tab to set the serial port parameters of each channel.

Select the channel number 1 or 2 to set the relevant parameters below.



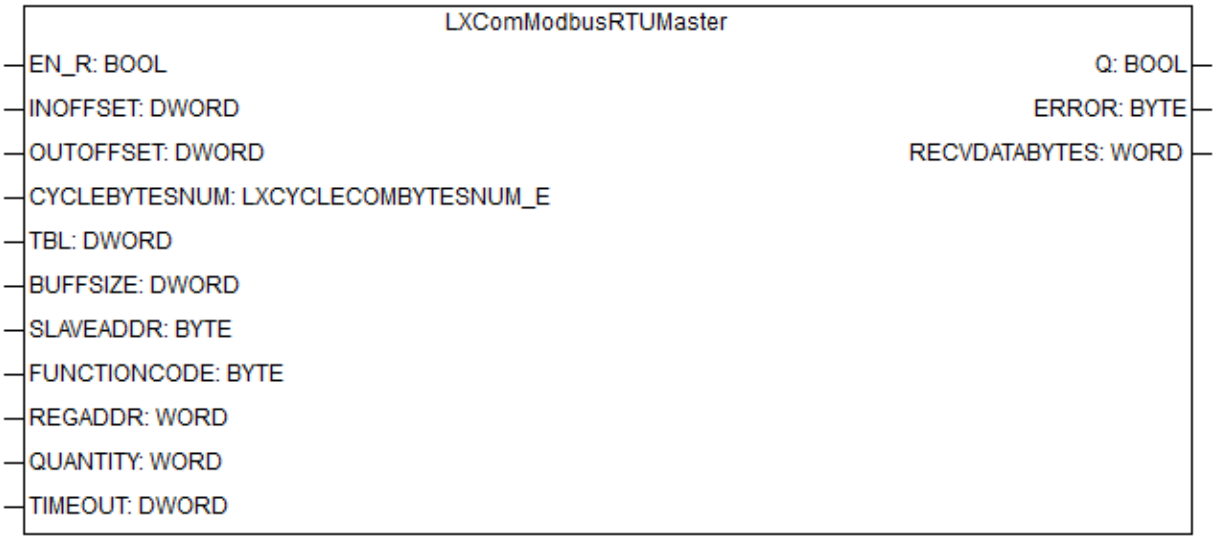
Parameter Name	Default Value	Description
Channel State	Enable	Enable: Enable serial port channel communication; Disable: Disable channel
Baudrate	9600	Communication baud rate (bps): 1200; 2400; 4800; 9600; 19.2 k; 38.4 k; 57.6 k; 115.2 k (optional)
Databits	8	Data bits
Parity	NONE	Data check mode: NONE - no check; EVEN - even check; ODD - odd check
Stopbits	1	Stop bits: 1; 2
Communication Type	RS485	It is related to the module interface type and cannot be changed.
SendContinuous State	Enable	Send continuous status: Enable; Disable When the data length to be sent by the CPU module to the external device is greater than the IO length allocated by the EtherCAT bus to the serial port module, the data cannot be sent within one EtherCAT cycle, that is, the same data frame needs to be split and sent. When this item is set to Enable, the serial port module will cache the current data until the complete data frame is received, and then send it to the external device through the serial port. When this item is set to Disable, the serial port module does not verify the integrity of the data frame and sends the current data immediately. For MODBUS protocols, this item must be Enable.
Frame Interval	0	After the serial port detects that the time interval between two frames of data is greater than the set value, it is considered that one frame has been received. MODBUS communication: 35; free protocol communication: 0

■ Communication functional block configuration

Add the MODBUS communication master station protocol functional block LXComModbusRTUMaster

or LXComMBMaster in the AutoThink task.

□ LXComModbusRTUMaster Description



LXComModbusRTUMaster Functional Block

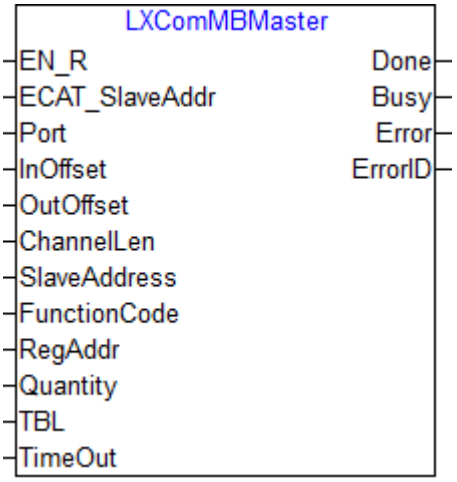
Parameter Description

Parameter	Statement	Data Type	Function Description	Description	Initial Value	Power Fail Safeguard
EN_R	Input	BOOL	Enabled, valid on rising edge	Each time an instruction action is executed, EN_R shall be set from FALSE to TRUE. When it is checked that Q changes from FALSE to TRUE, it means that the execution of this action is completed, and EN_R can be set to FALSE.	FALSE	FALSE
InOffset	Input	DWORD	Area I offset address of periodic input data	INOFFSET depends on the channel address in the EtherCAT IO mapping page in the hardware configuration. The channel address corresponding to the first status is the INOFFSET of the first channel, and the channel address corresponding to the second status is the INOFFSET of the second channel.	0	FALSE
OutOffset	Input	DWORD	Area Q offset address of periodic output data	OUTOFFSET depends on the channel address in the EtherCAT IO mapping page in the hardware configuration. The channel address corresponding to the first ctrl is the OUTOFFSET of the first channel, and the channel address corresponding to the second ctrl is the OUTOFFSET of the second channel.	0	FALSE
CycleBytesNum	Input	LXCYCLECOMBYTESNUM_E	Data size transmitted periodically	For CYCLEBYTESNUM, the enumeration variable of type LXCYCLECOMBYTESNUM_E shall be filled in, and the value depends on the process parameters in the hardware configuration. On the left side of the process parameter page, 16#1A00/16#1A01/16#1A02 respectively correspond to the data of 20/40/80 bytes input per cycle in channel 1, and	LXCycleComBytesNum_20	FALSE

				16#1A03/16#1A04/16#1A05 respectively correspond to the data of 20/40/80 bytes input per cycle in channel 2. On the right side, 16#1600/16#1601/16#1602 respectively correspond to the data of 20/40/80 bytes output per cycle in channel 1, and 16#1603/16#1604/16#1605 respectively correspond to the data of 20/40/80 bytes output per cycle in channel 2. When checked, the input and output of each channel must be consistent. For example, if 40 is selected for the input of channel 1, 40 must also be selected for the output, and if 20 is selected for the input of channel 2, 20 must also be selected for the output.		
TBL	Input	DWORD	First byte address of the master station to store data	It is the first byte address of area M for data storage, such as 200, which means that the storage address is a space starting from %MW200. If it is a read-type instruction, the data value read back is stored in this data area. If it is a write type instruction, the data value to be written is stored in this data area	0	FALSE
BuffSize	Input	DWORD	Byte size of the data memory stored in the master station	Byte size of the data memory to store data by the master station	0	FALSE
SlaveAddr	Input	BYTE	Slave station address	The address of MODBUS slave station is 1~247, and 0 is the broadcast address.	0	FALSE
Function Code	Input	BYTE	Instruction number	MODBUS instruction number, supporting instructions 01, 02, 03, 04, 05, 06, 15, and 16	0	FALSE
RegAddr	Input	WORD	Register start address	MODBUS register start address	0	FALSE
Quantity	Input	WORD	Number of registers	Number of MODBUS registers	0	FALSE
TimeOut	Input	DWORD	Slave station response timeout (ms)	Starting from the time when EN_R is enabled, if the correct slave station response frame is not received within the specified time, the receiving process will be aborted. Q is set to 1, and Error prompt times out.	0	FALSE
Q	Output	BOOL	Completion flag	1: Completed 0: Not completed	FALSE	FALSE
Error	Output	BYTE	Error	0: No error 2: Timeout 4: This module does not support function codes 10: Respond to unrequested responses 16: Incorrect slave station address 33: TBL or BuffSize out of range 34: BuffSize input 0 35: Timeout input 0 64: The slave station returns an invalid instruction 65: The slave station returns an invalid data address 66: The slave station returns an invalid data value 67: The slave station returns a slave station device fault 68: The slave station returns other error diagnosis	0	FALSE

				129: InOffset out of range 130: OutOffset out of range 131: CycleBytesNum is illegal 132: Error occurred in data acquisition		
RecvDataBytes	Output	WORD	Number of data bytes read	Number of data bytes actually read by the read operation	0	FALSE

- LXComMBMaster Description



LXComMBMaster Function Block

Parameter Description

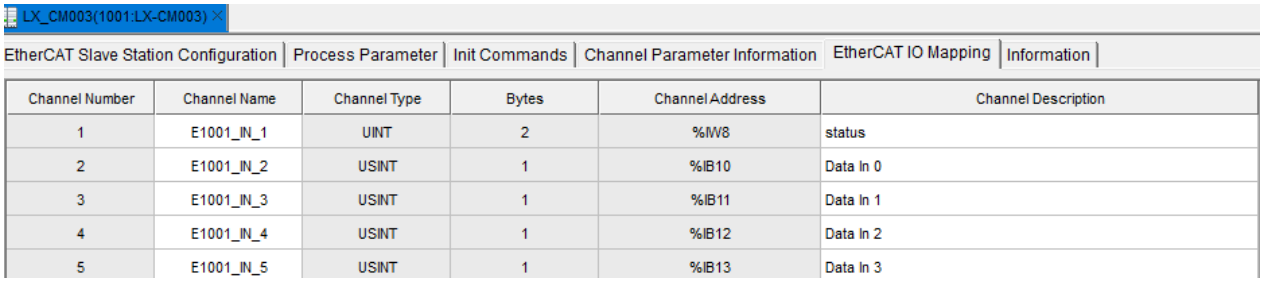
Parameter	Declaration	Data Type	Description	Default Value
EN_R	INPUT	BOOL	Rising edge enable.	FALSE
ECAT_SlaveAddr	INPUT	WORD	EtherCAT slave address of the serial port module for capture. Input range 1~65535.	0
Port	INPUT	BYTE	Serial port module channel number: 1: Channel 1 (channel1Setting) 2: Channel 2 (channel2Setting)	0
InOffset	INPUT	DWORD	Offset address in the EtherCAT cyclic input data I area.	0
OutOffset	INPUT	DWORD	Offset address in the EtherCAT cyclic output data Q area.	0
ChannelLen	INPUT	LXCHANNELLENNUM_E	Amount of data transmitted cyclically via EtherCAT.	LXChannelLenNum_20
SlaveAddress	INPUT	BYTE	Modbus RTU slave address.	0
FunctionCode	INPUT	BYTE	Modbus RTU slave function code.	0
RegAddr	INPUT	WORD	Starting address of the slave register.	0
Quantity	INPUT	WORD	Number of registers.	0
TBL	INPUT	DWORD	Offset address in the M area where the master stores data.	0
TimeOut	INPUT	WORD	Slave response timeout (ms).	0
Done	OUTPUT	BOOL	FALSE: Slave message processing not completed or not started TRUE: Slave message processing completed	FALSE

Busy	OUTPUT	BOOL	FALSE: Idle TRUE: Receiving or processing the slave's response message	FALSE
Error	OUTPUT	BOOL	FALSE: No error TRUE: Error occurred	FALSE
ErrorID	OUTPUT	WORD	Error code: 0: No error Other: Refer to Serial Port Capture Error Codes	0

- InOffset: Area I offset address of periodic input data

  InOffset depends on the channel address in the EtherCAT IO mapping page in the hardware configuration. The channel address corresponding to the first status is the InOffset of the first channel, and the channel address corresponding to the second status is the InOffset of the second channel.

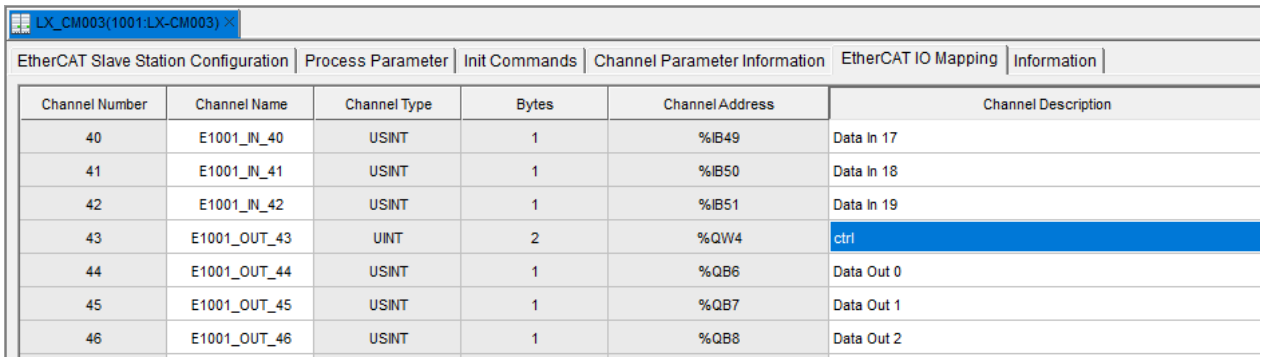
  For example, if the corresponding address of the first status is %IW8, fill in 8 for the InOffset pin.



- OutOffset: Area Q offset address of periodic output data

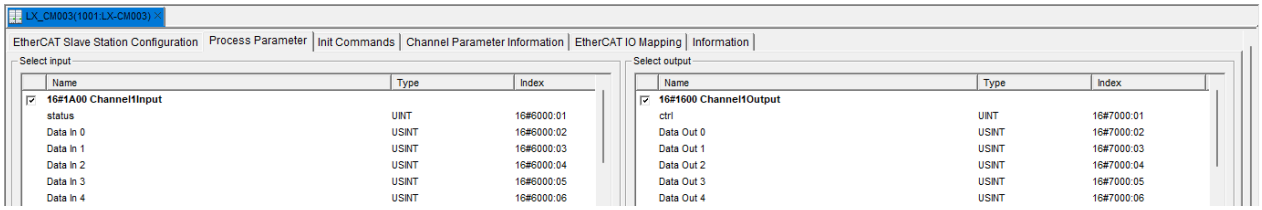
  OutOffset depends on the channel address in the EtherCAT IO mapping page in the hardware configuration. The channel address corresponding to the first ctrl is the OutOffset of the first channel, and the channel address corresponding to the second ctrl is the OutOffset of the second channel.

  For example, if the corresponding address of the first ctrl is %QW4, fill in 4 for the OutOffset pin.



- CycleBytesNum: Data type: LXCYCLECOMBYTESNUM_E

  This variable is related to the number of input and output bytes selected by the process parameters. For example, if 20BYTE is selected for input and output, the value of LXCYCLECOMBYTESNUM_E is 20 or LxcycleCOmBytesNum_20. Similarly, if 40BYTE is selected for input and output, the value of LXCYCLECOMBYTESNUM_E is 40 or LxcycleCOmBytesNum_40. If 40BYTE is selected for input and output, the value of LXCYCLECOMBYTESNUM_E is 80 or LxcycleCOmBytesNum_80.



- TBL: Pointing to the first byte address of area M for data storage

  For example, 200 means that the storage address is a space starting from %MW200. If it is a read type instruction, the data value read back is stored in this data area. If it is a write type instruction, the data value to be written is placed in this data area.

- BuffSize: length of master station data buffer area

  Since the maximum data size of each MODBUS communication frame is 250 bytes, the value can be fixed to be greater than or equal to 250.

  - SlaveAddr: Slave station address
  - FunctionCode: Function code
  - RegAddr: Starting address of the read and write slave station
  - Quantity: Data length of the read and write slave station
  - TimeOut: Slave station response timeout

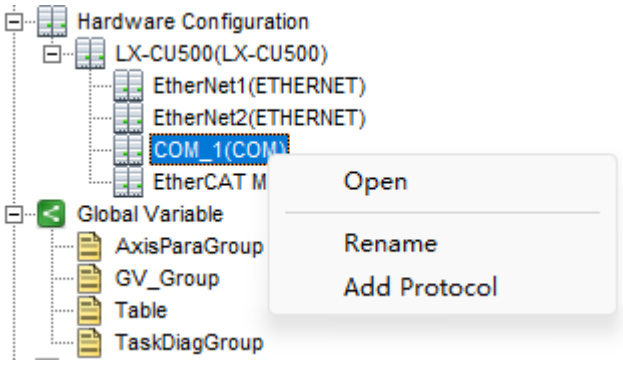
15.3.2 Configuring Modbus RTU Slave Station

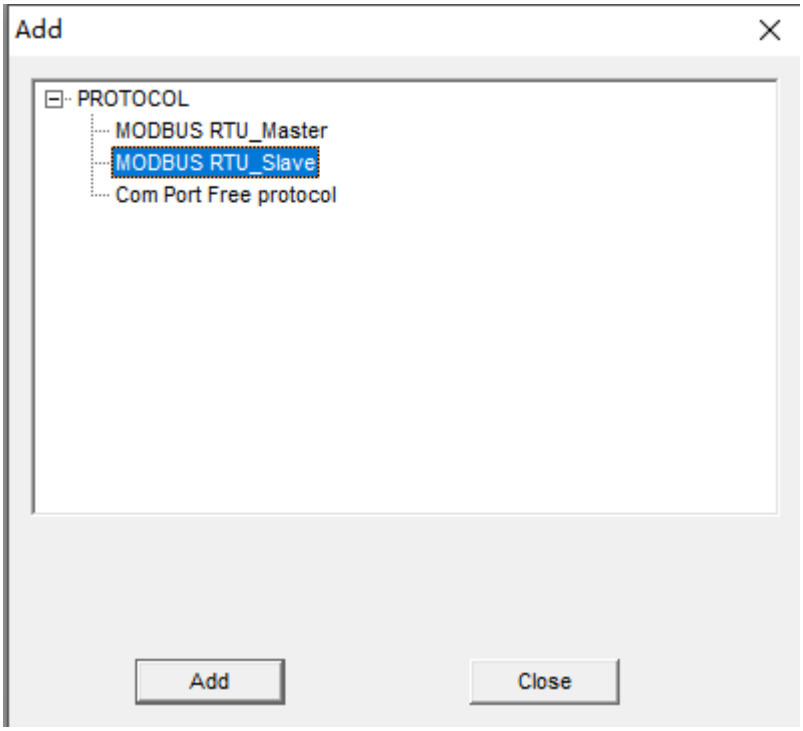
15.3.2.1 Configuration process using the CPU integrated RS485 interface

When the controller serves as the MODBUS RTU slave station, its configuration process is as follows:

- Adding protocol

  Select the Add Protocol command in the right-click menu of the COM_1 node and the Add dialog box is displayed.

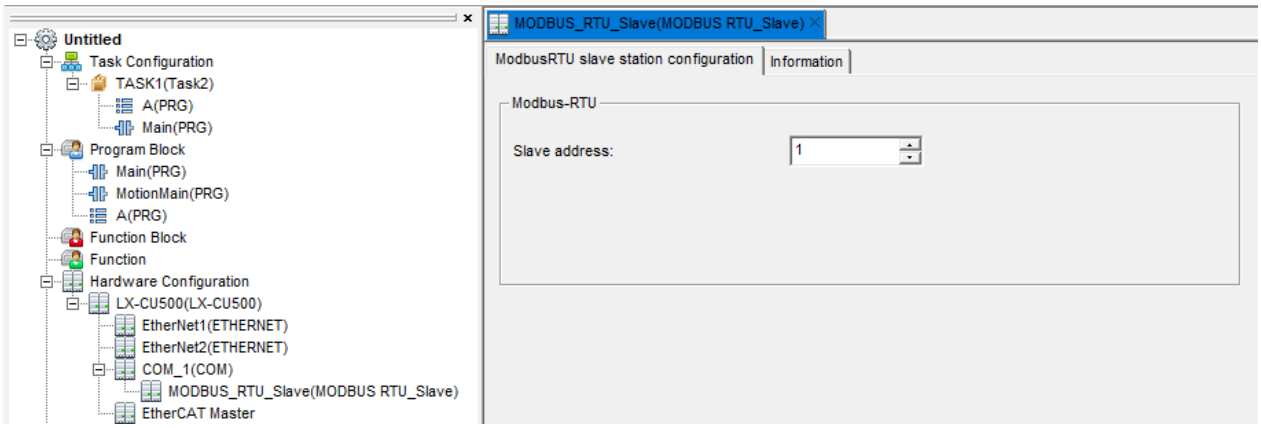




Select MODBUS RTU_SLAVE slave station protocol and click the Add button.

- Configuring slave station protocol

Double-click the MODBUS_RTU_Slave node or select the Open command in the right-click menu and the slave station configuration window is displayed, as shown in the figure.



