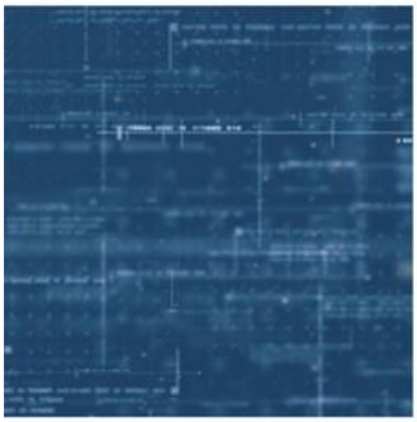
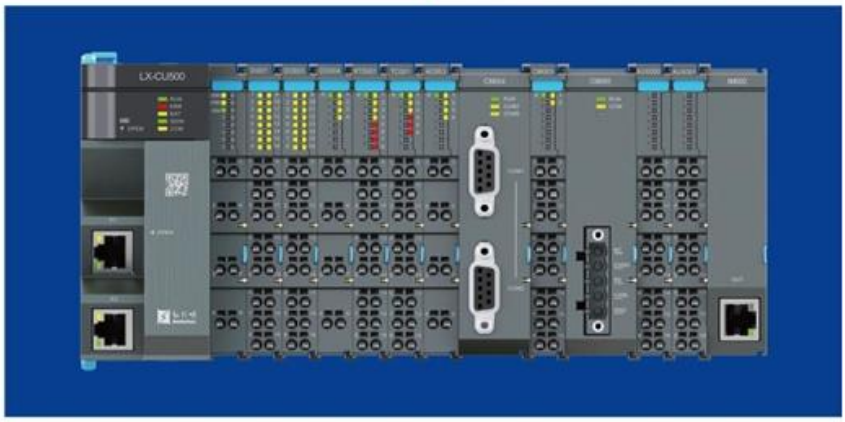
Programming Manual for LX Series Programmable Logic Controllers





版本：2025.03

Programming Manual for LX Series Programmable Logic Controllers

2025.03

V1.2.0

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
Hollysys Group Website: https://www.hollysys.com/
Factory Automation Business Website: 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 available manuals for this product are displayed in the table below. Please choose and refer to them based on your specific usage requirement.

Manual Name	Purpose	Content
Module Manual for LX Programmable Logic Controllers	Learn about the hardware modules of the LX system	Describes all hardware modules, including: Module functions Hardware interface Wiring Indicator Technical parameters
Communication Manual for LX Programmable Logic Controllers	Guide users to build the LX system communication network	Describe the communication networks supported by the LX system, including: Introduction to communication protocols and use of functions
Basic Instruction Manual for LX Programmable Logic Controllers	Learn about the use of basic instructions of the LX system	Describe the use of basic instructions, including: Instruction functions Parameter Example
Motion Instruction Manual for LX Programmable Logic Controllers	Learn about the use of motion control instructions of the LX system	Describe the use of motion control instructions, including: Instruction functions 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

2.2 Usage Convention

The following symbols are used in this manual:

- Tip: 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, kindly access it through our website.

- The official website of HollySys (https://cn.hollysys.com/other/download.html) to download the file.

Chapter 3 Revision History

Version	Revision Date	Revision Content
V1.0.0	July 2023	Created
V1.1.0	October 2023	Added the examples of LDS program and CFC programming
V1.1.1	November 2023	Modified the description of the option to install the USB driver, and the description of the 5.1.2 title bar
V1.1.2	December 2023	Changed illustrations
V1.1.3	August 2024	Added 10.3.12 Export Public Variables Table
V1.2.0	February 2025	New Additions: Multiple LX-LG901 Configuration Instructions Controller Backup Import and Export of Hardware Configuration Graphical Configuration Instructions Copy, Paste, and Cut for Hardware Configuration Support for Comment State Switching in LD/LDS Program Sections Associate IEC Variables Updates: DeviceNet Master/Slave Addition Method Update: When adding LX-CM005/LX-CM006, DeviceNet master/slave stations are automatically added.

Chapter 4 Software Installation

4.1 Requirements for Installation Environment

To install FA-AutoThink software, it is necessary to meet the following requirements for the software and hardware environments:

Environment	Type	Requirement
Hardware Environment	USB interface	At least one USB 2.0 interface
	Graphics card	Resolution 1920 ×1080 or above supported
	CPU	Intel Core i5 standard edition or above
	Memory	8 GB or above
	Hard disk	10 GB or above
Software Environment	Operating system	Windows Server 2012 or above Windows 7 Windows 10 Windows 11
	Utility software	Microsoft Office Excel 2003 or above

4.2 Installation

4.2.1 Introduction

The following installation is described using AutoThink V3.2.3 as an example.

4.2.2 Requirement

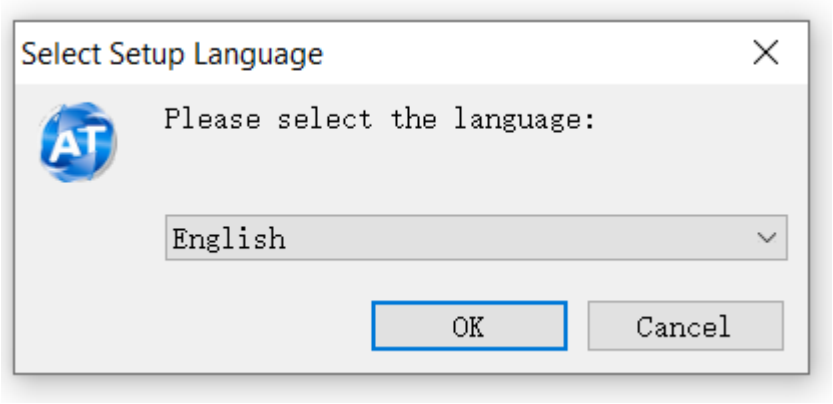
- No other applications occupying storage space.
- Storage space greater than 10 GB.

4.2.3 Steps

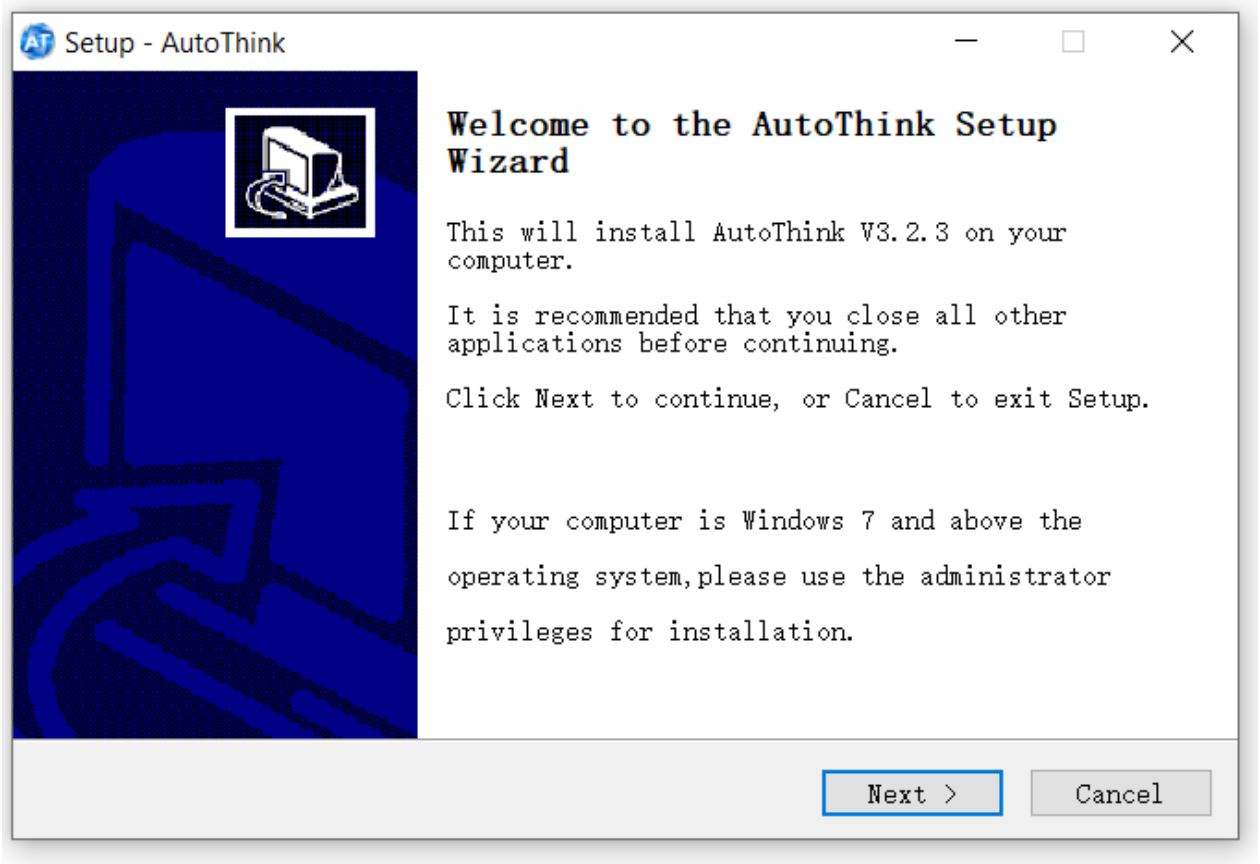
4.2.3.1 Select installation language

Double-click the installer 

. The **Select Installation Language** dialog box is displayed. In the dialog box, select **English** for the language, and click **OK**.

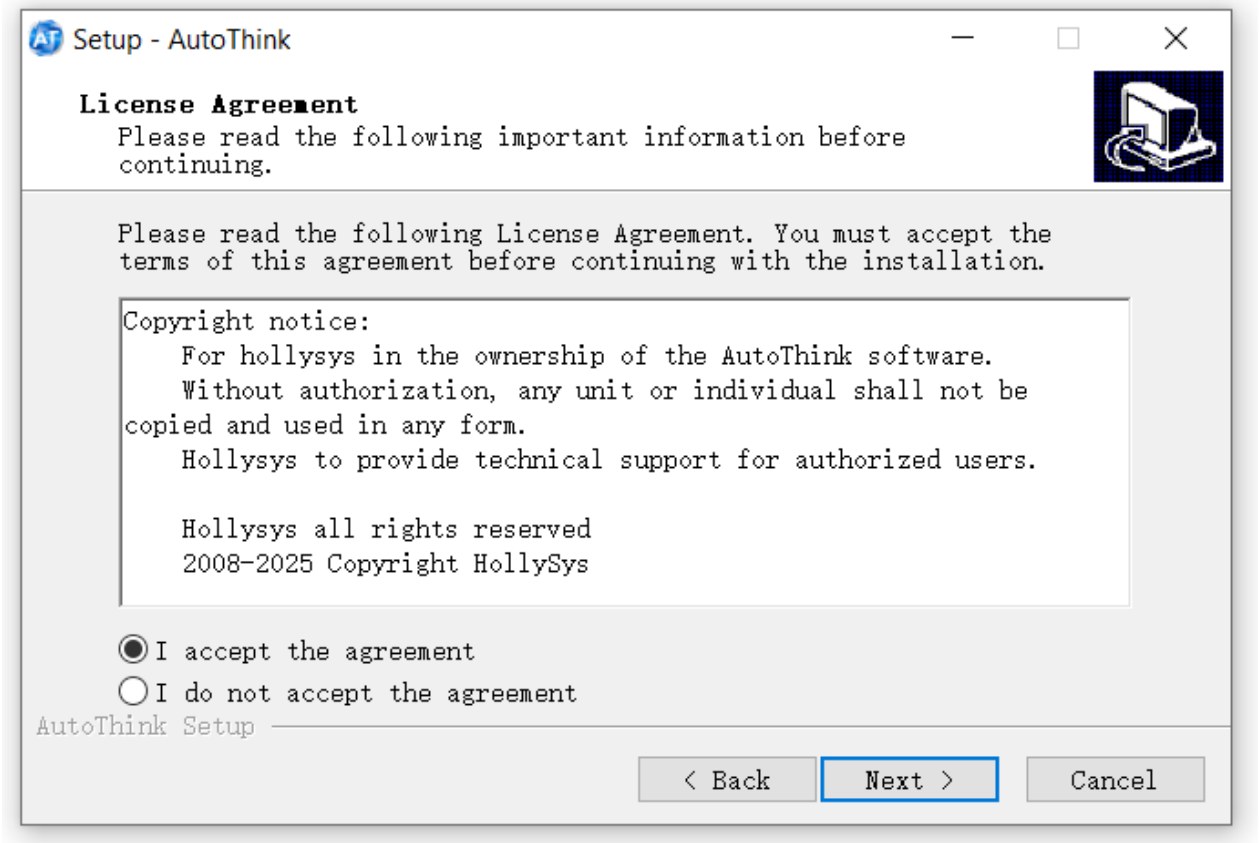


4.2.3.2 Start installation wizard



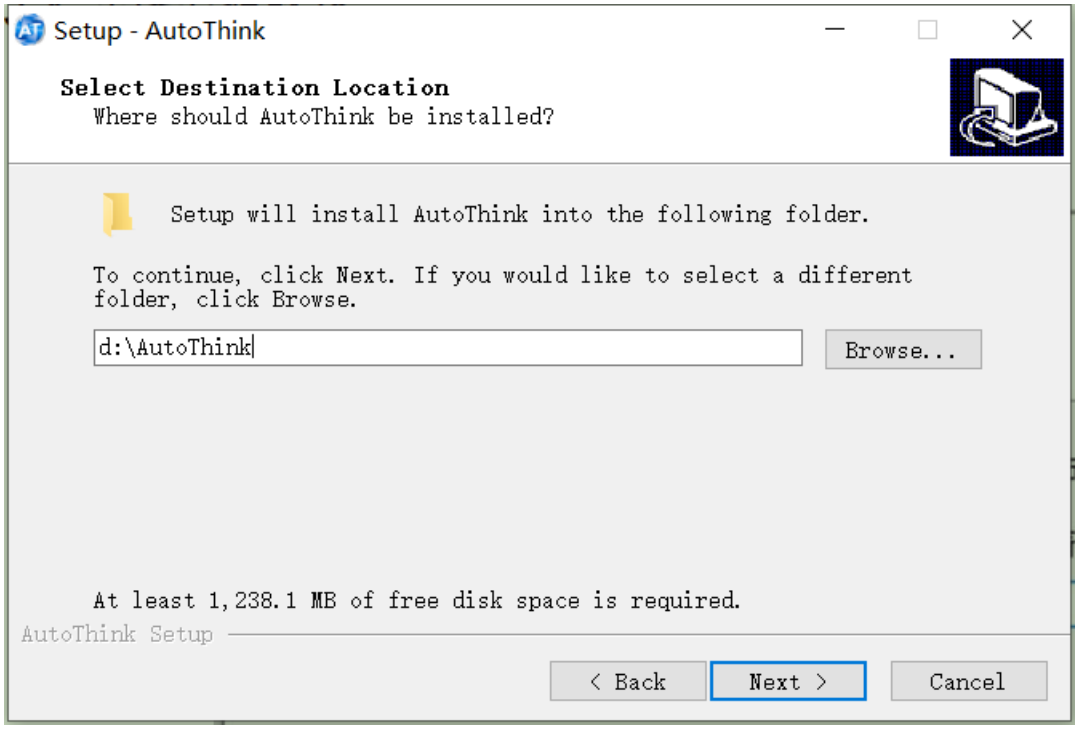
4.2.3.3 License agreement

Select **I accept the agreement (A)**, and click **Next (N)**.



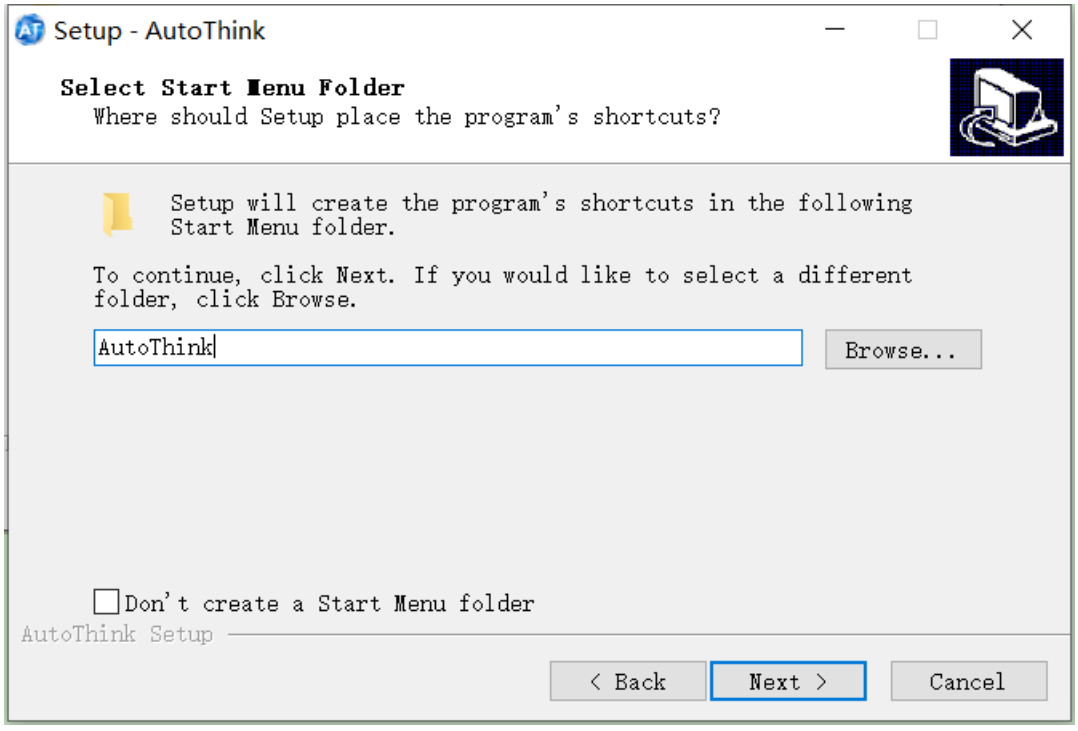
4.2.3.4 Select installation path

Click **Browse (R)...** to select installation path, and click **Next (N)**.



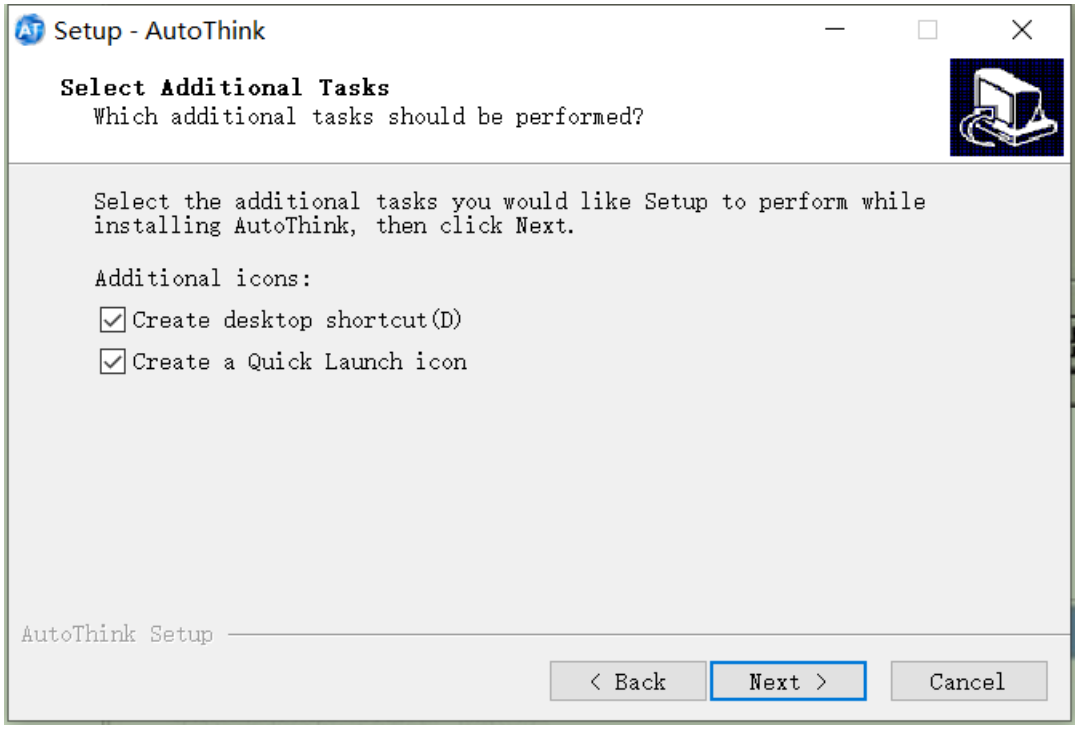
4.2.3.5 Select Start Menu folder

(1) Click **Browse (R)** ... to select the folder, and click **Next (N)**.



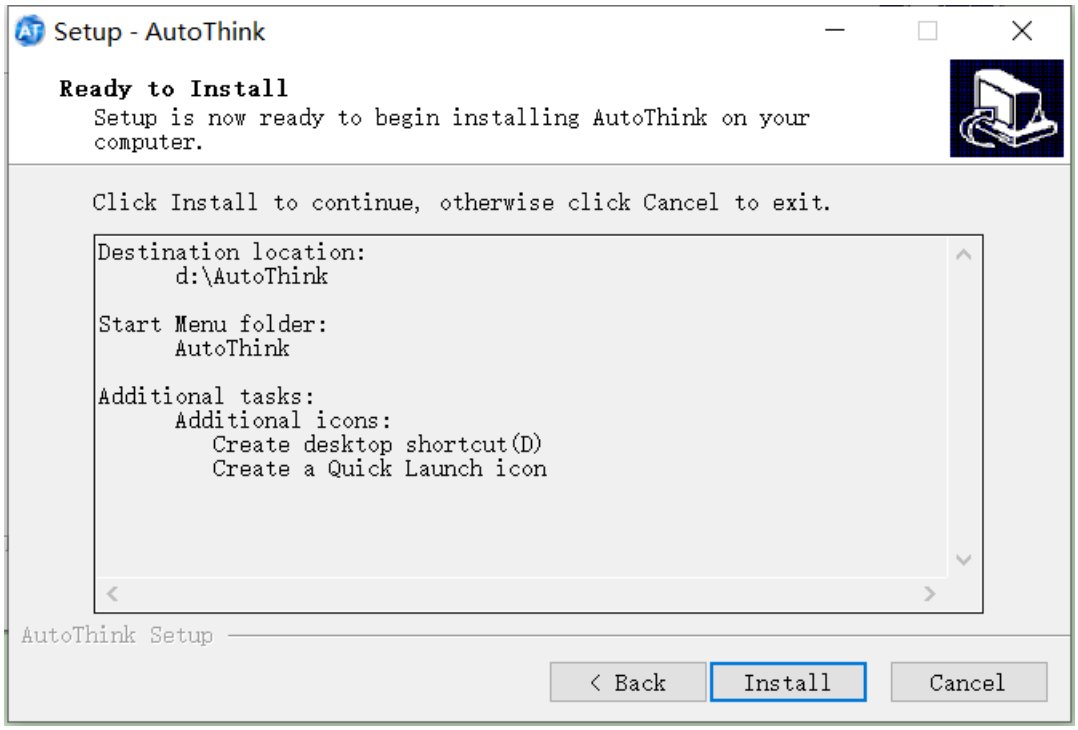
(2) Select additional tasks

Check the shortcut icons that need to be created, and click Next (N).



(3) Prepare for installation

After confirming the installation information, click **Install**. If you want to make changes, click .

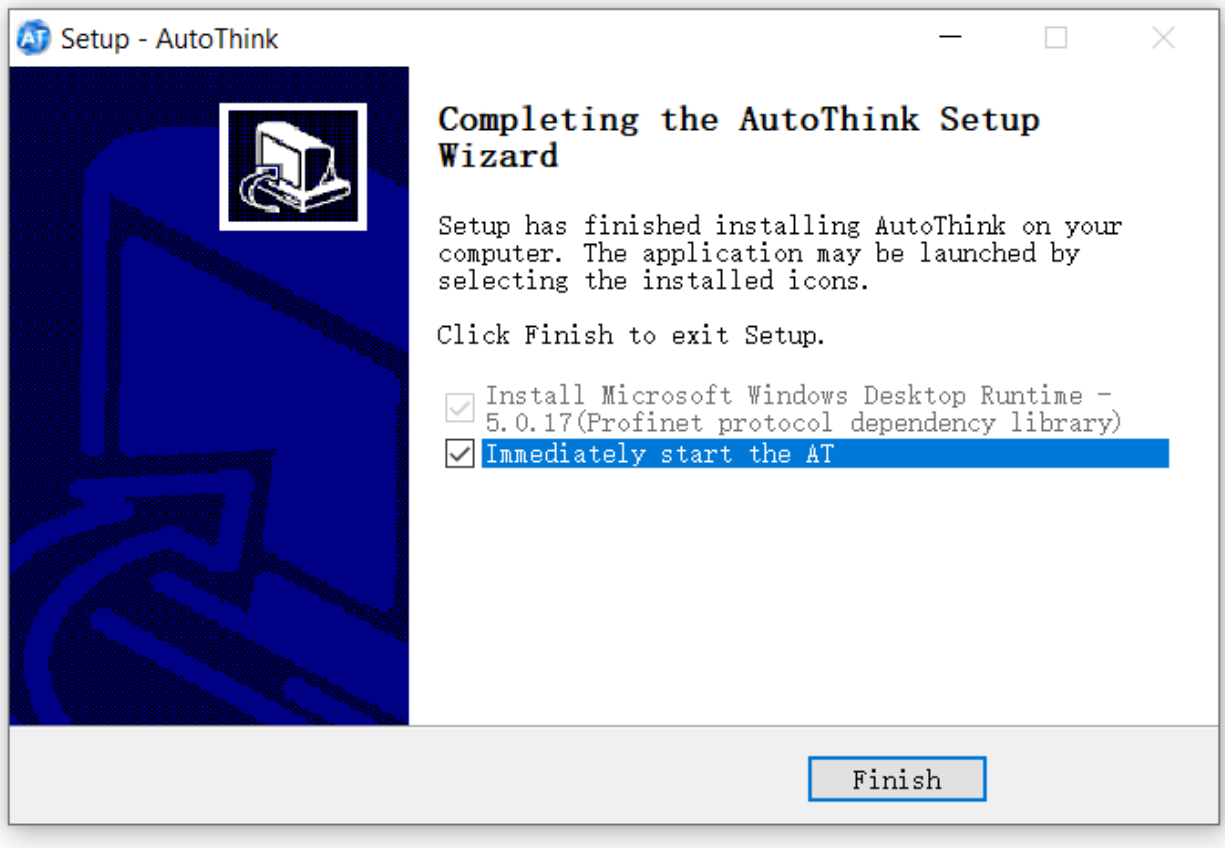


(4) Installation process

Installation progress is displayed during installation. To cancel the installation, click **Cancel**.

(5) Installation completion prompt

After the installation is complete, the AutoThink Installation Completed window is displayed.



Install Microsoft Windows Desktop Runtime-5.0.17(Profinet protocol depency library): In versions V3.2.3 and above, this option is set to mandatory by default. After clicking the 'Finish' button, an interface for installing the driver will pop up. It is recommended to install the driver; otherwise, the PN protocol will not be usable."

Start AutoThink immediately: After installation, if you need to run AutoThink software immediately, then check this option.

4.2.4 Result

By checking Start AutoThink immediately, the software startup interface is displayed, otherwise, no interface is displayed.

4.3 Uninstall

4.3.1 Steps

Uninstall AutoThink software via:

- On **Desktop**: Click **Start** menu > **All Programs** > **AutoThink** > **Uninstall AutoThink**.
- In **Control Panel**, uninstall it.

4.3.2 Result

Once the uninstallation is complete, AutoThink software is completely removed from the computer where it was installed.

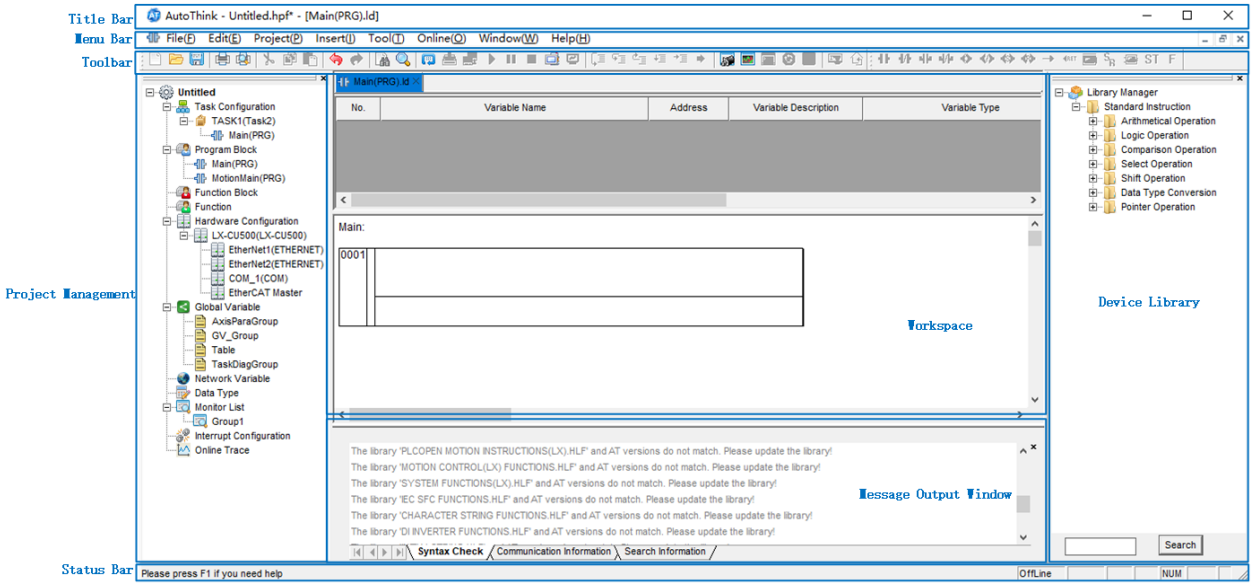
Chapter 5 Software Interface

5.1 Interface Composition

5.1.1 Software Interface View

5.1.1.1 Introduction

The programming interface of AutoThink software is composed as follows:



5.1.1.2 Reference message

- Title Bar
- Menu Bar
- Toolbar
- Workspace
- Project Management Window
- Message Output Window
- Status Bar

5.1.2 Title Bar

5.1.2.1 Introduction

In the Title Bar, there are the following items displayed from left to right: software logo (software icon and software name), project name (unsaved marking), and window control buttons (minimize, maximize, and close buttons).

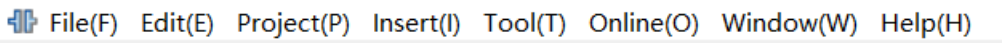
1. Software logo and project name;
2. Project name;
3. Unsaved marking;
4. Window control buttons.

For the modified project, the project name is displayed as **Project Name.hpf***. Where the project state * indicates that the modification has not been saved. After the modification is saved, * disappears.

5.1.3 Menu Bar

5.1.3.1 Introduction

The Menu Bar is below the Title Bar. Left-click each menu. In the drop-down menu, select each menu command item to perform operations related to algorithm configuration. Depending on the editing content of the workspace, the menus displayed in the Menu Bar are slightly different. Generally, the following groups of menus are included:



- **Icon**: Used to display the type of interface currently open, including: (CFC Language), (LD Language), (ST Language), (SFC Language), (Hardware Configuration), (Global Variables), (Network Variables), etc.
- **File**: Used to execute operations related to project files, such as open, save, and close.
- **Edit**: Used to perform operations on existing content, such as edit, replace, and find.
- **Project**: Used to perform operations on the project currently being edited, such as compile, setup, and view log.
- **Insert**: This menu is displayed when the program block is configured. It is used to insert elements in the graphical programming language.
- **Tools**: Used to provide various instruction wizards and usage wizards.

- **Online**: Used to provide commands related to the project's operations in the actual system or simulation system, such as install, online, and simulate.
- **Window**: Used to provide commands related to the operations on the currently opened window, such as arrange, switch display, and close.
- **Help**: Used to provide help information for the software.

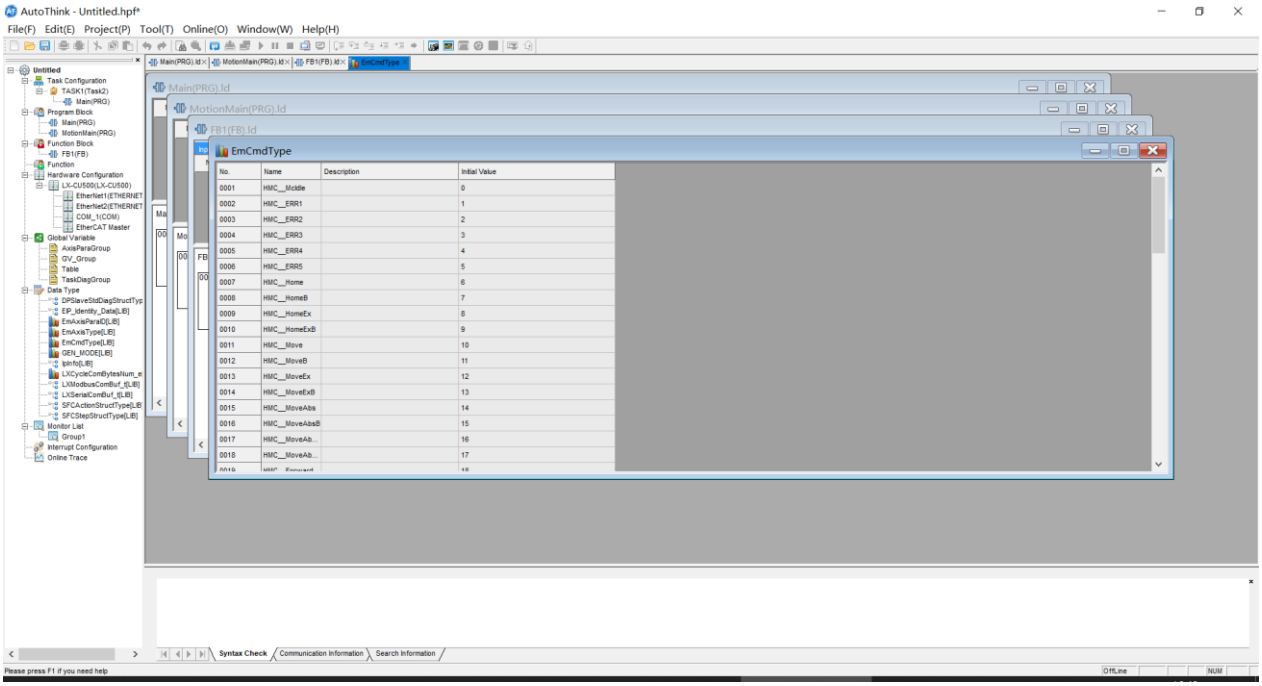
5.1.4 Toolbar

In the **Toolbar**, there are the following items displayed from left to right: Main Toolbar, Online Toolbar, ST Toolbar (available only in the ST Editor), Special Toolbar, and Input Assistant. Depending on the programming language, the programming language toolbar is displayed as LD Toolbar, ST Toolbar, and CFC Toolbar in different versions.

5.1.5 Workspace

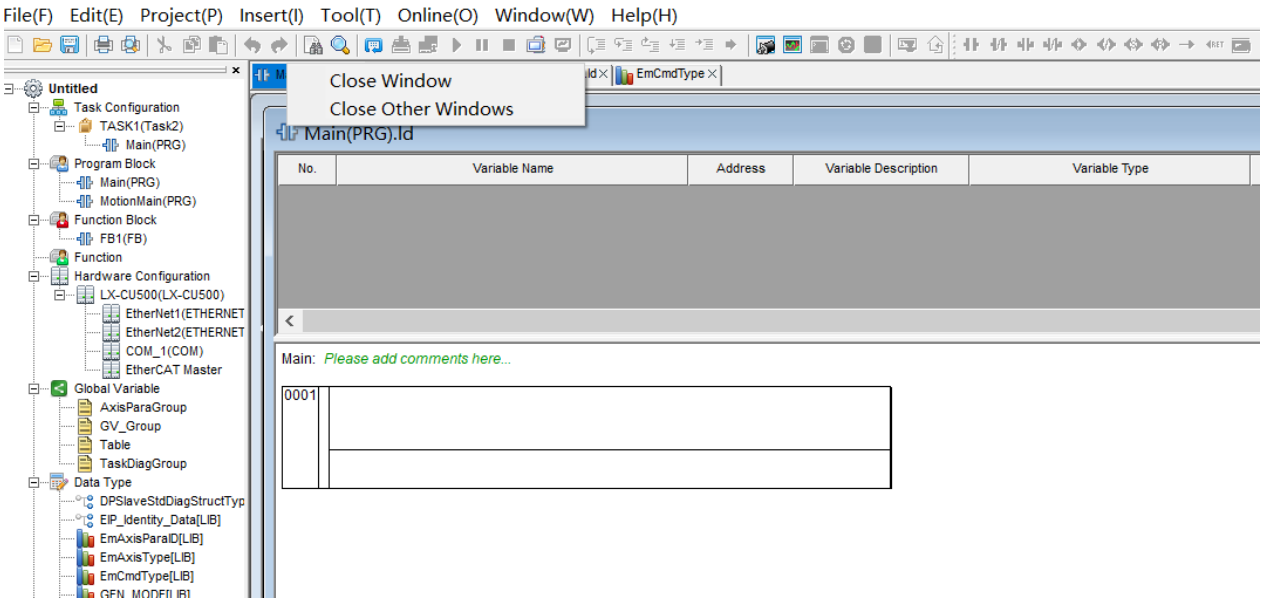
5.1.5.1 Introduction

The **Workspace** is a rectangular area used for specific operations such as variable definition, program editing, and hardware configuration, as marked in gray in the figure. Various editing windows are centrally displayed as tabs at the top of the **Workspace**. By clicking the tabs, you can access the corresponding editing windows.



In the **Workspace**, each window has a set of window control buttons in the upper right corner. By double-clicking the **Title Bar** of each window, you can switch the window between its floating and fixed state.

When the editing window is active, on the left side of the window's Title Bar, click the Window icon to display the window control drop-down menu, as shown in the figure below. This operation also switches and controls windows.



5.1.6 Project Management Window

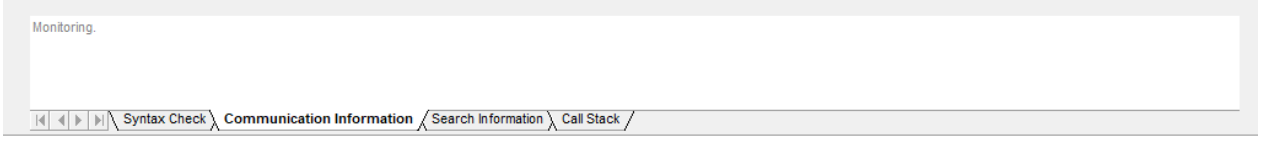
5.1.6.1 Introduction

Project management adopts a tree structure diagram for the management of the corresponding contents. This management method makes the whole project management with clear levels, clear structure and convenient management. The root node name of the Project Management Tree diagram is the name of the current project, which includes various sub-nodes below it: task configuration, program block, functional block, function, hardware configuration, global variables, data types, watch list, interrupt configuration, online tracking, etc.

5.1.7 Message Output Window

5.1.7.1 Introduction

The **Message Output Window** contains 4 tabs, **Syntax Check, Communication Message, Finding Message,** and **Call Stack**, which are used to display messages about compilation, error, warning, communication, lookup, and online call stack.



Double-click the line where the error message, warning message, or finding result is located to automatically locate the relevant position. Select the error message, and press **Shift + F9** or **F9** to quickly

locate it up or down. Click **Close** to close the **Message Output Window**.

When an exception occurs during operation, an exception message is indicated in red font in the **Message Output Window**.

In the **Message Window**, right-click to bring up the menu item Clear All Messages, which clears all the messages in the tab currently being displayed.

Hide/Show **Message Window**: In the **Window** menu, use the **Message Window** command to control whether to display the **Message Window**. After the Message Window is hidden, compiling and finding operations automatically load and display the **Message Window**.

In online mode, the **Call Stack** tab is visible. Users can view historical call stack information for the currently commissioned POU. See *Commissioning Modes* for details.

5.1.8 Status Bar

5.1.8.1 Introduction

The Status Bar is used to display information related to the current editing window, as shown in the figure.

1. Used to display help tips;
2. When editing a POU in ST language, the Status Bar displays information about the row number and column number where the current cursor is located;
3. Used to display the **Online + IP**, **Offline** and **Simulation** status of the project;
4. Reserved;
5. Reserved;
6. Used to show the uppercase state when the input method is in uppercase;
7. Used to display numbers;
8. Reserved.

5.1.9 Device Library

The window of Device Library displays all the module's models supported by current object configuration.

5.1.10 Interface in Online Mode

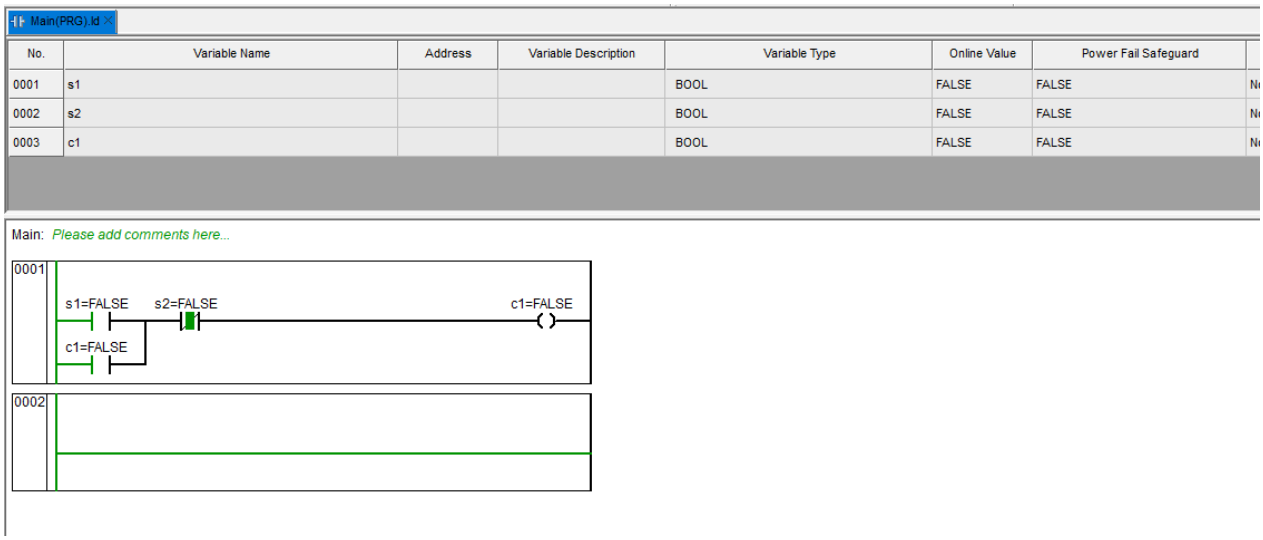
5.1.10.1 Introduction

In this mode, it is important to connect the hardware device and install the successfully compiled program into the main control through the correct channel address, and then the program can be commissioned.

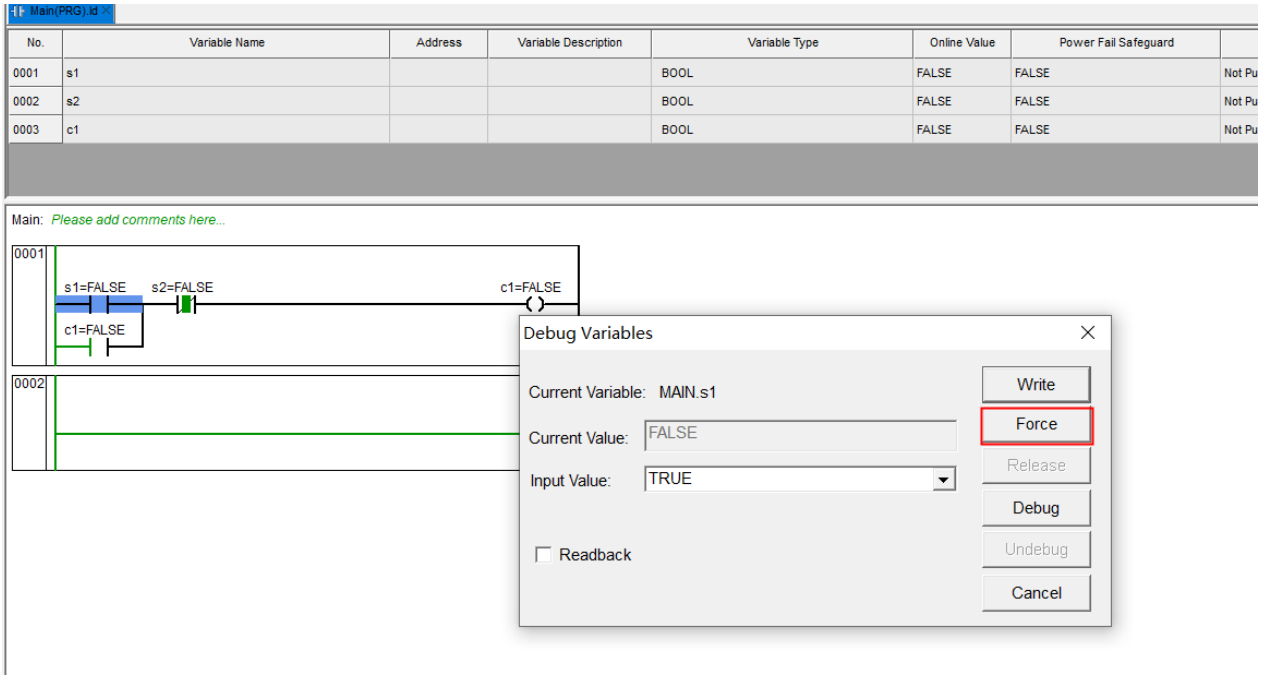
5.1.10.2 Reference message

To execute online operations, please follow the steps below:

(1) After installing the program to the CPU module, select the **Monitor** command to enter the monitoring state.

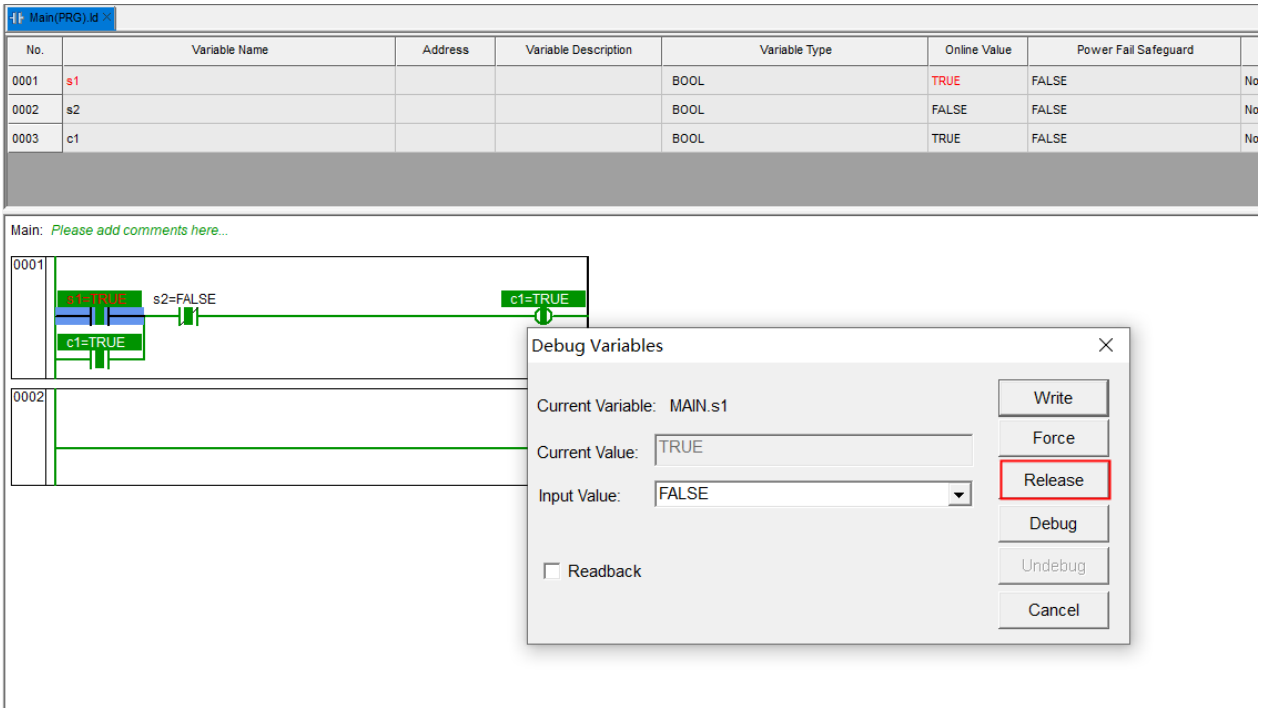


(2) Double-click the contact **S1**. In the pop-up **Commissioning Variables** window, preset S1 to **TRUE**, and click **Force** to determine the forced value.

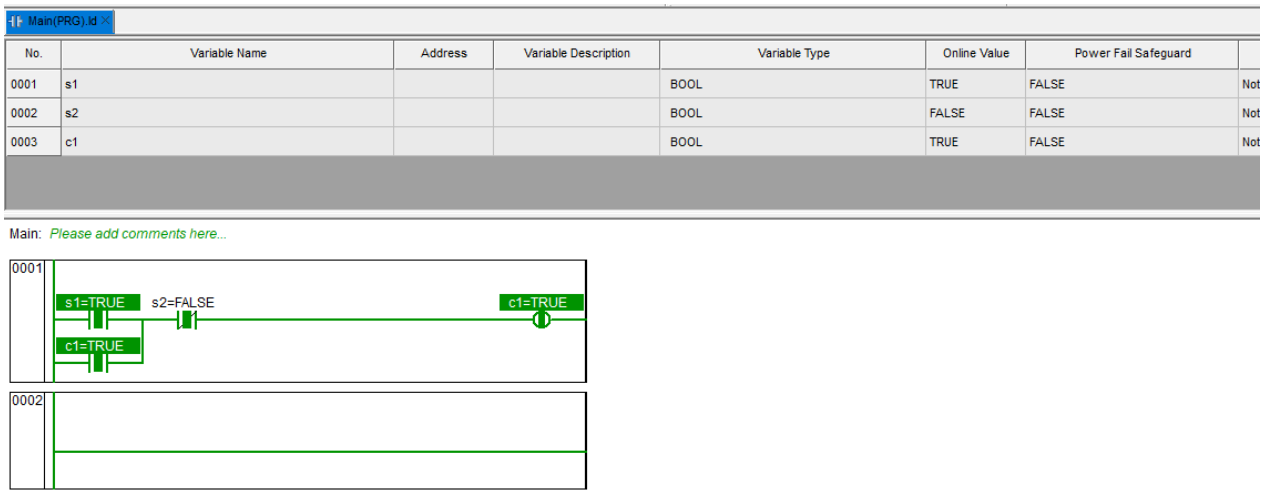


After forcing, S1 is displayed in red, and the **Online Value** is displayed as **TRUE** in its corresponding variables area.

(3) Double-click the contact **S1** again. In the pop-up window, click **Release** to release the S1 contact forced, as shown in the figure.



After the forcing is released, the **S1** contact changes from red to black, and the assignment item in the global variables area is also restored to black. The data shall be unforced after it has been forced. Otherwise, the forced data remains.



5.2 Shortcut Menu Commands

This section describes the shortcut keys in AutoThink configuration software. This helps users to better master the functions of AutoThink software and make it easy to use.

In AutoThink, the shortcut keys are divided into menu command shortcut keys and non-menu command shortcut keys. In addition, the shortcut keys are different for different function modules. A detailed description is given below.

5.2.1 Menu Command Shortcut Keys

1. General menu command shortcut keys

- Shortcut keys for offline editing

Function	Shortcut Key
File\New	Ctrl + N
File\Open	Ctrl + O
File\Save	Ctrl + S
File\Save as	F12
File\Print	Ctrl + P
Edit\Undo	Ctrl + Z
Edit\Restore	Ctrl + Y
Edit\Cut	Ctrl + X
Edit\Copy	Ctrl + C
Edit\Paste	Ctrl + V
Edit\Delete	Del
Edit\Find	Ctrl + F
Edit\Replace	Ctrl + H
Edit\Input Assistant	F2
Project\Compile	F11

- Shortcut keys for online editing

Function	Shortcut Key
Online\Monitor\Exit Monitor	Alt + F8\CTL+F8
Online\Single-step Commissioning	F10
Online\Jump	Ctrl+Shift+F8
Online\Jump In	Ctrl+F11
Online\Run at Breakpoint	Ctrl+F5
Online\Run	F5
Online\Stop	Shift + F8
Online\Write	Ctrl + F7
Online\Force	F7
Online\Release All	Shift + F7
Online\Show Forced Variable Table	Ctrl + Shift + F7

- LD menu command shortcut keys

Function	Shortcut Key
Insert\Contact (Append)	F3
Insert\Previous Section	Ctrl + M
Insert\Next Section	Ctrl + W
Insert\Parallel Contact	F10
Insert\Coil	F6
Insert\Block Element	Alt + F9
Insert\Invert	Ctrl + G
Set/Reset	Ctrl + T
Multiple Inputs (Add Block Pins)	Ctrl + D
Insert\Invert Contact	Alt + F3
Insert\Parallel Inverted Contacts	Alt + F10
Insert\Invert Coil	Alt + F6
Insert\Set Coil	Alt + R

Insert\Reset Coil	Alt + Ctrl +R

- CFC menu command shortcut keys

Function	Shortcut Key
Insert\Input Element	Ctrl + I
Insert\Output Element	Ctrl + U
Insert\Block Element	Ctrl + B
Insert\Jump Element	Ctrl + J
Insert\Return Element	Ctrl + R
Insert\Tag Element	Ctrl + L
Insert\Annotation Element	Ctrl + K
Insert\Invert	Ctrl + G
Insert\Set/Reset	Ctrl + T
Insert\Enable	Ctrl + E
Insert\Multiple Inputs	Ctrl + D

- SFC menu command shortcut keys

Function	Shortcut Key
Insert\Step-Transition (Forward)	Ctrl + T
Insert\Step-Transition (Backward)	Ctrl + E
Insert\Alternative Branch (Right)	Ctrl + A
Insert\Alternative Branch (Left)	Ctrl + D
Insert\parallel Branch (Right)	Ctrl + L
Insert\parallel Branch (Left)	Ctrl + M
Insert\Jump	Ctrl + J
Insert\Transition-Jump	Ctrl + R
Insert\Add Entry Action	Ctrl + I
Insert\Add Exit Action	Ctrl + G
More\Add parallel Branch Codename	Ctrl + Q
More\Add Action\Transition	Ctrl + W
More\Delete Action\Transition	Ctrl + U
More\Associated Action	Ctrl + K

5.2.2 Non-Menu Command Shortcut Keys

- General Non-menu command shortcut keys

Function	Shortcut Key
Help	F1
Modify Variable Attributes in the IEC View Area	Shift + F2
Exit Monitor	Ctrl + F8
Highlight Next Error Message	F9
Highlight Previous Error Message	Shift + F9
Automatically Declare Variables	Enter
Select All	Ctrl + A

- ST Keyboard Command Shortcut Keys

Function	Shortcut Key
Delete One Character Forward	Backspace
Delete One Character Backward	Delete
Next Line	↓
Previous Line	↑

Scroll to Next Line	Ctrl + ↓
Scroll to Previous Line	Ctrl + ↑
Select Downwards Line-by-Line	Shift + ↓
Select Upwards Line-by-Line	Shift + ↑
Cursor Jumps to End of Line	End
Cursor Jumps to End of Program	Ctrl + End
Select Entire Line after Cursor	Shift +End
Select Entire Program after Cursor	Ctrl + Shift + End
Cursor Jumps to Beginning of Line	Home
Cursor Jumps to Beginning of Program	Ctrl + Home
Select Entire Line before Cursor	Shift + Home
Select Entire Program before Cursor	Ctrl + Shift + Home
Insert\Overwrite Switching	INSERT
Cursor Jumps One Character to Left	←
Cursor Jumps One Word to Left	Ctrl + ←
Select One Character to Left of Cursor	Shift + ←
Select One Word to Left of Cursor	Ctrl + Shift + ←
Cursor Jumps One Character to Right	→
Cursor Jumps One Word to Right	Ctrl + →
Select One Character to Right of Cursor	Shift + →
Select One Word to Right of Cursor	Ctrl + Shift + →
Insert Tab	Tab
Cursor Rolls Back One Tab	Shift+ Tab

5.3 Use the Help Documentation

5.3.1 Online Help Composition

The online help consists of four main sections: Contents, Index, Search, and Favorites.

5.3.2 Open the Online Help

5.3.2.1 Introduction

The online help supplements this manual. It is intended to provide detailed support in the use of the software.

5.3.2.2 Instructions for Use

This help system is integrated into the software. Users can open it in two ways:

(1) Click the **Menu** bar, **Device Library, Library Manager**, and sub-nodes of the Project Management Tree, and press F1 to accurately locate the relevant help instructions.

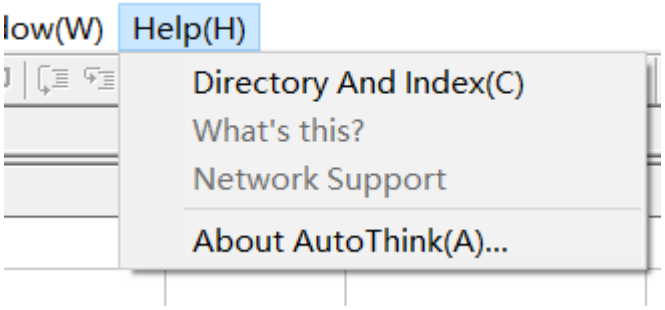
(2) In the Menu Bar, go to **Help** > **Contents and Index (C)** to call up the **Online Help**. In the **Index** and **Search** tabs, enter the keywords to find and locate them.

- **Index**: Used to traverse index pages for quick find.
- **Search**: Used for full-text searches.

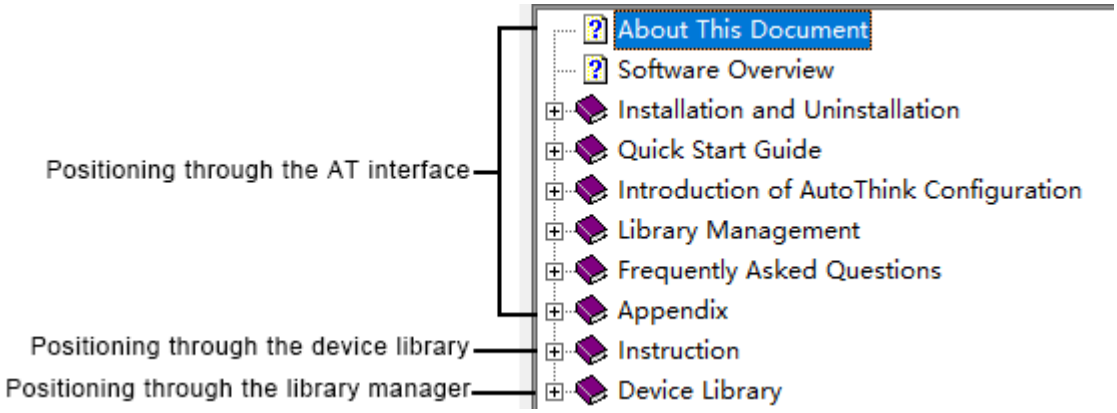
When users search for a keyword in the **Online Help**, only the pages that contain the keyword are searched, and the pages whose names contain the keyword are not searched.

When users search for a keyword in the **Online Help**, it is necessary to enter ***keyword or keyword*** to perform a fuzzy search. For example, to search for the **Target** instruction, enter: ***GET** in the **Search** tab; and to search for **GetTickCnt**, enter: **GET*** in the **Search** tab. This allows users to search for relevant pages.

In the AT software interface, **Library Manager** and **Device Library**, if you want to press **F1** to open the Online Help, you need to manually click the manual pages of the other two sections in order to load the keywords of the corresponding manuals. Otherwise, when using the **Index** and **Search** functions to search for keywords, the results are incomplete or cannot be searched.



- Locate the **Online Help** portal



1. Description of the legend displayed

The illustrations in this manual are normally displayed correctly. If appears in the legend position, press **F5** to refresh.

Chapter 6 Project Management

6.1 New Project

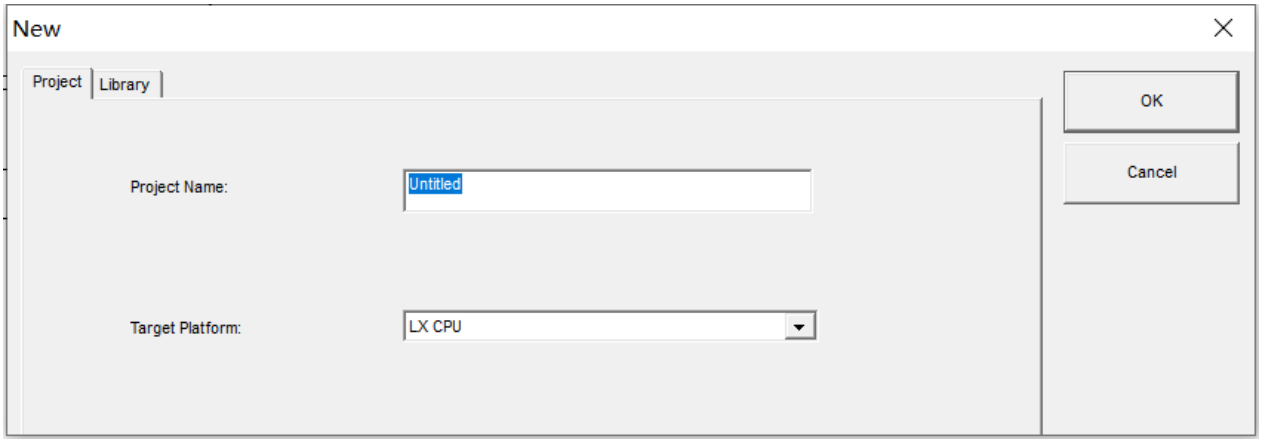
6.1.1 Introduction

This command is used to create a new project or library, which can be created via:

- Menu Bar: Click [File]- [New];
- Toolbar:  ;
- Shortcut key: **Ctrl + N**;
- Start page: New Project.