The slave parameters are described as follows:

Parameter	Value	Default Value	Description
Slave address	1~247	1	Modbus RTU Slave Device ID

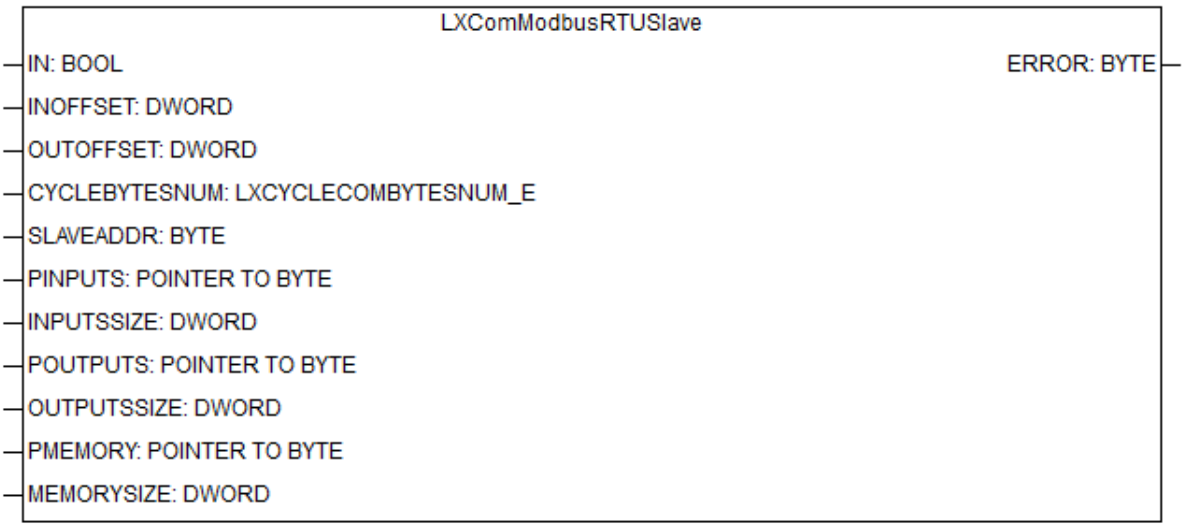
15.3.2.2 Configuration process using the serial port module LX-CM00X

- For the module configuration process, please refer to the Hardware Configuration.

■ Communication functional block configuration

Add the MODBUS communication slave station protocol functional block LXComModbusRTUSlave or LXComMBSlave in the AutoThink task.

□ LXComModbusRTUSlave Description



LXComModbusRTUSlave Functional Block

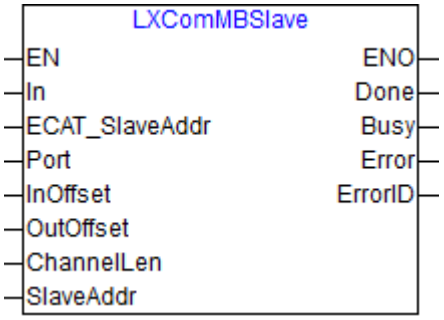
Parameter Description

Parameter	Statement	Data Type	Function Description	Description of Parameter Values	Default Value	Power Fail Safeguard
IN	Input	BOOL	Enable, valid at high level	Communication takes effect when the level is high	FALSE	FALSE
InOffset	Input	DWORD	Area I offset address of periodic input data	INOFFSET depends on the channel address in the EtherCAT IO mapping page in the hardware configuration. The channel address corresponding to the first status is the INOFFSET of the first channel, and the channel address corresponding to the second status is the INOFFSET of the second channel.	0	FALSE
OutOffset	Input	DWORD	Area Q offset address of periodic output data	OUTOFFSET depends on the channel address in the EtherCAT IO mapping page in the hardware configuration. The channel address corresponding to the first ctrl is the OUTOFFSET of the first channel, and the channel address corresponding to the second ctrl is the OUTOFFSET of the second channel.	0	FALSE
CYCLEBYTESNUM	Input	LXCYCLECOMBYTESNUM_E	Data size transmitted periodically	For CYCLEBYTESNUM, the enumeration variable of type LXCYCLECOMBYTESNUM_E shall be filled in, and this value depends on the process parameters in the hardware configuration. On the left side of the process parameter page, 16#1A00/16#1A01/ 16#1A02 respectively correspond to the data of 20/40/80 bytes	LXCycleComBytesNum_20	FALSE

				input per cycle in channel 1, and 16#1A03/ 16#1A04/16 #1A05 respectively correspond to the data of 20/40/80 bytes input per cycle in channel 2. On the right side, 16#1600/ 16#1601/16#1602 respectively correspond to the data of 20/40/80 bytes output per cycle in channel 1, and 16#1603/ 16#1604/ 16#1605 respectively correspond to the data of 20/40/80 bytes output per cycle in channel 2. When checked, the input and output of each channel must be consistent. For example, if 40 is selected for the input of channel 1, 40 must also be selected for the output, and if 20 is selected for the input of channel 2, 20 must also be selected for the output.		
SlaveAddr	Input	BYTE	Slave station address	MODBUS slave station address, 1~247	0	FALSE
pInputs	Input	POINTER TO BYTE	Pointer address of area I (if not filled in, the address of area I will be used by default)	If the pointer is filled in, the area pointed to by the pointer is considered to be area I when the communication is performed. If it is not filled in or filled with 0, the default address of area I is used for communication.	0	FALSE
InputsSize	Input	DWORD	Length of the filled-in address of area I	Byte size of area I filled in	0	FALSE
pOutputs	Input	POINTER TO BYTE	Pointer address of area Q (if not filled in, the address of area Q will be used by default)	If the pointer is filled in, the area pointed to by the pointer is considered to be area Q when the communication is executed. If it is not filled in or filled with 0, the default address of area Q is used for communication.	0	FALSE
OutputsSize	Input	DWORD	Length of the filled-in address of area Q	Byte size of area Q filled in	0	FALSE
pMemory	Input	POINTER TO BYTE	Pointer address of area M (if not filled in, the address of area M will be used by default)	If the pointer is filled in, the area pointed to by the pointer is considered to be area M when the communication is executed. If it is not filled in or filled with 0, the default address of area M is used for communication.	0	FALSE
MemorySize	Input	DWORD	Length of the filled-in address of area M	Byte size of area M filled in	0	FALSE
Error	Output	BYTE	Error	0: No error 1: Received a data packet that failed the CRC check 2: Received a data packet from a non-local address 16: Incorrect slave station address	0	FALSE

				36: pInputs and InputsSize are enabled but the address area is invalid 37: pOutputs and OutputsSize are enabled but the address area is invalid 38: pMemory and MemorySize are enabled but the address area is invalid 129: InOffset out of range 130: OutOffset out of range 131: CycleBytesNum is illegal 132: Error occurred in data acquisition		

□ LXComMBSlave Description



LXComMBSlave Function Block

Parameter Description

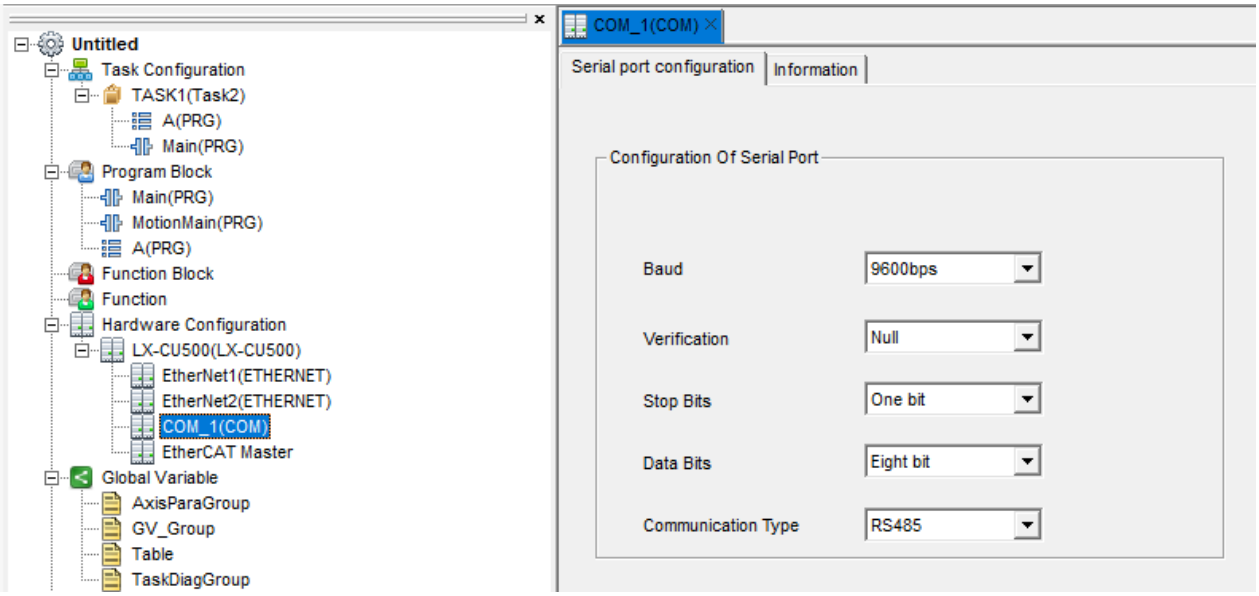
Parameter	Declaration	Data Type	Description	Default Value
In	INPUT	BOOL	Enable on high level.	FALSE
ECAT_SlaveAddr	INPUT	WORD	EtherCAT slave address of the serial port module for capture. Input range 1~65535.	0
Port	INPUT	BYTE	Serial port module channel number: 1: Channel 1 (channel1Setting) 2: Channel 2 (channel2Setting)	0
InOffset	INPUT	DWORD	Offset address in the EtherCAT cyclic input data I area.	0
OutOffset	INPUT	DWORD	Offset address in the EtherCAT cyclic output data Q area.	0
ChannelLen	INPUT	LXCHANNELLENNUM_E	Amount of data transmitted cyclically via EtherCAT.	LXChannelLenNum_20
SlaveAddress	INPUT	BYTE	Modbus RTU slave address.	0
Done	OUTPUT	BOOL	FALSE: Slave message processing not completed or not started TRUE: Message processing completed	FALSE
Busy	OUTPUT	BOOL	FALSE: Idle TRUE: Receiving or processing message	FALSE
Error	OUTPUT	BOOL	FALSE: No error TRUE: Error occurred	FALSE
ErrorID	OUTPUT	WORD	Error code: 0: No error Other: Refer to Serial Port Capture Error Codes	0

15.3.3 Configuring COM Port Free Protocol

15.3.3.1 Configuration process using the CPU module integrated COM port

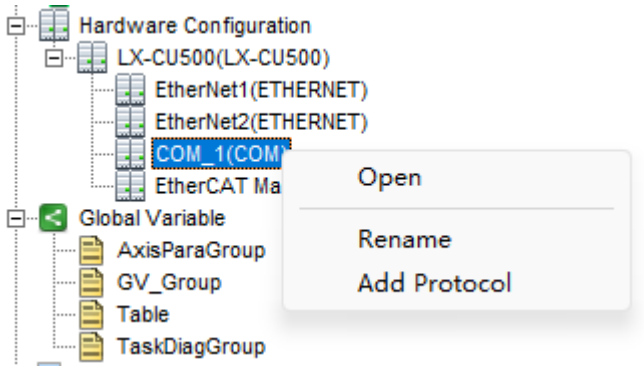
■ COM parameter configuration

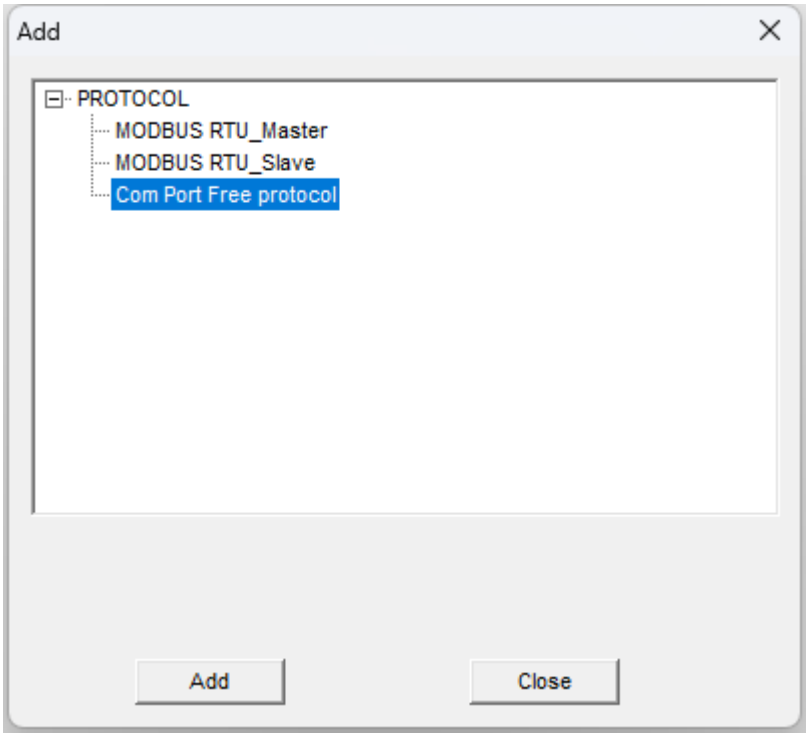
Double-click on the COM_1 node, and the serial port configuration window is displayed, then modify the serial port parameters.



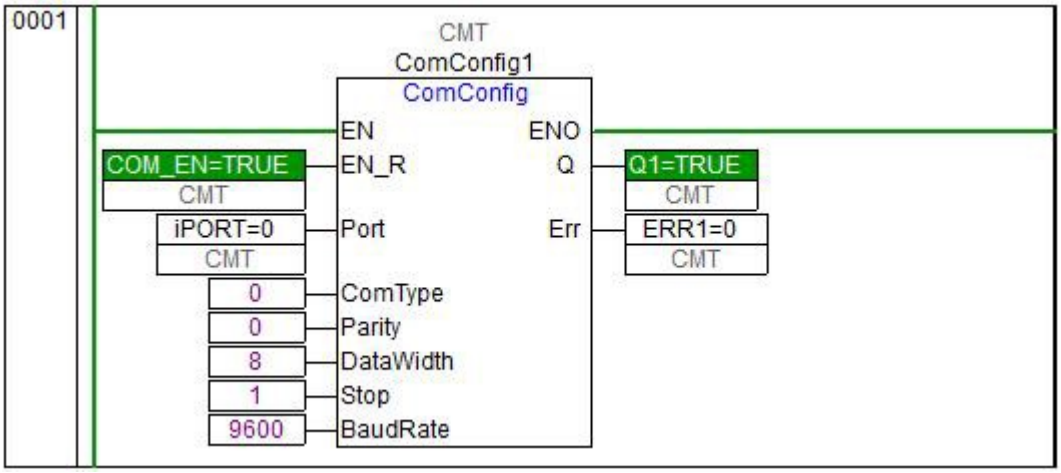
■ Protocol configuration

Right-click the COM_1 node, select Add Protocol, and the Add Protocol dialog box is displayed, as shown in the figure, then select the required COM communication module, click the Add button, and the module is displayed under the node.





- For the COM port that comes with the CPU itself, the serial port parameters need to be configured using the communication instruction ComConfig, as shown in the figure below:

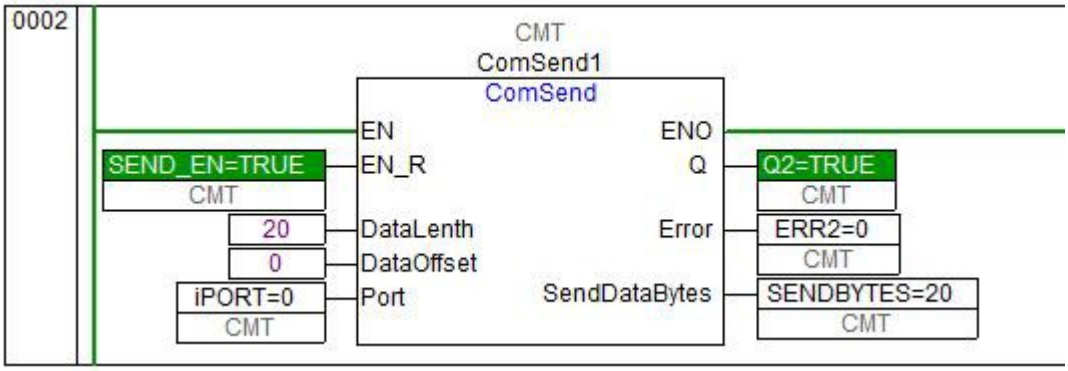


The parameters are described as follows:

Parameter	Data Item	Data Type	Description
Input Parameters	EN_R	BOOL	Trigger valid on rising edge
	Port	USINT	0: Interface of CPU body 1: CPU body terminal communication port (not supported at 14 points) 2: Extended function board communication port (not supported at 14 points) 3: The first communication expansion module communication port 4~N: The second to Nth communication expansion module communication ports
	ComType	USINT	Communication type 0: RS485 1: RS422 2: RS232

	Parity	USINT	Check mode 0: No check; 1: Even parity check; 2: Odd parity check
	DataWidth	USINT	Data bit width: 8: 8 bits; 7: 7 bits
	Stop	USINT	Stop bit 1: Stop bit 1; 2: Stop bit 2
	BaudRate	UDINT	Baud rate (bps) 9600 19200 38400 57600 115200 2400000
Output Parameters	Q	BOOL	Terminal Q is TRUE, which indicates that the instruction is executed.
	Err	USINT	Error code 0: No error 1: Port communication port error 2: ComType error 3: BaudRate error 4: DataWidth error 5: Parity error 6: Stop error 7: Internal error inside the controller

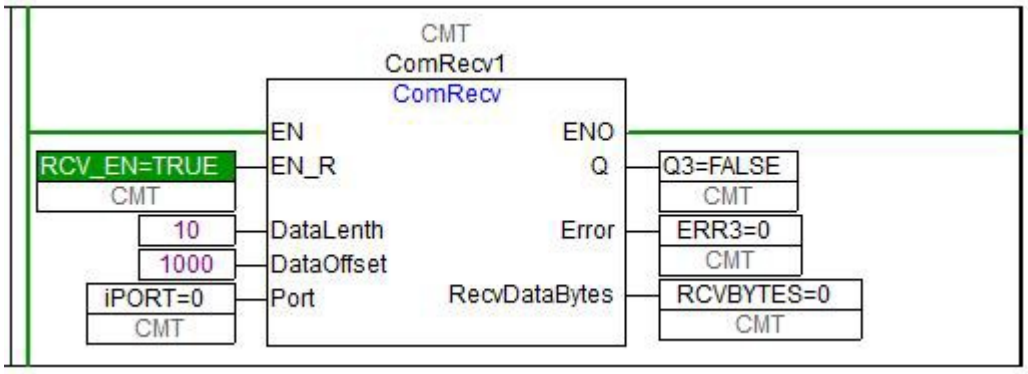
■ Use the ComSend instruction to send data, as shown in the figure below:



The parameters are described as follows:

Parameter	Data Item	Data Type	Description
Input Parameters	EN_R	BOOL	Trigger valid on rising edge
	DataLenth	UDINT	Length of data to send (number of bytes)
	DataOffset	UDINT	Offset address of area M where the data to be sent is located For example, 0——%MB0
	Port	USINT	0: Interface of CPU body
Output Parameters	Q	BOOL	Terminal Q is TRUE, which indicates that the instruction is executed.
	Error	USINT	Error code 0: No error 1: The input parameter configuration is incorrect 2: This port is not configured as a free port protocol
	SendDataBytes	UDINT	Actual number of bytes sent

■ Use the ComRecv instruction to receive data, as shown in the figure below:



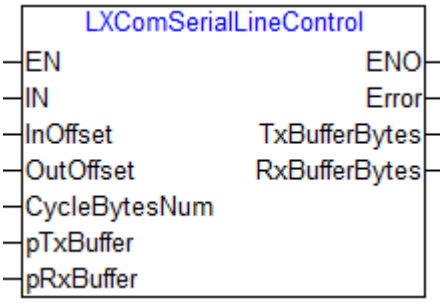
The parameters are described as follows:

Parameter	Data Item	Data Type	Description
Input Parameters	EN_R	BOOL	Trigger valid on rising edge
	DataLenth	UDINT	Length of data to receive (number of bytes)
	DataOffset	UDINT	Offset address of area M where the received data is stored For example, 1000——%MB1000
	Port	USINT	0: Interface of CPU body
Output Parameters	Q	BOOL	After the instruction is executed, the output is TRUE and only remains for one task cycle.
	Error	USINT	Error code 0: No error 1: The input parameter configuration is incorrect 2: This port is not configured as a free port protocol
	SendDataBytes	UDINT	Actual number of bytes received

15.3.3.2 Configuration process using the serial port module LX-CM00X

- For the module configuration process, please refer to the Hardware Configuration.
- Communication functional block configuration

(1) Add the free communication control functional block to the AutoThink task:



LXComSerialLineControl Functional Block

Parameter Description

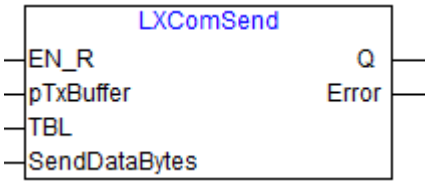
Parameter	Statement	Data Type	Function	Description	Initial	Power Fail

			Description		Value	Safeguard
IN	Input	BOOL	Enable, valid at high level	Communication takes effect when the level is high	FALSE	FALSE
InOffset	Input	DWORD	Area I offset address of periodic input data	INOFFSET depends on the channel address in the EtherCAT IO mapping page in the hardware configuration. The channel address corresponding to the first status is the INOFFSET of the first channel, and the channel address corresponding to the second status is the INOFFSET of the second channel.	0	FALSE
OutOffset	Input	DWORD	Area Q offset address of periodic output data	OUTOFFSET depends on the channel address in the EtherCAT IO mapping page in the hardware configuration. The channel address corresponding to the first ctrl is the OUTOFFSET of the first channel, and the channel address corresponding to the second ctrl is the OUTOFFSET of the second channel.	0	FALSE
CycleBytesNum	Input	LXCYCLECOMBYTESNUM _E	Data size transmitted periodically	For CYCLEBYTESNUM, the enumeration variable of type LXCYCLECOMBYTESNUM_E shall be filled in, and this value depends on the process parameters in the hardware configuration. On the left side of the process parameter page, 16#1A00/ 16#1A01/ 16#1A02 respectively correspond to the data of 20/40/80 bytes input per cycle in channel 1, and 16#1A03/ 16#1A04/ 16#1A05 respectively correspond to the data of 20/40/80 bytes input per cycle in channel 2. On the right side, 16#1600/ 16#1601/ 16#1602 respectively correspond to the data of 20/40/80 bytes output per cycle in channel 1, and 16#1603/ 16#1604/ 16#1605 respectively correspond to the data of 20/40/80 bytes output per cycle in channel 2. When checked, the input and output of each channel must be consistent. For example, if 40 is selected for the input of channel 1, 40 must also be selected for the output, and if 20 is selected for the input of channel 2, 20 must also be selected for the output.	LXCycleComBytesNum_20	FALSE
pTxBuffer	Input	POINTER TO LXSERIALCOMBUF_T	Tx communication buffer pointer	Pointer to Tx buffer	0	FALSE
pRxBuffer	Input	POINTER TO LXSERIALCOMBUF_	Rx communication buffer pointer	Pointer to Rx buffer	0	FALSE
Error	Output	BYTE	Error	0: No error 1: Incorrect buffer pointer input 129: InOffset out of range 130: OutOffset out of range 131: CycleBytesNum is illegal 132: Incorrect cycle data acquisition	0	FALSE
TXBUFFERBYTES	Output	DWORD	Existing data length of Tx communication buffer	Existing data length of Tx communication buffer	0	FALSE

RXBUFFERBYTES	Output	DWORD	Existing data length of Rx communication buffer	Existing data length of Rx communication buffer	0	FALSE

(2) Add the free communication Tx data functional block LXComSend Functional or LXComFreePortSend to the AutoThink task.