6.1.2 Steps

(1) Select the New command, and the New dialog box is displayed.

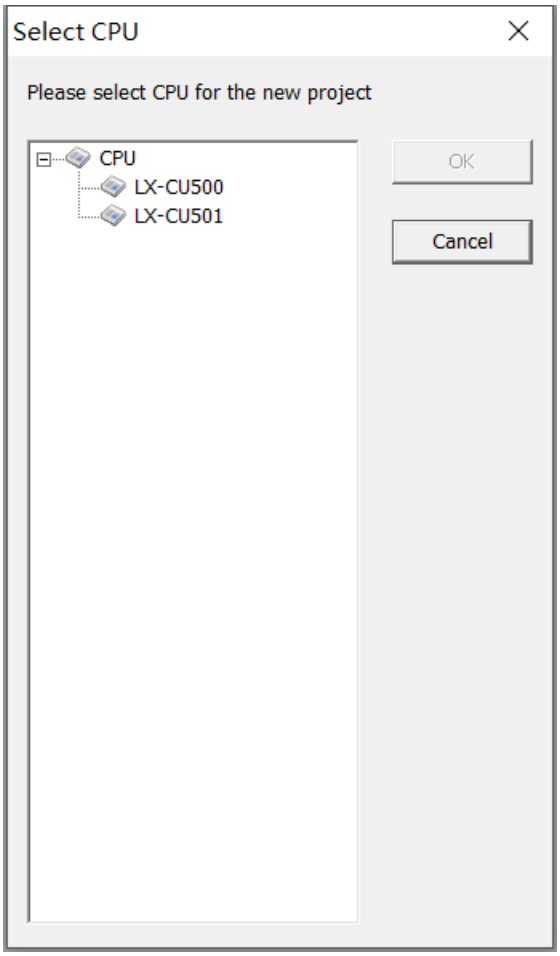


Project name: The default is **Untitled**. It is recommended to modify this name when saving this project.

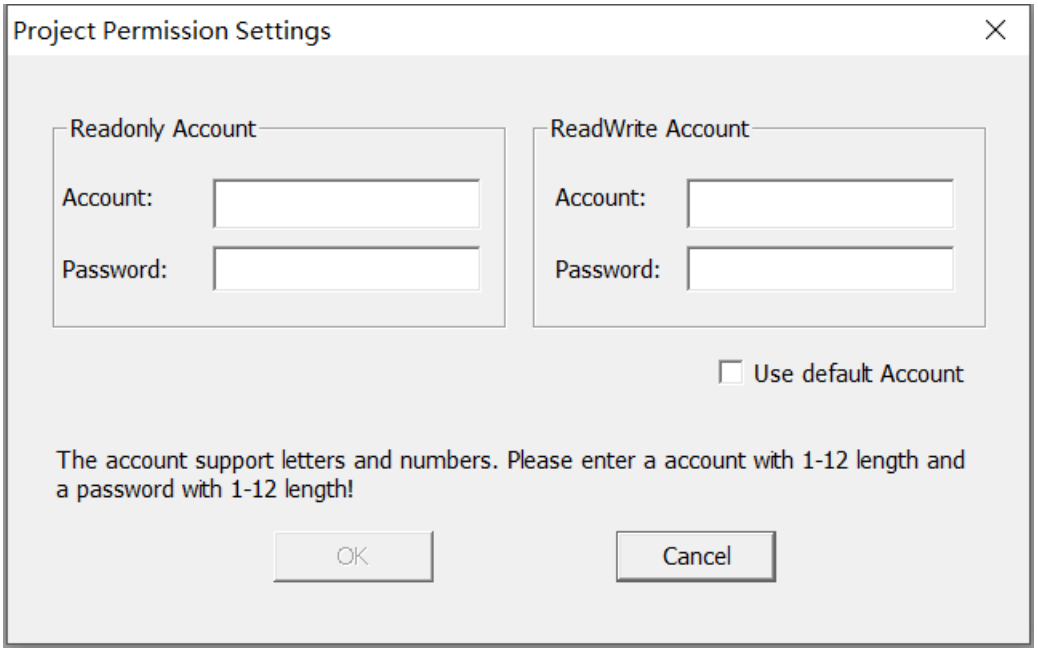
- The 10 characters, namely, \, /, :, *, ?, ", <, >, | and space, shall not be used in the project name.
- The project name shall not be null.
- The length of the project name shall not exceed 32 bytes, and the part that exceeds the range cannot be entered.

- When the project name is invalid, the **OK** button cannot be operated.
- Target platform: Users can select the corresponding platform according to the type of CPU used. The platform cannot be changed after the project has been created.

(2) Enter the project name, select the target platform, click **OK**, and the **Select CPU** dialog box is displayed, as shown in the figure.



(3) Select the desired CPU, and click **OK** to create a new project. The **Project Permission Settings** dialog box is displayed.



(4) Set Usernames and Passwords for Read-only Permission and Read/write Permission respectively.

- Naming: The username and password can be entered in both letters and numbers.
- Length: 1-12 characters for username, and 1-12 characters for password.
- The default permission can also be used as the project permission. Users need to check Use default account password, making the corresponding default account information to be synchronized to the project permission account information.

(5) Click OK, and the Setting Succeeded dialog box is displayed.

6.2 Open Project

6.2.1 Introduction

This command is used to open an existing project or library.

6.2.2 Steps

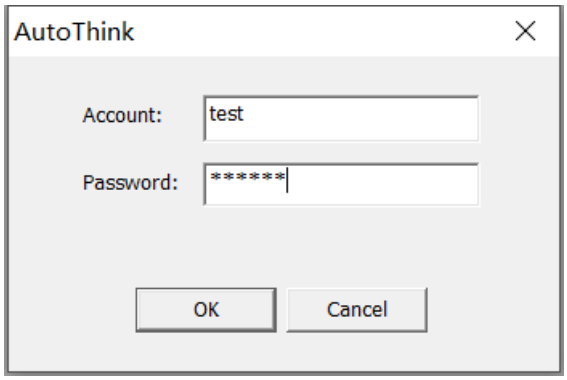
Users can open it via:

- Menu Bar: Click [File]-[Open];
- Toolbar: ;
- Shortcut key: Ctrl + O;

- Start page: Click [Open Project] - [Browse].

The Open dialog box is displayed. Select the target file of the project or library. Click Open. The software loads the file and displays the progress of opening in the Status Bar at the same time.

Select the target file of the project or library. Click **Open**. The Project Permission Login dialog box is displayed.



Enter the Account and Password for the project permission. Users can use either read/write or read-only permission accounts to open projects with different permissions.

6.2.3 Result

After executing the operation, users can open a project file.

6.3 Open a Recently Used Project

6.3.1 Introduction

It is used to open a recently used project.

6.3.2 Steps

Users can execute it via:

- Menu Bar: Click [File] - [Recent Projects list];
- Start page: Click Open Project. Select a recently opened record, and double-click or click Open.

The Recent Projects list is used to display the most recently opened projects, including the project path and project name. This command allows you to quickly switch projects.

6.3.3 Result

It is used to open a recently used project.

6.4 Save Project

6.4.1 Introduction

After modifying a project or library, users can use this command to save the changes.

6.4.2 Steps

- Save

  Users can execute it via:

  - Menu Bar: Click [File] - [Save];
  - Toolbar:  ;
  - Shortcut key: **Ctrl + S**.

    If the project or library is saved as Untitled, make sure to change the name. For the project or library saved to the corresponding folder under the installation path by default, its saving progress is displayed in the Status Bar when it is saved.

- Save as

  To modify save path of current project or library, you can use the order of [Save as].

  - Menu bar: Click [File]-[Save as].
  - Shortcut: **F12**.

    The dialog box of **Save as** pops up when this order is selected. Select an existing file name or enter a new file name, select file type and click the button of **OK** to complete.

6.4.3 Result

The project and library files are saved to the specified path.

6.5 Close Project

Users can execute it via:

- Menu Bar: [Click File] - [Close].

Use this command to close the currently opened project or library without exiting AutoThink software.

If the project or library has been changed, the system prompts whether to save the current project or library.

6.6 Configure Project Options

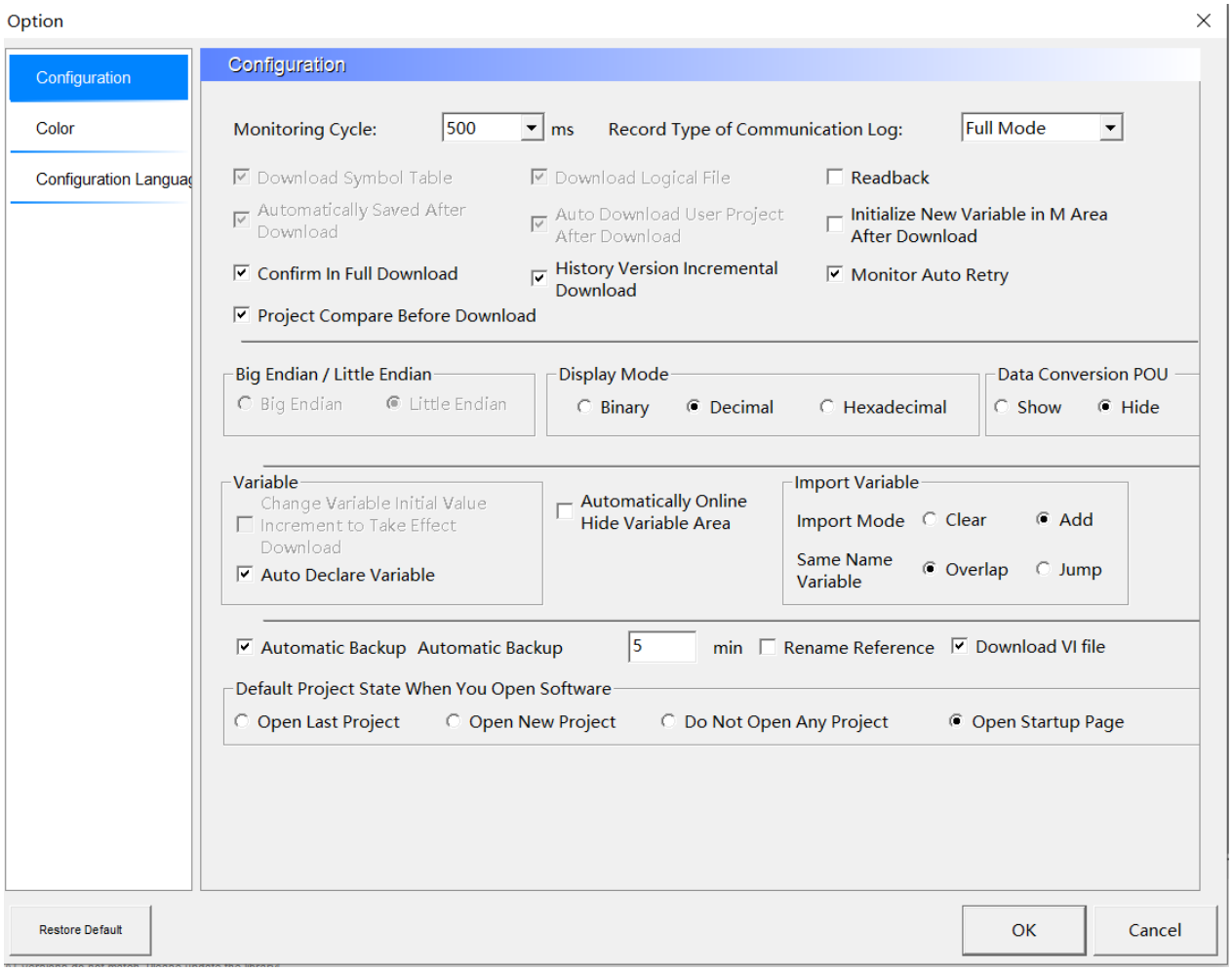
Open the Options dialog box via:

- Menu Bar: Click [Project] - [Option].

The Options setting contains settings for Configuration, Color, and Configuration Language.

6.6.1 Configure

The Configuration dialog box is used to set the monitoring period, log, online display, variables, and save related to the project



- **Monitoring Cycle**: The default is 500 ms. 250 ms, 500 ms, and 1000 ms are optional based on the actual needs.
- **Recording Type of Communication Log**: Used to set whether the communication log is recorded and the format of recording. The default is Full mode. Detailed communication information can be recorded in Full mode. Only the name of the communication message is recorded in Lite mode, and no information such as the code value is recorded.
- **Download Symbol Table**: Install the symbol table generated from project compilation to the specified communication station. The default is checked and this option cannot be modified.
- **Download Logic File**: Install the project file whose compilation passed to the specified communication station. The default is checked, and this option cannot be modified.
- **Initialize New Variables in M Area After Download**: Used to set whether to initialize new variables in M Area after installing, which takes effect after compilation.
- **Automatically Saved After Download**: Used to set whether to save the project automatically after the installation. the default is checked. This option is grayed out by default. By unchecking the Compare projects before installing checkbox, this option is enabled.
- **Auto Download User Project After Download**: The default is checked. After the project is installed, the project source file is automatically installed to the controller. Uncheck this option to clear the

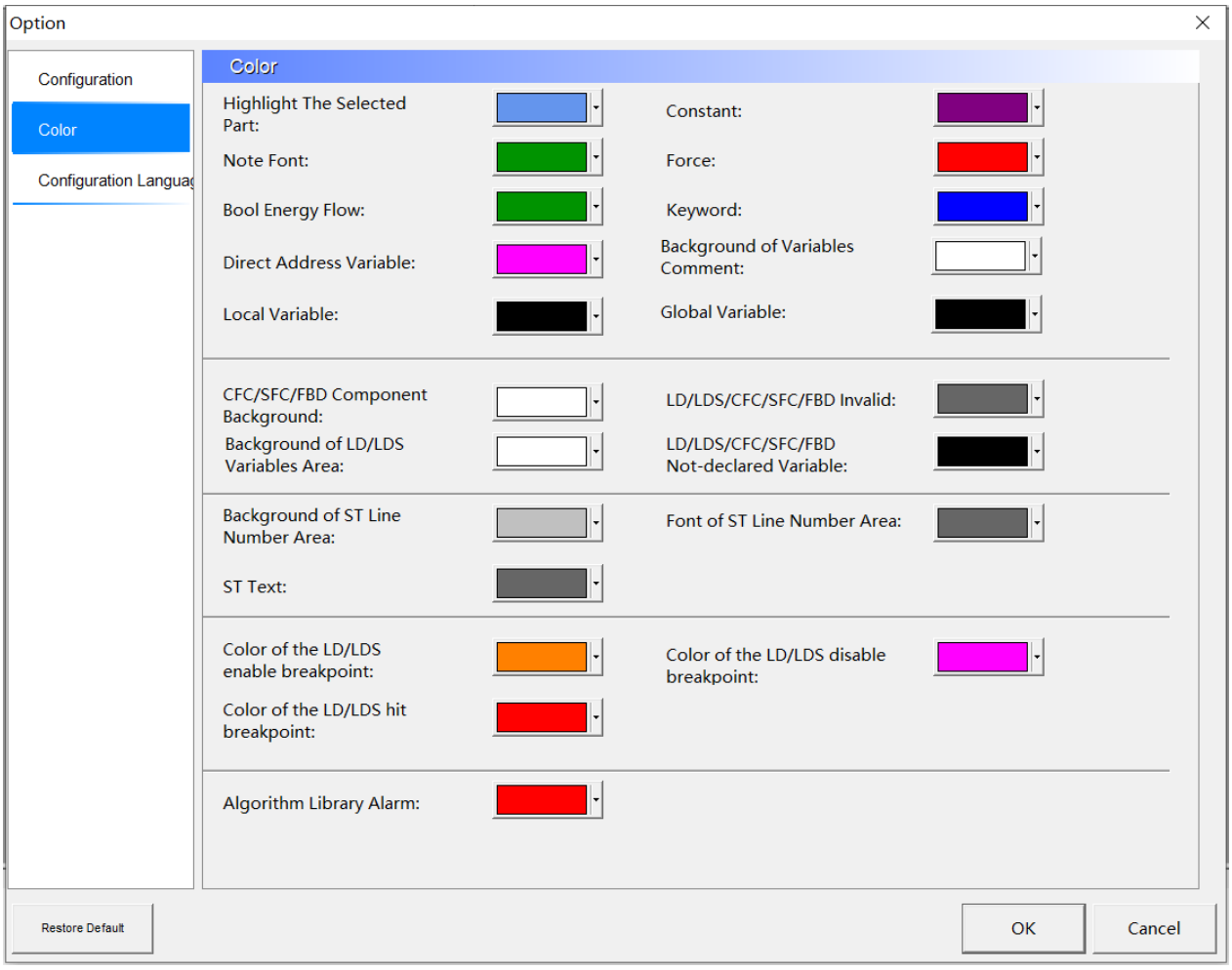
historical user projects from the controller. Before installing, the system prompts whether to delete the user project on the controller. This option is grayed out by default. By unchecking the Compare projects before installing checkbox, this option is enabled.

- **Confirm in Full Download**: Used to set whether the user is required to enter a verification code for confirmation in full installing.
- **Project Compare Before Download**: The default is checked. When installing, the current project is compared with the controller project.
- **Monitor Auto Retry**: Check this option, and AT automatically tries to be online once in case the link is abnormally disconnected during the online monitoring of the project.
- **Readback**: Used to set whether the data can be read back to the control station project during commissioning. The default is unchecked for commissioning readback. This option can be modified.
- **Bid Endian/Little Endian**: It refers to the data storage mode. The default is Small end mode. This option cannot be modified.
- **Display Mode**: In online or simulation state, set the data format of online display. It can be selected as Binary, Decimal, or Hexadecimal. The default setting is Decimal. This option can be modified.
- **Variable**: Used to set incremental installing and auto-declaration settings for variables.
  - **Change Variable initial Value Increment to Take Effect Download**: Used to set the incremental installing to take effect when the initial value of a variable is modified. This option is unchecked by default. That is, in the normal condition, if the project has been installed, the initial value of the variable is modified offline. After the incremental installing is performed, the modified initial value does not affect the online running value. This option cannot be modified;
  - **Auto Declare Variables**: Used to set whether to automatically display the Variable Auto-declaration dialog box for variable declaration if an undefined variable is used when a program is being written. The default is unchecked, that is, it is automatically defined as a declared variable. This option can be modified. Once this option is unchecked, the Variable Auto-declaration dialog box is no longer displayed when an undefined variable is entered;
  - **Automatically Online Hide Variables Area**: Used to set whether to automatically hide the variables area in the POU if online. This option is unchecked by default.
- **Import Variable**: Used to set how variables are imported and how variables with the same name are imported during incremental import.
  - **Import Mode**: There are two types of import modes, namely, Clear and Incremental. If Clear is selected, when importing variables, the variables in the corresponding POU or global variable group are cleared before the import is executed. If Incremental is selected, when importing variables, the incremental variables are imported, and variables with the same name are imported in two ways: overwrite or skip;
  - **Same Name Variable**: This option can be set when the import mode is Incremental. If Overwrite is selected, when importing variables, the variables with the same name in the project are overwritten, and the message Group XX: X is overwritten. is displayed in the Message Output Window. If Skip is selected, when importing variables, the variables with the same name are not imported, and the message Group XX: X is skipped! is displayed in the Message Output Window.

- **Automatic Backup**: Used to set the time and interval for auto backup of a project in offline state, designed for project backup when the software is shut down abnormally.
- **Rename Reference**: Check this option to rename the referenced variable. In the Rename dialog box displayed, users can choose to rename the variable.
- **Download VI file**: Check this option to automatically generate a .csv file of variable table after the project is compiled successfully. If the project has not been saved for the first time, the File Save dialog box is displayed; And if the project has been saved, a .csv file is automatically generated to the project directory with the file named VarInfo (project name).csv.
- **Default Project Status When You Open software**: Used to set the project opened when the software is started, with options: open the last project, open a new project, do not open any project, or open the start page. The default setting is to open the start page. This option can be modified.

6.6.2 Color

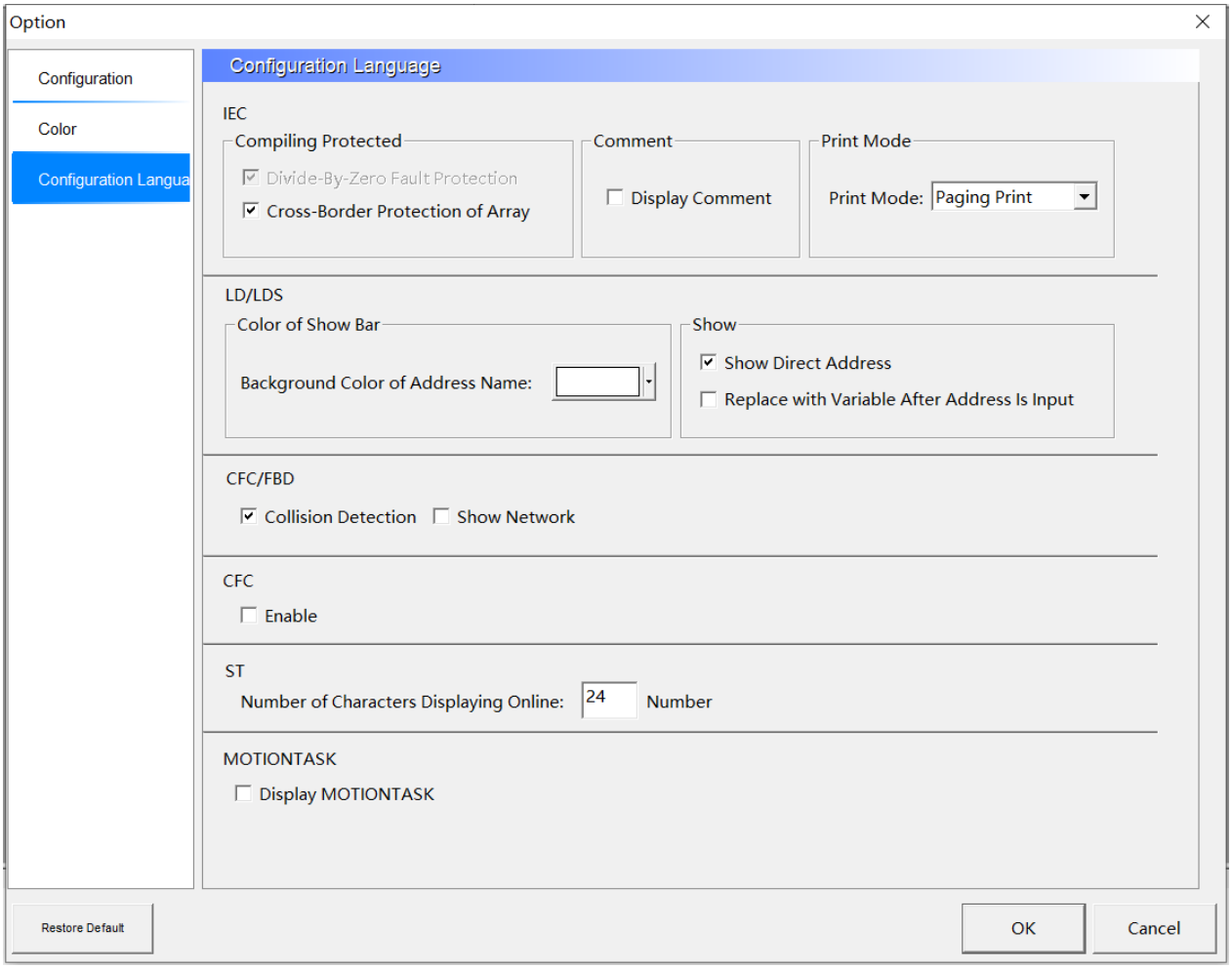
When configuring a program in the editor, use the Color options to freely set the colors of text, variables, and background requiring special display.



- The actual use of various color marks in the project can be adjusted according to the actual project needs. The color descriptions covered in this document are based on the colors set by default and will not be repeated.

6.6.3 Configuration Language

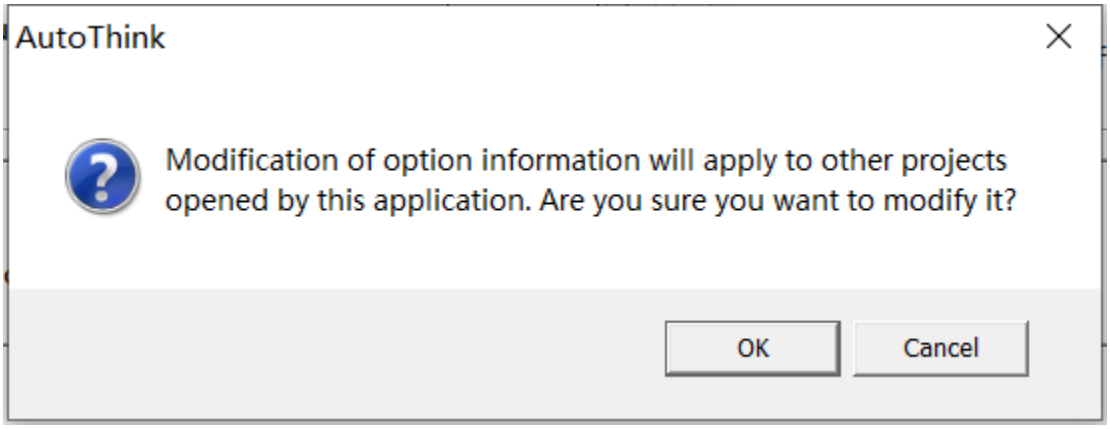
This option is used to set the display characteristics of various IEC configuration languages, as well as the display characteristics specific to each language.



- IEC
  - Compile-time protection: Used to set whether to perform Divide-by-zero protection and Array index out-of-bounds protection. Divide-by-zero protection is checked by default. When a divide-by-zero operation occurs in the controller, the result of the operation maintains the value of the previous cycle. By checking Array index out-of-bounds protection, the result of the operation maintains the value of the previous cycle when the array index exceeds the upper or lower limit.
  - Comment: Used to display the variable description contents of common type and functional block type variables.

- Printing mode: Used to set the printing mode, including Printing in pages and Printing on a single page. The default setting is Printing in pages.

- LD/LDS: Used to display parameter settings related to the LD/LDS language.
  - Display bar color: Used to set the background color of the address name in the program block of LD language configuration.
  - Display direct address: The default setting displays the address after adding a contact or coil.
  - Replace the address with a variable after entering it: In the variables area, define the variable A1 with the direct address %MX100.0. In the program area, define a contact, associate %MX100.0. After pressing Enter (left-clicking), A1 is displayed.
- CFC/FBD: Used to display parameter settings related to the CFC/FBD language.
  - Collision detection: Used to detect whether each element is allowed to overlap with each other when writing a program block in CFC language. If this option is unchecked, when two or more elements overlap, the later placed element partially or completely obscures the first placed element.
  - Show grid: If this option is checked, when the program block is written in CFC language, the grid mark is shown in the programming area. Otherwise, no grid is shown in the editing area. This option is unchecked by default.
- CFC: Used to display parameter settings related to the CFC language.
  - Enable: If this option is checked, when the program block is written in CFC language, the block element with enabled input and output is added. This option is unchecked by default.
- ST: Used to display parameter settings related to the ST language.
  - Number of characters displayed online: Used to set the number of characters displayed online for variable names and operation results. The default setting is 24. The display length ranges from 20-100 characters.
- MOTIONTASK: Used to display parameter settings related to the MOTIONTASK.
  - Display MOTIONTASK: Used to display the MOTIONTASK in Project Management area of AT. The default is unchecked.
- Restore default
  - By click Restore Default, a prompt box is displayed as shown in the figure. By clicking OK, the parameters in the options are restored to the system default settings.

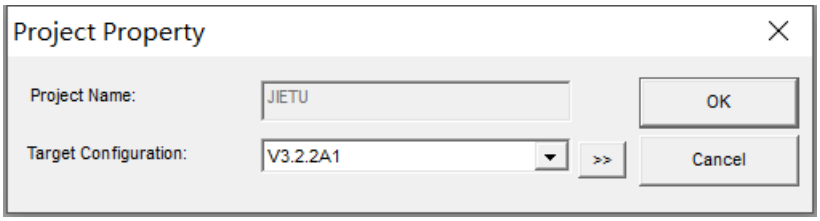


6.7 View Project Property

Users can open it via:

Project Management Tree: Right-click the root node (project name), and click Property.

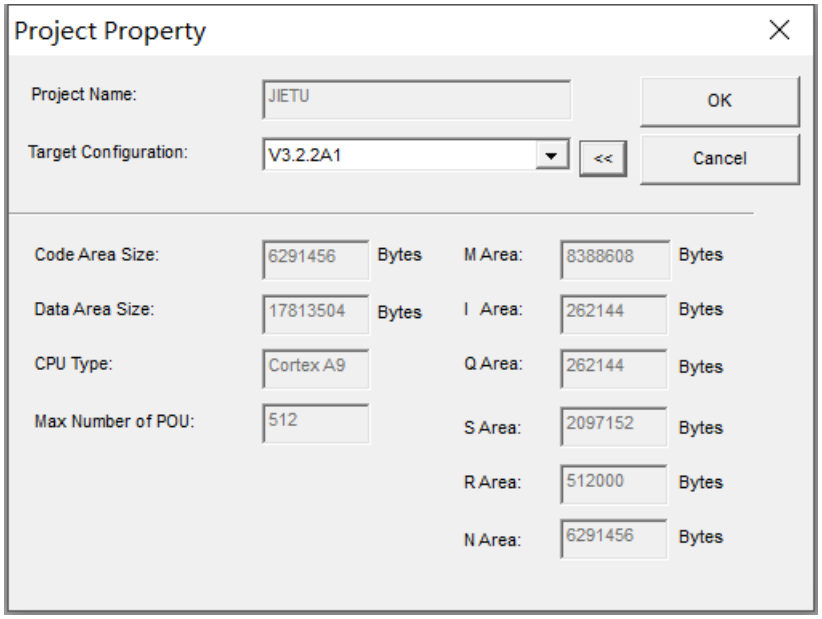
Open the configuration information for the current project, as shown in the figure.



Target Configuration: It refers to the AutoThink software version supported by the current CPU module. Users can switch the configuration between different versions by click ▼.

The functions supported by each software version may differ. Be sure to learn about the version before switching.

Click » button to view specific configuration information for this project.



6.8 Import Sample Project

6.8.1 Steps

Follow the steps below:

(1) Click the [Project] menu.

(2) Select the [Import Sample Project] command.

(3) Check the corresponding algorithm block and click Import.