□ LXComSend Description

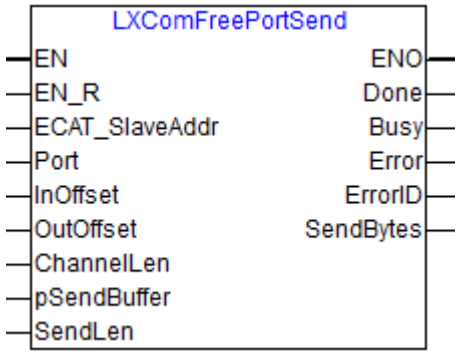


LXComSend Functional Block

Parameter Description

Parameter	Statement	Data Type	Function Description	Description of Parameter Values	Initial Value	Power Fail Safeguard
EN_R	Input	BOOL	Enabled, valid on rising edge	Each time a Tx action is executed, EN_R shall be set from FALSE to TRUE. When it is checked that Q changes from FALSE to TRUE, it means that the execution of this transmission is completed, and EN_R can be set to FALSE.	FALSE	FALSE
pTxBuffer	Input	POINTER TO LXSERIALCOMBUF_T	Tx communication buffer pointer	Pointer to Tx buffer	0	FALSE
TBL	Input	DWORD	Tx data buffer address	Offset of area M where the data address is to be sent, such as 200, which means that the address to send data is a space starting from %MB200.	0	FALSE
SendDataBytes	Input	DWORD	Tx data length	Length of data to be sent	0	FALSE
Q	Output	BOOL	Completion flag	1: Completed 0: Not completed	FALSE	FALSE
Error	Output	BYTE	Error	0: No error 1: Incorrect buffer pointer input 2: TBL or BuffSize out of range 3: SendDataBytes exceeds 1000	0	FALSE

□ LXComSend Functional Description



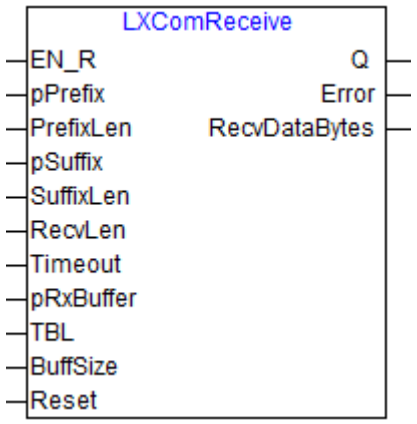
LXComFreePortSend Function Block

Parameter Description

Parameter	Declaration	Data Type	Description	Default Value
EN_R	INPUT	BOOL	Rising edge enable.	FALSE
ECAT_SlaveAddr	INPUT	WORD	EtherCAT slave address of the serial port module for capture. Input range 1~65535.	0
Port	INPUT	BYTE	Serial port module channel number: 1: Channel 1 (channel1Setting) 2: Channel 2 (channel2Setting)	0
InOffset	INPUT	DWORD	Offset address in the EtherCAT cyclic input data I area.	0
OutOffset	INPUT	DWORD	Offset address in the EtherCAT cyclic output data Q area.	0
ChannelLen	INPUT	LXCHANNELLENNUM_E	Amount of data transmitted cyclically via EtherCAT.	LXChannelLenNum_20
pSendBuffer	INPUT	POINTER TO BYTE	Send data storage buffer.	0
SendLen	INPUT	WORD	Send data length.	
Done	OUTPUT	BOOL	Send complete flag.	FALSE
Busy	OUTPUT	BOOL	Sending flag.	FALSE
Error	OUTPUT	BOOL	FALSE: No error TRUE: Error occurred	FALSE
ErrorID	OUTPUT	WORD	Error code: 0: No error Other: Refer to Serial Port Capture Error Codes	0
SendBytes	OUTPUT	WORD	Actual length of data sent.	

(3) Add the free communication Rx data functional block LXComReceive or LXComFreePortReceive to the AutoThink task.

□ LXComReceive Functional Description



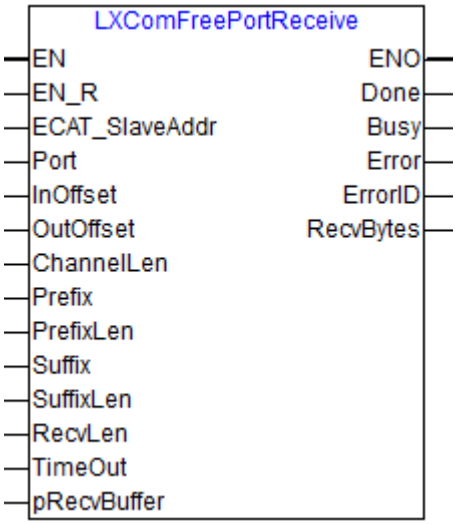
LXComReceive Functional Block

Parameter Description

Parameter	Statement	Data Type	Function Description	Description	Initial Value	Power Fail Safeguard
EN_R	Input	BOOL	Enabled, valid on rising edge	Each time an Rx action is executed, EN_R shall be set from FALSE to TRUE. When it is checked that Q changes from FALSE to TRUE, it means that the execution of this reception is completed, and EN_R can be set to FALSE.	FALSE	FALSE
pPrefix	Input	POINTER TO BYTE	Prefix pointer	If a variable pointer is passed to an input variable prefix, the characters of the received data must be the same as this prefix, while other characters are discarded. After the prefix is used, if no suffix is specified, it will be received with the incoming RecvLen data length. If no RecvLen is passed in, it will be received with the incoming Timeout. If no Timeout is passed in, it will be considered an invalid Rx mode.	0	FALSE
PrefixLen	Input	BYTE	Prefix length	Number of data bytes in the prefix	0	FALSE
pSuffix	Input	POINTER TO BYTE	Suffix pointer	If a variable pointer is passed to the input variable suffix, the input data is read until the end of the received data coincides with the suffix.	0	FALSE
SuffixLen	Input	BYTE	Suffix length	Number of data bytes in the suffix	0	FALSE
RecvLen	Input	DWORD	Receiving length	If no prefix or suffix is specified, the data of a RecvLen length is received starting from the first data.	0	FALSE
Timeout	Input	DWORD	Receiving timeout	If the prefix and suffix and the Rx data length are not specified, the data will be received within the timeout period. If there is no new data after the timeout period after the completion of one byte reception, the reception is considered completed.	0	FALSE
pRxBuffer	Input	POINTER TO LXSERIALCOMBUF_T	Rx communication buffer pointer	Pointer to Rx buffer	0	FALSE
TBL	Input	DWORD	Rx data buffer address	Offset of area M where the data address is to be stored, such as 200, which means that the storage address is a space starting from %MB200.	0	FALSE

BuffSize	Input	DWORD	Rx data buffer length	Byte size of memory area to store data	0	FALSE
Reset	Input	BYTE	Reset	Reset signal to restore functional block status	0	FALSE
Q	Output	BOOL	Completion flag	1: Completed 0: Not completed	FALSE	FALSE
Error	Output	BYTE	Error	0: No error 1: Incorrect buffer pointer input 2: No valid Rx mode 3: Incorrect input of prefix or suffix pointer 4: Sum of prefix and suffix exceeds 1000 5: RecvLen exceeds 1000 6: TBL or BuffSize out of range 7: Actual data length is greater than BuffSize	0	FALSE
RecvData Bytes	Output	WORD	Received data length	Actual length of data received after the reception is completed	0	FALSE

□ LXComFreePortReceive Functional Description



LXComFreePortReceive Function Block

Parameter Description

Parameter	Declaration	Data Type	Description	Default Value
EN_R	INPUT	BOOL	Enable on rising edge.	FALSE
ECAT_SlaveAddr	INPUT	WORD	EtherCAT slave address of the serial port module for capture. Input range 1~65535.	0
Port	INPUT	BYTE	Serial port module channel number: 1: Channel 1 (channel1Setting) 2: Channel 2 (channel2Setting)	0
InOffset	INPUT	DWORD	Offset address in the EtherCAT cyclic input data I area.	0
OutOffset	INPUT	DWORD	Offset address in the EtherCAT cyclic output data Q area.	0
ChannelLen	INPUT	LXCHANNELLENNUM_E	Amount of data transmitted cyclically via EtherCAT.	LXChannelLenNum_20
Prefix	INPUT	POINTER TO BYTE	Pointer to prefix..	0

PrefixLen	INPUT	BYTE	Prefix length in bytes.	0
Suffix	INPUT	POINTER TO BYTE	Pointer to suffix..	0
SuffixLen	INPUT	BYTE	Suffix length in bytes.	0
RecvLen	INPUT	DWORD	Specifies the length of data to receive.	0
TimeOut	INPUT	DWORD	Specifies the timeout period (ms).	0
pRecvBuffer	INPUT	POINTER TO BYTE	Pointer to the receive data buffer.	0
Done	OUTPUT	BOOL	FALSE: Slave message processing not completed or not started TRUE: Message processing completed	FALSE
Busy	OUTPUT	BOOL	FALSE: Idle TRUE: Receiving or processing message	FALSE
Error	OUTPUT	BOOL	FALSE: No error TRUE: Error occurred	FALSE
ErrorID	OUTPUT	WORD	Error code: 0: No error Other: Refer to Serial Port Capture Error Codes	0
RecvBytes	OUTPUT	WORD	Actual length of data received.	0

15.4 Communication Example of Serial Port

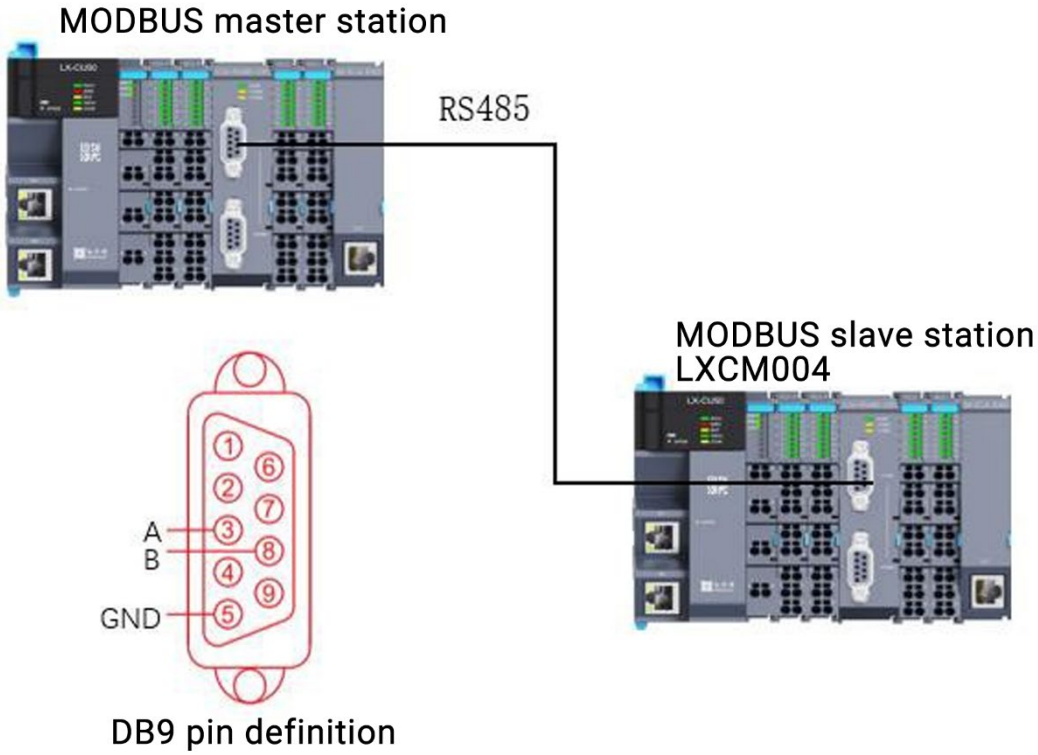
15.4.1 Example Function

This example uses two LX-CM004 (one as the master station and one as the slave station) to demonstrate the MODBUS RTU communication function.

15.4.2 System Composition

- MODBUS RTU master station: communication module LX-CM004
- MODBUS RTU slave station: communication module LX-CM004

15.4.3 Example Topology



15.4.4 Programming

Create two projects (one master station project and one slave station project), and program and configure the two projects respectively.

15.4.4.1 Modbus RTU slave station communication

- The parameters of channel 1 of module LX-CM004 are configured as follows:

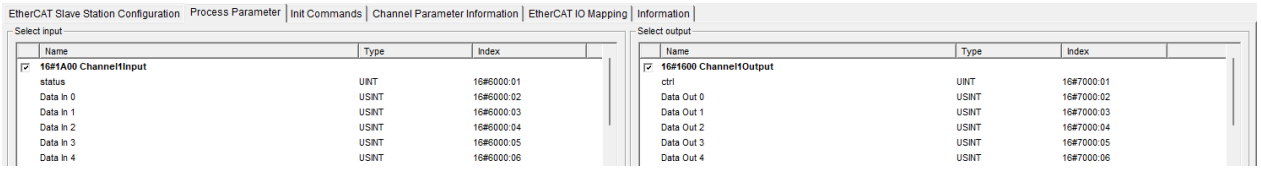
EtherCAT Slave Station Configuration | Process Parameter | Init Commands | Channel Parameter Information | EtherCAT IO Mapping | Information

Channel basic parameters

Channel Number	Channel Name	Index	Channel Description
1	channel1Setting	0x8000:00	CH1 Config Data
2	channel2Setting	0x8010:00	CH2 Config Data

Parameters of channel channel1Setting

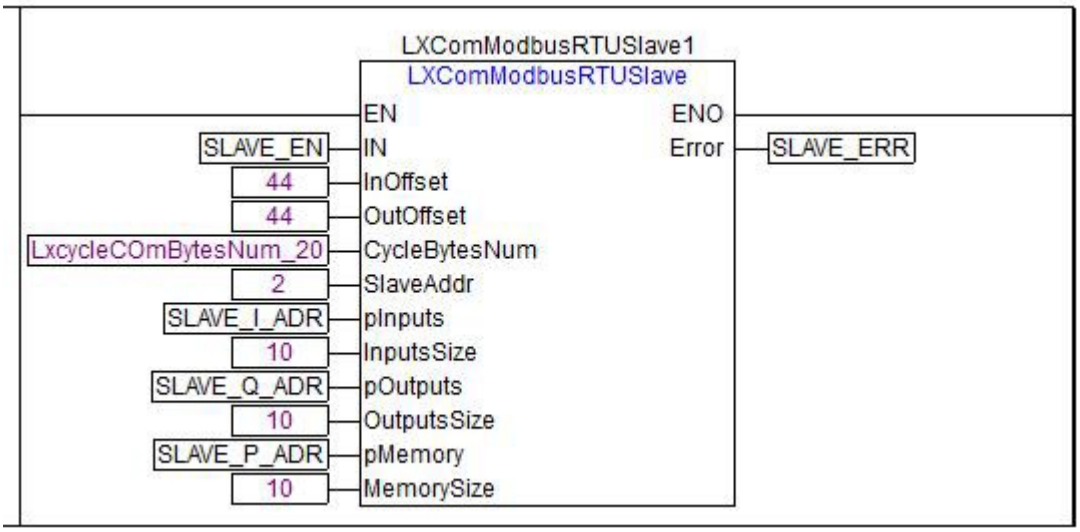
Number	Parameter name	Parameter value	Data Type	Default value	Maximum	Minimum
1	Channel State	Enable	USINT	Enable		
2	Baudrate	9600 Baud	USINT	9600 Baud		
3	Databits	8	USINT	8		
4	Parity	NONE	USINT	NONE		



EtherCAT Slave Station Configuration | Process Parameter | Init Commands | Channel Parameter Information | EtherCAT IO Mapping | Information

Channel Number	Channel Name	Channel Type	Bytes	Channel Address	Channel Description
1	E1001_IN_1	UINT	2	%IW0	status
2	E1001_IN_2	USINT	1	%IB2	Data In 0
3	E1001_IN_3	USINT	1	%IB3	Data In 1
41	E1001_IN_41	USINT	1	%IB42	Data In 18
42	E1001_IN_42	USINT	1	%IB43	Data In 19
43	E1001_OUT_43	UINT	2	%QW0	ctrl
44	E1001_OUT_44	USINT	1	%QB2	Data Out 0
45	E1001_OUT_45	USINT	1	%QB3	Data Out 1

- Slave station program:



- Input parameter description
  - IN: Slave station program enable, valid at high level;
  - InOffset: Corresponding to the Status address of channel 1 in the IO mapping table; 44——%IW44;
  - OutOffset: Corresponding to the Ctrl address of channel 1 in the IO mapping table; 44 - %QW44;
  - CycleBytesNum: Corresponding to the number of IOs selected in the process parameter table;
  - SlaveAddr: Slave station address of channel 1 corresponding to the InOffset and OutOffset addresses of the slave station module;
  - pInputs: Pointer address mapping area I;
  - InputsSize: Data length of area I (Words);
  - pOutputs: Pointer address mapping area Q;
  - OutputsSize: Data length of area Q (Words);
  - pMemory: Pointer address mapping area M;
  - MemorySize: Data length of area M (Words);
- Output parameter description
  - Error: Slave station communication fault code.