6.9 Compare Offline Project

6.9.1 Introduction

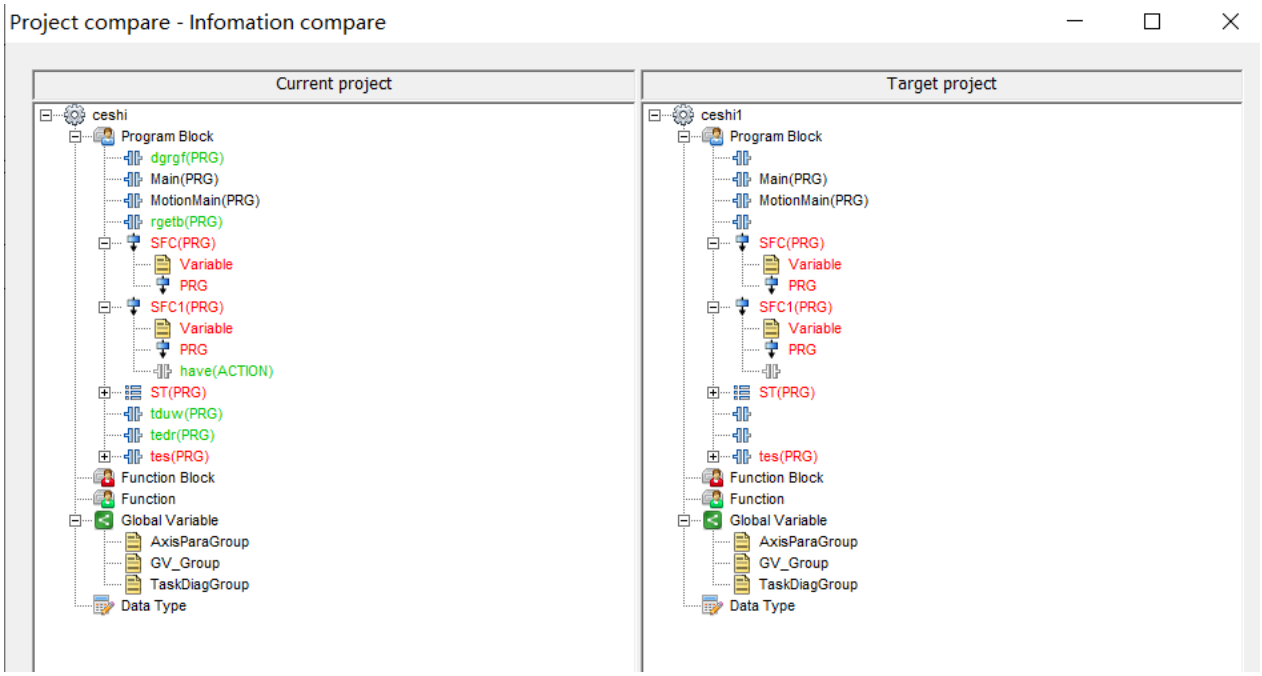
AutoThink software supports the comparison of two different local user projects to determine the data differences between the two projects. The comparison contains user programs, variables and data types. Detailed comparisons can be performed on the difference data compared. For POUs, only ST and LD language POUs currently support detailed comparisons.

Difference data is displayed as Project Management Tree nodes. The tree nodes with differences are displayed in four different colors with the following meanings:

- Data in red: Indicates that the data in the same POU or variable group is different in the two projects.
- Data in green: Indicates that the data only exists in one of the two projects, and the data node does not

exist in the other project.

- Data in purple: Only for SFC type POU. Indicates that the subprograms under the same SFC program are different in the two projects.
- Data in blue: Only for SFC POUs. Indicates that the SFC programs in the two projects are different, and the subprograms under the SFC programs are also different.

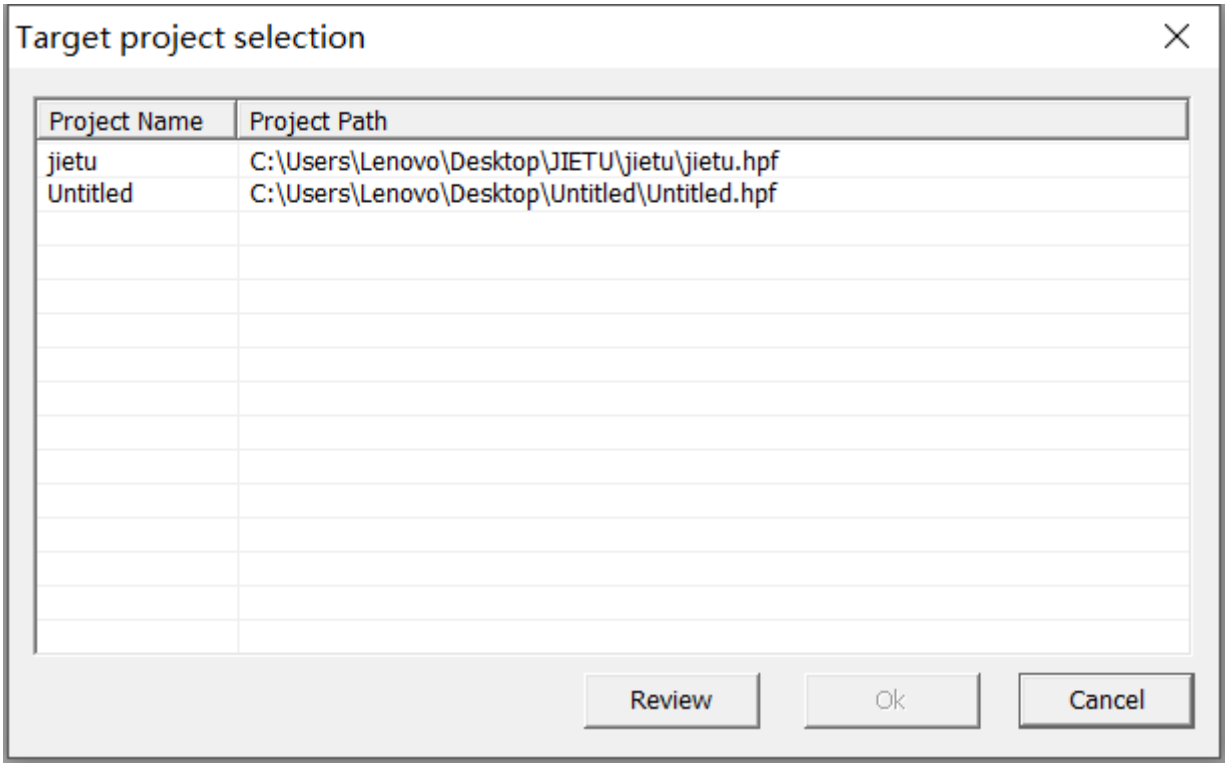


6.9.2 Requirement

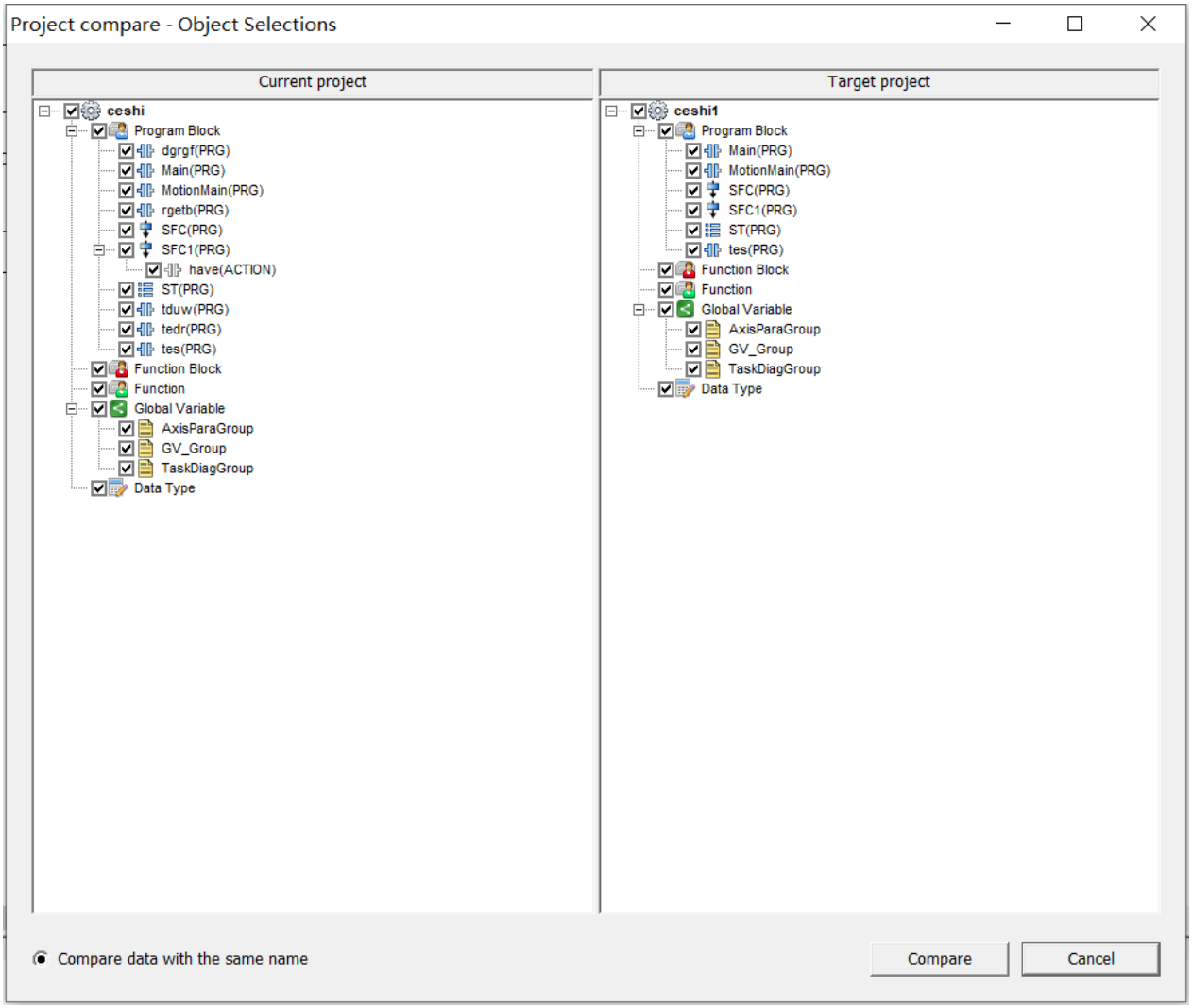
AutoThink is offline.

6.9.3 Steps

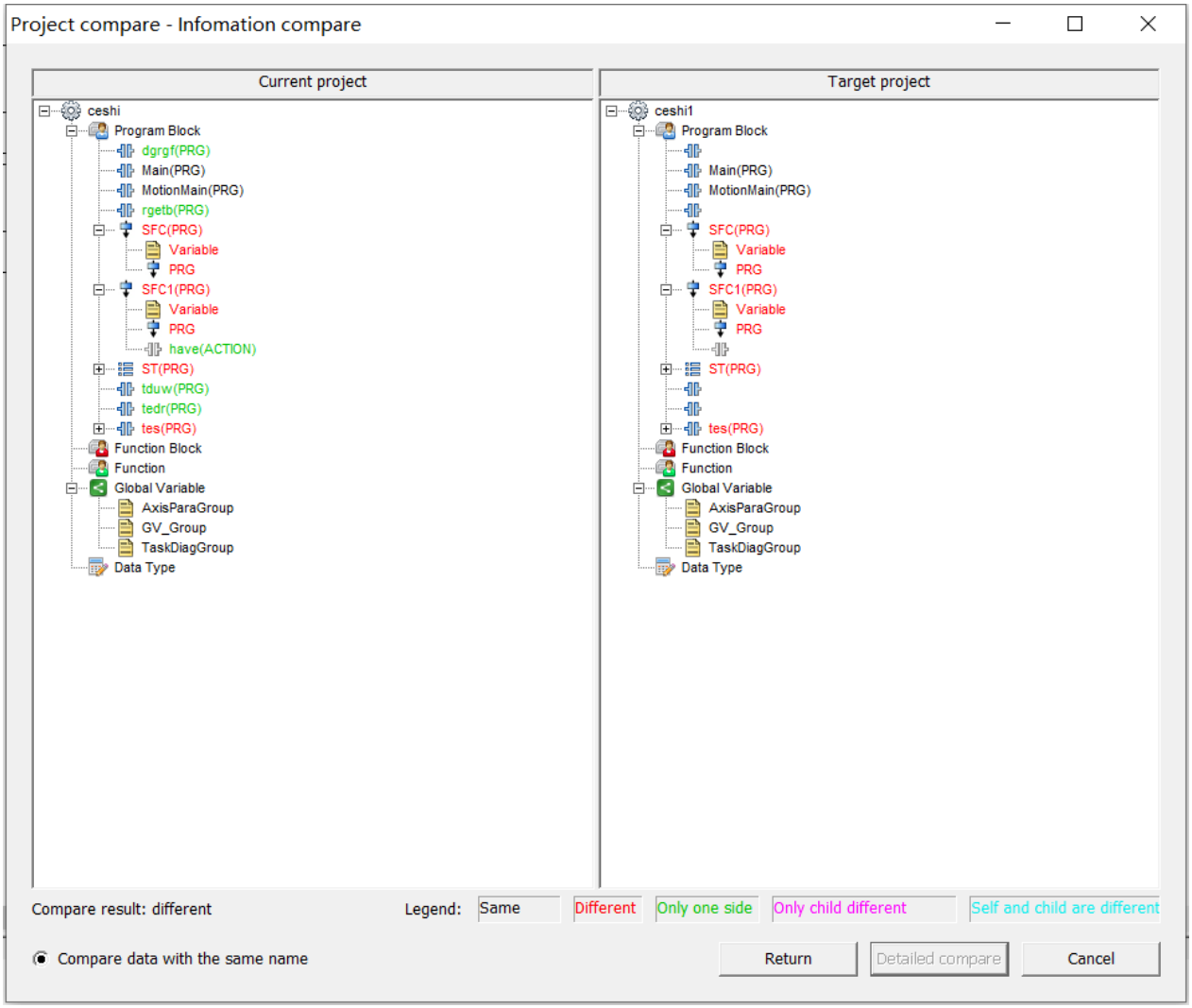
(1) In the Menu Bar, click [File] - [Project Compare]. The Target Project Selection dialog box is displayed.



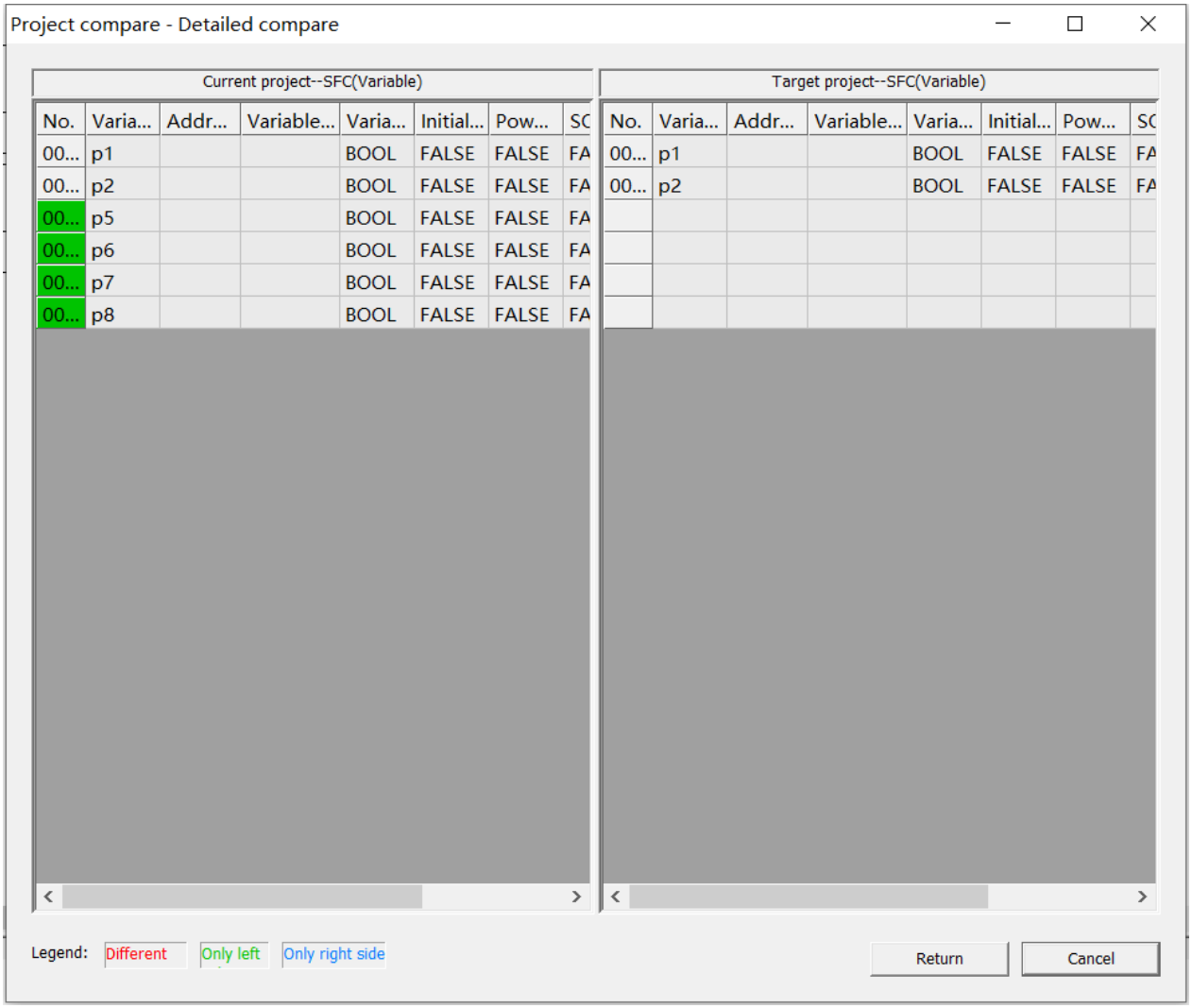
(2) Click Browse. Select a target project or a previously compared project. Click Open. The Comparison dialog box is displayed.



(3) Check the project nodes that need to be compared. Click Compare. The difference nodes of the two projects and the comparison result are displayed.



(4) Check the different nodes. Click Detailed Comparison to list the detailed differences.



6.9.4 Result

It enables comparison of different projects.

6.10 Axis Parameter

6.10.1 Axis Parameter Display Settings

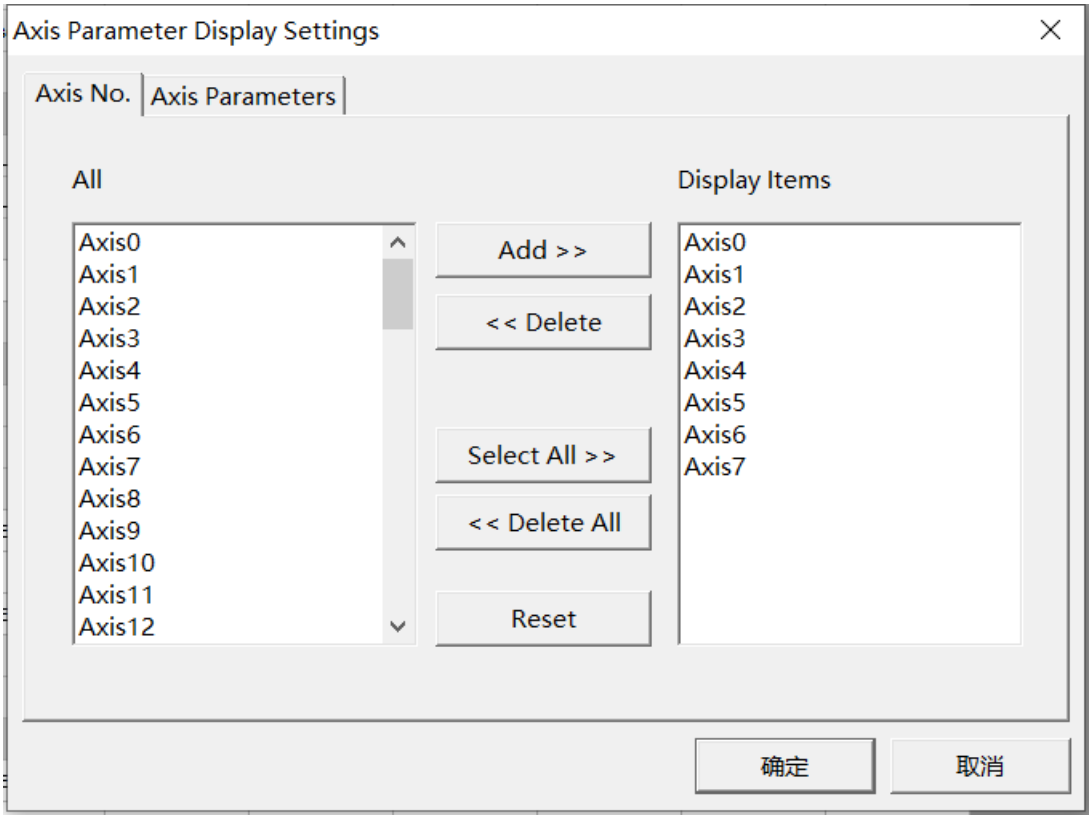
6.10.1.1 Introduction

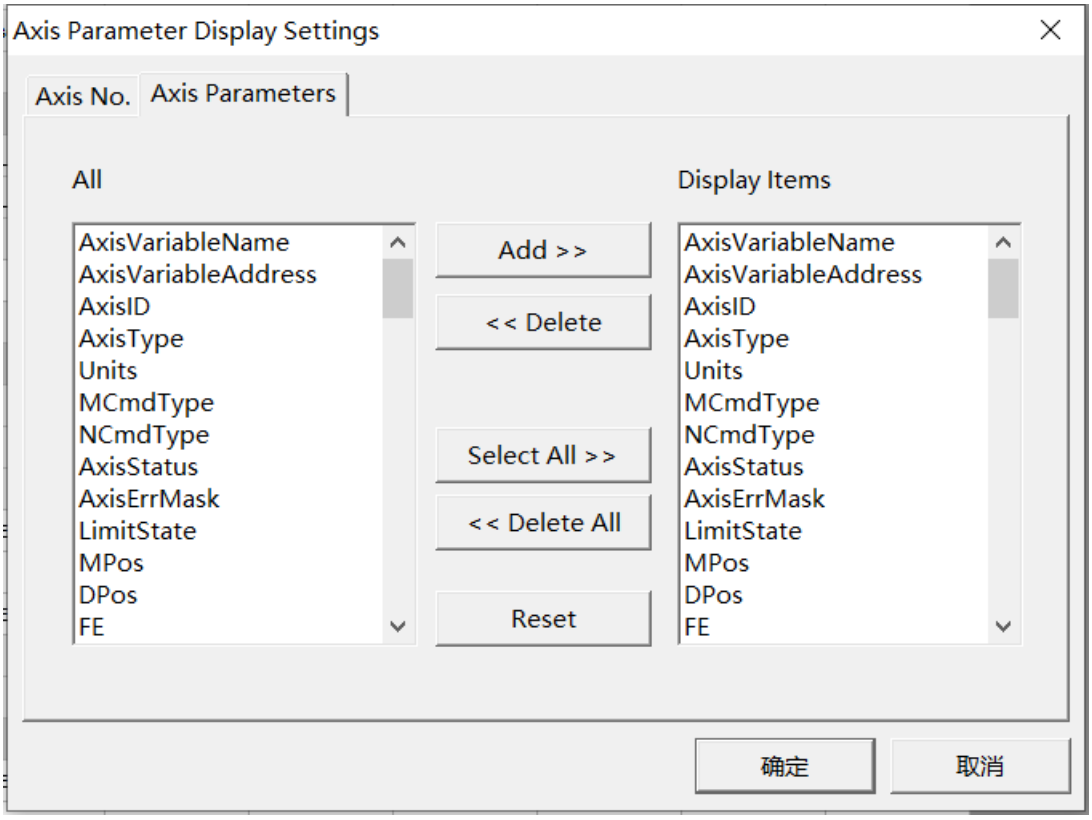
Users can customize the displayed Axis Numbers and Axis Parameters. The configured Axis Numbers and Axis Parameters are synchronized in the axis parameter list.

6.10.1.2 Steps

Menu Bar: Click [Tool] - [Axis Parameter Display Settings].

Open the Axis Parameter Display Settings dialog box, as shown in the figure. The LX-CU500 supports 64 axes, and only the first few axes are displayed by default. The Axis Parameter Display Settings supports user-defined display of Axis Numbers and Axis Parameters. The Axis Numbers and Axis Parameters configured here are synchronized in the axis parameter list.





6.10.2 View Axis Parameters

6.10.2.1 Introduction

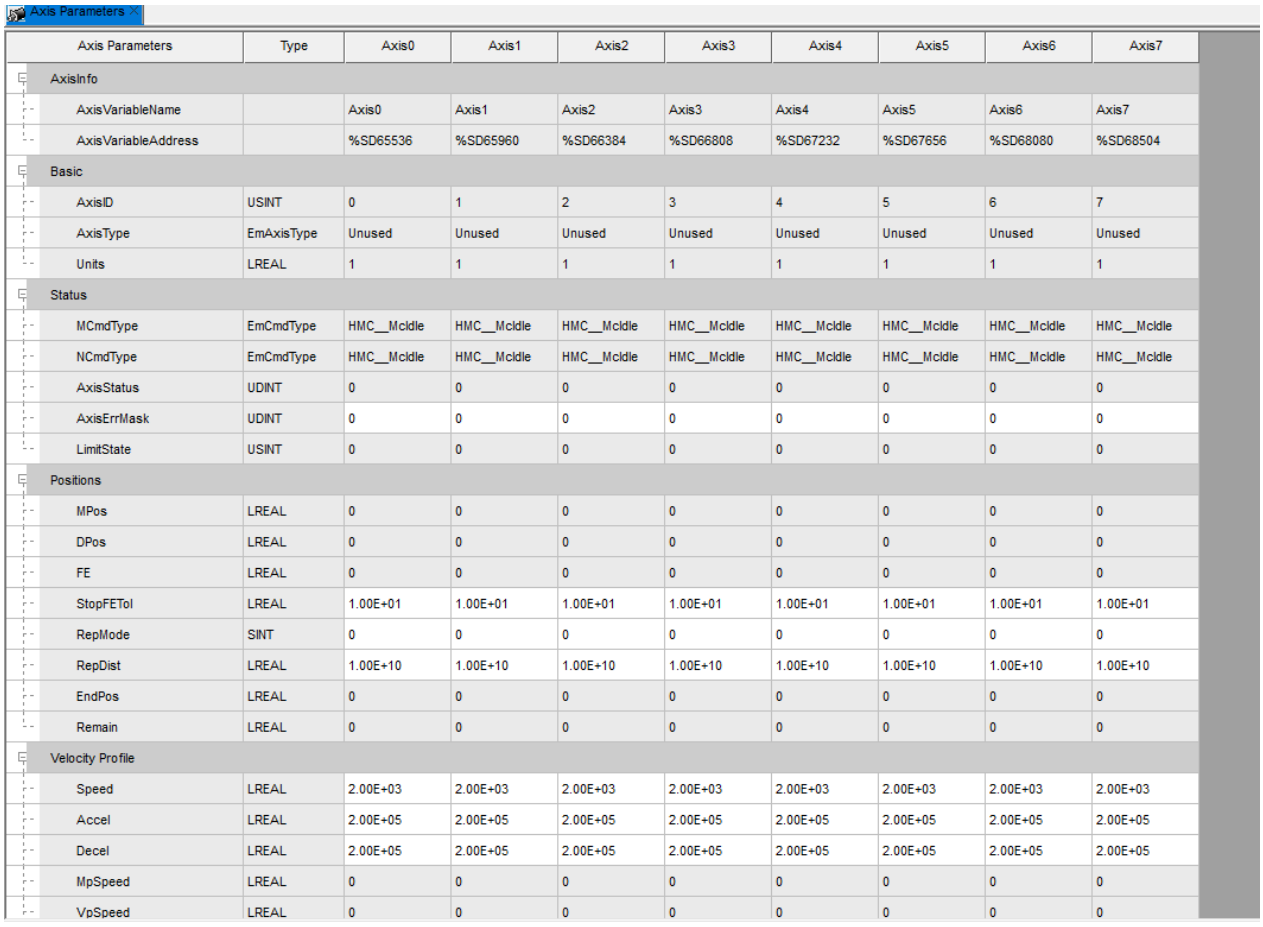
This function allows users to view the set Axis Parameters.

6.10.2.2 Requirement

Menu bar: Click [Tool] - [View Axis Parameters];

Toolbar:  .

Open the Axis Parameter. The window is shown in the figure.



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

The axes and axis parameters displayed in the axis parameter list are configured by the user in the Axis Parameter Display Settings. The first row shows the axis number of each axis, the first column shows the name of the axis parameter, and the second column shows the corresponding type of the axis parameter.

Axis parameters are displayed in groups for easy viewing. By clicking  and  in front of the axis parameter group, users can collapse and expand the axis parameter group.

The read/write property of the axis parameter is predefined by the system. In the axis parameter list, the white area indicates the axis parameters with writable property, whose initial and online values can be modified; And the gray area indicates the axis parameters with read-only property, which can only be

viewed but not modified.

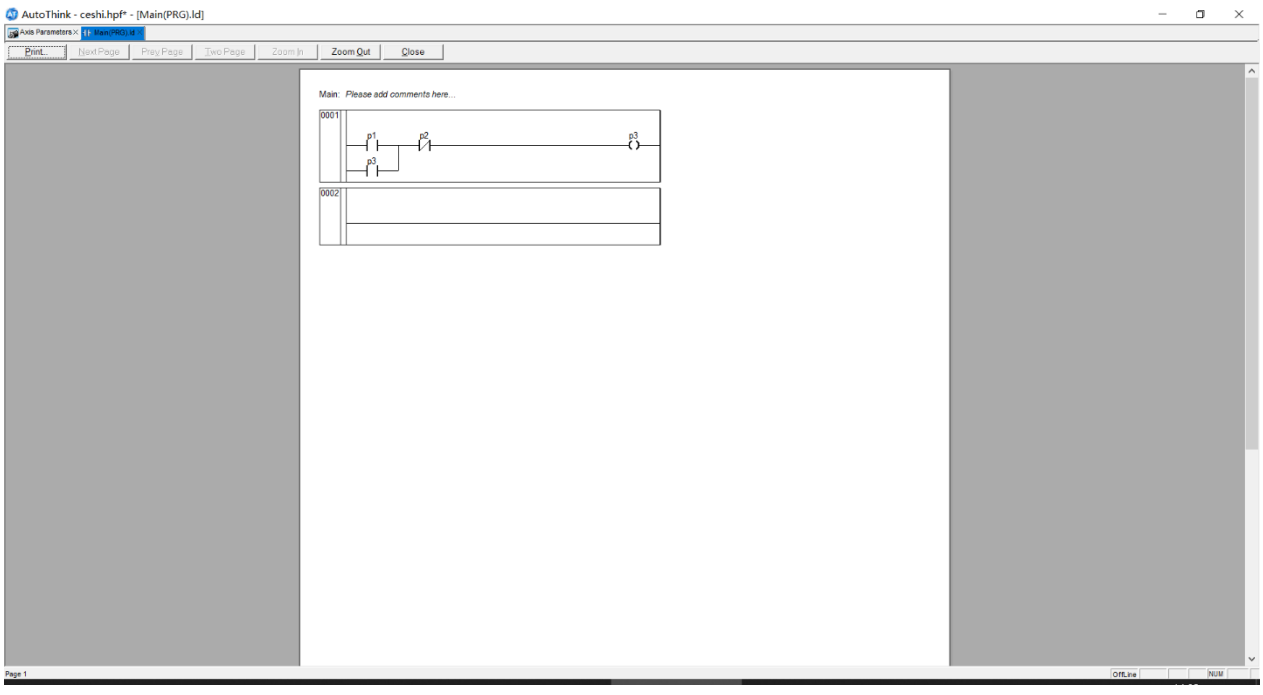
- In the LX project, the axis parameters have been predefined as system variables in the global variable group AxisParaGroup. The default variable names of the axis parameters for the 64 axes are Axis0-Axis63, and users can rename the axis parameter variables.
- The axis parameter variables are of functional block type, and the pins are all parameters for the axis. For example, to set the Speed of Axis1 to 1000, users can use the following statement: Axis01.Speed := 1000.
- The function Axis Parameter Display Configuration has been integrated into the right-click menu of the axis parameter list.