15.4.4.2 Modbus RTU master station communication program

- The parameters of channel 1 of module LX-CM004 are configured as follows:

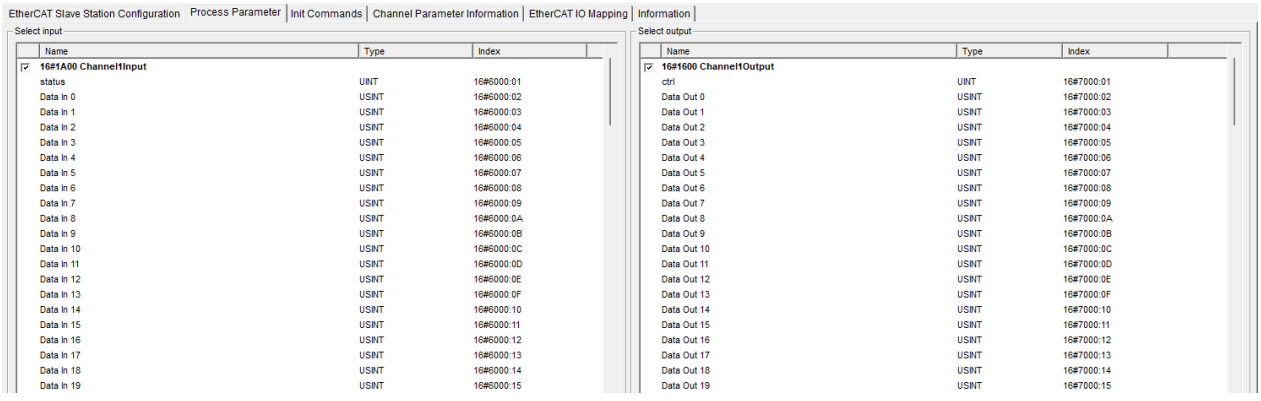
EtherCAT Slave Station Configuration | Process Parameter | Init Commands | Channel Parameter Information | EtherCAT IO Mapping | Information

Channel basic parameters

Channel Number	Channel Name	Index	Channel Description
1	channel1Setting	0x8000:00	CH1 Config Data
2	channel2Setting	0x8010:00	CH2 Config Data

Parameters of channel channel1Setting

Number	Parameter name	Parameter value	Data Type	Default value	Maximum	Minimum
1	Channel State	Enable	USINT	Enable		
2	Baudrate	9600 Baud	USINT	9600 Baud		
3	Databits	8	USINT	8		
4	Parity	NONE	USINT	NONE		



LX_CM004(1001:LX-CM004)

EtherCAT Slave Station Configuration | Process Parameter | Init Commands | Channel Parameter Information | EtherCAT IO Mapping | Information

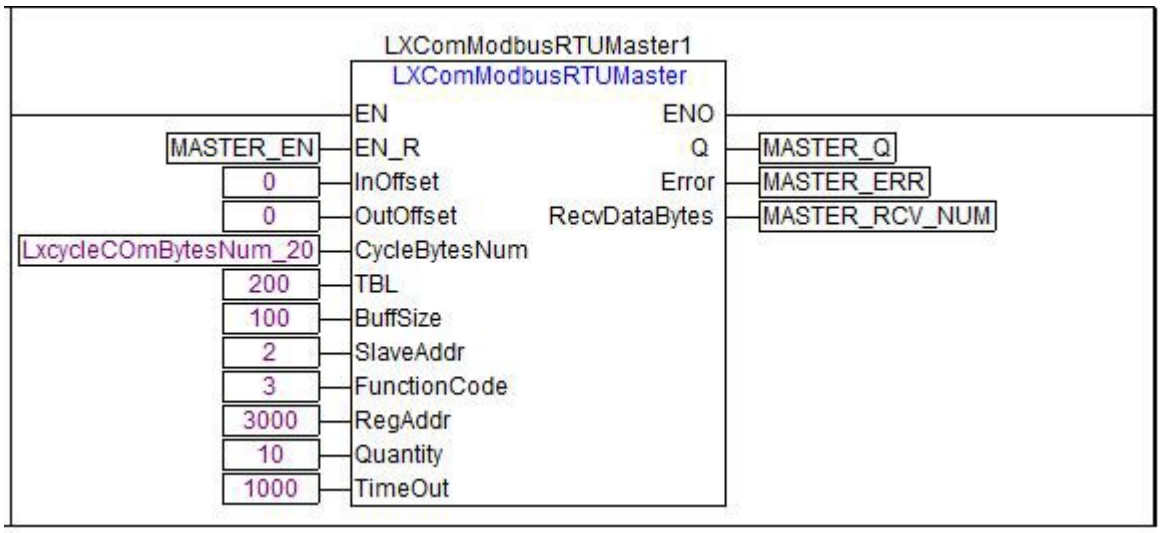
Channel Number	Channel Name	Channel Type	Bytes	Channel Address	Channel Description
1	E1001_IN_1	UINT	2	%IW0	status
2	E1001_IN_2	USINT	1	%IB2	Data In 0
3	E1001_IN_3	USINT	1	%IB3	Data In 1

LX_CM004(1001:LX-CM004)

EtherCAT Slave Station Configuration | Process Parameter | Init Commands | Channel Parameter Information | EtherCAT IO Mapping | Information

Channel Number	Channel Name	Channel Type	Bytes	Channel Address	Channel Description
40	E1001_IN_40	USINT	1	%IB41	Data In 17
41	E1001_IN_41	USINT	1	%IB42	Data In 18
42	E1001_IN_42	USINT	1	%IB43	Data In 19
43	E1001_OUT_43	UINT	2	%QW0	ctrl
44	E1001_OUT_44	USINT	1	%QB2	Data Out 0
45	E1001_OUT_45	USINT	1	%QB3	Data Out 1

■ Master station program:



□ Input parameter description

□ EN_R: Master station program enable, valid on rising edge;

□ InOffset: Corresponding to the Status address of channel 1 in the IO mapping table; 0 - %IW0;

□ OutOffset: Corresponding to the Ctrl address of channel 1 in the IO mapping table; 0——%QW0;

□ CycleBytesNum: Corresponding to the number of IOs selected in the process parameter table;

□ TBL: The first byte number of the data receiving area corresponding to the read slave station command; the first byte number of the data sending area corresponding to the write slave station command; 200——%MB200;

□ BuffSize: Number of bytes in area TBL;

□ SlaveAddr: Slave station device address;

□ FunctionCode: Modbus function code; 3——read holding register;

□ RegAddr: The first address of the register operating on the slave station; 3000——the register %MW0 of the slave station LX;

□ Quantity: Number of registers operating on the slave station; 10——10 words;

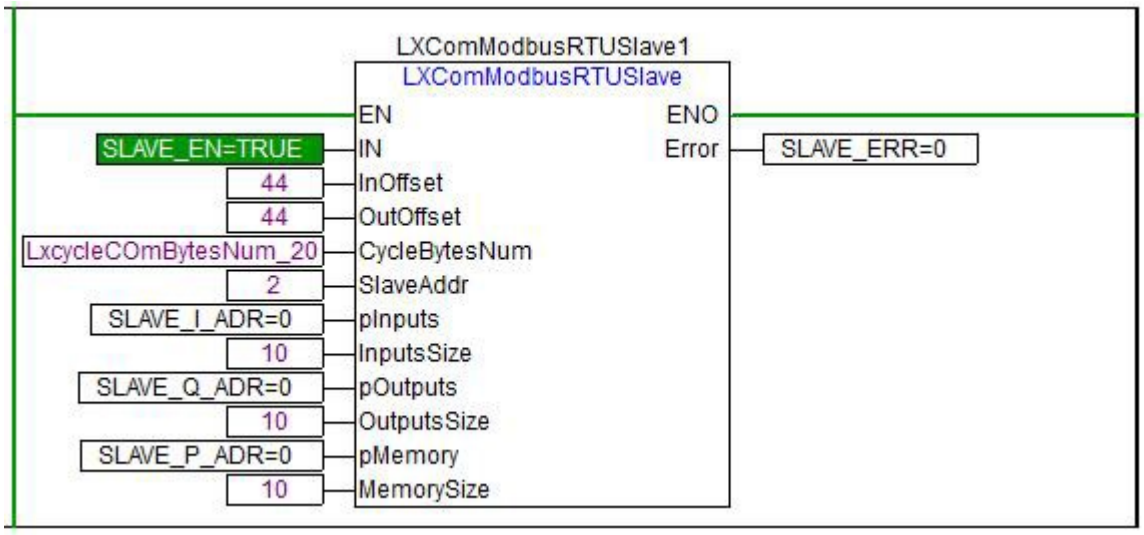
□ TimeOut: Timeout; 1000——1000 ms.

- Output parameter description
  - Q: Communication completion flag bit;
  - Error: Master station communication fault code;
  - ReceiveBytes: Number of bytes received.

- When the slave station is non-standard and causes abnormal data link communication, the MbSetComFrameEndDelay function block can be used to increase the message end delay.

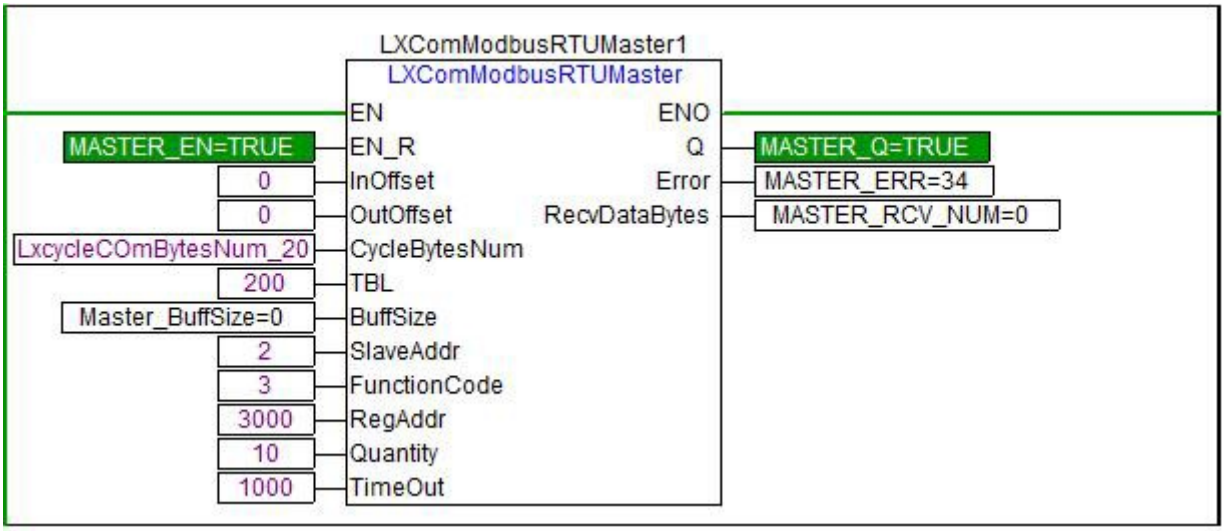
15.4.5 Online Debugging

15.4.5.1 Modbus RTU slave station program online debugging



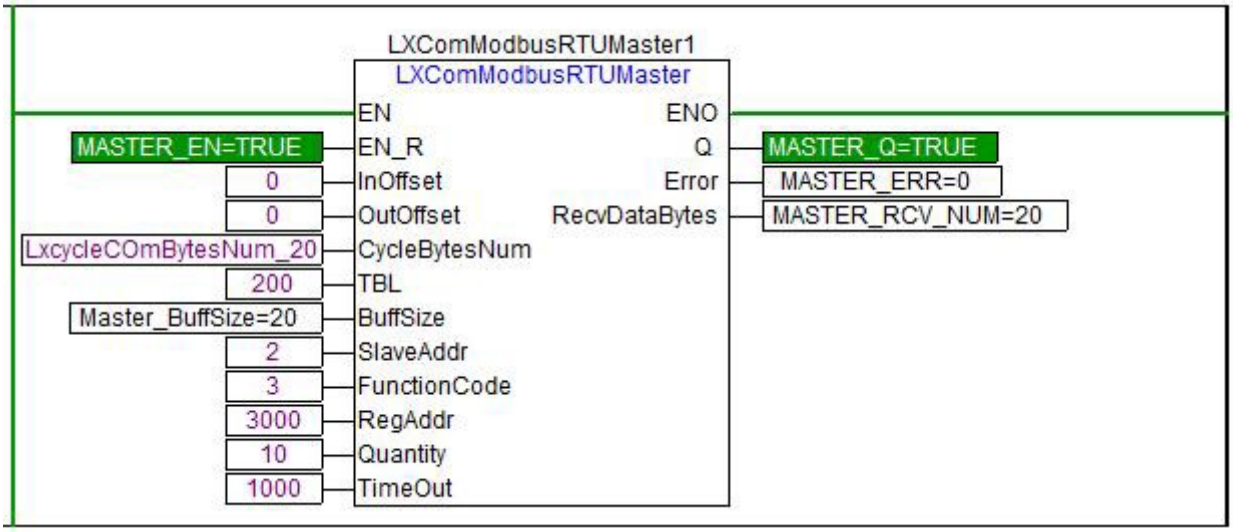
Set the enable variable SLAVE_EN. It can be seen that the slave station is configured successfully and the fault code output is 0: no error.

15.4.5.2 Modbus RTU master station program online debugging



Set the enable variable MASTER_EN. It can be seen that the master station communication is completed, the Q output is 1, and the fault code output is 34: BuffSize input 0. Manually reset the MASTER_EN variable to reset the functional block.

Modify the value of Master_BuffSize to 20 and set the enable variable MASTER_EN. It can be seen that the fault code output is 0 and the RecvDataBytes output is 20, indicating that 20 bytes are successfully received, which is consistent with the reading quantity (10 words) of the input parameter Quantity, and the master station communication instruction is successfully executed.



Monitoring values of the data sending area of the slave station program:

send_tbl

No.	Variable Name	Address	Variable Description	Variable Type	Online Value	Power Fail Safeguard	NetWork Public	Enable const
0001	send_tbl[0]	%MW0		WORD	11	FALSE	Not Public	
0002	send_tbl[1]	%MW2		WORD	22	FALSE	Not Public	
0003	send_tbl[2]	%MW4		WORD	3	FALSE	Not Public	
0004	send_tbl[3]	%MW6		WORD	4	FALSE	Not Public	
0005	send_tbl[4]	%MW8		WORD	5	FALSE	Not Public	
0006	send_tbl[5]	%MW10		WORD	6	FALSE	Not Public	
0007	send_tbl[6]	%MW12		WORD	7	FALSE	Not Public	
0008	send_tbl[7]	%MW14		WORD	0	FALSE	Not Public	
0009	send_tbl[8]	%MW16		WORD	0	FALSE	Not Public	
0010	send_tbl[9]	%MW18		WORD	0	FALSE	Not Public	

Monitoring values of the data receiving area of the master station program:

rcv_tbl

No.	Variable Name	Address	Variable Description	Variable Type	Online Value	Power Fail Safeguard	NetWork Public	Enable co
0001	rcv_tbl[0]	%MW200		WORD	11	FALSE	Not Public	
0002	rcv_tbl[1]	%MW202		WORD	22	FALSE	Not Public	
0003	rcv_tbl[2]	%MW204		WORD	3	FALSE	Not Public	
0004	rcv_tbl[3]	%MW206		WORD	4	FALSE	Not Public	
0005	rcv_tbl[4]	%MW208		WORD	5	FALSE	Not Public	
0006	rcv_tbl[5]	%MW210		WORD	6	FALSE	Not Public	
0007	rcv_tbl[6]	%MW212		WORD	7	FALSE	Not Public	
0008	rcv_tbl[7]	%MW214		WORD	0	FALSE	Not Public	
0009	rcv_tbl[8]	%MW216		WORD	0	FALSE	Not Public	
0010	rcv_tbl[9]	%MW218		WORD	0	FALSE	Not Public	
0011	rcv_tbl[10]	%MW220		WORD	0	FALSE	Not Public	
0012	rcv_tbl[11]	%MW222		WORD	0	FALSE	Not Public	
0013	rcv_tbl[12]	%MW224		WORD	0	FALSE	Not Public	
0014	rcv_tbl[13]	%MW226		WORD	0	FALSE	Not Public	
0015	rcv_tbl[14]	%MW228		WORD	0	FALSE	Not Public	
0016	rcv_tbl[15]	%MW230		WORD	0	FALSE	Not Public	

15.5 Fault Diagnosis

15.5.1 Function Diagnosis Description of MODBUS RTU Master Station

For the MODBUS RTU master station function, when the LXComModbusRTUMaster main functional block in the LXCom_Functions library is used, if there is a communication fault, the Error pin of the functional block will report a response prompt, with the specific information as follows:

Functional Block	Output Parameters	Description
LXComModbusRTUMaster	Error	0: No error 2: Timeout 4: This module does not support function codes 10: Respond to unrequested responses 16: Incorrect slave station address 33: TBL or BuffSize out of range 34: BuffSize input 0 35: Timeout input 0 36: pInputs and InputsSize are enabled but the address area is

		invalid 37: pOutputs and OutputsSize are enabled but the address area is invalid 38: pMemory and MemorySize are enabled but the address area is invalid 64: The slave station returns an invalid instruction 65: The slave station returns an invalid data address 66: The slave station returns an invalid data value 67: The slave station returns a slave station device fault 68: The slave station returns other error diagnosis 129: InOffset out of range 130: OutOffset out of range 131: CycleBytesNum is illegal 132: Error occurred in data acquisition

15.5.2 Function Diagnosis Description of MODBUS RTU Slave Station

For the MODBUS RTU slave station function, when the LXComModbusRTUSlave functional block in the LXCom_Functions library is used, if there is a communication fault, the Error pin of the functional block will report a response prompt, with the specific information as follows:

Functional Block	Output Parameters	Description
LXComModbusRTUSlave	Error	0: No error 1: Received a data packet that failed the CRC check 2: Received a data packet from a non-local address 16: Incorrect slave station address 36: pInputs and InputsSize are enabled but the address area is invalid 37: pOutputs and OutputsSize are enabled but the address area is invalid 38: pMemory and MemorySize are enabled but the address area is invalid 129: InOffset out of range 130: OutOffset out of range 131: CycleBytesNum is illegal 132: Error occurred in data acquisition

15.5.3 Function Diagnosis Description of Serial Port Free Protocol

For free protocols, when the LXComSerialLineControl, LXComReceive LXComSend, and LXComClearBuf function blocks in the LXCom_Functions library are used, if there is a communication fault, the Error pin of the functional block will report a response prompt, with the specific information as follows:

Functional Block	Output Parameters	Description
LXComSerialLineControl	Error	0: No error 1: Incorrect buffer pointer input 129: InOffset out of range 130: OutOffset out of range 131: CycleBytesNum is illegal 132: Error occurred in data acquisition

LXComReceive	Error	0: No error 1: Incorrect buffer pointer input 2: No valid Rx mode 3: Incorrect input of prefix or suffix pointer 4: Sum of prefix and suffix exceeds 1000 5: RecvLen exceeds 1000 6: TBL or BuffSize out of range 7: Actual data length is greater than BuffSize
LXComSend	Error	0: No error 1: Incorrect buffer pointer input 2: TBL or BuffSize out of range 3: SendDataBytes exceeds 1000

For modules, when the LXComClearCom functional block in the LXCom_Functions library is used to implement module initialization, if there is a communication fault, the Error pin of the functional block will report a response prompt, with the specific information as follows:

Functional Block	Output Parameters	Description
LXComClearCom	Error	0: No error 129: InOffset out of range 130: OutOffset out of range 131: CycleBytesNum is illegal

15.5.4 Modbus RTU Protocol Fault Diagnosis

When online, the status of each instruction can be diagnosed through the value of the diagnostic status:

Diagnostic Value	Meaning	Possible Causes or Solutions
0	No error	-
1	Data CRC error	1. Check communication parameters: Ensure baud rate, data bits, stop bits, and parity are identical on master and slave. 2. Check physical connection: Verify A/B wiring is correct, not reversed or loose; measure termination resistance with a multimeter (120Ω resistor at each bus end, total approx. 60Ω). 3. Minimal system test: Test one-to-one with the slave using short, high-quality shielded twisted pair cable in an interference-free environment. 4. Check interference and grounding: Ensure communication cables are away from interference sources like power cables; ensure shield is grounded at a single point (usually at the master). 5. Check bus topology and load: Ensure a daisy-chain topology with no branches; check if the number of slaves exceeds RS-485 drive capacity (typically 32 loads), add a repeater if necessary. 6. Check signal quality: If possible, observe the signal waveform with an oscilloscope for overshoot, ringing, or distortion. 7. Adjust frame interval: Try increasing or decreasing the Modbus frame interval (e.g., 3.5 character time) to see if the issue improves. 8. Contact device manufacturer: If the above steps fail, contact the device manufacturer.
2	Slave response timeout	1. Check physical connection: Verify RS-485 cable connection is correct and reliable; measure A/B voltage difference to confirm signal integrity. 2. Check device status: Verify device power supply and network connection status are normal. 3. Check parameter matching: Confirm communication parameters (baud rate, etc.) match between master and slave. 4. Check master timeout setting: Increase the master's timeout (Response Timeout) to exceed the total data transmission time and slave response time. 5. Check slave status: Ensure slave power and operational status are normal; verify slave address is correct; check if slave is busy or in a fault state.

		6. Reduce bus load: Try a one-to-one connection test to isolate other slaves. 7. Comparative testing: Test the same slave with a third-party standard Modbus master, or test the master with a standard slave. 8. Check master transmission: Verify the message sent by the master (address, function code, etc.) is correct using a serial monitor tool. 9. Contact device manufacturer: If the above steps fail, contact the device manufacturer.
4	Function code in the slave response does not match the master request	1. Check master timeout setting: Increase the master's Response Timeout. 2. Check master command polling frequency, adjust it based on different types (periodic, high-level, rising edge). 3. Check if the slave returned an exception response (function code 0x80 + request function code). 4. Confirm if the slave supports the function code (consult the slave manual). 5. Check for multiple masters or slave address conflicts on the bus causing response confusion. 6. One-to-one connection test to eliminate interference from other devices. 7. Contact device manufacturer: If the above steps fail, contact the slave device manufacturer.
8	Data length in the slave response does not match the master's expected length	1. Check request message: Ensure the data length requested by the master (number of registers/coils) is within the slave's allowable range. 2. Check master timeout setting: Increase the master's Response Timeout. 3. Check master command polling frequency, adjust it based on different types (periodic, high-level, rising edge). 4. Check response message: Monitor messages to see if the slave returns an exception response (shorter length); if so, handle according to the exception code. 5. Eliminate interference and conflicts: Test one-to-one between master and slave; check for multiple masters or slave address conflicts on the bus. 6. Adjust timing: Increase the Modbus frame interval to allow sufficient response time for the slave. 7. Check communication parameters: Ensure baud rate, etc., match to avoid data truncation or misalignment due to baud rate deviation. 8. Contact device manufacturer: If the above steps fail, contact the slave device manufacturer.
16	Coil or register address in the slave response does not match the address in the request message	1. Check address mapping: Consult the slave manual to confirm the address format used (PLC address, 0-Based or 1-Based) and if there is a fixed offset (e.g., 40001 maps to internal address 0). 2. Check command configuration: Check if the master has multiple duplicate write operations to the same address on the current slave. 3. Check master timeout setting: Increase the master's Response Timeout. 4. Check master command polling frequency, adjust it based on different types (periodic, high-level, rising edge). 5. Simple test: Test one-to-one with the slave, test a single address (e.g., address 0) and monitor messages, comparing request and response addresses. 6. Pattern test: Test consecutive addresses (e.g., 0, 1, 2), observe address change patterns to determine fixed offset or non-linear mapping. 7. Eliminate interference: Check if the slave is connected to multiple masters; perform comparative testing with a standard master or slave. 8. Check slave status: Ensure the slave has no timeout or exception responses. 9. Contact the slave device manufacturer: Obtain an accurate address mapping table or firmware update.

15.5.5 Serial Port Capture Error Codes

ErrorID	Description
0	No error
1	Invalid EtherCAT slave address input (1~65535)
2	Input EtherCAT address does not exist

3	Input EtherCAT address is not a serial port module
4	Invalid communication serial port number input (1, 2)
5	Input In Offset out of bounds
6	Input Out Offset out of bounds
7	When serial port number is 1, input offset does not match the module's actual offset
8	When serial port number is 1, output offset does not match the module's actual offset
9	Input cyclic data length is not 20/40/80
10	Invalid Modbus slave address input (1~247)
11	Invalid timeout input (must be an integer greater than 0)
20	Invalid function code input
21	Invalid number of slave registers input
40	Input pSendBuffer pointer error
41	Input send data length exceeds the 1000-byte limit
60	Input pRecvBuffer pointer error
61	Input prefix length exceeds receive length
62	Input prefix pointer is null
63	Input suffix pointer is null
64	Input receive length exceeds the 1000-byte limit
100	Communication timeout
120	Master send process error
121	Master receive process error
122	Received message CRC check error
123	Data length or register address in slave response does not match the request
124	Slave returned illegal instruction
125	Slave returned illegal data address
126	Slave returned illegal data value
127	Slave returned slave device failure
128	Slave returned other error diagnostic
140	Error during EtherCAT cyclic message reception
141	Received message CRC check error
142	Error sending data from controller (as slave) to master
160	Free port send process error
180	Error during data reply process after free port reception
181	Free port receive buffer full or prefix pointer mismatch
182	Received data length is zero, receive timeout
183	Receive mode not supported
184	Free port cyclic data reception error
200	Not a HollySys LX series serial port module
201	Not a serial port module
202	Capture port not enabled
203	EtherCAT_Read failed. Check module status or confirm if input address is a configured slave
204	Capture via AT already enabled, function block capture is disabled
205	Capture function block stopped
206	Capture via AT has stopped the function block, waiting for AT to start capture
207	Capturing with new parameters
208	Capture function block called repeatedly
209	Capture function serial port parameter is empty
255	Maximum error code value

Chapter 16 CANopen Communication

16.1 Network Overview

16.1.1 Introduction

The LX-CM007 is a CANopen master protocol module that communicates with other CANopen slave devices on the field via CAN interface. It supports up to 64 slave nodes.

16.1.2 Network Characteristics

- 1 isolated CAN channel with CANopen protocol support
- Built-in 120Ω termination resistor
- Maximum of 64 connectable slave devices

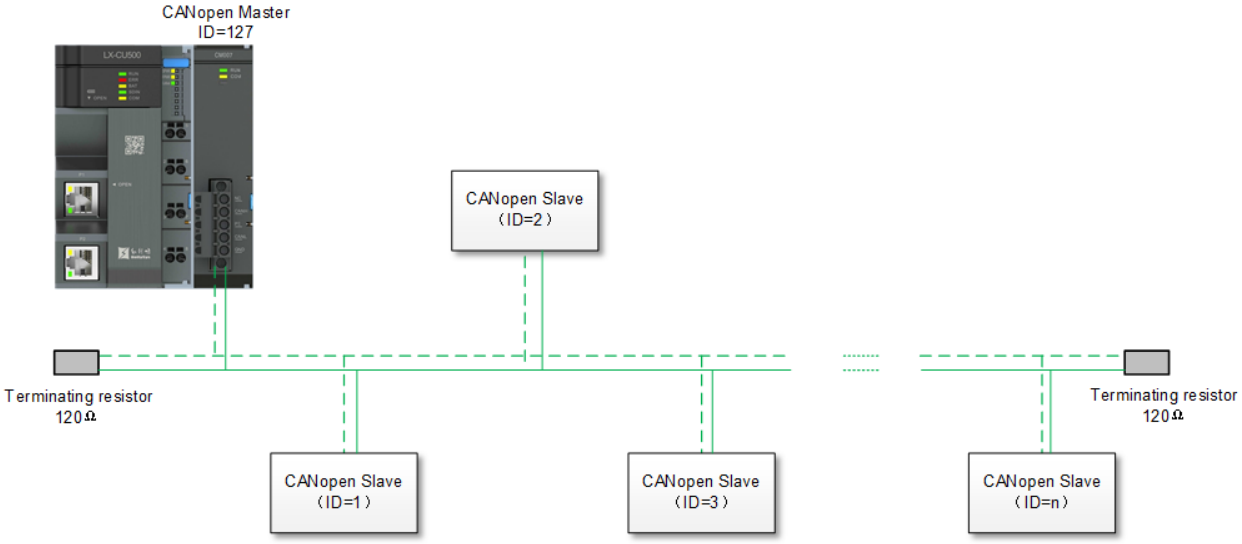
16.2 Network Topology

The LX-CM007 functions as a CANopen master, communicating with the LX-CUxxx controller via an internal bus. It converts signals into the CANopen protocol and facilitates data exchange between the master and CANopen slave devices.

A single CANopen master can support up to 64 slave nodes, compatible with linear, star, tree, and hybrid network topologies. Shielded twisted pair cables are used for communication lines.

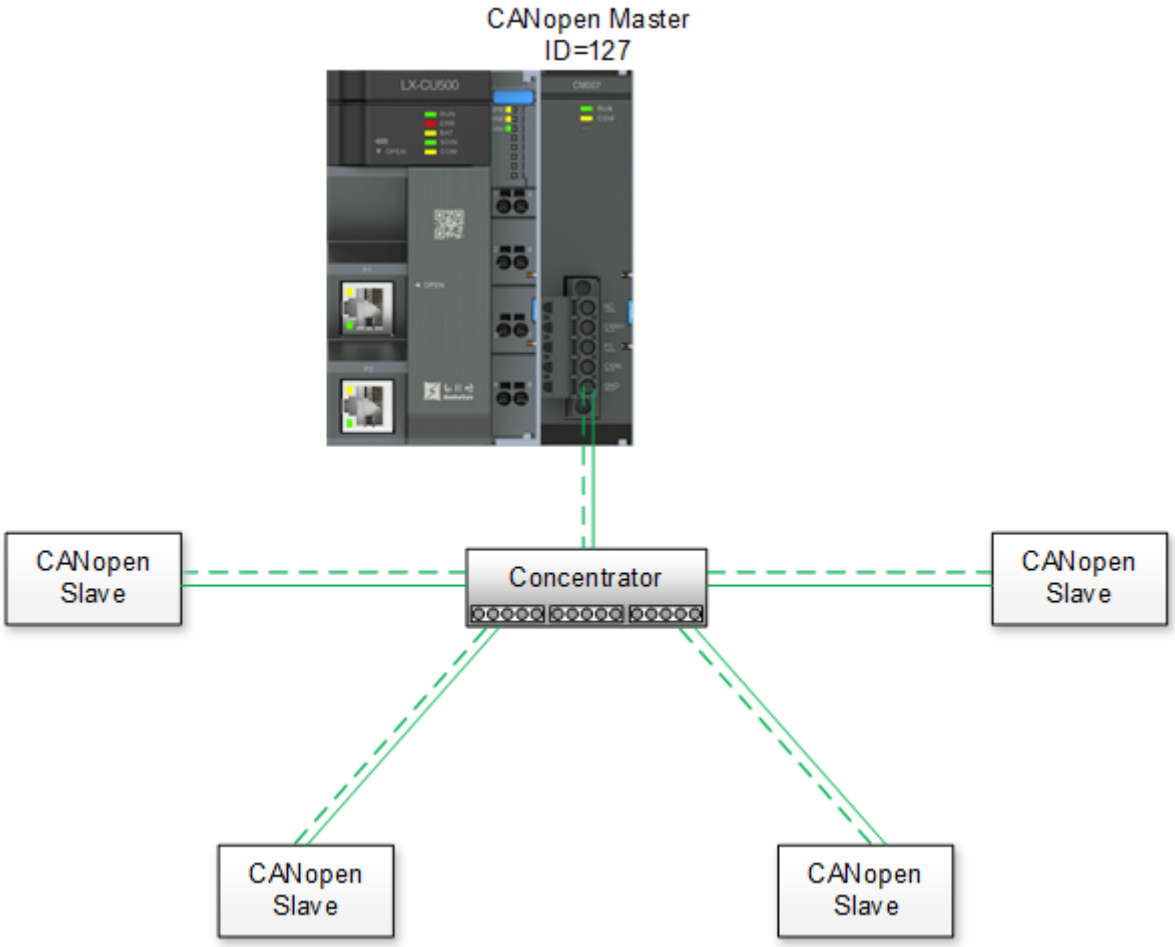
16.2.1 Linear Network Topology

In a linear topology (as shown in the diagram below), a terminating resistor (120Ω) must be connected at both ends of the bus. This topology employs a multi-drop connection structure, where all CANopen slave devices are connected to a shared transmission medium.



16.2.2 Star Network Topology

As shown in the figure below, this is a star topology structure. All CANopen slave devices are connected point-to-point to the hub (central node), which is responsible for physical signal relaying or management.



16.3 Configuration

CANopen configuration includes the addition of master and slave stations as well as parameter configuration instructions.

16.3.1 Configuring the CANopen Master

16.3.1.1 Adding a CANopen Master

The LX-CM007 supports both automatic and manual addition. For detailed instructions, refer to EtherCAT Automatic Scanning Device and Adding Device Manually.

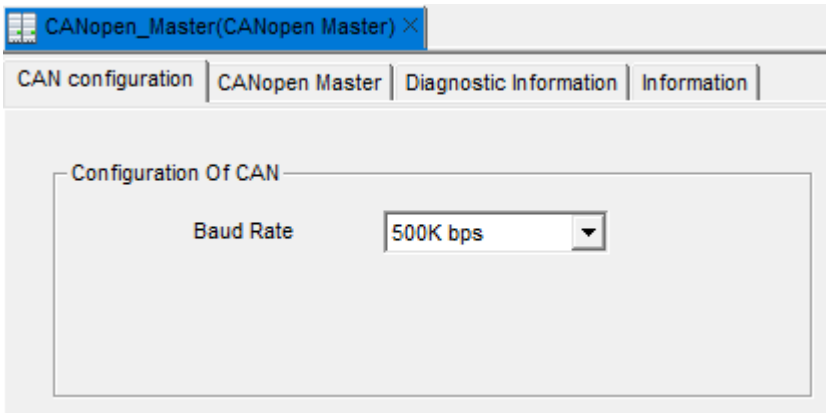
16.3.1.2 Configuring DeviceNet Master Station

Double-click the added "CANopen_Master" module tree node to configure the master station protocol parameters.

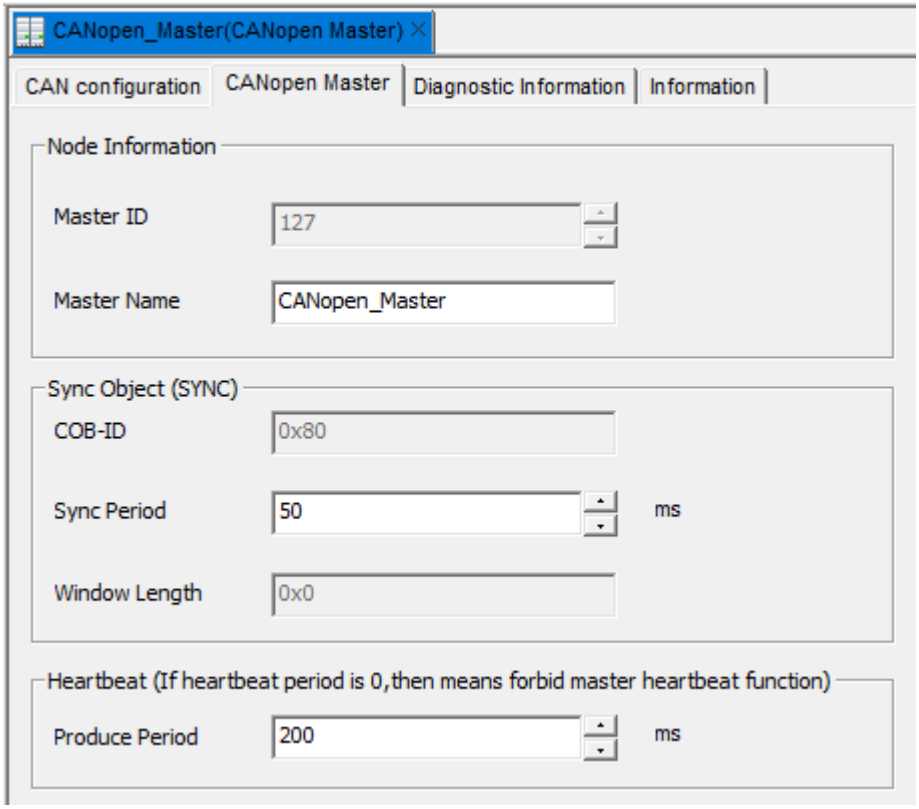
The configuration interface includes "CAN Configuration," "CANopen Master," "Diagnostic Information," and "Information." The detailed descriptions are as follows:

- CAN Configuration

  Baud Rate: The communication baud rate for CANopen. Available options: 10Kbps, 20Kbps, 50Kbps, 100Kbps, 125Kbps, 250Kbps, 500Kbps (default), 800Kbps, 1Mbps.



- CANopen Master
  - Node Information
    - Master ID: The ID of the CANopen master. Default value is 127 and cannot be modified.
    - Master Name: The name of the CANopen master. After modification, the master name on the left navigation tree will update accordingly. Default value is CANopen_Master. Supports uppercase and lowercase letters, numbers, and underscores, but cannot start with a number.
  - Sync Object (SYNC)
    - COB-ID: The communication identifier of the CAN frame. Cannot be modified.
    - Sync Period: The cycle period of CANopen communication, unit: ms. Default value is 50, valid range [2, 65535].
    - Window Length: The validity period for sending PDOs from the slave station, meaning that PDOs sent within this time window are valid. The default value is 0x0 and cannot be modified. A value of 0 indicates that the synchronization window is disabled, meaning there is no time window restriction for the slave station to process PDO messages.
  - Heartbeat
    - Production Period: The cycle period for the master to send heartbeat messages, unit: ms. Default value is 200, valid range [0, 65535]. Setting it to 0 disables the master heartbeat function.

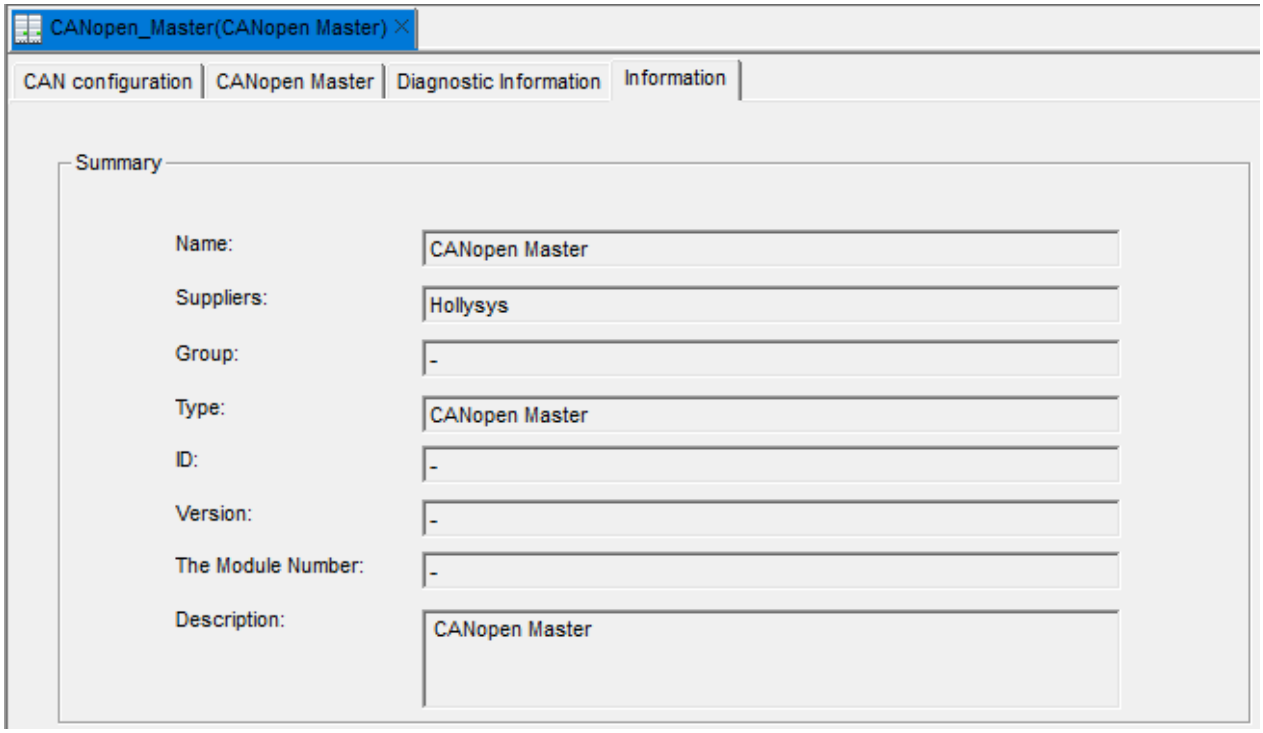


- Diagnostic Information

  For detailed descriptions, please refer to the Fault Diagnosis section.

- Information

  Displays relevant information about the CANopen_Master.