6.11 Print Project

6.11.1 Print Preview

This command allows you to preview the printing effect of the currently active window. Users can execute it via:

- Menu Bar: Click [File] - [Print Preview];
- Toolbar: .



On the Print Preview page, users can Zoom In and Zoom Out on the current page. For multiple pages, click Previous or Next to page forward and backward.

6.11.2 Print

This command allows you to print the contents of the active window. Users can execute it via:

Menu Bar: Click [File] - [Print];

Toolbar:  ;

Shortcut key: **Ctrl + P**.

In the pop-up Print dialog box, select the Printer Name, set the Print Range and Number of Copies, and click OK to print the active window.

When the Number of Copies exceeds 1, users can check Auto Pagination. Printed content can also be output to a file.

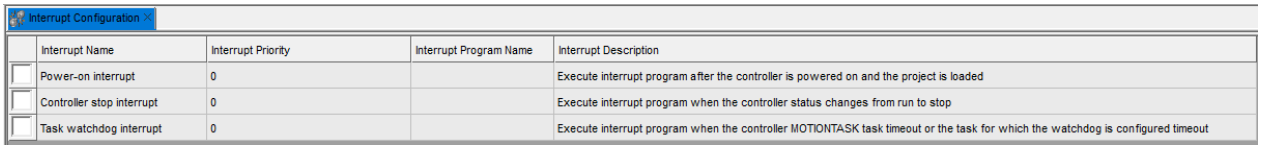
6.12 Interruption Configuration

Interruption refers to some special triggering conditions that making the program can be executed preferentially in the process of program execution.

Interruption configuration refers to configuring the interruption signals to trigger calling of corresponding subprogram. The interruption signals with different functions are all provided by the system. Interruption is executed as per the default priority in turn in the course of execution, from high priority to low priority.

Main program will be stopped when interruption occurs in the process of program running, so as to call the program associated with this interruption.

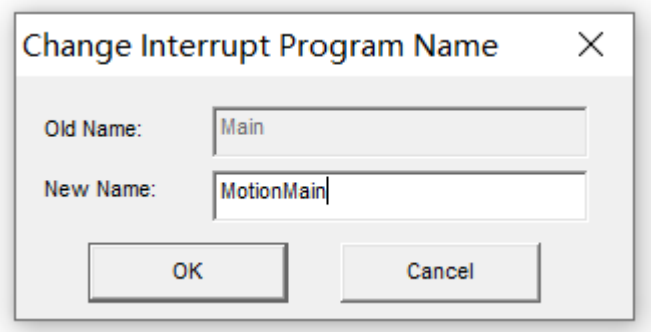
Double click the node of [Interruption Configuration] in management tree to display the edit window of [Interruption Configuration], as shown in figure.



Edit window of Interruption Configuration

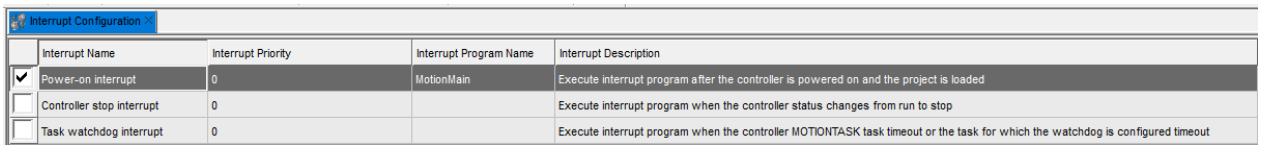
Check the corresponding **Interrupt Name** to enable an interrupt signal. However, you must set the **Interrupt Program Name** for the interrupt before you decide whether to start the interrupt.

Double click the field of **Interrupt Program Name** to pop up the dialog box, as shown in figure.



Dialog Bx of Add Interruption Program

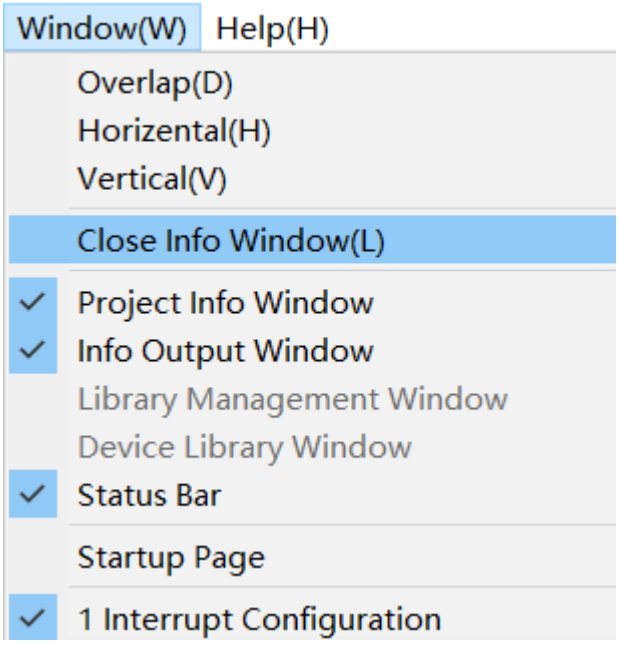
Enter POU name of [Program Block] in the **Name** box, such as **MotionMain** program. Automatically defaults to the language type of the POU, and click **OK** button to close the dialog. Then check the interruption name, as shown in following figure. The **MotionMain** program will be executed when an interruption occurs .



Start Interruption

6.13 Window

[Window (W)] menu provides orders used to set layout of windows in work area, hides or displays designated windows, and provides orders of switching display or switching to windows opened recently, as shown in figure, with detailed description of functions as follows:



Window Menu

The operation orders and their functions in window menu are as shown in Table.

Description of orders in window menu

Order	Description
Overlap	one superposed on another, and staggered one by one.
Horizontal	All the windows are laid horizontally filling the whole work area and without being cascaded
Vertical	All the windows are laid vertically filling the whole work area and without being cascaded.
Close all	Close all the windows opened currently in work area.

6.13.1 Common windows

This area lists commonly used auxiliary windows: [Project Info Window], [Info Output Window], [Library Management Window], [Device Library Window], [Status Bar], [Startup page], which are selected by checking.

The bottom of dropdown list of [Window] displays names of each edit window opened currently in work area. Clicking the window name to make it as current active window .

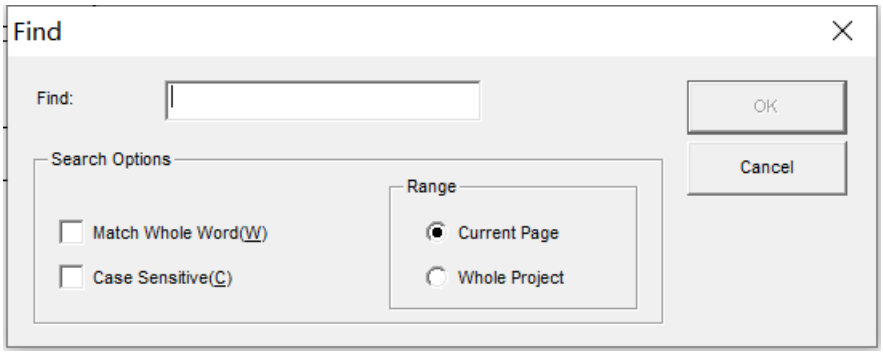
6.14 Find

6.14.1 Introduction

Users can execute it via:

- Menu Bar: Click [Edit] - [Find];
- Toolbar: ;
- Shortcut key: **Ctrl + F**.

This command can be used to search a certain text in the current editor or the whole project. Open the Find dialog box. In the Find content input box, enter the character sequence to be found. In addition, users can set Search options: whether to Match all, whether to be Case sensitive, and whether to search Whole project or Current page.



Click **OK** to start finding. The finding always starts from the current page.

The findings are displayed in the Finding Message column of the message window, as shown in the figure. By double-clicking the finding item, users can locate the desired text.

6.14.2 Result

It enables target finding.

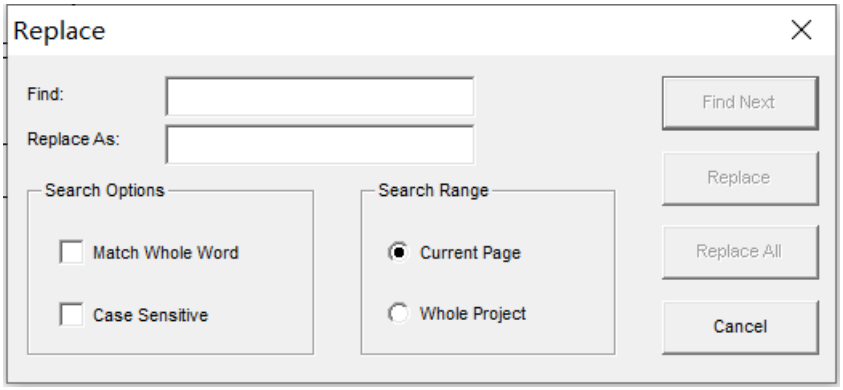
6.15 Replace

6.15.1 Introduction

This command can be used to search a text just like the command Find, and it can replace the text with another one. Users can execute it via:

- Menu Bar: Click [Edit] - [Replace];
- Shortcut key: **Ctrl + H**.

Open the Replace dialog box. In the Find content input box, enter the character sequence to be found. In the Replace with input box, enter the replacement. Set the Search options and Search range. By clicking Replace, users can replace the fields that meet the finding criteria, one at a time, on the current page or in the whole project. Click Find Next again to continue down the find and replace; By clicking Replace All, users can replace all fields that meet the finding criteria on the current page or the whole project. After the replacement is complete, a pop-up prompt displays the total number of completed replacements.



6.15.2 Result

It enables target replacement.

6.16 Editing Operations

6.16.1 Undo

6.16.1.1 Introduction

This command allows you to cancel the most recently executed action in the current editor. By repeating this command, all actions can be executed up to the point where the window opens, which applies to all actions in the currently open editor.

6.16.1.2 Steps

Users can execute it via:

- Menu Bar: Click [Edit] – [Undo];
- Toolbar:  ;
- Shortcut key: Ctrl + Z.

6.16.2 Redo

6.16.2.1 Introduction

This command allows you to restore the action that has been **undone** in the currently open editing window.

6.16.2.2 Steps

Users can execute it via:

- Menu Bar: Click [Edit] – [Redo];
- Toolbar:  ;
- Shortcut key: Ctrl + Y.

6.16.3 Cut

6.16.3.1 Introduction

This command allows you to transfer the currently selected object from the editor to the clipboard and to delete the selected object from the editor. In the project tree, this command can be applied to the currently selected object. However, not all selected objects can be cut.

6.16.3.2 Steps

Users can execute it via:

- Menu Bar: Click [Edit] – [Cut];
- Toolbar:  ;
- Shortcut key: Ctrl + X.

6.16.4 Copy

6.16.4.1 Introduction

This command allows you to copy the currently selected object from the editor to the clipboard without changing the contents of the editor window. In the project tree, this command can only be applied to selected objects. However, not all objects can be copied, such as hardware configuration and tasks.

6.16.4.2 Steps

Users can execute it via:

- Menu Bar: Click [Edit] – [Copy];
- Toolbar:  ;
- Shortcut key: Ctrl + C.

6.16.5 Paste

6.16.5.1 Introduction

This command allows you to paste the contents of the clipboard into the current position of the editor.

6.16.5.2 Steps

Users can execute it via:

- Menu Bar: Click [Edit] – [Paste];
- Toolbar:  ;
- Shortcut key: Ctrl + V.

6.16.6 Delete

6.16.6.1 Introduction

This command allows you to delete the selected area from the editing window, which does not change the contents of the clipboard. In the project tree, this command can also be applied to selected objects. However, not all objects can be deleted, such as hardware configuration.

6.16.6.2 Steps

Users can execute it via:

- Menu Bar: Click [Edit] – [Delete];
- Shortcut key: Delete.

6.17 BackUp

6.17.1 Controller BackUp

6.17.1.1 Introduction

To back up the project, network configuration, etc., in the controller and generate a disk file.

6.17.1.2 Requirements

The controller is connected.

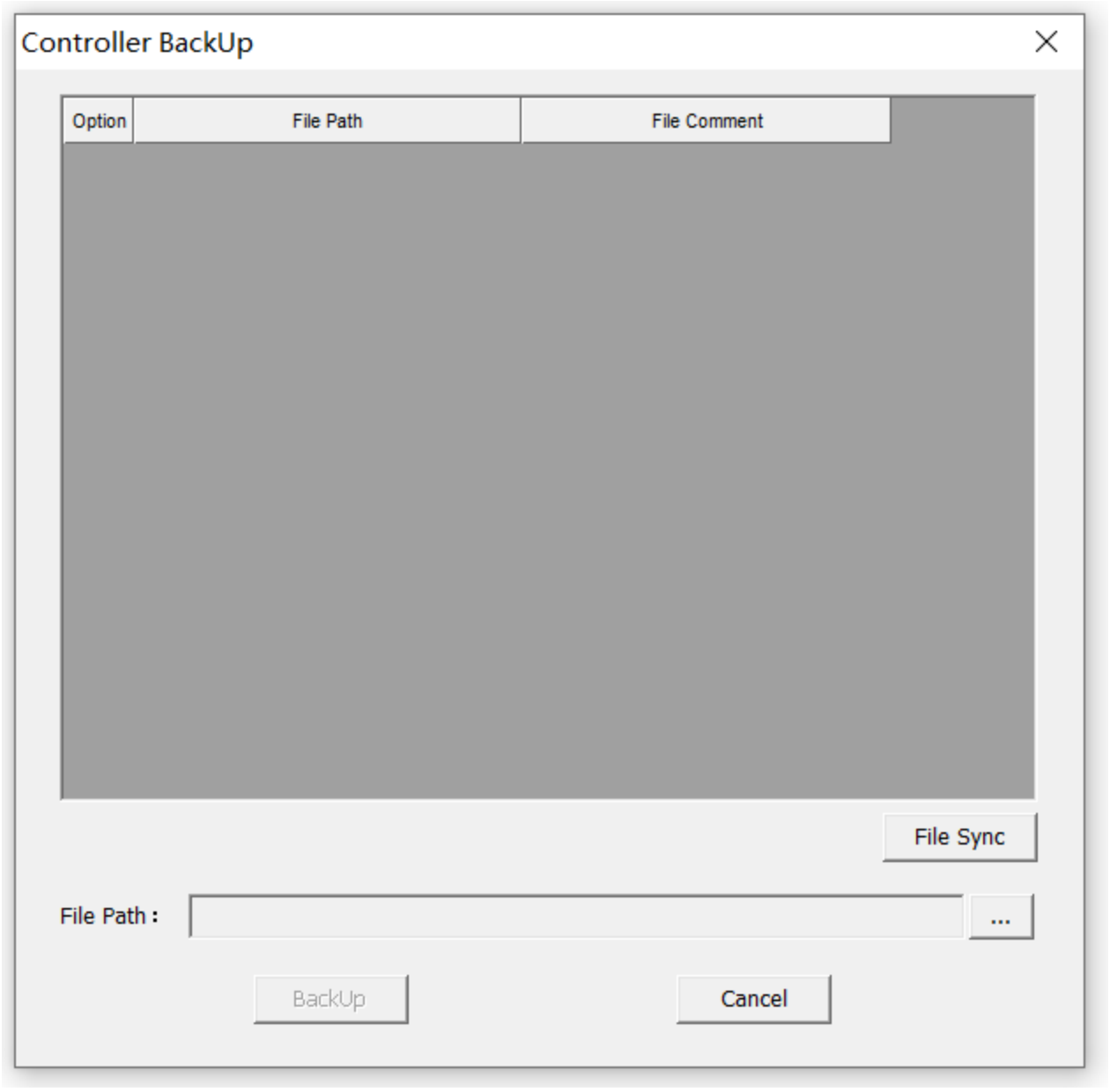
The project has entered the offline state.

6.17.1.3 Steps

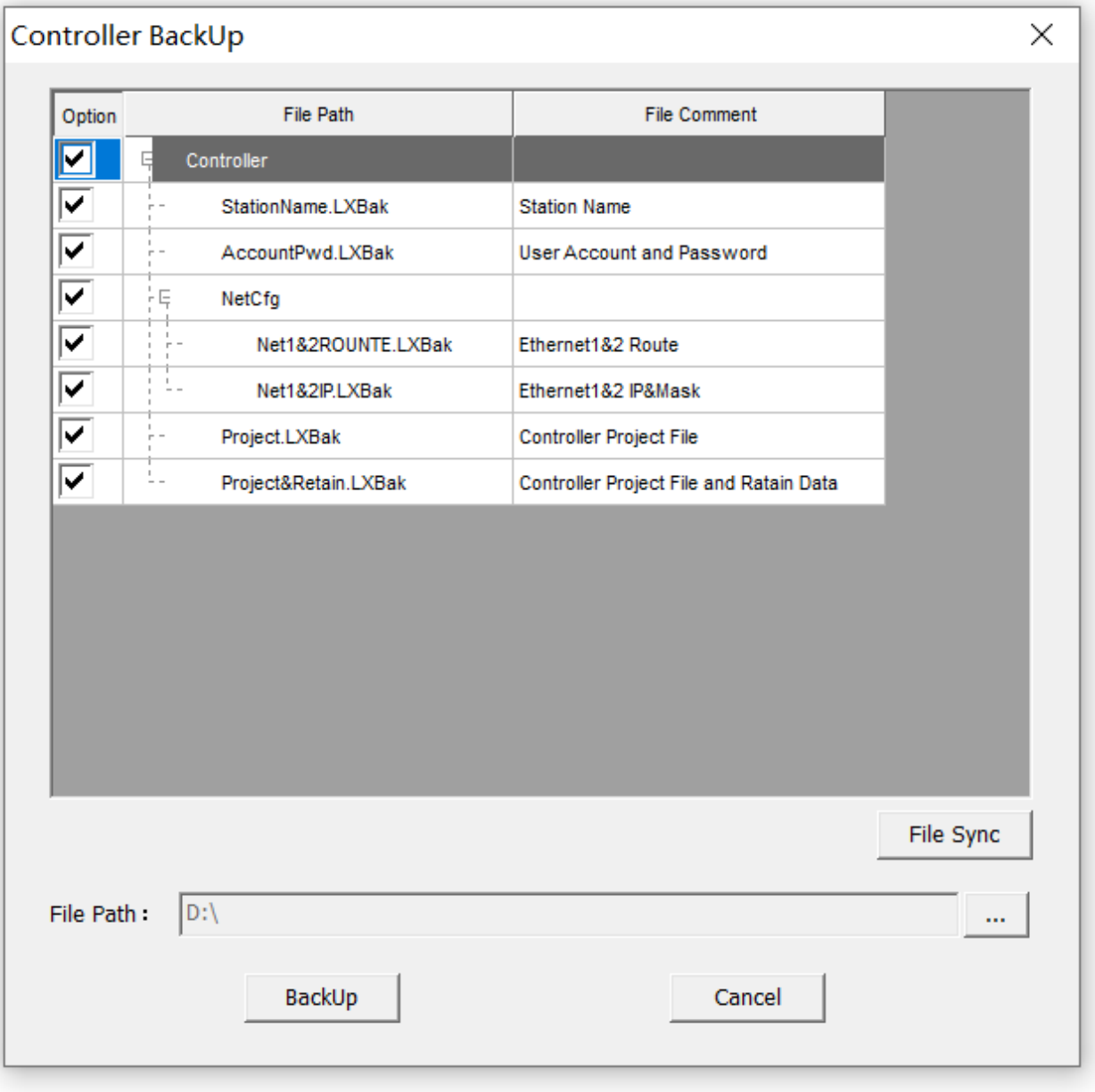
To back up the controller, follow these steps:

(1) Click the "Project" menu.

(2) Select "Backup" - "Controller Backup" command. A controller backup dialog box will appear, as shown in the figure below.



(3) Click the File Sync button, enter the controller account password, and a list of files that can be backed up will be displayed. Select the files you want to back up and choose an output directory, then click the Backup button. As shown in the figure below.



6.17.2 Controller Recover

6.17.2.1 Introduction

To restore the backed-up files, verification is not supported during the restoration process.

6.17.2.2 Requirements

The controller is connected.

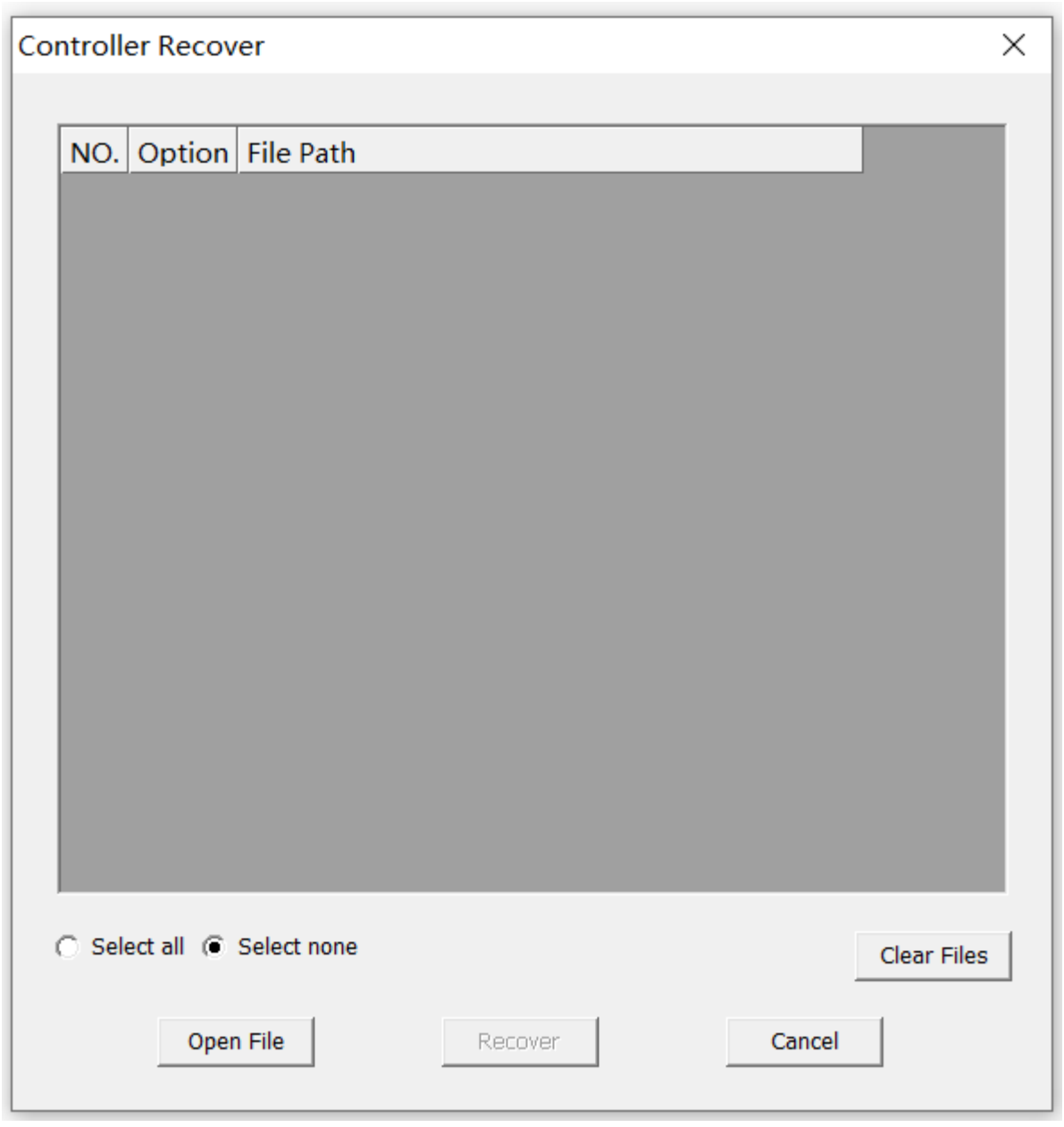
The project has entered the offline state.

6.17.2.3 Steps

To restore the controller, follow these steps:

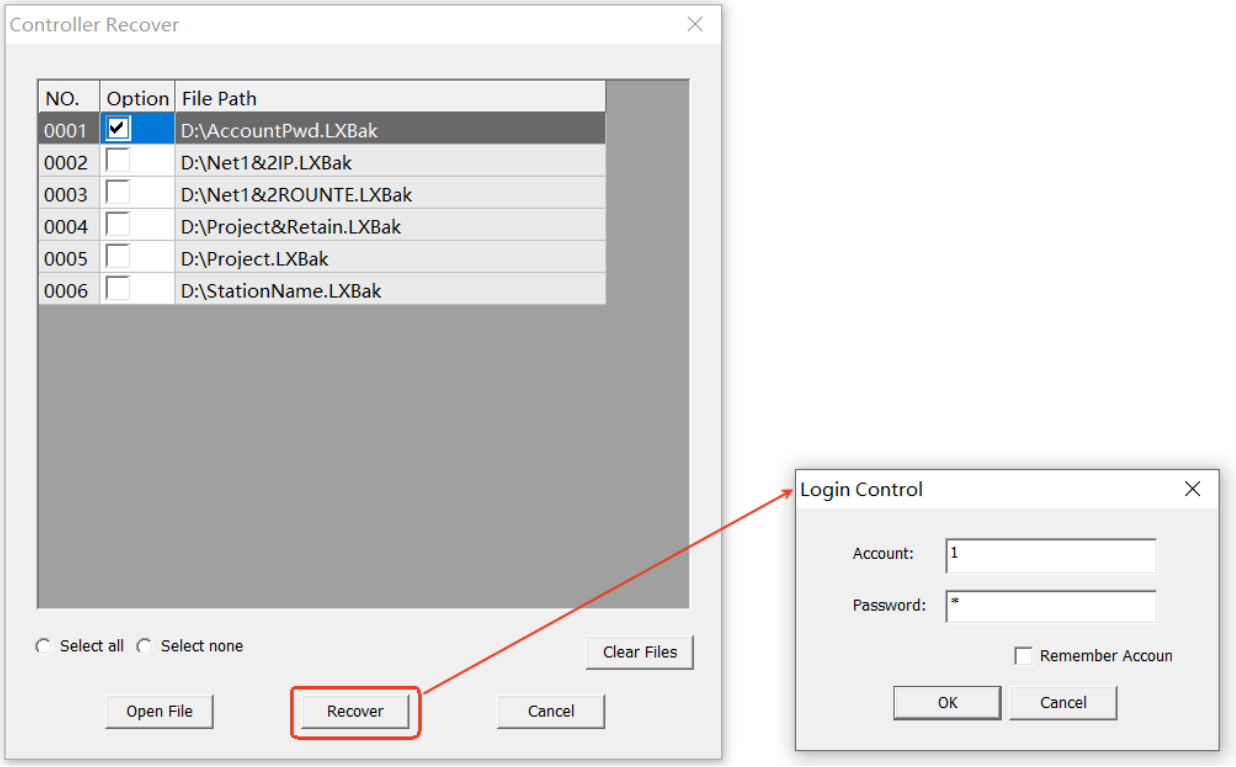
(1) Click the "Project" menu.

(2) Select "Backup" - "Controller Restore" command. A controller restore dialog box will appear, as shown in the figure below.



(3) To restore the backed-up files, verification is not supported during the restoration process. Click the Open File button, import the file(s) you want to restore and check them, then click the Recover

button and enter the controller account password. As shown in the figure below.



(4) A prompt dialog box will appear; click OK. The controller restoration is now complete.

6.17.3 Retain Var BackUp

6.17.3.1 Introduction

Back up the power failure protection variables of the controller. It supports backing up either part of the power failure retention variables or all power failure retention data. The power failure retention data backup file is in XML format.

6.17.3.2 Requirements

The controller is connected.

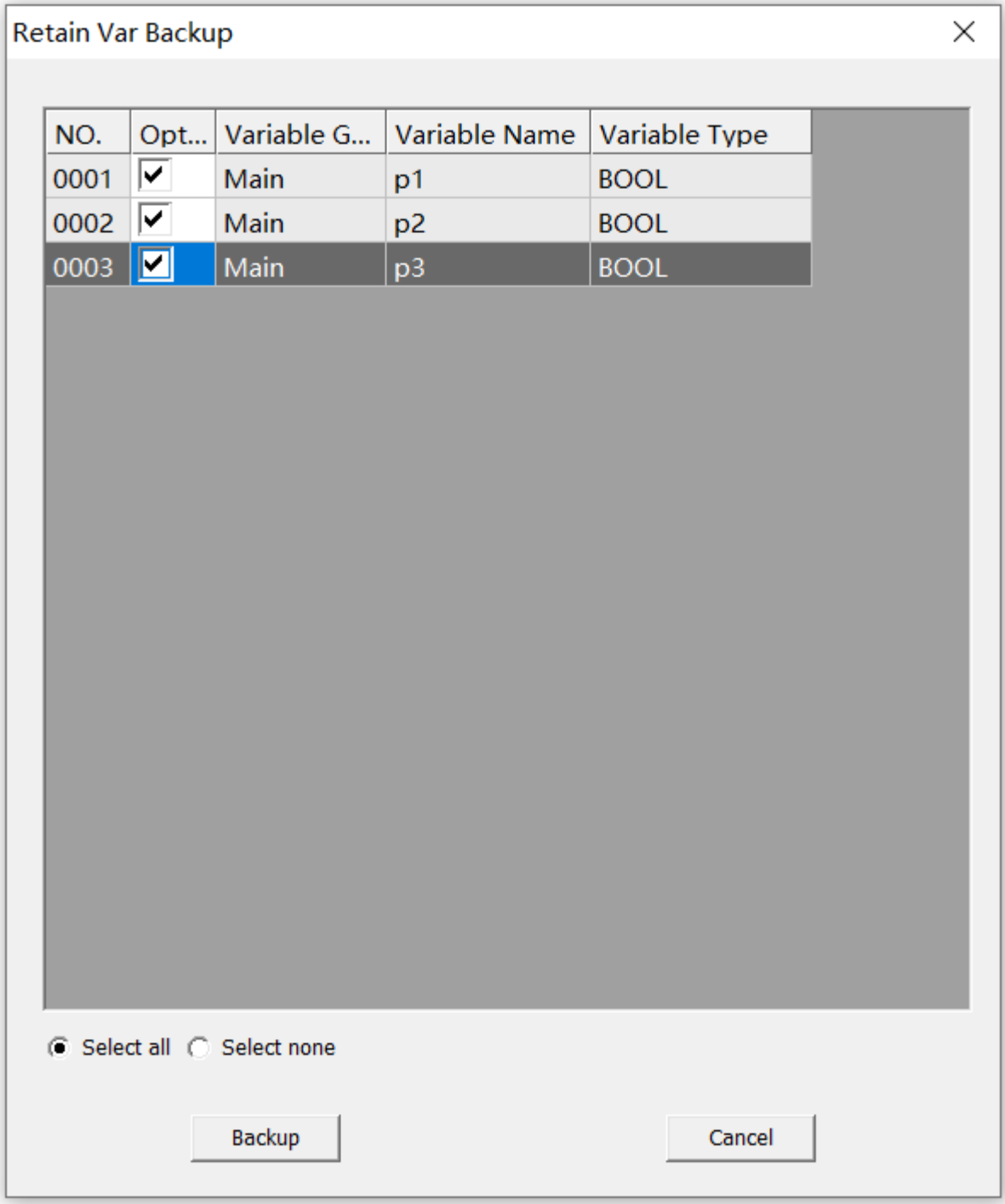
The project has entered the online monitoring status.