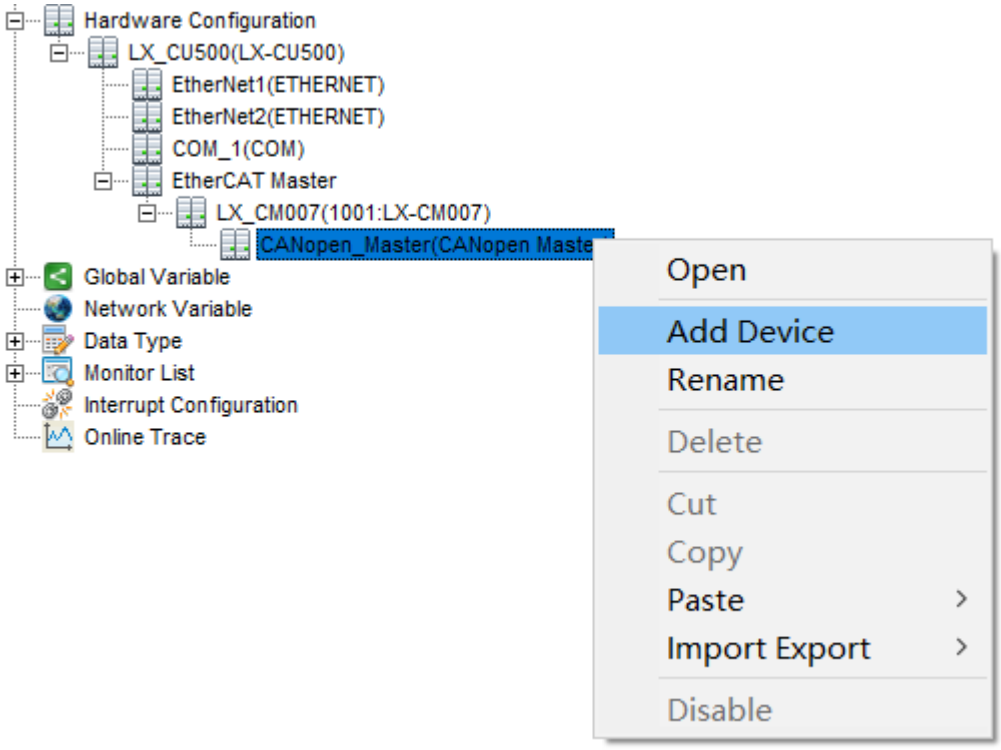


16.3.2 Configuring CANopen Slave Devices

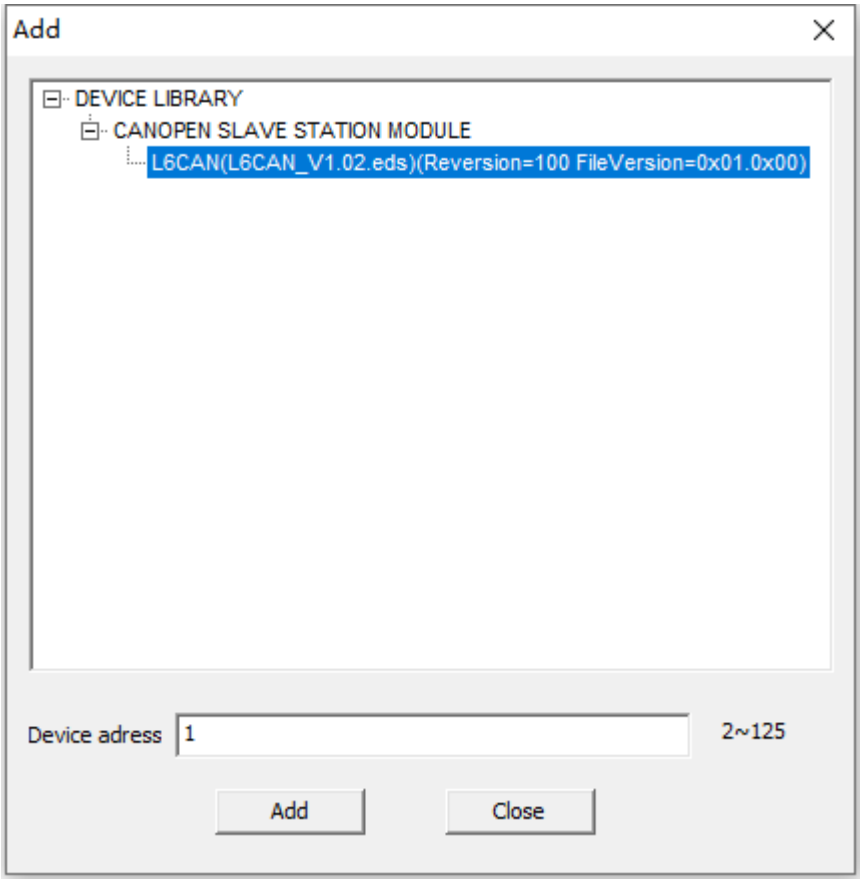
16.3.2.1 Adding a CANopen Slave

(1) Import Configuration File: Go to the menu bar: "Project" - "Import Device Description File", select and import the CANopen device configuration file.

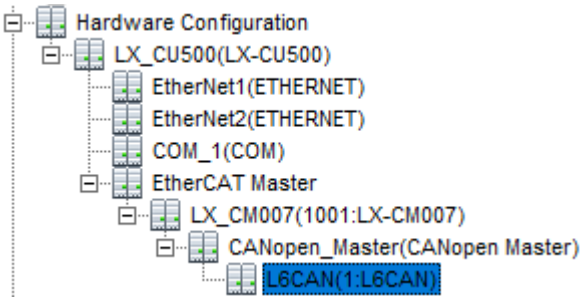
(2) Right-click the "CANopen_Master" node. Select the "Add Device" command (as shown in the figure below).



(3) In the pop-up "Add" dialog box, select the device imported in step (1) and click the Add button.



(4) The CANopen slave station is added under the "CANopen_Master" node.



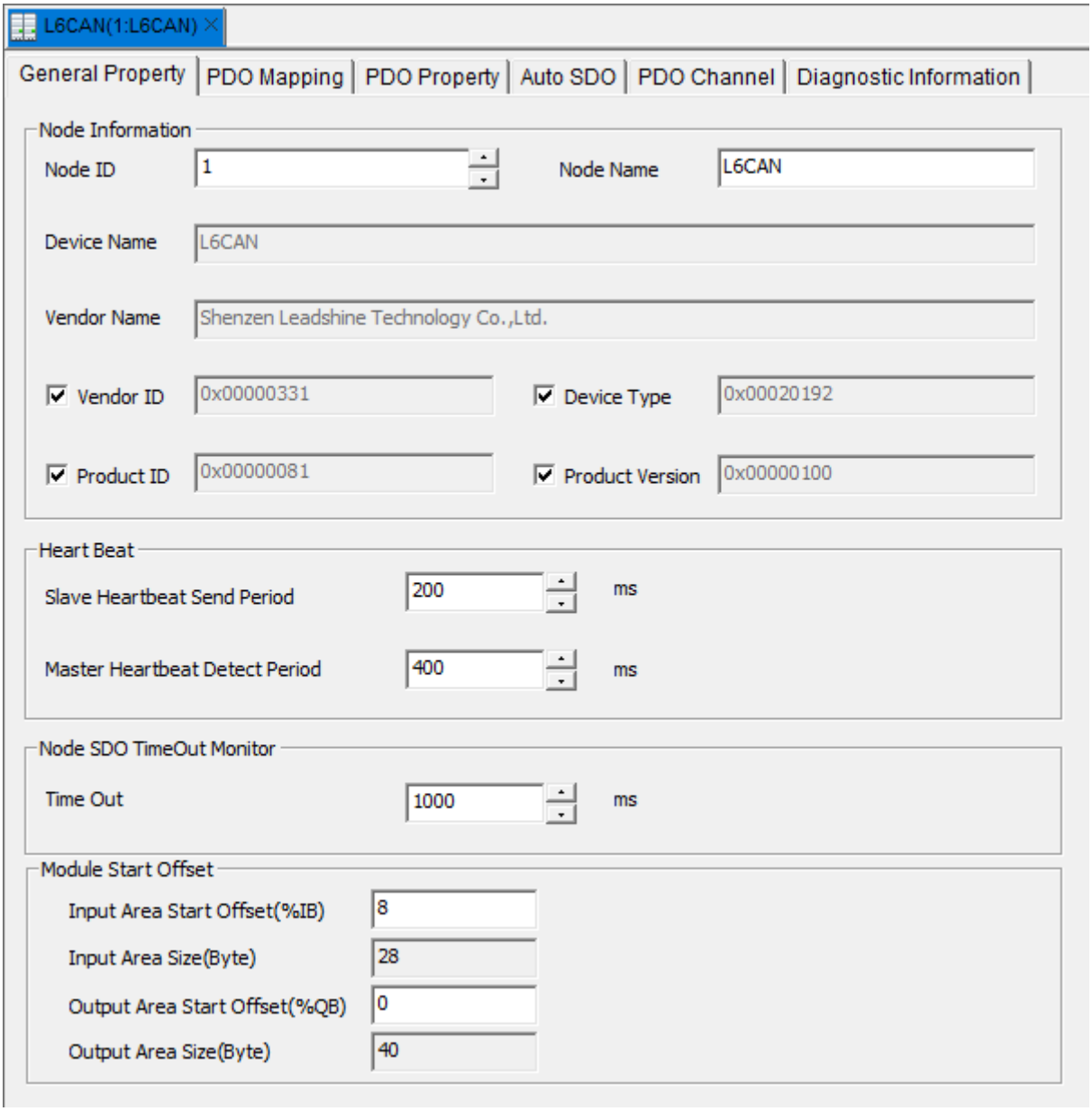
16.3.2.2 Configuring a CANopen Slave

Double-click the CANopen slave device node to configure slave parameters: "General Property", "PDO Mapping", "PDO Property", "Auto SDO", "PDO Channel", and "Diagnostic Information." The detailed descriptions are as follows:

- General Property
  - Node Information
    - Node ID: The ID of the CANopen slave, automatically assigned by the system. The valid range is [1, 126]. Ensure the ID is globally unique during configuration.
    - Node Name: The name of the CANopen slave. After modification, the slave name in the left navigation tree will update accordingly. Automatically assigned by the system, it supports uppercase/lowercase letters, numbers, and underscores but cannot start with a number.
    - Device Name: The device name of the CANopen slave, defined by the slave manufacturer and cannot be modified.
    - Vendor Name: The manufacturer name of the CANopen slave, defined by the slave manufacturer and cannot be modified.
    - Vendor ID/Device Type/Product ID/Product Version: Detailed information about the CANopen slave device, defined by the slave manufacturer and cannot be modified. If checked, the controller will perform a compatibility check during the slave startup configuration. If the check fails, the configuration for the current slave will be terminated, and a diagnostic error will be reported.
  - Heart Beat
    - Slave Heartbeat Send Period: The interval at which the slave sends heartbeat messages to the master, in milliseconds (ms). The default value is 200, with a valid range of [0, 65535]. When set to 0, it indicates that the slave station's heartbeat function is disabled. After disabling, the master station cannot detect the status of the slave station, which will result in the slave station being reported as offline. It is not recommended to configure this parameter to 0 in practical use.
    - Master Heartbeat Detect Period: The heartbeat detection cycle of the master station for slave stations. If the master station does not receive a heartbeat message from a slave station within this cycle, the slave station will be considered offline. Unit: ms. Default value: 400. Valid range: [1, 65535]. The value of this parameter must be greater than the heartbeat transmission cycle

of the slave station. It is recommended that the value of this parameter be ≥ 2 * the heartbeat transmission cycle of the slave station.

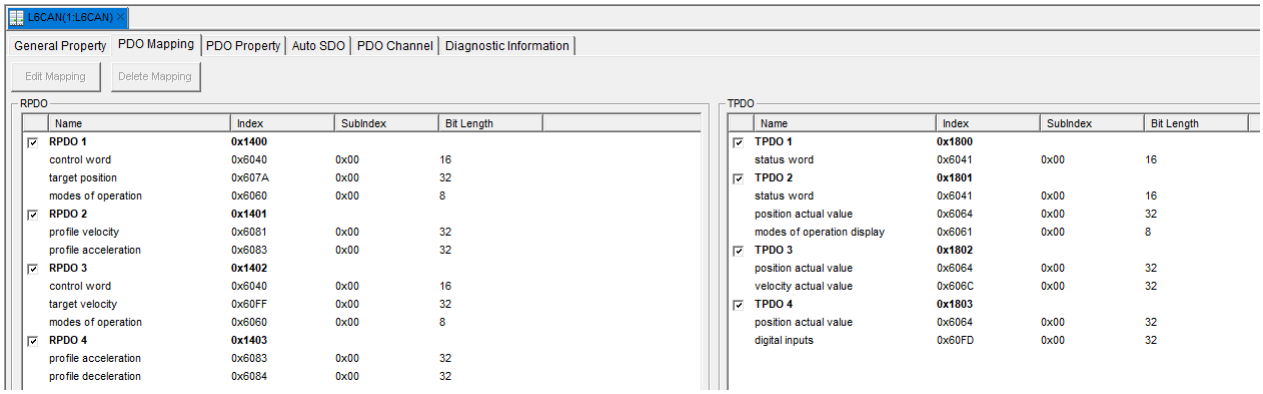
- Node SDO Timeout Monitor
  - Time Out: The timeout period for the master to read/write the slave's SDO, in milliseconds (ms). The default value is 1000, with a valid range of [100, 65535].
- Module Start Offset
  - Input Area Start Offset: Configuration related to the input area data. The default value is 0, and the valid range must not exceed the controller's input area size.
  - Input Area Size: Automatically filled based on the TPDO data configured in the "PDO Mapping" tab and cannot be modified.
  - Output Area Start Offset: Configuration related to the output area data. The default value is 0, and the valid range must not exceed the controller's output area size.
  - Output Area Size: Automatically filled based on the RPDO data configured in the "PDO Mapping" tab and cannot be modified.



- PDO Mapping

Select PDO Parameters: On the "PDO Mapping" tab, choose the PDO parameters to be added to the I/O mapping. Check the PDO parameters in the RPDO (Receive PDO parameters) or TPDO (Transmit PDO parameters) list box, and all process data under the selected PDO parameters will be added to the PDO channels. The added PDO parameters can be viewed on the "PDO Channels" tab. Simultaneously, these PDO parameters will be mapped to the "PDO Properties" tab.

The cumulative total of RPDOs for all slave devices must not exceed 256 entries, and the cumulative total of TPDOs for all slave devices must not exceed 256 entries. The combined data size of RPDOs for all slave devices must not exceed 1280 bytes, and the sum of TPDO data and online status diagnostic data for all slave devices must not exceed 1280 bytes.



□ Edit Mapping

By default, each PDO parameter is preconfigured with object data. You can manually add or remove object data for TPDOs and RPDOs.

(a) Select the PDO parameter row, then click the Edit Mapping button or right-click and choose "Edit Mapping" from the menu. This will open the PDO Parameter Configuration window.

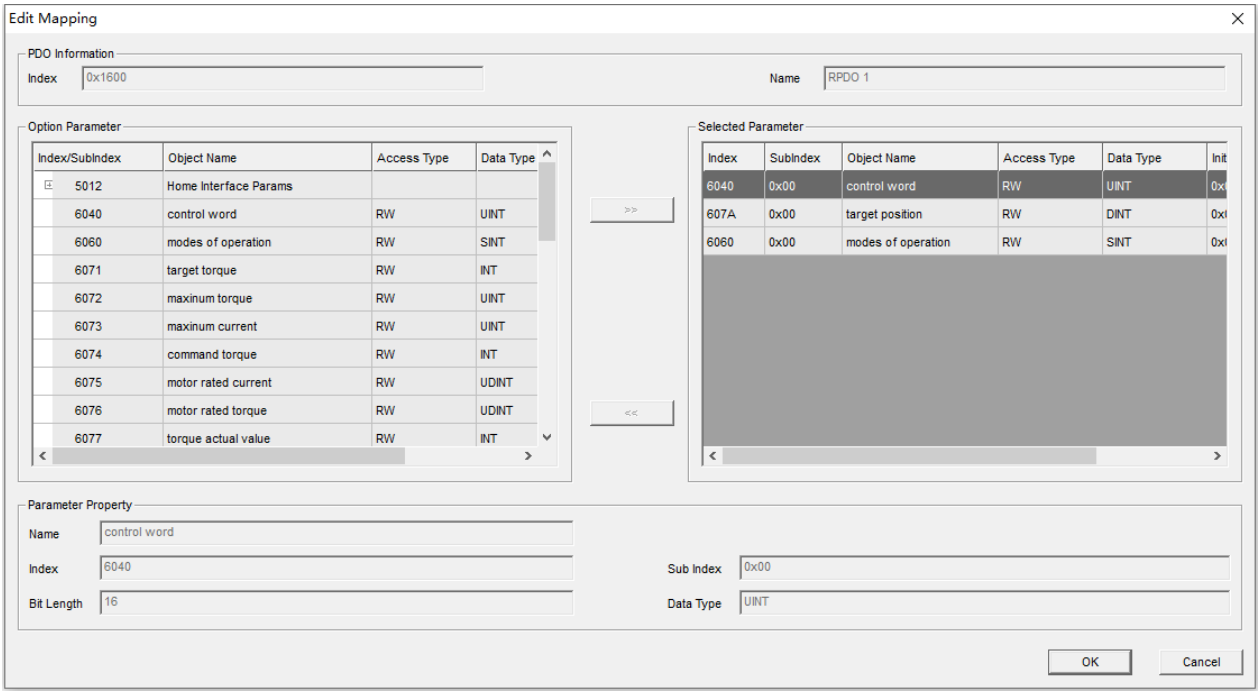
(b) In the "Available Parameters" list, click the "Index/Subindex" dropdown ⊞ to expand the parameter list. Select the desired object data and click the Add button >> to move the parameter to the "Selected Parameters" list.

Configuration Guidelines:

RPDO parameters can only be configured with RW (Read-Write) object data. TPDO parameters can be configured with RO (Read-Only) or RW (Read-Write) object data.

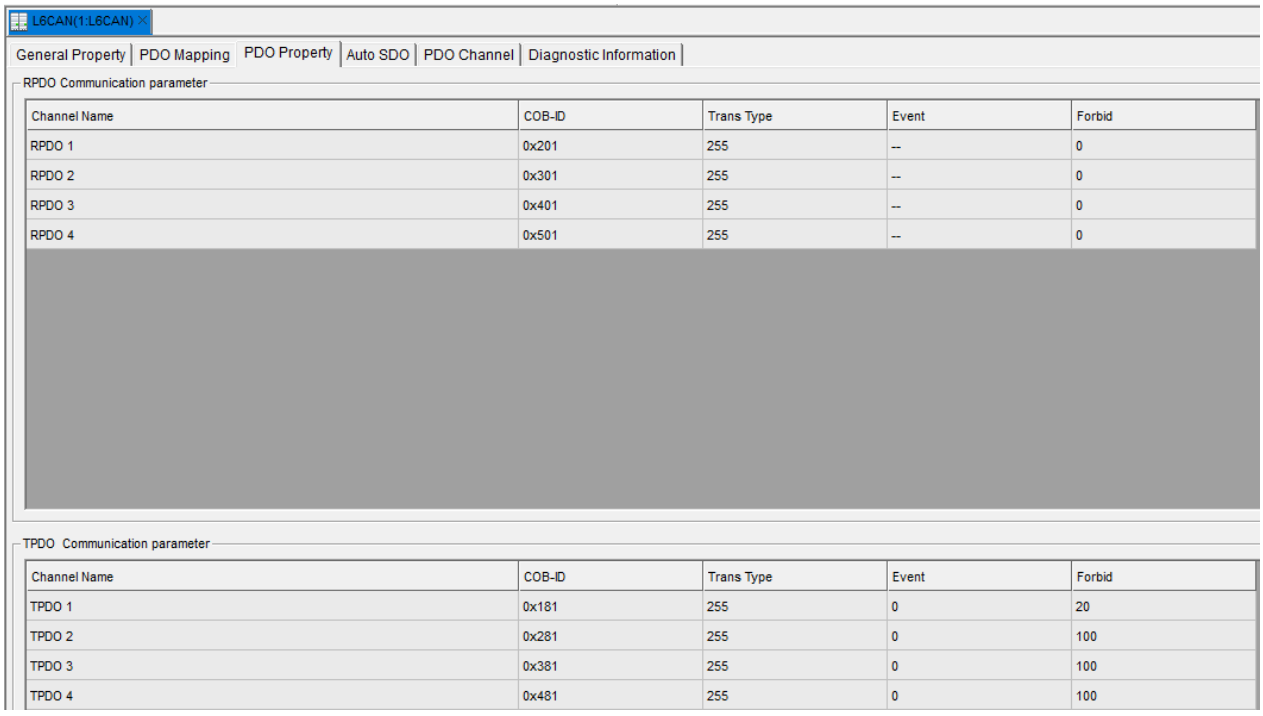
Each PDO parameter supports a maximum of 8 bytes of data.

(c) Click OK to confirm the object data configuration. The newly added object data will be displayed under the PDO parameters.



■ PDO Property

On the "PDO Properties" tab, you can configure the communication attributes of PDO parameters, including COB-ID, transmission type, event, and inhibit attributes.



PDO Properties Information

Parameter	Description	Principle
COB-ID	The message ID uniquely identifies the PDO and determines its priority (a smaller ID value indicates higher priority).	Default value: Defined in the EDS file Valid range: [0x180, 0x57F]
Trans	The triggering method for transmitting PDO data is detailed in	Default value: Defined in the EDS file

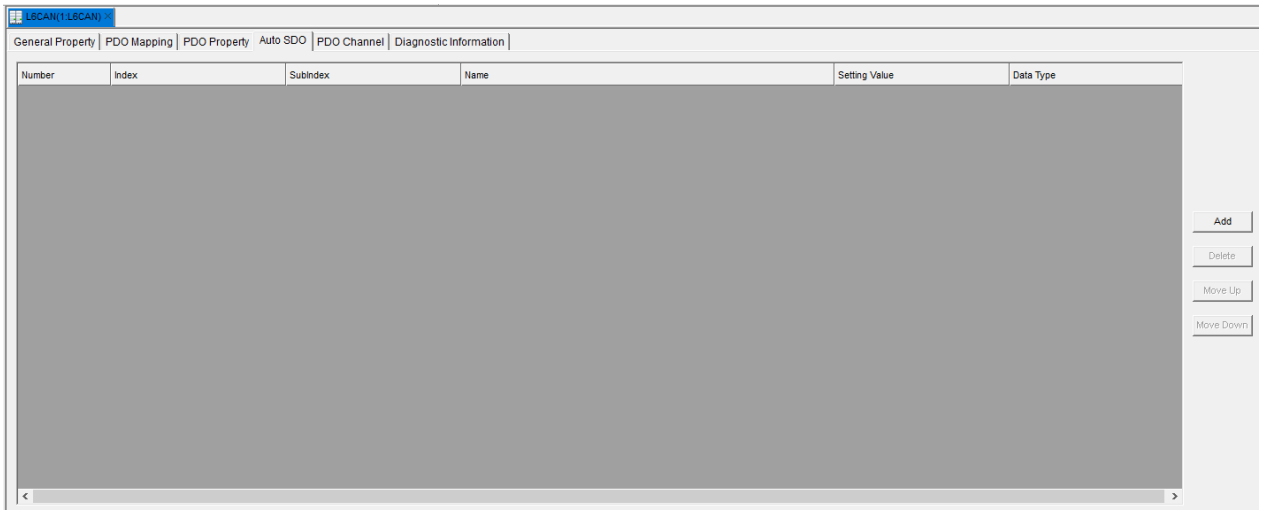
Type	the transmission type description.	Valid range: • 0: Non-cyclic synchronization • 1~240: Cyclic synchronization, indicating transmission is triggered every specified number of SYNC messages • 254: Asynchronous, triggered by manufacturer-specific events • 255: Asynchronous, triggered by device sub-protocol-specific events
Event	Configurable TPDO parameter: Indicates the sending interval for periodic PDOs. The PDO is automatically triggered upon timeout, even if the data remains unchanged. Unit: ms.	Default value: Defined in the EDS file Valid range: [0, 65535]
Forbid	Configurable TPDO parameter: Indicates the minimum time interval between two PDO transmissions to prevent event-triggered PDOs from frequently occupying the bus. Unit: 100μs.	Default value: Defined in the EDS file Valid range: [0, 65535]

Transmission Type Description

Transmission Type		Description
0: Non-cyclic synchronization	RPDO	The master station sends a SYNC message to the slave station every synchronization cycle. When RPDO data changes, the master transmits it to the slave, but the received data only takes effect upon the next SYNC message. If RPDO data remains unchanged, the master does not transmit it to the slave.
	TPDO	The master station sends a SYNC message to the slave station every synchronization cycle. When TPDO data changes and the slave receives the SYNC message, it transmits the TPDO data to the master, which takes effect immediately upon reception. If TPDO data remains unchanged, the slave does not transmit it to the master.
1~240: Cyclic synchronization	RPDO	The master station sends a SYNC message to the slave station every synchronization cycle. A value of 1~240 indicates how many SYNC cycles trigger a transmission (e.g., the master sends RPDO data every *n* cycles). The slave applies the received RPDO data only after the next SYNC message.
	TPDO	The master station sends a SYNC message to the slave station every synchronization cycle. The slave transmits TPDO data to the master every *n* received SYNC messages, and the master applies the data immediately upon reception.
254: Asynchronous (triggered by manufacturer-specific events) 255: Asynchronous (triggered by device sub-protocol-specific events)	RPDO	When RPDO data changes, the master transmits it to the slave, and the data takes effect immediately. If RPDO data remains unchanged, the master does not transmit it to the slave.
	TPDO	• If both event time and inhibit time are 0, the slave transmits TPDO data to the master upon change, and the master applies it immediately. If TPDO data is unchanged, the slave does not transmit it. • If event time and inhibit time are non-zero, the slave sends TPDO data to the master at each event time interval (but not again within the inhibit time after transmission). If TPDO data changes, it is transmitted immediately, and the master applies it upon reception.

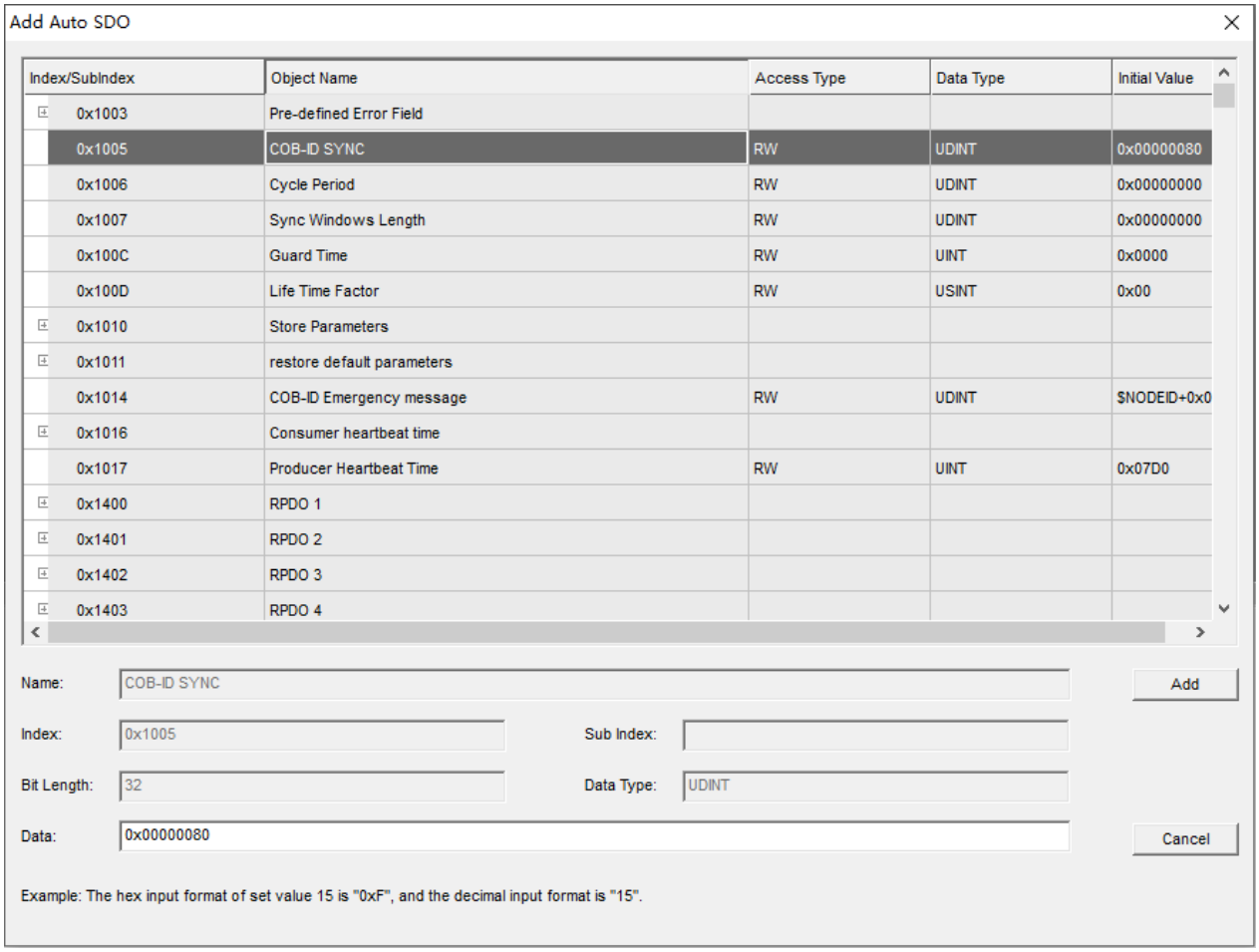
■ Auto SDO

Each slave device supports up to 32 auto SDO entries. Once the master establishes a connection with the slave, these parameters are written once. If the controller reboots or the slave disconnects, the master will re-establish the connection and rewrite the auto SDO parameters.



Click the Add button on the right to open the Auto SDO configuration dialog.

Select the object data, nnter the desired value in the Data field. Click Add to include the entry in the Auto SDO list.



■ PDO Channel

The PDO channel data is generated from the PDO object entries selected in the [PDO Mapping] tab.

The channel addresses are automatically assigned by the system.

L6CAN(1:L6CAN)

General Property | PDO Mapping | PDO Property | Auto SDO | PDO Channel | Diagnostic Information

Channel Number	Channel Name	GroupName	Channel Type	Index	SubIndex	Channel Adress	Channel Description
RPDO 1							
1	CANOPEN_8_1		UINT	0x6040	0x00	%QW0	control word
2	CANOPEN_8_2		DINT	0x607A	0x00	%QD4	target position
3	CANOPEN_8_3		SINT	0x6060	0x00	%QB8	modes of operation
RPDO 2							
4	CANOPEN_8_4		UDINT	0x6081	0x00	%QD12	profile velocity
5	CANOPEN_8_5		UDINT	0x6083	0x00	%QD16	profile acceleration
RPDO 3							
6	CANOPEN_8_6		UINT	0x6040	0x00	%QW20	control word
7	CANOPEN_8_7		DINT	0x60FF	0x00	%QD24	target velocity
8	CANOPEN_8_8		SINT	0x6060	0x00	%QB28	modes of operation
RPDO 4							
9	CANOPEN_8_9		UDINT	0x6083	0x00	%QD32	profile acceleration
10	CANOPEN_8_10		UDINT	0x6084	0x00	%QD36	profile deceleration
TPDO 1							
11	CANOPEN_8_11		UINT	0x6041	0x00	%IW8	status word
TPDO 2							
12	CANOPEN_8_12		UINT	0x6041	0x00	%IW10	status word
13	CANOPEN_8_13		DINT	0x6064	0x00	%ID12	position actual value
14	CANOPEN_8_14		SINT	0x6061	0x00	%IB16	modes of operation display
TPDO 3							
15	CANOPEN_8_15		DINT	0x6064	0x00	%ID20	position actual value
16	CANOPEN_8_16		DINT	0x606C	0x00	%ID24	velocity actual value
TPDO 4							
17	CANOPEN_8_17		DINT	0x6064	0x00	%ID28	position actual value
18	CANOPEN_8_18		UDINT	0x60FD	0x00	%ID32	digital inputs

- Diagnostic Information

For detailed descriptions, please refer to the Fault Diagnosis section.

16.4 Fault Diagnosis

LX-CM007 includes slave station online status diagnostics, with specific definitions as shown below.

LX_CM007(1001:LX-CM007)

EtherCAT Slave Station Configuration | Process Parameter | Init Commands | EtherCAT IO Mapping | Information

Channel Number	Channel Name	GroupName	Channel Type	Bytes	Channel Address	Channel Description
1	E6_IN_1		UINT	2	%IW0	Slave1-16 online status
2	E6_IN_2		UINT	2	%IW2	Slave17-32 online status
3	E6_IN_3		UINT	2	%IW4	Slave33-48 online status
4	E6_IN_4		UINT	2	%IW6	Slave49-64 online status

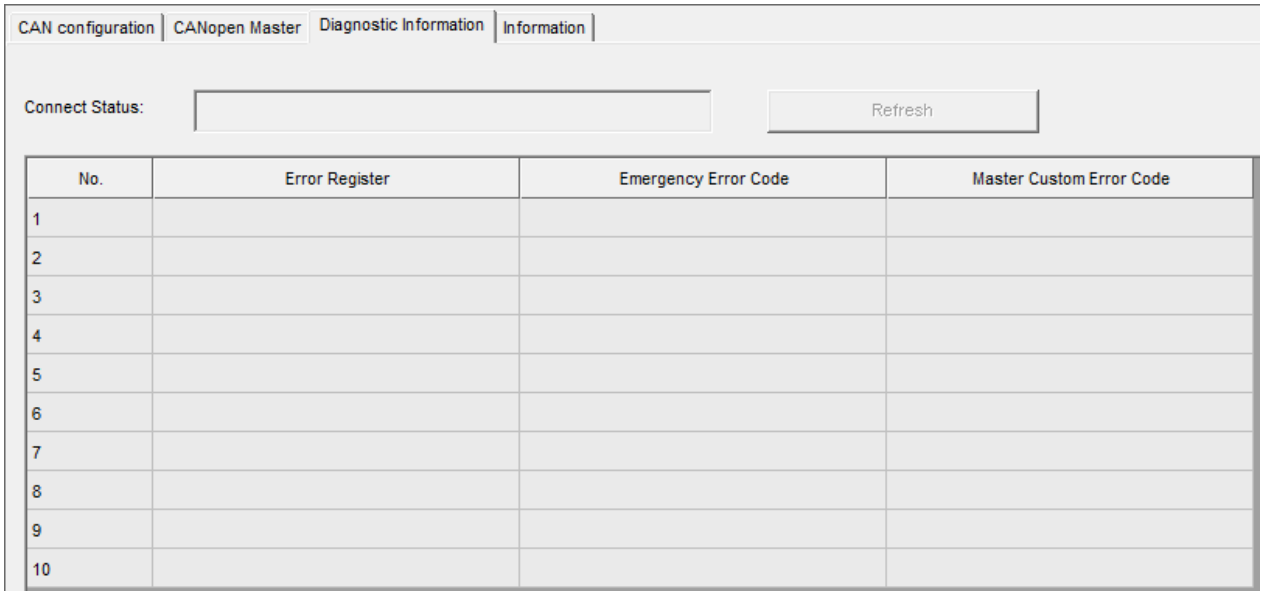
Fault Diagnosis Description

Parameters	Description
Slave1-16 online status	Slave station online status: Bit0 ~ Bit15 correspond to stations 1-16 (the station numbers here refer to the AT configuration sequence, not the slave station addresses): • 1: Indicates the slave station is online • 0: Indicates the slave station is offline or not configured
Slave17-32 online status	Slave station online status: Bit0 ~ Bit15 correspond to stations 17-32 (the station numbers here refer to the AT configuration sequence, not the slave station addresses): • 1: Indicates the slave station is online • 0: Indicates the slave station is offline or not configured
Slave33-48 online	Slave station online status: Bit0 ~ Bit15 correspond to stations 33-48 (the station numbers here

status	refer to the AT configuration sequence, not the slave station addresses): • 1: Indicates the slave station is online • 0: Indicates the slave station is offline or not configured
Slave49-64 online status	Slave station online status: Bit0 ~ Bit15 correspond to stations 49-64 (the station numbers here refer to the AT configuration sequence, not the slave station addresses): • 1: Indicates the slave station is online • 0: Indicates the slave station is offline or not configured

Double-click the CANopen master or slave node, then click the "Diagnostic Information" tab to view the diagnostic information of the master or slave station. The diagnostic information reads the master/slave connection status and reports the error register, emergency error code, and master-defined error codes. There are 10 sets of information in total, group 10 shows the current diagnostics. Groups 1–9 record historical diagnostics, logged in chronological order of reporting.

Click the "Refresh" button to update the diagnostic information.



Master Diagnostic Information

Diagnostic Data	Diagnostic Information Description
Connection Status	The operating status of the CANopen master: 0: Initializing 1: Offline 2: Connecting 4: Stopped 5: Running 0x7F: Pre-Operational 0xF: Unknown
Error Register	Diagnostic reporting registers.
Emergency Register	Refer to Master Station Emergency Error Codes.
Custom Error Codes for Master	Refer to Master Custom Error Codes.

Slave Diagnostic Information

Diagnostic Data	Diagnostic Information Description
Connection Status	The current operating status of the slave station: 0: Initializing 1: Offline 2: Connecting 4: Stopped

	5: Running 0x7F: Pre-Operational 0xF: Unknown
Error Register	Diagnostic reporting registers.
Emergency Register	Defined by the slave station manufacturer.
Custom Error Codes for Master	Refer to Master Custom Error Codes.

16.4.1 Error Codes

Master Station Emergency Error Codes

Error Register	Emergency Error Code	Master Custom Error Code
0x0	0x0: Error reset	0
0x11	0x8130: Slave offline	Slave's nodeID

Master Custom Error Codes

Error Code	Description
0x00	No error.
0x01	NodeID error.
0x02	Index error.
0x04	Write object dictionary parameter length error.
0x05	NMT command word error.
0x06	EtherCAT bus read/write error.
0x1105	Object dictionary read/write operation - slave no response.
0x1106	CAN bus error passive state.
0x1107	Master information storage memory error.
0x1108	Master state Eerror.
0x1109	Master configuration data error.
0x110A	Master initialization error.
0x110B	Master object dictionary read/write error.
0x110C	Slave node does not exist.
0x110D	Slave state error.
0x110E	Slave configuration return error.
0x110F	Slave PDO parameter error.
0x1110	Slave SDO read/write return error.
0x1111	Slave SDO read/write parameter error.
0x1112	CAN bus disconnected.
0x1113	CAN bus reconnection restored.
0x1114	CAN bus data transmission failed.
0x1115	CAN bus receive overflow.
0x1116	Slave device type mismatch error.
0x1117	Slave vendor code mismatch error.
0x1118	Slave product code mismatch error.
0x1119	Slave product version mismatch error.
0x111A	Slave heartbeat exception.
0x111B	The total number of RPDOs exceeds 256.
0x111C	The total number of TPDOs exceeds 256.
0x111D	The total number of SDOs exceeds 8192.
0x111E	The number of slave PDOs exceeds 64.
0x111F	The number of slave SDOs exceeds 1000.

SDO Error Codes

Error Code	Description
0x05030000	Toggle bit not alternated.
0x05040000	SDO protocol timed out.
0x05040001	Client/server command specifier not valid or unknown.
0x05040002	Invalid block size (block mode only).
0x05040003	Invalid sequence number (block mode only).
0x05040004	CRC error (block mode only).
0x05040005	Out of memory.
0x06010000	Unsupported access to an object.
0x06010001	Attempt to read a write only object.
0x06010002	Attempt to write a read only object.
0x06020000	Object does not exist in the object dictionary.
0x06040041	Object cannot be mapped to the PDO.
0x06040042	The number and length of the objects to be mapped would exceed PDO length.
0x06040043	General parameter incompatibility reason.
0x06040047	General internal incompatibility in the device.
0x06060000	Access failed due to an hardware error.
0x06070010	Data type does not match, length of service parameter does not match.
0x06070012	Data type does not match, length of service parameter too high.
0x06070013	Data type does not match, length of service parameter too low.
0x06090011	Sub-index does not exist.
0x06090030	Value range of parameter exceeded (only for write access).
0x06090031	Value of parameter written too high.
0x06090032	Value of parameter written too low.
0x06090036	Maximum value is less than minimum value.
0x08000000	General error.
0x08000020	Data cannot be transferred or stored to the application.
0x08000021	Data cannot be transferred or stored to the application because of local control.
0x08000022	Data cannot be transferred or stored to the application because of the present device state.
0x08000023	Object dictionary dynamic generation fails or no object dictionary is present (e.g. object dictionary is generated from file and generation fails because of an file error).

Chapter 17 Network Variable

17.1 Introduction

Network variables are used for variable data exchange between different controllers, supporting the simultaneous transmission of variable data from one controller to multiple other controllers. Has the following characteristics:

- Supports up to 32 network variable group senders.
- Supports up to 32 network variable group receivers.
- Each network variable group supports a maximum of 512 network variables, with a total variable byte count not exceeding 512 bytes.
- Support simultaneous deployment of sender and receiver.
- The network type supports UDP unicast and broadcast transmission, with a default port number of 48040.
- Support package variable transmission, package and transmit variable data within network variable groups, reduce the number of Ethernet transmission messages, and improve business performance.
- Support transmission verification, where the sender calculates a checksum for each data packet. The receiving end checks the checksum and discards packets that do not match the checksum.
- Support transmission confirmation, with the receiving end replying with a confirmation message for each received data packet.
- The sender supports setting transmission modes (loop transmission, change transmission, event transmission).
- The minimum support for cyclic transmission interval is 1ms, and the minimum support for variable transmission interval is 1ms.
- Event transmission supports associating BOOL variables or complex type BOOL sub variables in the project, and is only transmitted when the BOOL variable is TRUE.
- Supports IEC basic data types and basic type arrays, as well as structures and arrays that do not include FB and unions. FB, unions, and other data types that include these types are not supported.
- Provide diagnostic information for network variable groups.

17.2 Precautions

- To ensure data consistency, the IEC task only updates the network variable receiver data before the

start of each cycle, and updates the network variable transmitter data after the end of each cycle. Therefore, the actual transmission and reception cycle of network variable data is determined by the protocol configuration and the IEC task cycle.

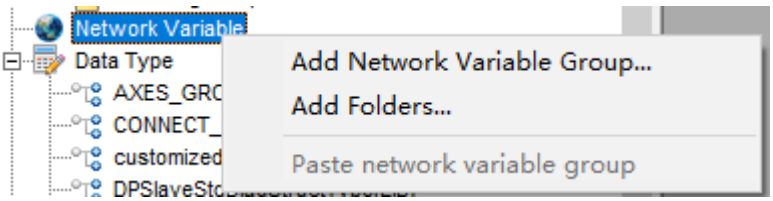
- When the amount of data is large or the system load is heavy, the actual transmission cycle of network variable data may not meet the configured configuration.
- Network variables belong to private protocols, and their implementation may not be completely the same among manufacturers. Therefore, LX series controllers do not support network variable communication with other models of controllers.
- The minimum transmission interval for the network variable group associated with a full-speed task is 1 ms.
- When network variables are used exclusively for LXS FSoE communication, it is recommended to associate them with a MotionTask; for other application scenarios, it is recommended to associate them with a normal IEC task.