6.17.3.3 Steps

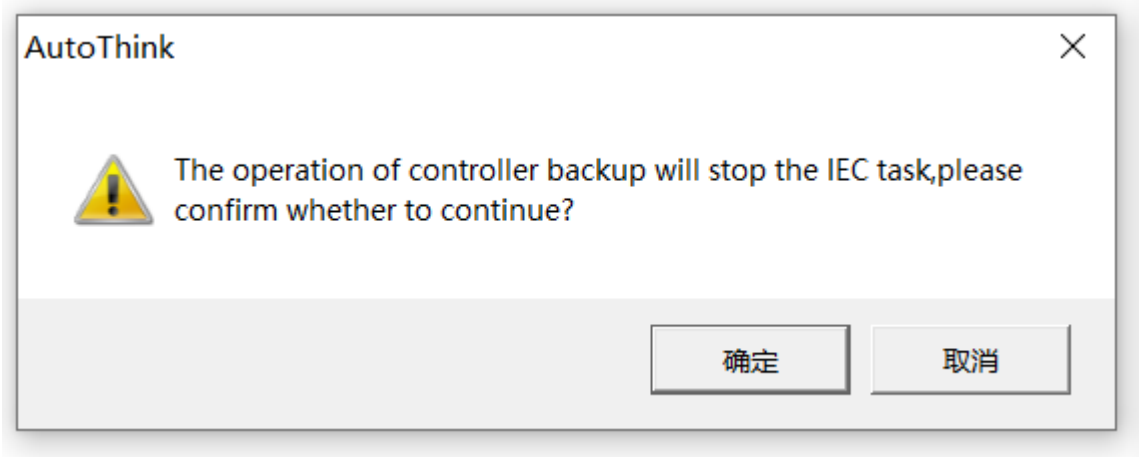
To perform a power failure protection variable backup, follow these steps:

(1) Click the "Project" menu.

(2) Select "Backup" - "Retain Var BackUp" command. A Power Failure Protection Variable Backup dialog box will appear. Choose the variables you want to back up and click the Backup button, as shown in the figure below.



(3) In the prompt dialog box that appears, click OK.



(4) Save the file to a local path to complete the power failure protection variable backup.

6.17.4 Retain Var Recover

6.17.4.1 Introduction

Restore the power failure retention data using the backup file.

6.17.4.2 Requirements

The controller is connected.

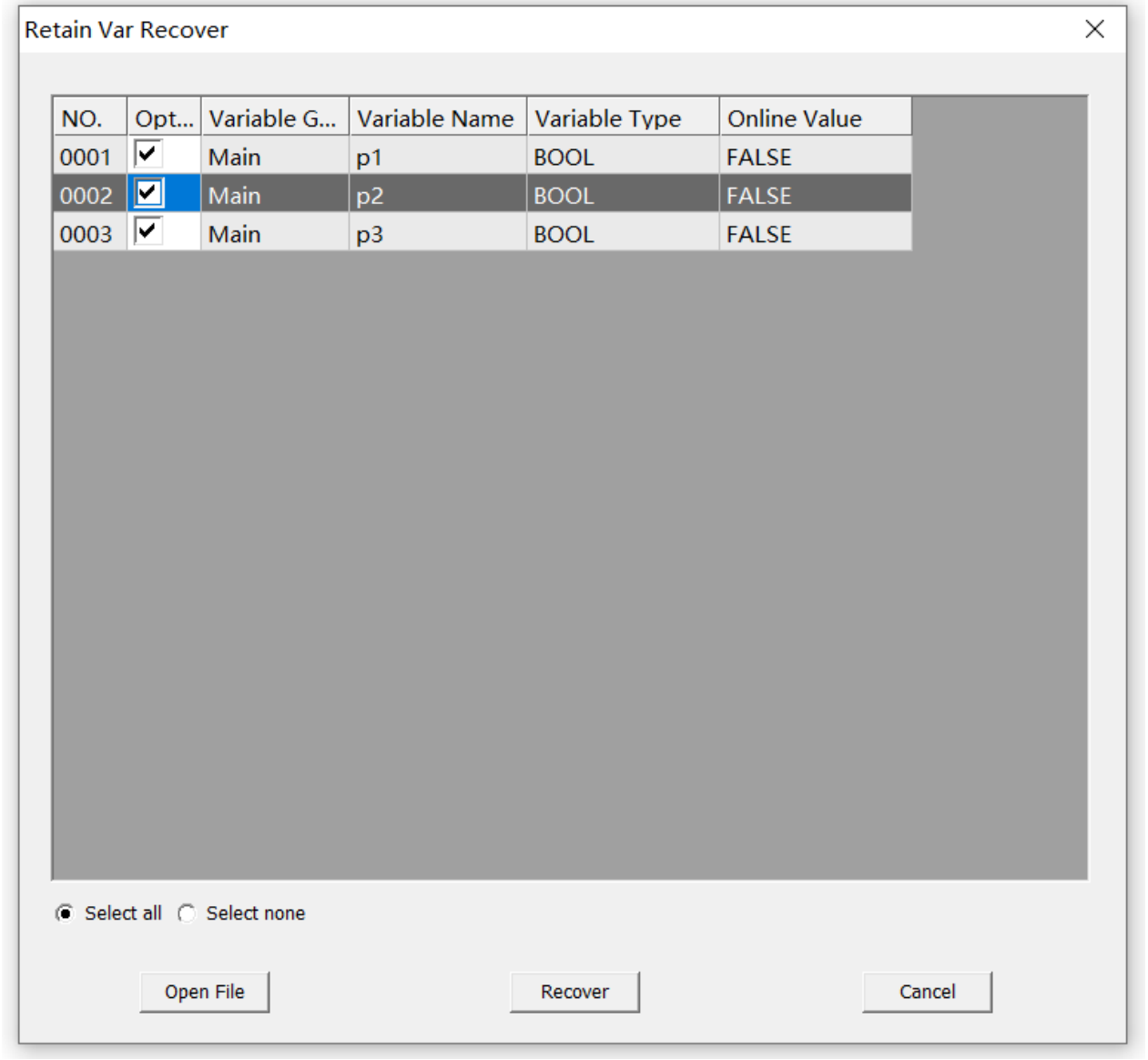
The project has entered the online monitoring status.

6.17.4.3 Steps

To perform a power failure protection variable recovery, follow these steps:

(1) Click the "Project" menu.

(2) Select "Backup" - "Retain Var Recover" command. A Power Failure Protection Variable Recovery dialog box will appear. Click the Select File button, import the file you want to restore, and check the variables. Click the Restore button to complete the power failure protection variable recovery, as shown in the figure below.


Retain Var Recover
NO.
Opt...
Variable G...
Variable Name
Variable Type
Online Value
0001
Main
p1
BOOL
FALSE
0002
Main
p2
BOOL
FALSE
0003
Main
p3
BOOL
FALSE
Select all
Select none
Open File
Recover
Cancel

Chapter 7 Permission Management

7.1 Set Default Project Permission

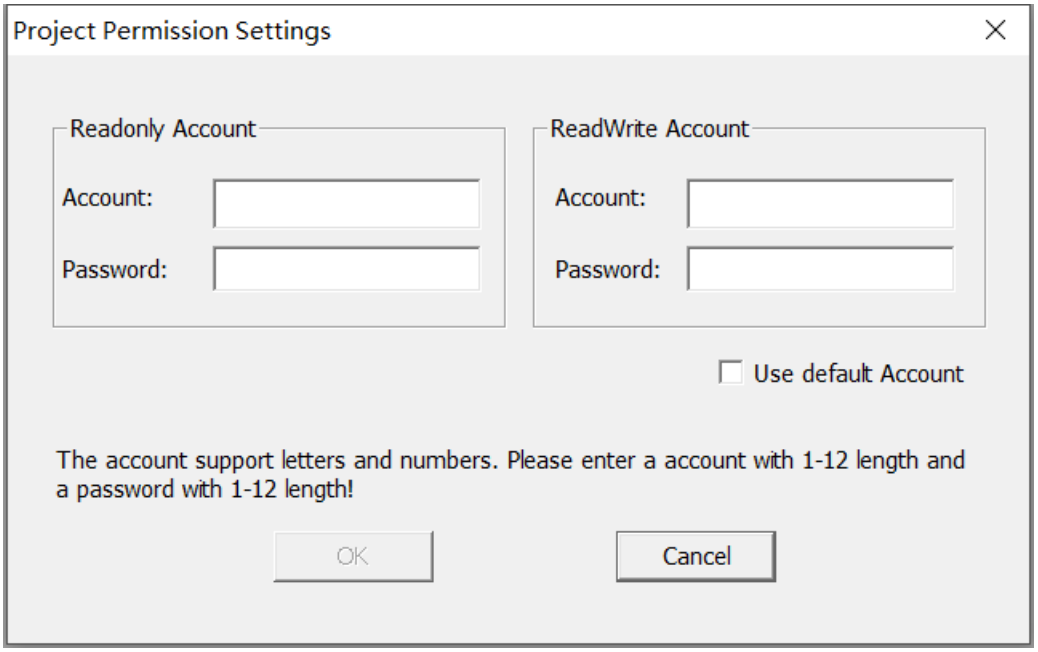
7.1.1 Introduction

AT software that runs for the first time allows users to set the default project permission. These permissions include read-only and read/write permissions. The permission can be used as a default permission to access any project. Users can also set their own project permission account for the project when they create a new project, which is only valid for accessing this project.

7.1.2 Requirement

- **Read/write permission**: It enables users to perform valid edits and operations on the project without constraints.
- **Read-only permission**: It enables users to view or monitor the project only. Under read-only permission, offline projects cannot be edited, and online projects that have been successfully installed can only be monitored.

After the default permissions have been set successfully, AT software starts running. When the software runs automatically or starts manually, the Default Permission Settings dialog box is displayed.



7.1.3 Steps

To set the default permissions, follow the steps below.

(1) Set Account and Passwords for Read-only Permission and Read/write Permission respectively.

- Naming: The username and password can be entered in both letters and numbers. Note that two accounts shall not have the same name.
- Length: 1-12 characters for username, and 1-12 characters for password.

(2) Click OK, and the Setting Succeeded dialog box is displayed.

(3) Click OK to complete the settings.

At this time, AT software opens. That means users can New Project or Open History Project.

7.1.4 Reference Message

Modify Project Permission

Create the new project

7.2 Modify Default Project Permission

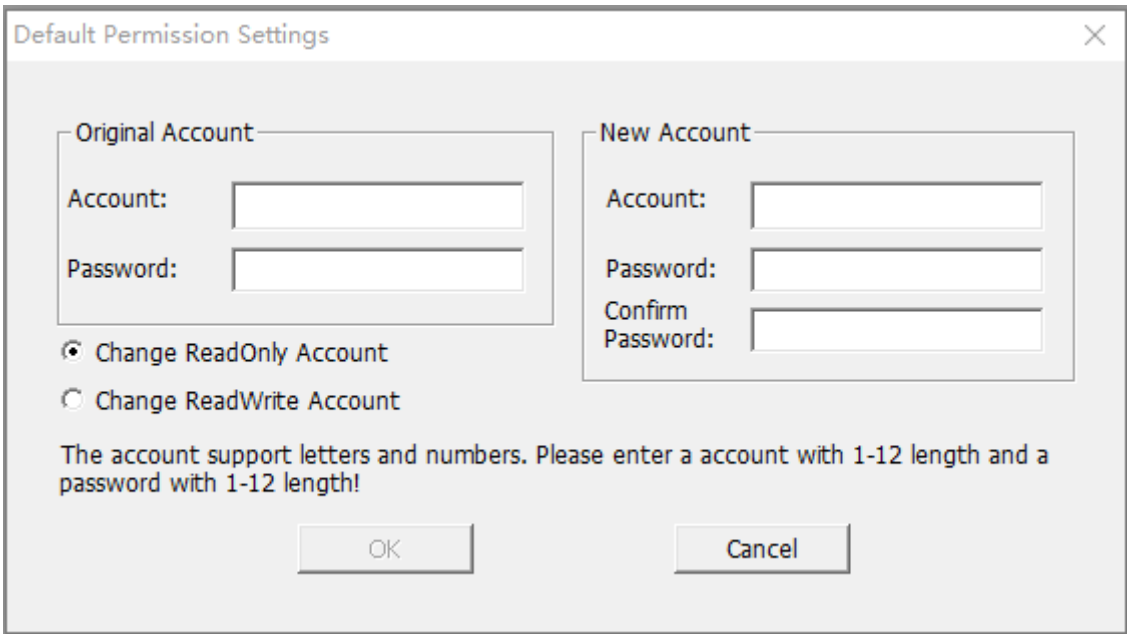
7.2.1 Introduction

Users can modify the default permissions for the project set in Set Default Project Permission.The permissions include Read-only permission and Read/write permission.

7.2.2 Steps

Modify the "Username" and "Password" for both "Read-Only Permission" and "Read-Write Permission" respectively.

(1) Click [Project] - [Default Permission settings], and the Default Permission settings dialog box is displayed.



(2) Select the permission account that needs to be modified, enter the current account information for the corresponding permission, and set up a new account.

- Naming: The username and password can be entered in both letters and numbers. Note that two accounts shall not have the same name.
- Length: 1-12 characters for username, and 1-12 characters for password.

The new account information must meet the setting requirements, and the password and confirmation password must be the same before the "OK" button can be used.

(3) Click OK, and the Setting Succeeded dialog box is displayed.

(4) Click OK to complete the settings.

7.2.3 Result

It enables default permission settings for projects.

7.3 Set Project Permission

7.3.1 Introduction

Users can use the Project Permission Settings command to modify the operating permissions of the project. The permissions are divided into read/write and read-only.

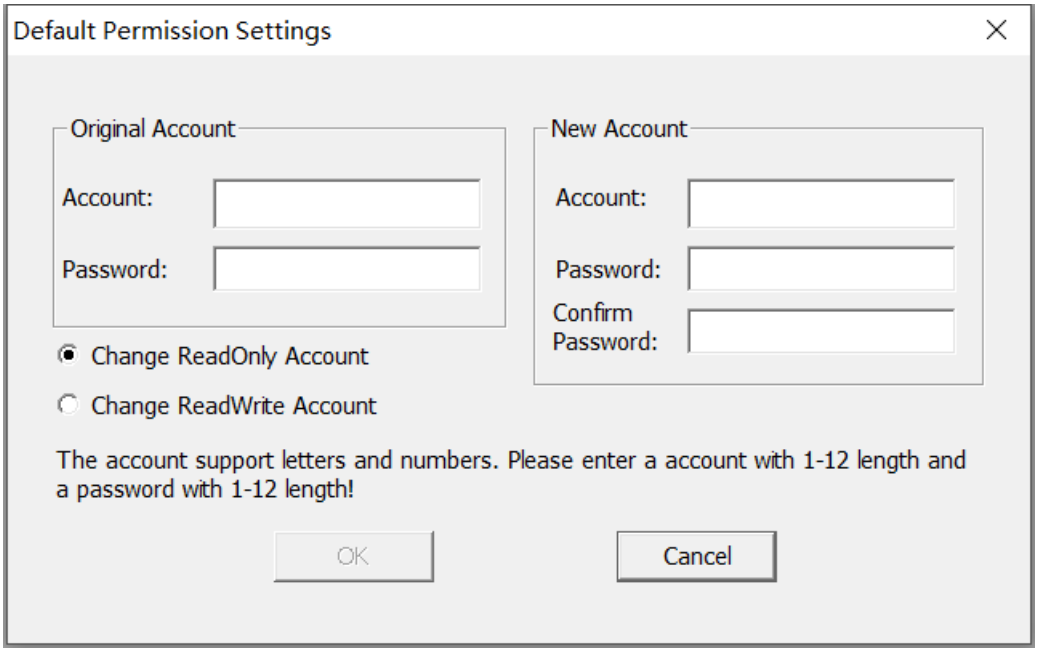
- **Read/write permission**: It means that users are not constrained to make valid edits and operations on the project. The Title Bar style of the project opened with this permission is shown in the figure below.

- **Read-only permission**: It means that users can only view or monitor the project. The Title Bar of a project opened with this permission is marked with ReadOnly, as shown in the figure below. Under read-only permission, offline projects cannot be edited, and online projects that have been successfully installed can only be monitored.

7.3.2 Steps

Set Account and Passwords for Read-only Permission and Read/write Permission respectively.

(1) Menu Bar: Click [Project] - [Project Permission Settings]. The Project Permission Settings dialog box is displayed.



(2) Enter the account information corresponding to the respective permissions. You can also check "Use Default Username and Password" to use the default account created in the Set Project Default Permission section.

- Content: The username and password can be entered in both letters and numbers.
- Length: 1-12 characters for username, and 1-12 characters for password.

Only if the new account information meets the setup requirements and the Password and Confirm Password are the same, the OK button can be operated.

(3) Click OK. The system prompts that the permission is set successfully and the new account information takes effect. If the system prompts that the account information is incorrect, please check whether the account information entered is consistent with the account information that needs to be checked.

(4) In the prompt box, click OK.

7.3.3 Result

It enables settings for project permissions.

7.4 Modify Project Permission

7.4.1 Introduction

Users can use the Project Permission Settings command to modify the operating permissions of the project. The permissions are divided into read/write and read-only.

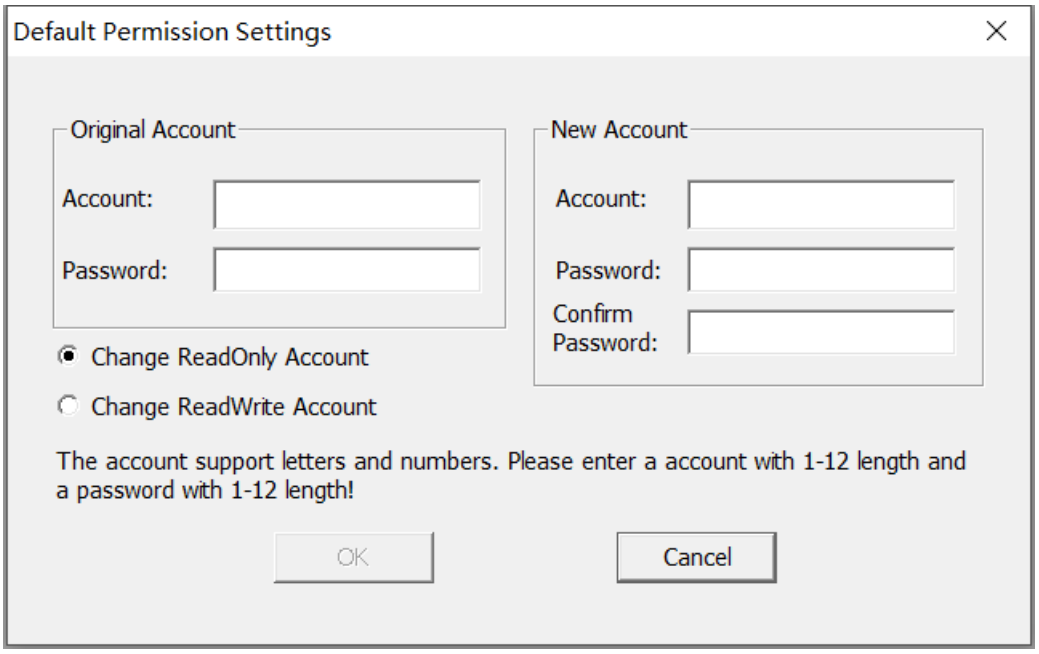
- **Read/write permission**: It means that users are not constrained to make valid edits and operations on the project. The Title Bar style of the project opened with this permission is shown in the figure below.

- **Read-only permission**: It means that users can only view or monitor the project. The Title Bar of a project opened with this permission is marked with ReadOnly, as shown in the figure below. Under read-only permission, offline projects cannot be edited, and online projects that have been successfully installed can only be monitored.

7.4.2 Steps

Set Usernames and Passwords for Read-only Permission and Read/write Permission respectively.

(1) Menu Bar: Click [Project] - [Project Permission Settings]. The Project Permission Settings dialog box is displayed.



(2) Check the permission accounts that need to be modified. Enter the current account information

corresponding to the permissions. And set up the new account.

- Content: The Username and Password can be entered in both letters and numbers.
- Length: 1-12 characters for Username, and 1-12 characters for Password.

Only if the new account information meets the setup requirements and the Password and Confirm Password are the same, the OK button can be operated.

(3) Click OK. The system prompts that the permission is set successfully and the new account information takes effect. If the system prompts that the account information is incorrect, please check whether the account information entered is consistent with the account information that needs to be checked.

(4) In the prompt box, click OK.

7.4.3 Result

It enables modification of project permissions.

7.5 Set Controller Login Account

7.5.1 Introduction

All online operations executed on the controller require a successful login to the controller in order to proceed. Before operating online, users need to set up a login account for the controller, which is valid for all projects accessing the controller.

The following two operations clear the controller account. That means users need to reset the controller account after executing the operations.

- To reset factory settings
- To upgrade the firmware in the tool disk via SD card

7.5.2 Requirement

The Controller Operation tool is successfully connected to the controller.

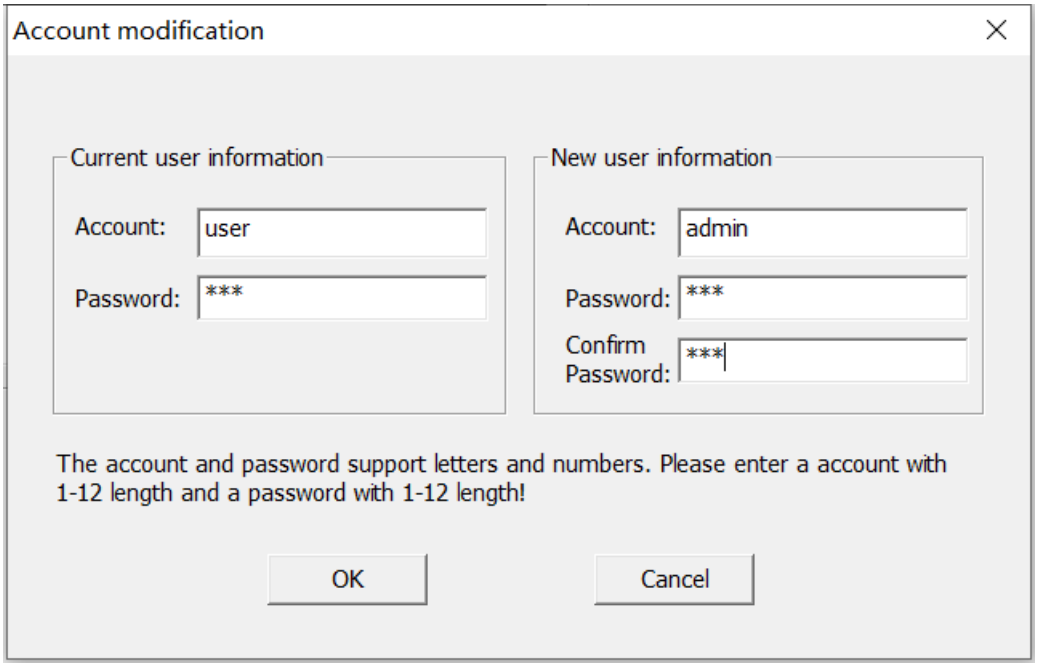
7.5.3 Steps

To set up the controller login account, follow the steps below.

(1) Click [Tool] - [Assistant tool] - [Controller Operation].

(2) In the Controller Information tab, click the Modify Account button. The Account modification dialog

box is displayed.



(3) For the first time setup, just enter the new account information. To modify the account, please enter the Current account information and New account information.

(4) Enter Usernames and Passwords. And click OK. The Controller account information setting successful prompt box is displayed.

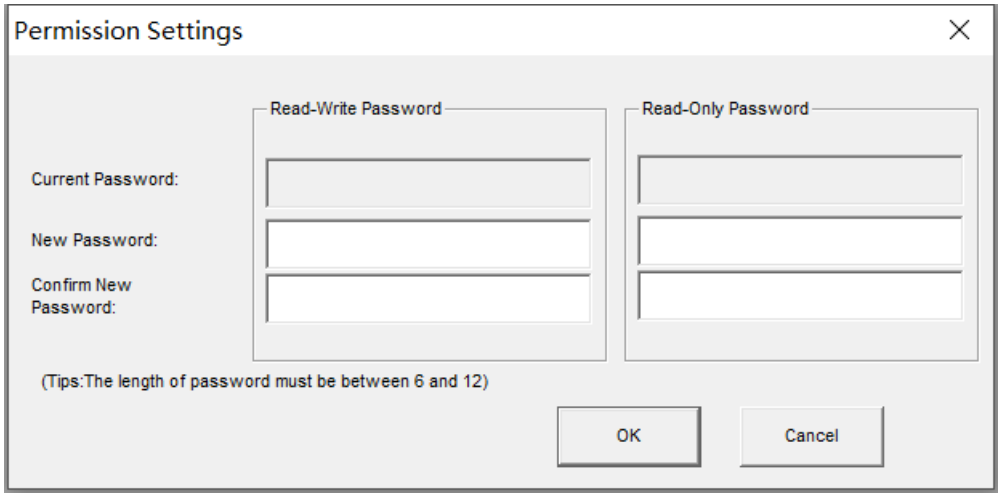
(5) Click OK to have the new account take effect.

7.6 POU Permissions

7.6.1 Introduction

Users can set it via:

Project Management Tree: Right-click the POU name. Click Permission Settings. The Permission Settings dialog box is displayed, as shown in the figure below. Users can set Read/write password and Read-only password for POU respectively.



- After users set permissions on a POU in SFC language, all actions added under that POU are also granted the corresponding permissions.

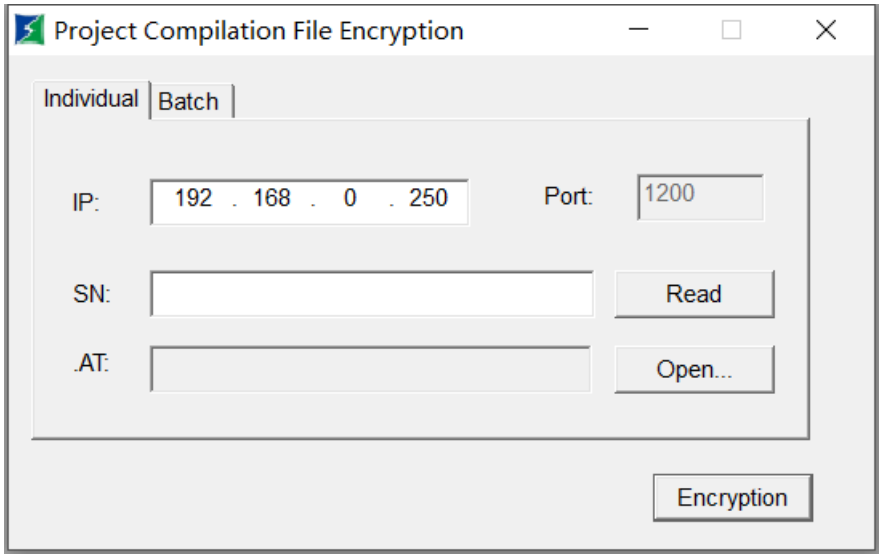
7.7 Encrypt Target Project Files

7.7.1 Separate Encryption

7.7.1.1 Introduction

Users can associate the SN number of the current controller with the .at file. The target project file encrypted can only be installed or transferred to the controller corresponding to this SN number.

Open the encryption tool. The dialog box is displayed as shown in the figure below.



7.7.1.2 Result

It enables encryption of target project files.

7.7.2 Bulk Encryption

7.7.2.1 Introduction

Users can bind the SN numbers of multiple controllers with .at files, encrypt them, and generate multiple .key files.

7.7.2.2 Steps

(1) Create a .csv file

Create a new .txt file. In the document, enter the controller SN numbers to be bulk encrypted. The SN numbers can be separated by English commas (e.g., 1, 2, 3) or by direct line feeds, as shown in the figure.

```
1, 2, 3
```

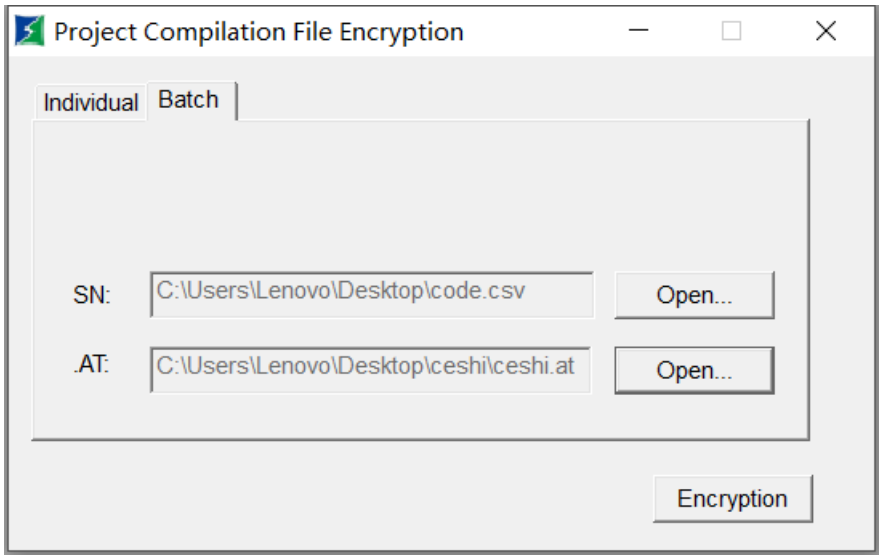
```
1
2
3
```

After saving, change the suffix of .txt to .csv. The .csv file containing the batch SN information is

generated, as shown in the figure.

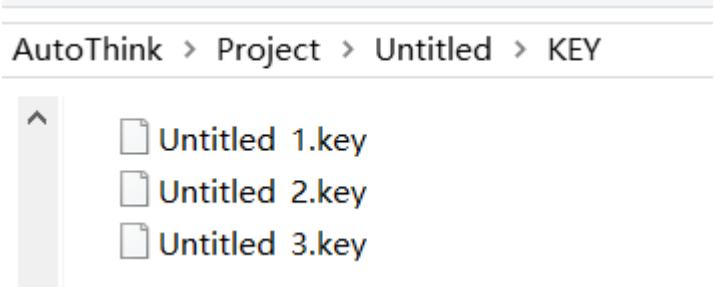
(2) Encrypt

Click the Batch tab to open the Project File Encryption window, as shown in the figure below.



Click Open after SN. Select the .csv file saved before. Click Open after .AT. Select the .at file to be loaded. After selecting, click the Encrypt button to batch encrypt the .sn files corresponding to the .at files. After the encryption is completed, the system prompts Generate encrypted file successfully.

The .at file encrypted is named in the same way as the separate encrypted file. Multiple .key files corresponding to the SN numbers are generated under the key folder. Users can transfer the encrypted .key file to the controller with the corresponding SN number via the Project Upgrade tool. An inconsistency between the SN number of the controller and that of the .key file may cause the transfer to fail.



- When the encryption is repeated for .at files of the same project (the project name and SN number are the same), the replacement prompt is displayed for separate encryption, while no prompt is displayed for bulk encryption and the replacement operation is executed directly.

7.7.3 Result

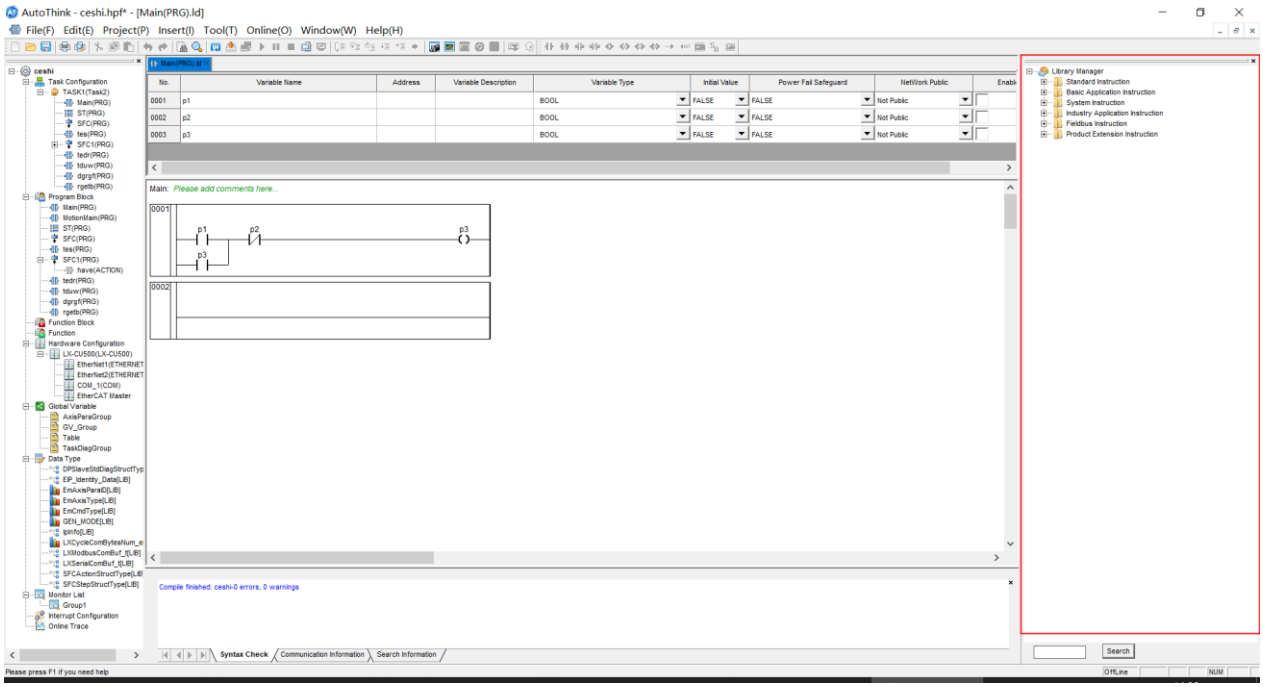
It enables bulk encryption of target project files.

Chapter 8 Library Manager

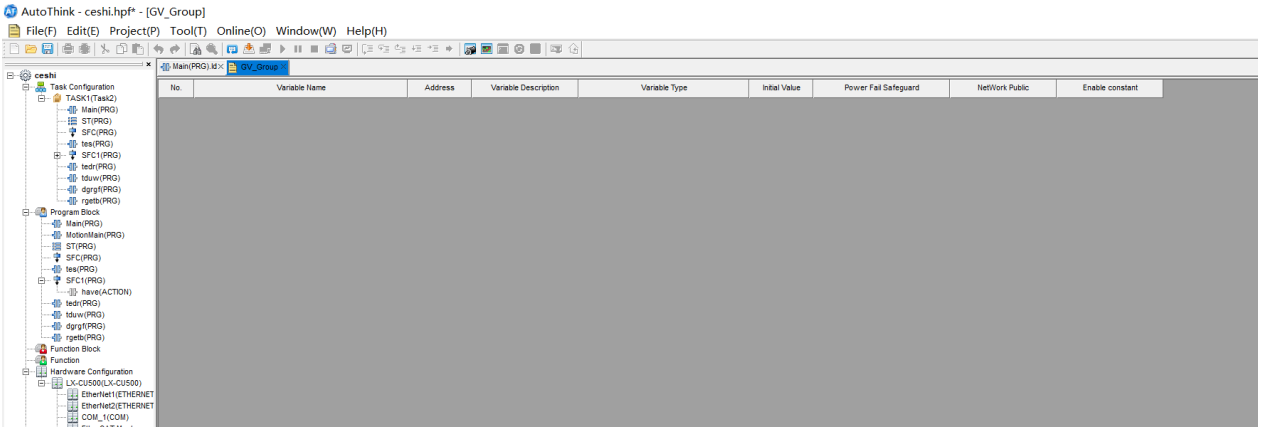
8.1 Library Window

8.1.1 Open Library Window

When the POU editing window is opened for the first time, the system automatically loads the Library Manager window.



If the editing window of another node in the Project Management Tree is opened, such as the Global Variables editing window, the POU editing window and Library Manager window are automatically switched to an inactive state and hidden behind the Global Variables editing window.



8.1.2 Introduction

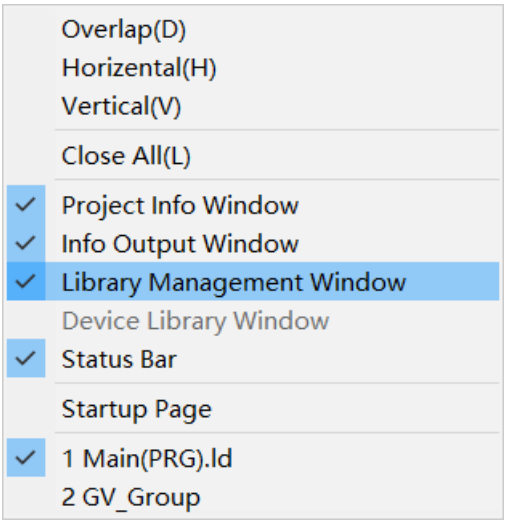
By clicking the POU editing window tag, users can switch back to the POU editing window display, which displays both the POU editing window and the Library Manager window at the same time.

8.1.3 Steps

Before switching, if the Library Manager window has been closed manually (by clicking Close in the upper-right corner of the Library Manager window) or by the Window menu > Library Manager Window command, the Library Manager window is no longer loaded automatically when switching back to the POU editing window.

That is, the Library Manager window has the state memory function. Before and after the window switching operation, it remembers whether the previous POU editing window is associated with the Library Manager window. If there is no previous association, then even if users switch back to the POU editing window, the Library Manager window is not loaded automatically.

At this time, users can use the Menu command to manually load the Library Manager window, as shown in the figure.



The relationship between the Device Library window and the Hardware Configuration window is similar and will not be repeated.

8.2 Library Configuration

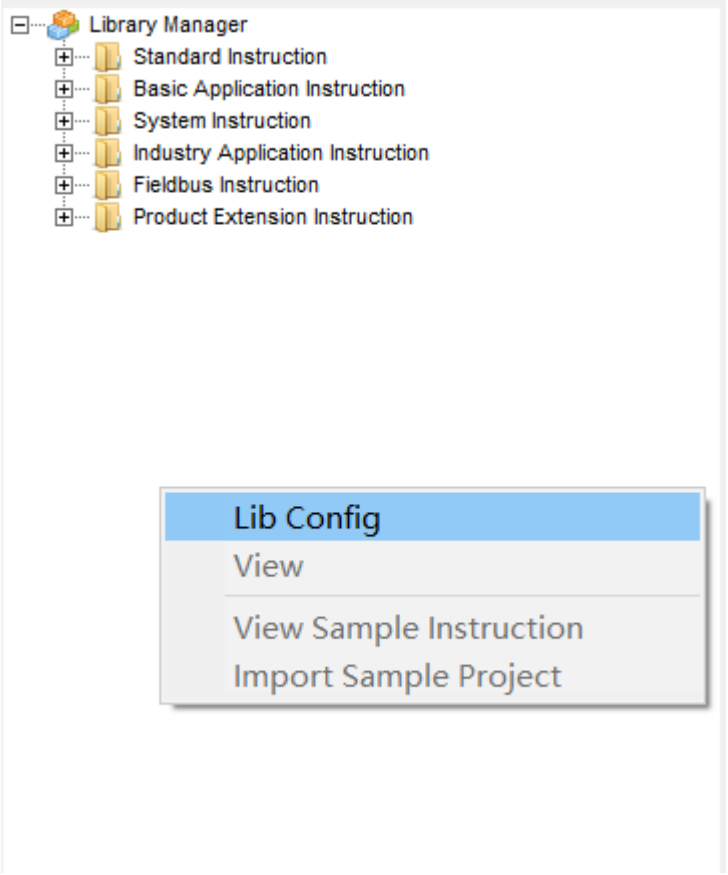
8.2.1 Introduction

In the Library Configuration window, users need to enable the relevant library files before using the corresponding library instructions. Otherwise, the compilation error may occur when using the instructions.

8.2.2 Steps

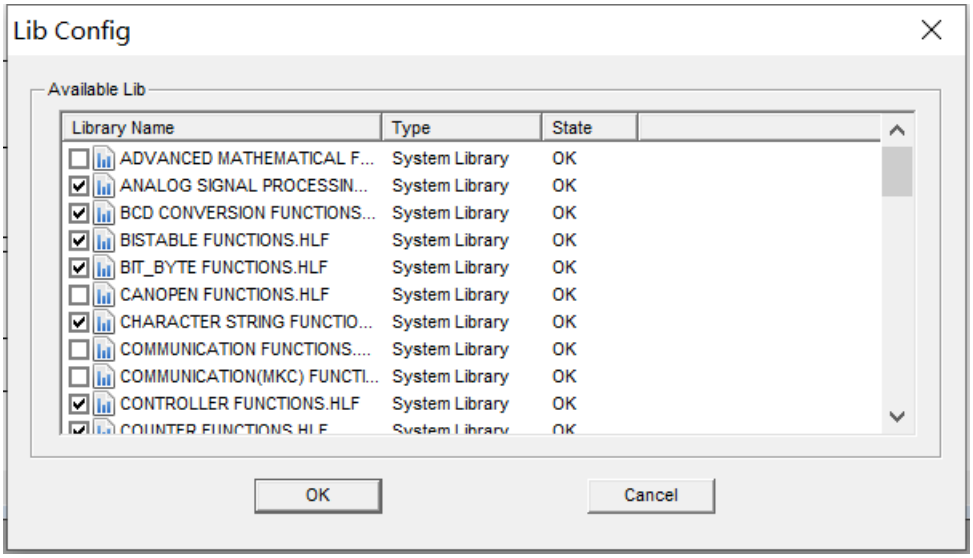
Users can open the library configuration via:

- Library Manager: Right-click an empty area, and click Library Configuration.



The Library Configuration dialog box is displayed, as shown in the figure. Check the library files that need to be enabled. And click OK to have the library files take effect. Users can call instructions under the validated library file.

- In the newly created project, the library files displayed by default under the Library Manager are enabled and can be used normally.

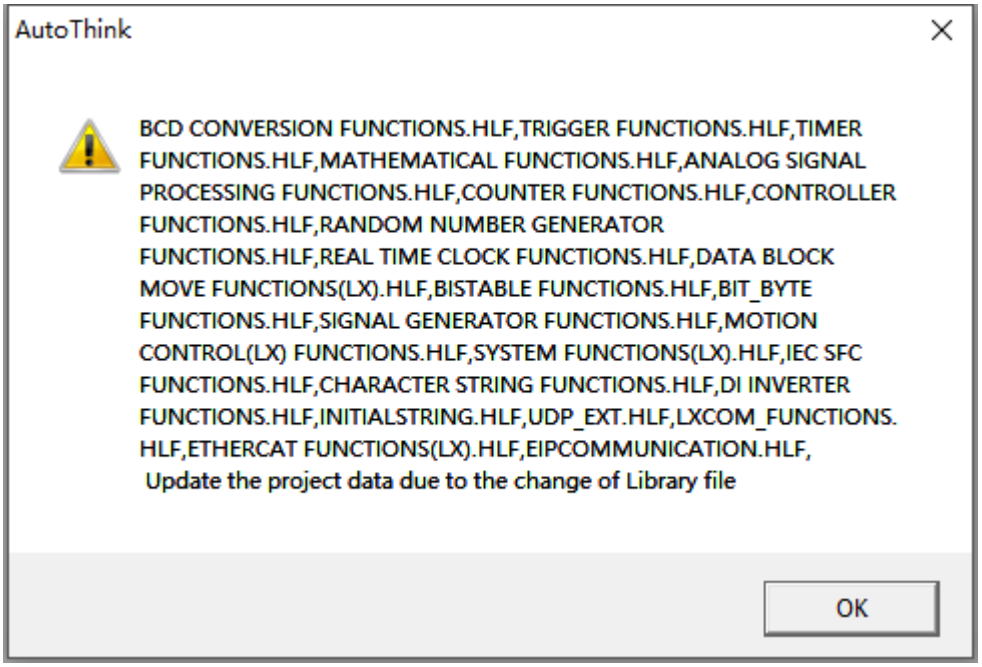


8.3 Update Library

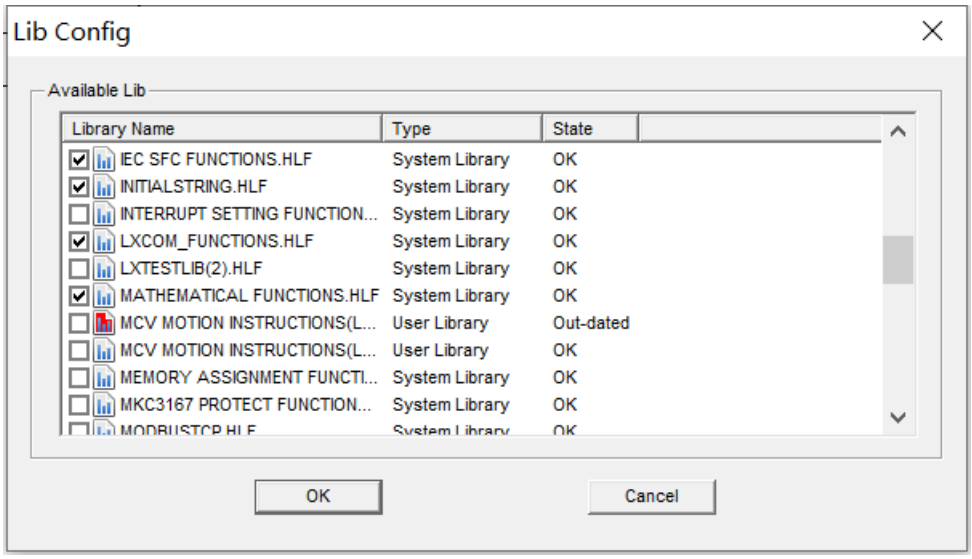
8.3.1 Library Detection

8.3.1.1 Introduction

After the project is opened, AutoThink software automatically detects whether the version information of the library referenced in the project is consistent with the version information of the library file that exists under the default path. When the detected version information is consistent, the system does not prompt to update the library; And when the version information of the library changes, the system prompts to update the library in sequence, as shown in the figure.



Click OK to close the prompt box. In the Library Configuration dialog box, the abnormal library file is displayed at this time, as shown in the figure. When compiling, the message window prompts for an update to the library file.



- After the library has been updated, check whether the library status in the Library Configuration window is normal. Please contact our professionals for assistance if any abnormal status is found.

8.3.2 Automatic Library Updates

8.3.2.1 Introduction

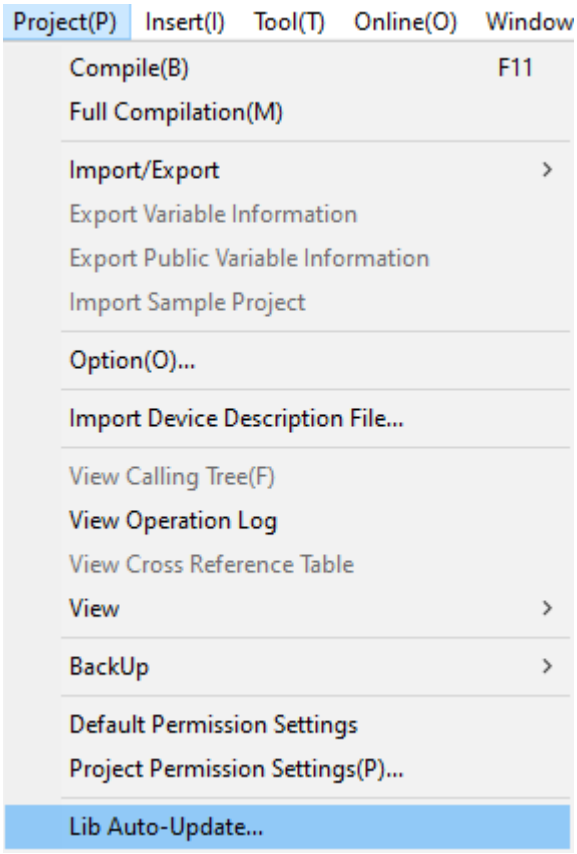
The function of Automatic Library Updates is intended for the situation that the .trg file of AutoThink software has been modified, or the software has been upgraded. When an old version of the project is opened, if the library files in the Library directory are not updated, these library files fail to be added to the current project normally, resulting in the inability to refer to the various instructions in the library. This version of the software enables automatic update of library files in the Library directory without manual replacement or configuration.

There are three scenarios in which Automatic Library Updates are required:

- When a newly created library file is added, and the library file has inconsistency between its .trg file ID number and the ID number recorded when it was compiled and saved, the library file fails to be added normally.
- When the software has been upgraded and the old version of the project is opened with the new version, the library file fails to be added normally and the library cannot be used in the project due to the inconsistency of the two serialized version numbers.
- When there is a reference between library files, that is, one library references another library, if the latter change, the former also needs to be automatically updated.

The following is an example of how to use Automatic Library Updates in a scenario where there is a reference relationship between libraries.

When there is a reference relationship and the referenced library changes, the referenced library cannot be used normally. At this time, the software brings up a prompt box, recommending the use of automatic updates in menu Bar: [Project] - [Lib Auto-Update].



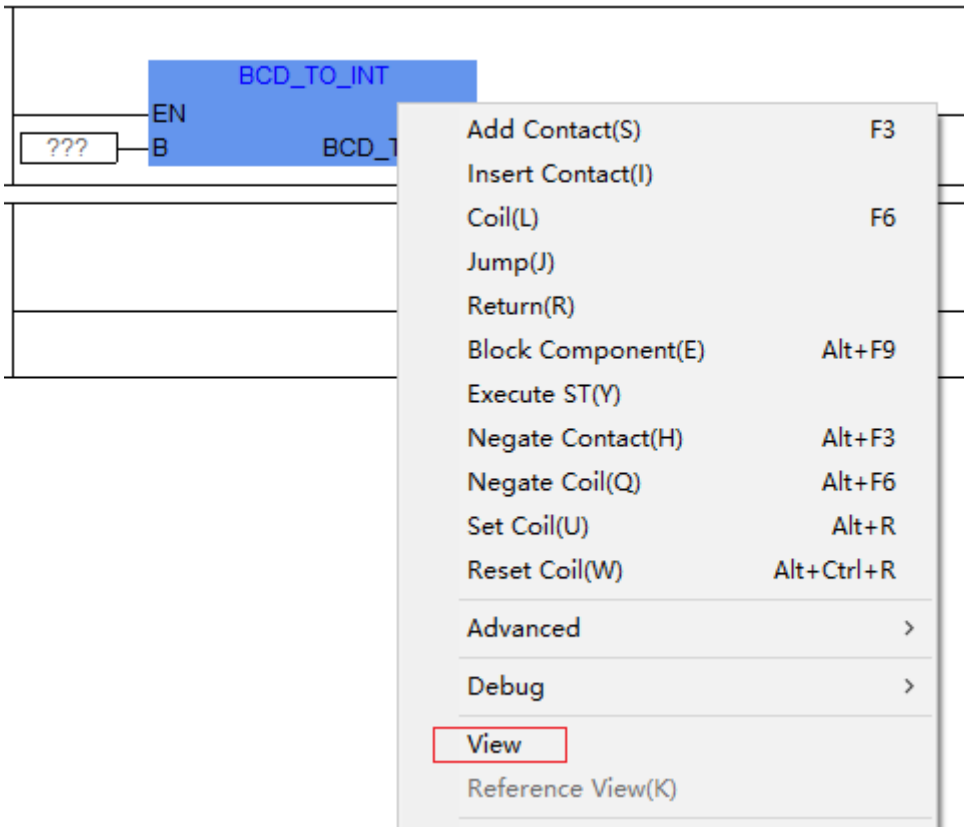
8.4 View Library Information

8.4.1 Introduction

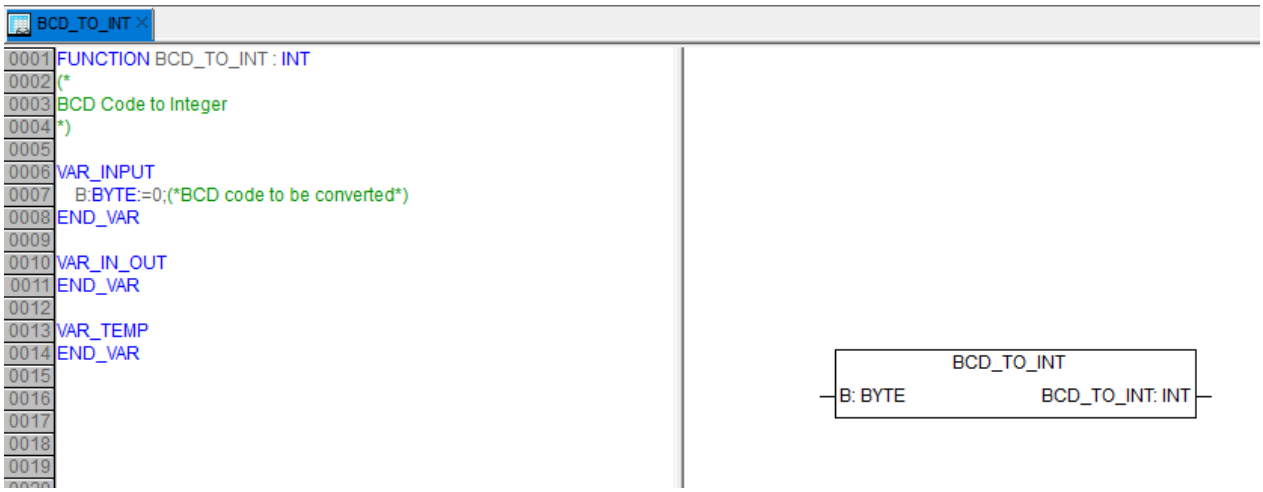
This command allows you to view information about library instructions.

8.4.2 Requirement

Library Manager: Right-click the instruction name, and click View.



The message window for this library instruction is displayed in the left editing area: Name (function type), Comments, Input Variables, Output Variables, Input-Output Variables, Local Variables, Legend, and other messages.



Chapter 9 Hardware Configuration

9.1 Hardware Configuration Operation

9.1.1 Copy, Paste, and Cut in Hardware Configuration

Configuration under the controller supports copy, paste, and cut operations, allowing these actions between multiple engineering projects, but not across different versions.

9.1.1.1 Requirements

The node under the controller must have a configuration project.

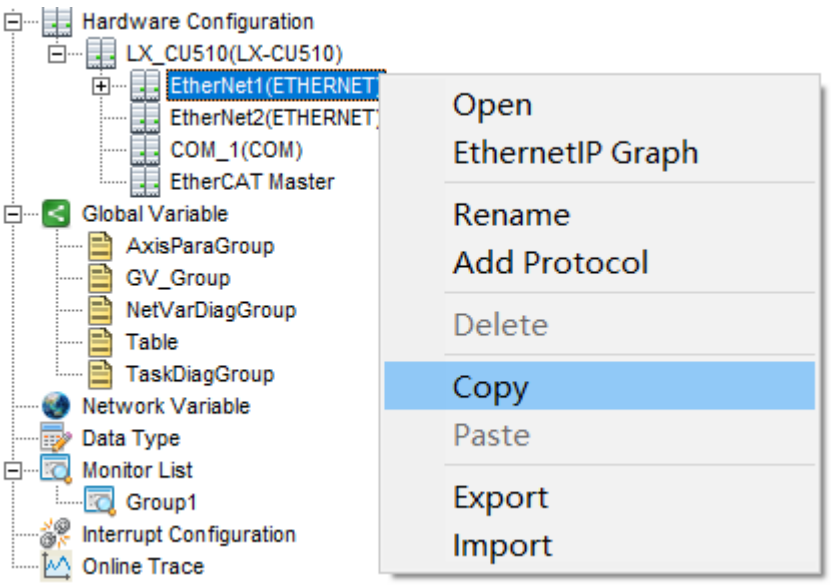
9.1.1.1.1 Steps

- Full Paste

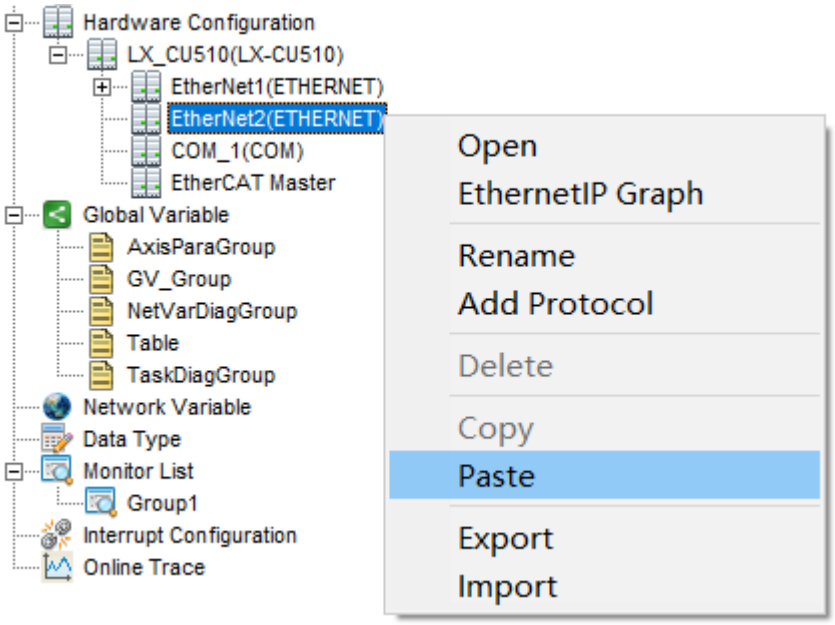
  Ethernet1/2, COM_1, and EtherNet_Master nodes support copy and paste operations.

  Example with the Ethernet node: Perform the configuration copy, paste, and cut operations as follows:

(1) Right-click on the “Ethernet1” node and select the “Copy” command.

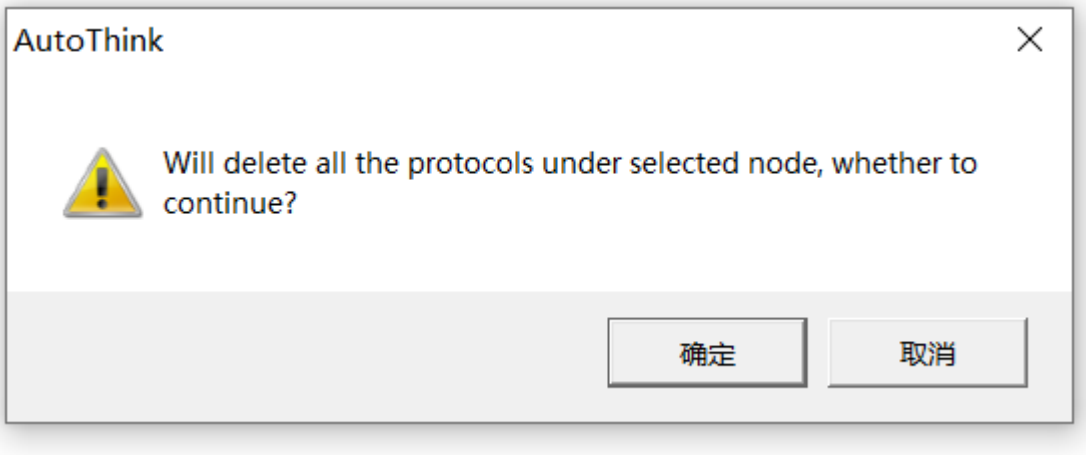


(2) Right-click on the “EtherNet2” node and select the “Paste” command.

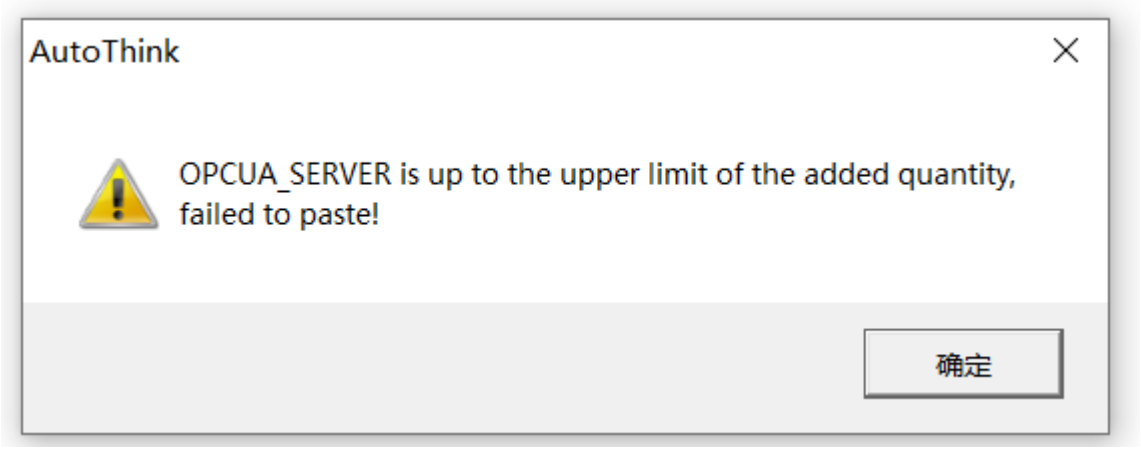


(3) The configuration from EtherNet1 will be pasted under EtherNet2.