17.3 Adding Network Variable Group

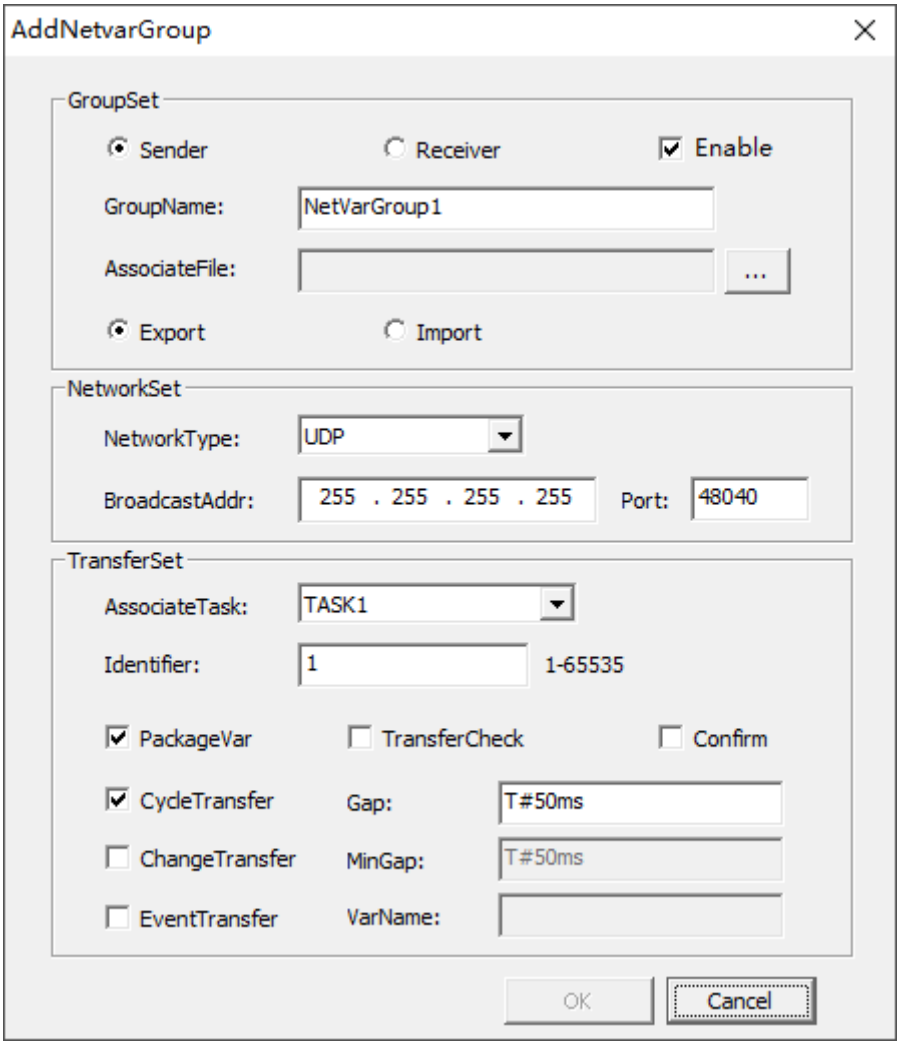
17.3.1 Adding sender network variable group

To add a sender network variable group, please follow these steps:

(1) In the engineering management tree, right-click on the "Network Variables" node and click "Add Network Variable Group".



(2) The "Add Network Variable Group" dialog box will pop up.



Sender parameter configuration:

- GroupSet
  - Enable: Set the flag for whether the network variable group should send or receive. On versions that do not support network variables, the "Enable" button is not selected by default and cannot be enabled. If not selected, the variable group will not be included in the network variable group download file and will be used as a regular global variable group in the project. Switch import and export properties without changing the enabled state of the "Enable" option and its checked state.
  - AssociateFile: Associate the variable group with a disk file. You can choose an existing file or manually enter a file name to create a new file.
  - Export: You can export the configuration and variable list of the current sender network variable group to an associated file.
  - Import: The system creates a sender network variable group based on the associated file, without importing the configuration information of the variable group during creation, and supports editing after creation. When importing repeatedly, the saved variable group information will not be changed.
- NetworkSet

- NetworkType: Only supports UDP transmission.
- BroadcastAddr: The destination IP address for sending UDP packets by the network variable sender, with a default value of 255.255.255.255, which means that data will be exchanged with all network units. When the target subnet is clear, it is recommended to modify the broadcast address to the subnet broadcast address of the specified network segment (for example, when communicating with all 192.168.0. x network segment controllers through the P1 network port, the broadcast address should be entered as 192.168.0.255).

  When the broadcast address and the IP address of the controller belong to different subnets, it is necessary to configure routing information for the broadcast address in the "Network Operation" tab of "Tool > Asistant tool > Controller Operation", otherwise the network variable group may fail to send.
- Port: The default UDP port number for the sender to send UDP packets is 48040. Be careful to avoid duplicating UDP port numbers with other applications.

■ TransferSet

- AssociateTask: Can be associated with tasks in the configuration other than motiontask.
- Identifier: 1-65535. The identifiers of all network variable groups within the controller cannot be duplicated. If there is an input error when the focus leaves, it will revert back to the previous correct value. If the variable group is not enabled, the identifier of the variable group will not be checked during compilation.
- PackageVar: It is recommended to select package variables to reduce the pressure on the controller system and improve performance. When checked, the network variable sender packages and sends all variables within the group in groups to minimize the number of messages to be sent and improve performance. When unchecked, the controller generates a separate message for each variable to be sent.
- TransferCheck: The sender provides a checksum for each message. The receiver checks the checksum and does not accept data messages with mismatched checksum to ensure the correctness of variable data.
- Confirm: The receiver sends a confirmation message to the sender for each received data packet.
- CycleTransfer: The sender sends network variable data in a loop for a period of time. When checked, the time interval for loop transmission can be defined. Definition example: "T # 50ms". Because the LX controller sends network variable data during the IEC task cycle, due to the jitter of IEC task, when the cyclic transmission interval of the network variable group is equal to the IEC task cycle, it cannot be guaranteed that the IEC task will send network variable data for each cycle, it is not recommended to set the loop transmission time to the same time as the IEC task cycle. To ensure data consistency, the IEC task only updates the network variable sender data after the end of each cycle. Therefore, the actual sending cycle of network variable data is a time greater than the set time interval and an integer multiple of the IEC task cycle (periodic task) (for example, if the network variable loop transmission interval is 50ms and the associated task cycle is 20ms, then the actual sending cycle of network variable data is 60ms).
- ChangeTransfer: Variable data is only sent when the value of the variable changes. The 'minimum interval' can be used to define the shortest time between two transmissions, for example: 'T #

50ms'.

- EventTransfer: Send variables when the value of the specified variable is TRUE, only BOOL variables or BOOL member variables in complex data types are supported.

- During compilation, it is necessary to perform error checks on the above information. If there are any exceptions, an error will be reported.

(3) Click ... on the associated file in the "Add Network Variable Group" dialog box, and select the associated file (with a suffix of. nvg) to import or export in the pop-up window. You can also create a. nvg file in the file name. Click to open, the associated file has been successfully added.

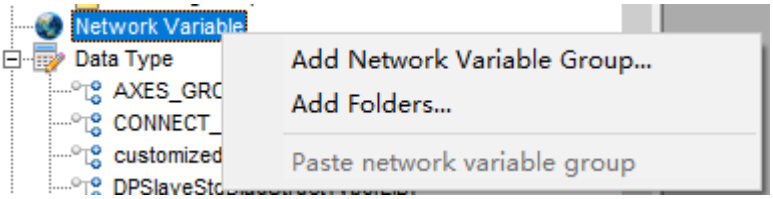
(4) Click 'OK' to successfully add the sender network variable group.

The sender icon  is , if the group is not enabled, the icon will be grayed out.

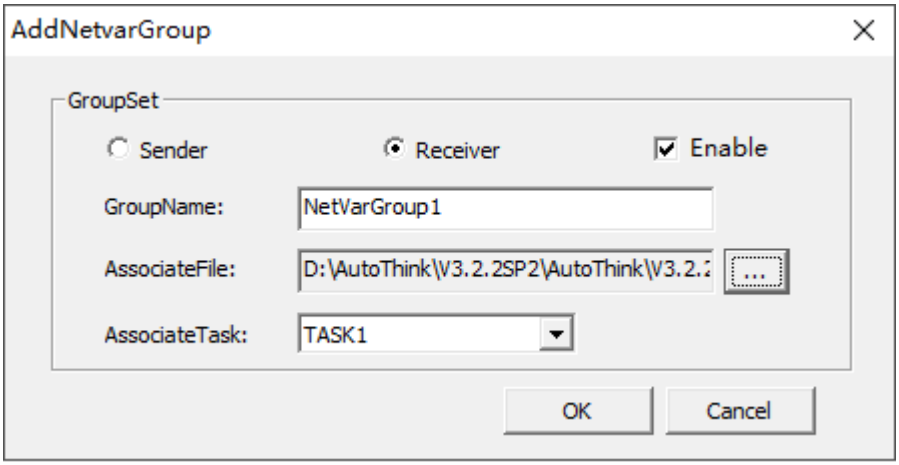
17.3.2 Adding receiver network variable group

To add a recipient network variable group, please follow these steps:

(1) In the engineering management tree, right-click on the "Network Variables" node and click "Add Network Variable Group".



(2) In the "Add Network Variable Group" dialog box, change the group settings to "Recipient" and select the associated file. Click OK.



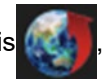
Receiver parameter configuration:

- When adding a receiver in the add network variable dialog box, the dialog box interface is as

shown in the figure above. The sender and receiver interfaces switch according to the user's selection of the radio button.

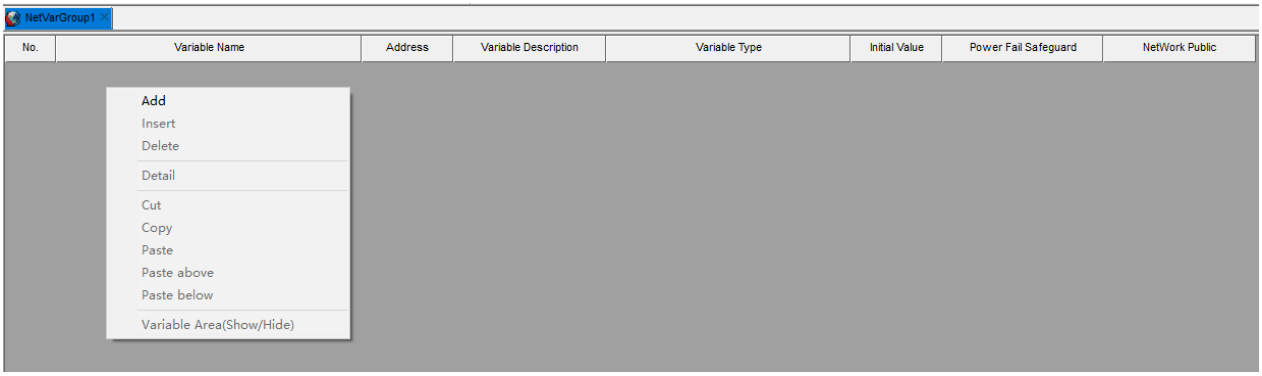
- The GroupName and AssociateTask in the receiver interface are consistent with those of the sender.
- AssociateFile: The system creates a recipient network variable group based on the associated file, and imports the configuration information of the variable group during creation. Editing is not supported after creation. When importing repeatedly, the same variables will be overwritten.

(3) Click 'OK' to successfully add the recipient network variable group.

The recipient icon is  , if the group is not enabled, the icon will be grayed out.

17.4 Editing Network Variable Group

Double click the network variable group that needs to be edited, and in the opened network variable group editing interface, right-click the mouse to perform operations such as adding, inserting, and deleting variables. The method is the same as for global/local variables, and will not be repeated here.



When editing a network variable group, it is important to note:

- If the variable group is the recipient or sender and the associated file is imported, the variable group interface does not allow variable editing, and all variables are automatically generated based on the associated file.
- When importing a file, if the variable contains a direct address, the direct address information is not within the import range, and the variable defaults to the G area. The power-off attribute is consistent with the exported variable in the file.
- In the same network variable group, use the variable number as the variable identifier:
  - Two variables with the same serial number, if their variable type or power failure protection attribute changes, are considered new variables and will re-enter the initialization list. If other attributes are different, only the variable attributes will be updated.
  - If there are only old variables and no new variables with the same serial number, it is considered

that the variable has been deleted.

- If there are only new variables and no old variables with the same serial number, it is considered that the variable has been added. It will re-enter the initialization list.

17.5 Network Variable Management

17.5.1 Add Folders

To add a network variable folder, please follow these steps:

(1) Right click on the "Network Variable" node and select "Add Folders".

(2) Enter the folder name and click the 'OK' button.

17.5.2 Delete network variable group

To delete a task, please follow these steps:

(1) In the project management tree, right-click on the task node, click "Delete network variable group", and a confirmation dialog box for deletion will pop up.

(2) Click "Confirm" to delete the network variable group.

17.5.3 Copy and paste network variable groups

To copy and paste a network variable group, please follow these steps:

(1) In the engineering management tree, right-click on the network variable group name and click "Copy Network Variable Group".

(2) Right click on the target network variable group and select "Paste Network Variable Group".

(3) The network variable group is pasted into the selected position.

- When pasting, it is necessary to determine the same name for the group name and variables within the group. If the group name or variable name overlaps with an existing name in the project, it should be renamed according to the rules. The default renaming of the group name is "NewNetVarGroupN", where "N" is a number. Starting from 1, it is compared with all names one by one until a variable group name that does not have the same name is found, which is the current pasted network variable group name. The variable name is assumed to be "nN" by default, where "n" is the default character in the network variable name and "N" is a number starting from 1.

17.5.4 Rename network variable group

To rename a network variable group, please follow these steps:

(1) In the engineering management tree, right-click on the network variable group name and click "Rename".

(2) Pop up the "Rename" dialog box.

(3) Enter the new network variable group name in the new name box and click the "OK" button.

17.5.5 Synchronize files

Mainly used to synchronize information between network variable groups and nvg file contents when there are changes.

1. Manual synchronization

   Manual synchronization is mainly used when creating a network variable group, importing variables through associated files, and manually clicking the menu item "Synchronize Files" after editing the network variable group

   To manually synchronize network variable groups, please follow these steps:

   - In the engineering management tree, right-click on the network variable group name and click "Synchronize Files".

2. Automatic synchronization

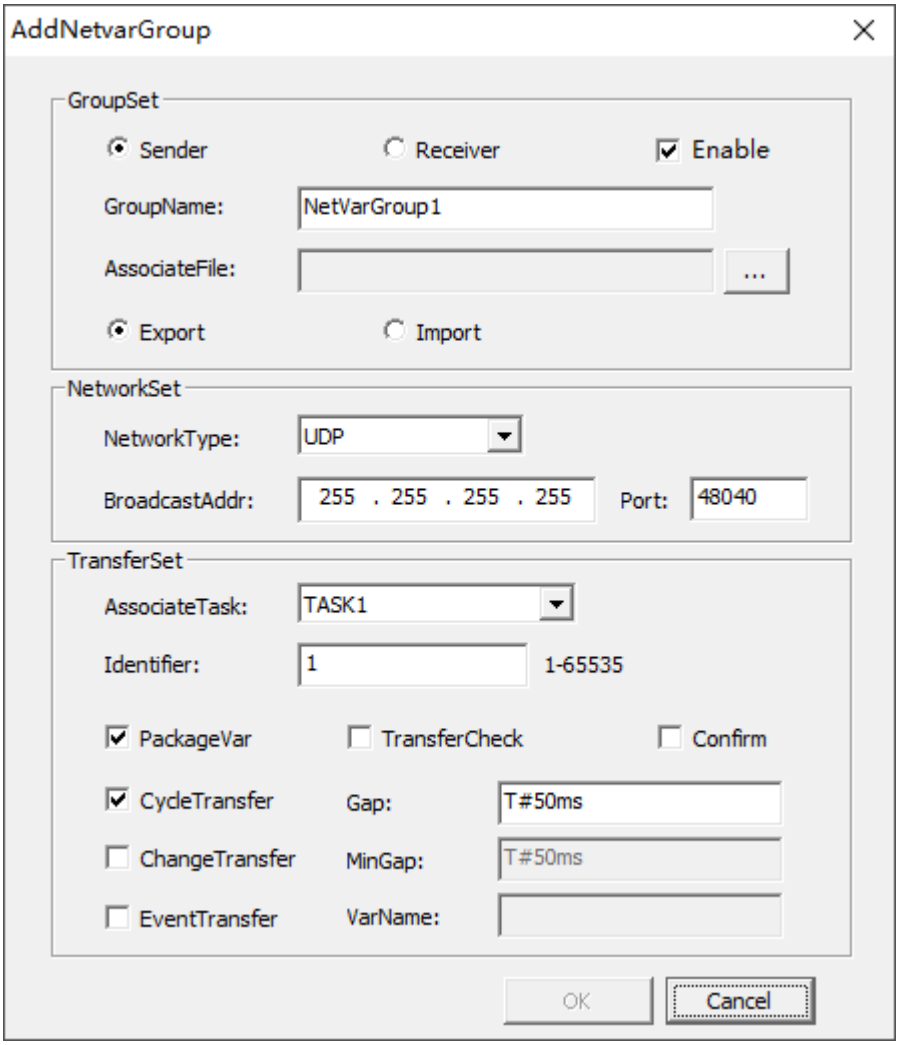
   Automatic synchronization refers to the mandatory synchronization of network variable groups and nvg file contents during compilation, ensuring strict consistency of downloaded files.

17.5.6 Attribute

To modify the network variable group name property, please follow these steps:

(1) In the engineering management tree, right-click on the network variable group name and click on "Properties".

(2) Pop up the dialog box for network variable group name properties, as shown in the following figure.



17.6 Diagnostic instructions

Explanation of Network Variable Group Diagnostic Information:

Diagnostic Name	Diagnostic Description	Diagnostic Information Description		Possible Causes or Solutions
NetVarGroupN_SendSuccessCounter	N Send Success Counter for Network Variable Group N	-		-
NetVarGroupN_SendErrorCounter	N Send Failure Counter for Network Variable Group N	-		-
NetVarGroupN_SendLastErrorNumber	N Send Failure Error Code for Network Variable Group N	0	Send successful	-
		100	Socket send failed	-
NetVarGroupN_ReceiveSuccessCounter	N Receive Success Counter	-		-

	for Network Variable Group N			
NetVarGroupN_ReceiveErrorCounter	N Receive Failure Counter for Network Variable Group N	-		-
NetVarGroupN_ReceiveLastErrorNumber	N Receive Failure Error Code for Network Variable Group N	0	No fault	-
		2	Invalid received length	1. Verify that the variable information in the sending group and receiving group is consistent. 2. Check for duplicate destination ports among all sending groups in the current network topology. 3. Re-export the associated .nvg file from the sending group, re-import it into the receiving group, then compile and download. 4. Test using a minimal system: directly connect the P1 or P2 Ethernet ports of the sending and receiving modules with an Ethernet cable to check if the fault persists. 5. If the above measures are ineffective, contact the device manufacturer.
		3	Incorrect request ID in the request header	1. Check if the Identifier in the sending group's Transmission Settings matches the one in the receiving group. 2. Check if the destination port of any sending group in the network topology duplicates the port of the current receiving group. 3. Re-export the associated .nvg file from the sending group, re-import it into the receiving group, then compile and download.
		4	Incorrect transmission mode in the request header	Internal error, contact the device manufacturer.
		5	Incorrect transmission direction in the request header	
		6	Incorrect variable group identifier in the request header	
		7	Incorrect number of variables in the group in the request header	
		8	CRC error in the request header	1. Check if there are any modifications to variables in the sending group from an unassociated IEC task. 2. Re-export the associated .nvg file from the sending group, re-import it into the receiving group, then compile and download.

				3. Disable the 'Transmission Checksum' option in the sending group's Transmission Settings to check if the fault persists. 4. Check for duplicate destination ports among all sending groups in the current network topology. 5. If the above measures are ineffective, contact the device manufacturer.
		9	Incomplete fragment reception	1. Sending group configuration check: (1) If 'Packet Transmission' is not checked, enable it. (2) If 'Cyclic Transmission' is checked: Increase the transmission interval. (3) If 'On Change' transmission is checked: Increase the minimum interval. (4) If 'On Event' transmission is checked: Adjust the change frequency of the associated variable. Note: The priorities for Cyclic, On Change, and On Event transmission are triggered in descending order. The send operation will be triggered if any one condition is met. 2. Adjust the IEC task cycle associated with the sending group. 3. Check network latency: Use the ping command to test network latency between the sending and receiving modules, and check the load on network devices (switches, routers). 4. Test using a minimal system: directly connect the P1 or P2 Ethernet ports of the sending and receiving modules with an Ethernet cable to check if the fault persists.
		10	Received packets out of order	
		11	Received fragment packets out of order	
		12	Duplicate sequence number	1. If both Ethernet ports of the controller are connected to the switch, disconnect one port. 2. Check the network topology for duplicate sending group configurations (identical network configuration, transmission settings, and variable tables). 3. Test using a minimal system: directly connect the P1 or P2 Ethernet ports of the sending and receiving modules with an Ethernet cable to check if the fault persists.
		100	Transmission failed	1. Check if the switch used in the network topology is a Layer 3 switch (or router).

				2. Check if the switch has disabled broadcast packet forwarding or enabled VLAN. 3. Test using a minimal system: directly connect the P1 or P2 Ethernet ports of the sending and receiving modules with an Ethernet cable to check if the fault persists. 4. Power cycle the module to check if the fault persists. 5. If the above measures are ineffective, contact the device manufacturer.

Beijing HollySys Intelligent Technologies Co., Ltd..

Di Sheng Middle Road, No.2

Economic-Technological Development Area

100176 Beijing, China

Tel: 010-5898 1588

Hotline: 400-811-1999

Fax: 010-5898 1558

Web: http://www.hollysys.com

FA Web: https://fa.hollysys.com/