If there is already a configuration under EtherNet2, a prompt dialog box will appear when performing the paste operation, as shown in the figure.



When the number of configuration protocols added in the project reaches the limit, a paste failure dialog box will pop up, as shown in the figure.

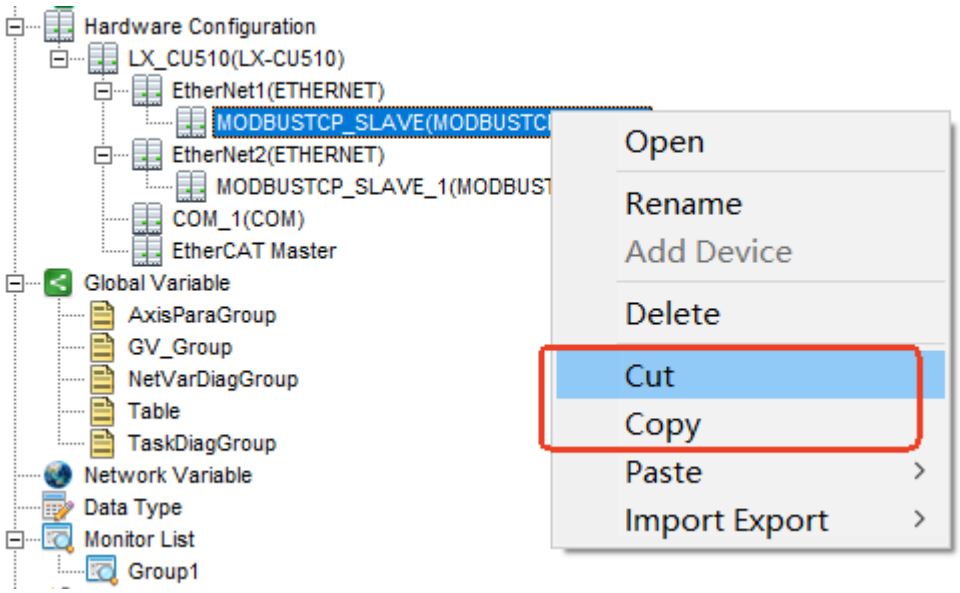


■ Partial Paste

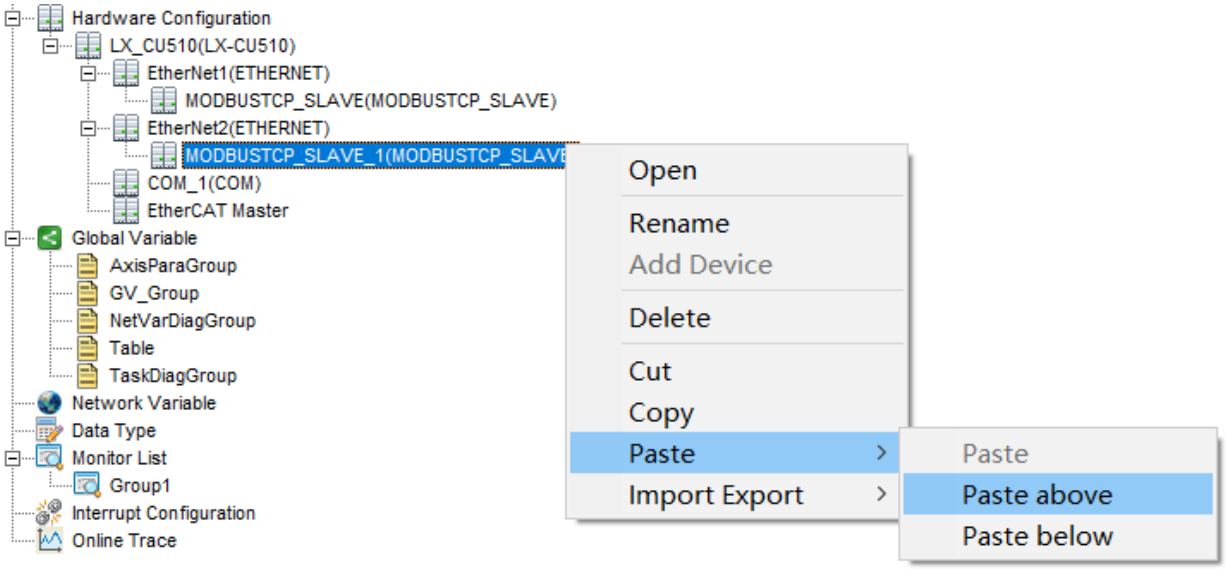
Configuration for Ethernet1/2, COM_1, and EtherNet_Master nodes supports copy, paste, and cut operations.

Example with the EtherNetIP_Scanner configuration under the EtherNet node: Perform the configuration copy, paste, and cut operations as follows:

(1) Right-click on the “EtherNetIP_Scanner” node and select the “Copy” or “Cut” command.



(2) Right-click on the target location node and select the “Paste” command. You can choose to paste the content above or below the target location.



(3) The EtherNetIP_Scanner configuration will be pasted to the target location.

9.1.2 Modify Hardware Configuration Node Name

After a node is added, you can modify its name using the "Rename" command in the right-click menu.

When renaming hardware configuration nodes, the following rules should be adhered to:

- The hardware configuration node name recognizes underscores.
- The hardware configuration node name is case-insensitive; for example, COM_1 and Com_1 refer to the same hardware configuration node.
- The hardware configuration node name cannot be empty and must not contain spaces. For instance, COM 1 is an incorrect hardware configuration node name.
- The hardware configuration node name must not contain special characters such as hyphens "-" and plus signs "+". For example, COM-1 and COM+1 are incorrect hardware configuration node names.
- Hardware configuration nodes must not have identical names.

- When opening a project created with FA-AT V3.2.2SP1 or earlier versions using FA-AT V3.2.2SP1 or a later version, if the hardware configuration node name contains a "-", it will automatically be replaced with "_".

9.1.2.1 Requirement

The node has already been added.

9.1.2.2 Steps

To modify the hardware configuration node name:

- Right-click the node whose name you want to modify, select "Rename" from the context menu.Enter the new name for the node.

  The hardware configuration node name cannot contain special characters, such as hyphens "-" or plus signs "+".

9.1.3 Export Hardware Config

9.1.3.1 Full Export

To perform a full export of the hardware configuration, follow these steps:

- Menu Bar: Click on [Project] - [Import/Export] - [Export Hardware Configuration].
- Hardware Configuration Node: Right-click on [EtherCAT Master] - [Export].

9.1.3.2 Partial Export

To perform a partial export of the hardware configuration, follow these steps:

- Hardware Configuration Node: Right-click on the EtherCAT Master slave device - [Import/Export]—[Export].

9.1.4 Import Hardware Config

9.1.4.1 Full Import

To perform a full import of the hardware configuration, follow these steps:

- Menu Bar: Click on [Project] - [Import/Export] - [Import Hardware Configuration].
- Hardware Configuration Node: Right-click on [EtherCAT Master] - [Import].

9.1.4.2 Partial Import

To perform a partial import of the hardware configuration, follow these steps:

- Hardware Configuration Node: Right-click on the EtherCAT Master slave device - [Import/Export] - [Import Above]/[Import Below].

9.1.5 Graphical Configuration

Graphical configuration is divided into full network graphical display and single protocol network graphical display, supporting zoom in and zoom out of the view. You can open it through the following methods:

- Full Network Graphical Display: Double-click the "Hardware Configuration" tree node .
- Single Protocol Network Graphical Display:
  - EtherNet/IP: Right-click the EtherNet node and select “EthernetIP Graph”.
  - EtherCAT: Right-click the EtherCAT Master node and select “EtherCAT Graph”.

The graphical configuration interface allows you to add nodes by dragging and dropping, and supports adding, deleting, and entering the hardware configuration parameter interface for individual nodes by double-clicking. Any changes to node information (such as adding, deleting, or renaming) in the graphical configuration interface will automatically update the configuration tree. For operations performed on the configuration tree (such as adding, deleting, or renaming), you need to manually refresh the graphical configuration interface to achieve real-time data synchronization and view refresh.

9.2 Configure Ethernet Port

9.2.1 Introduction

The LX controller Ethernet port supports Modbus TCP master-slave protocol, free communication protocol, and EtherNetIP slave station protocol.

9.2.2 Configure Modbus TCP

9.2.2.1 Configure Modbus TCP master station

9.2.2.1.1 Introduction

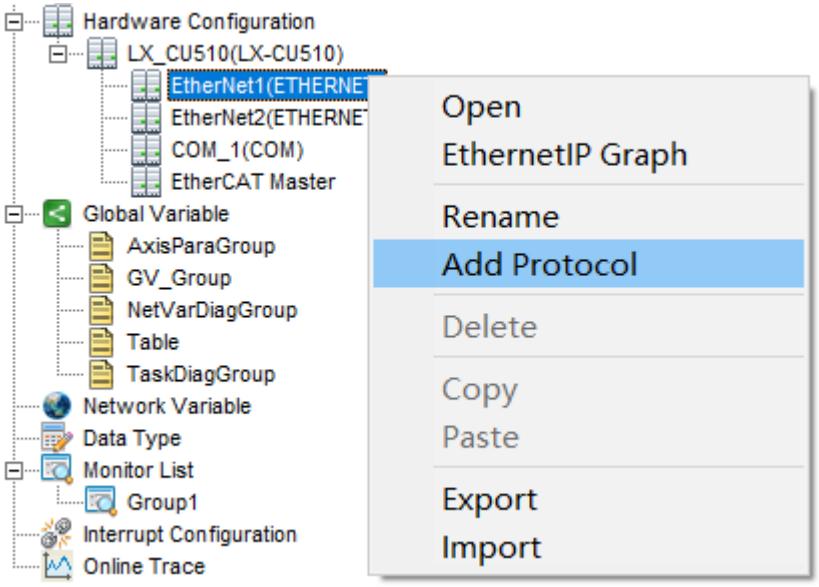
After the controller is added, the ETHERNET node is automatically generated under the Controller Tree node. The master and slave station configuration of Modbus TCP is completed under this node.

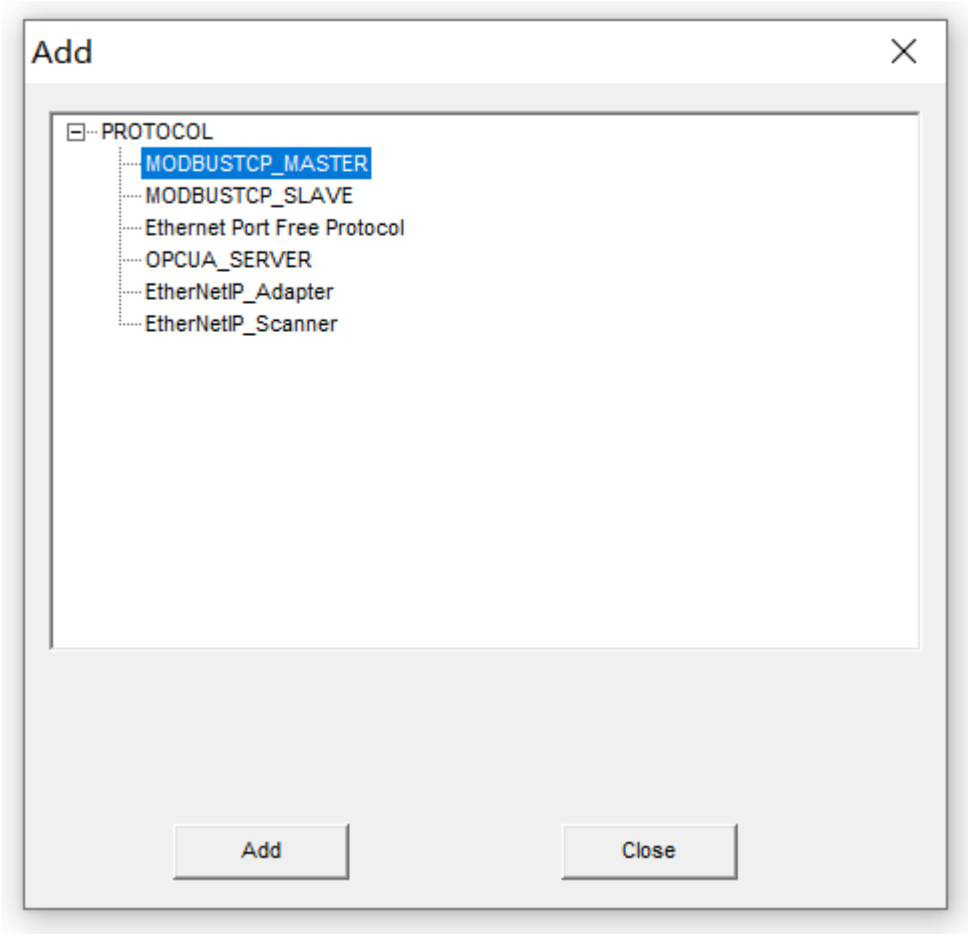
9.2.2.1.2 Steps

To use the controller as a master station for Modbus TCP, follow the steps below:

(1) Add master station protocol

In the right-click menu of the ETHERNET node, select the Add Protocol command. The Add dialog box is displayed, as shown in the figure. And select the MODBUSTCP_MASTER protocol.

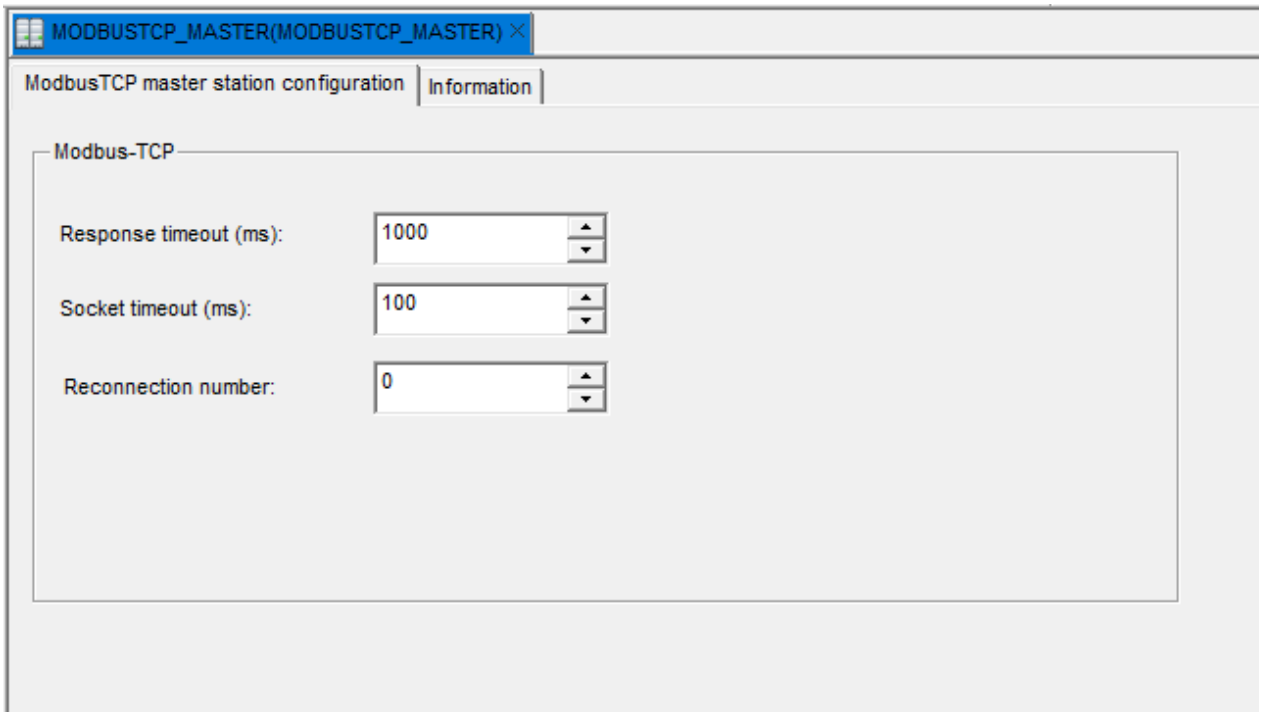




(2) Configure Modbus TCP master station

By double-clicking the MODBUSTCP_MASTER node or selecting the Open command in the right-

click menu, open the MODBUSTCP_MASTER window, as shown in the figure.



Configuration Parameters for Master Station Communication

Parameter	Parameter Value	Default Value	Description	Remarks
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	

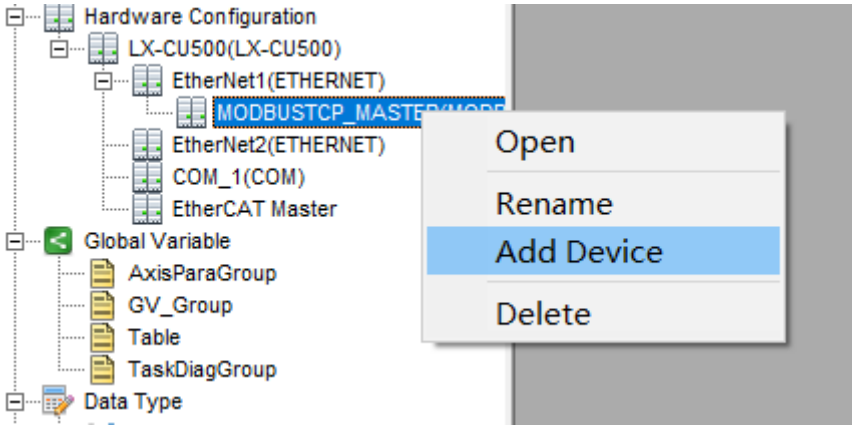
(3) Configure Modbus TCP slave station

When the current CPU is the master station, one or more slave stations can be configured for data communication.

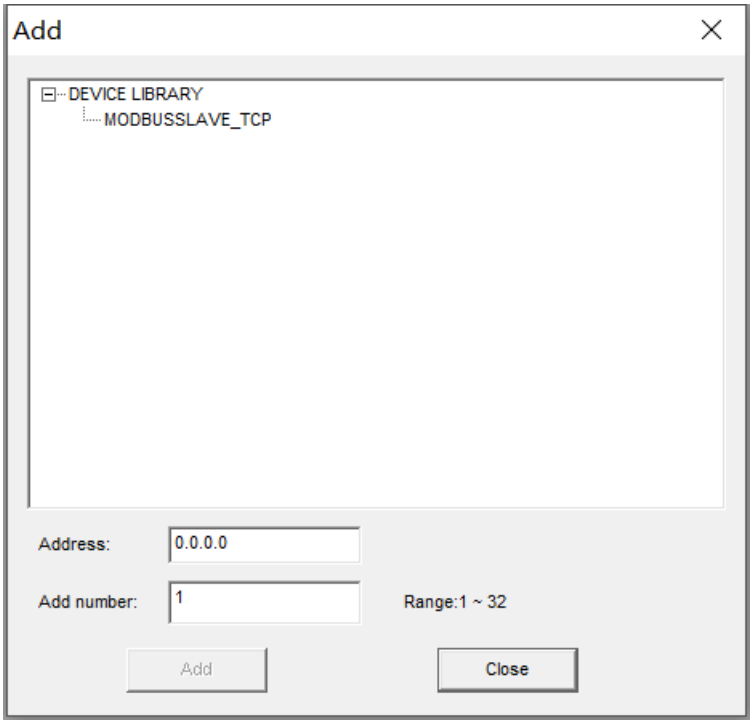
When adding Modbus TCP slave stations, please note that there is an upper limit on the total number of Modbus TCP master-slave connections for LX series controllers. If the total number of connections exceeds the upper limit, a compilation error may be reported. The total number of Modbus TCP connections includes the number of master-slave connections created on the controller network port and the number of master-slave connections created under the Ethernet communication module.

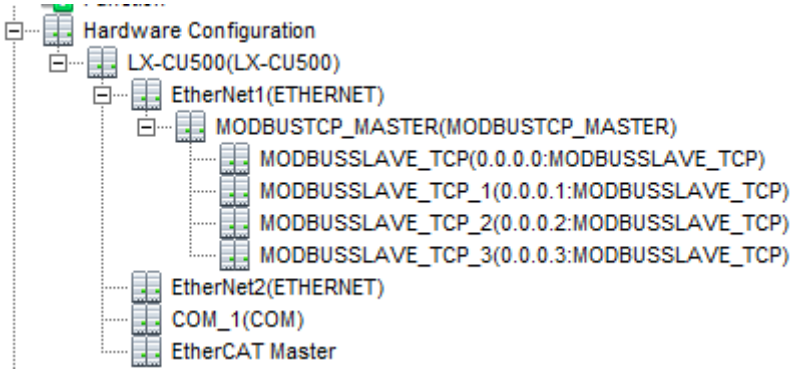
Controller Model	Maximum Total Number of Master-Slave Connections (pcs)
LXCU500	32

In the right-click menu of the MODBUSTCP_MASTER node, select the Add Device command, as shown in the figure.



The Add dialog box is displayed. Select the slave station device. The default IP address of the slave station displayed in the Address box is 0.0.0.0, which can be modified. Enter the Number of Additions. Click OK to add the Modbus TCP slave station under the master station node, as shown in the figure below.





32 slave stations can be added under Modbus TCP master station. The added slave stations are displayed by default: slave station name (slave station address: device name). Users can modify the slave station address.

9.2.2.2 Configure Modbus TCP slave station

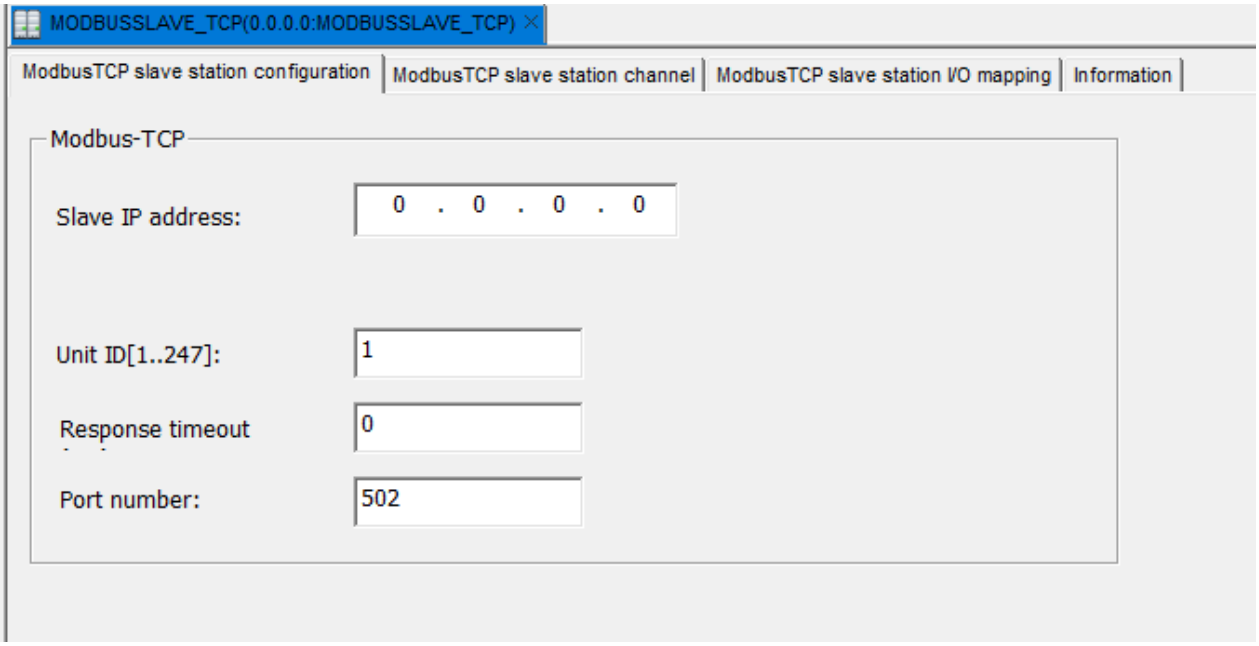
9.2.2.2.1 Introduction

The LX controller Ethernet port supports Modbus TCP master-slave protocol, free communication protocol, and EtherNetIP slave station protocol.

9.2.2.2.2 Steps

(1) Open the configuration window

By double-clicking the MODBUSSLAVE_TCP node or selecting the Open command in the right-click menu, open the MODBUSSLAVE_TCP window, as shown in the figure.



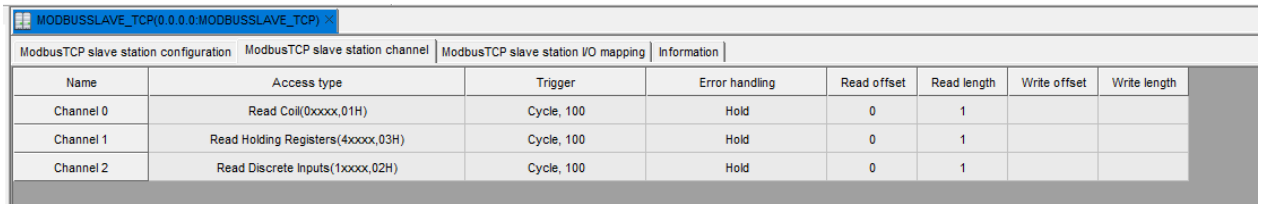
Configuration Parameters for Slave Station Communication

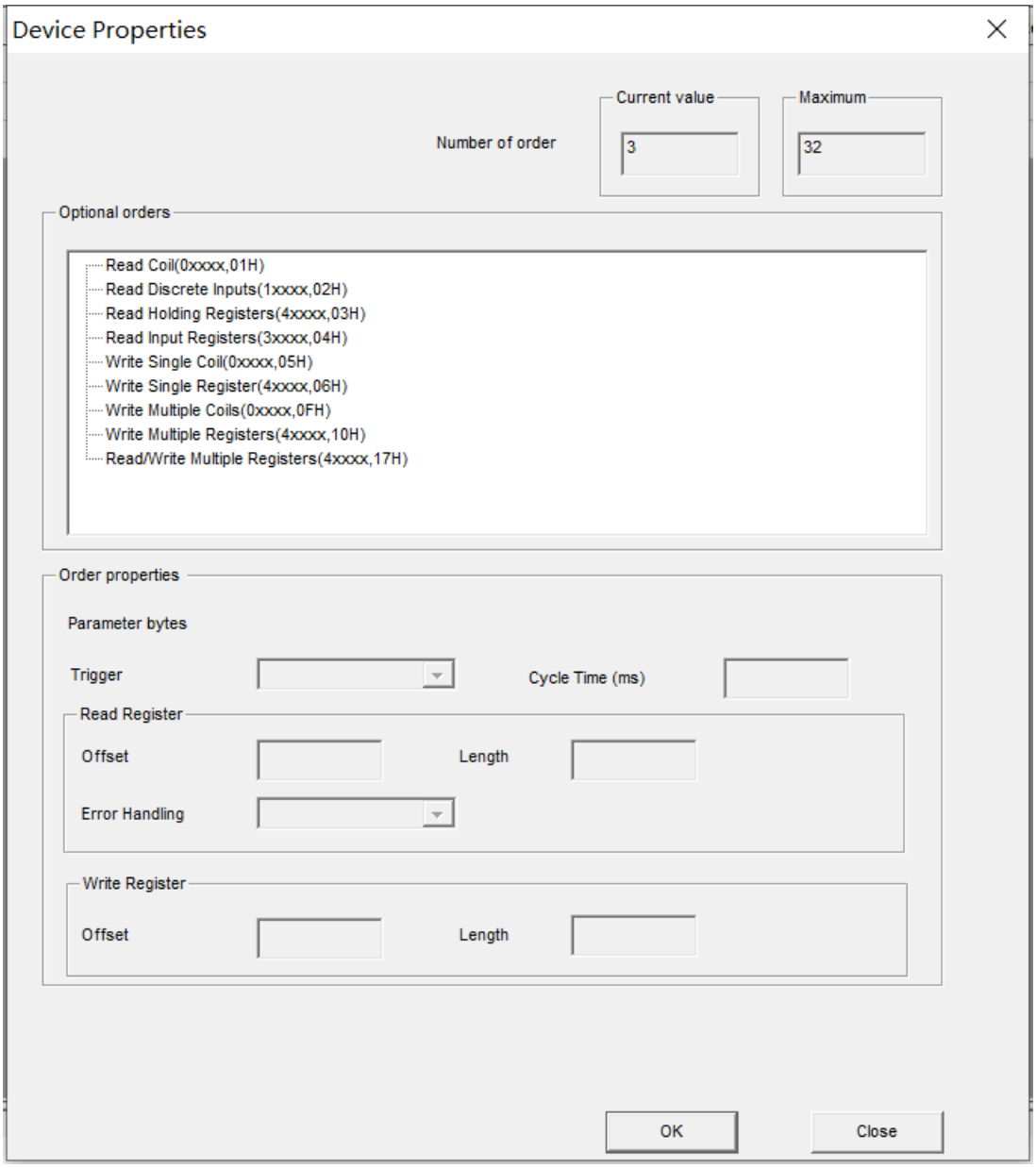
Parameter	Parameter Value	Default Value	Description
Slave station IP address	Set the address based on the actual IP address of the slave station	0.0.0.0	Slave station IP address requested by the master station When the LX-CU500 controller is used as the master station, it can be configured with two IP addresses, selecting one for connection and the other as a backup. If the first connection fails, the system tries to connect with the second one.
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. Users can individually configure the response timeout of a slave station. If the parameter is configured to be greater than zero, the timeout of the slave station shall prevail over the current slave station configuration
Port number	1~65,535	502	Modbus TCP protocol port

(2) Instruction configuration for slave station

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

In an LX project, if the number of added instructions exceeds 1000, the instruction diagnostic data may be biased, resulting in an error being reported during compilation. Therefore, it is recommended that the number of instructions does not exceed 1000.





Instruction Property Settings

Parameter	Parameter Value	Default Value	Description
Trigger	Cycle, rising edge, high level	Cycle	Cycle: The master station polls the slave station periodically to send and receive instruction data Rising edge: If the rising edge trigger is selected, a BOOL trigger variable is automatically generated under the corresponding channel of Modbus TCP Slave I/O Mapping. When this variable has a rising edge, the master station polls the slave station to send and receive

				instruction data. Users need to configure the trigger logic of the rising edge, and set the high and low level time properly according to the number of configured instructions and the set instruction timeout. Otherwise, the rising edge cannot be detected High level: If the high level trigger is selected, a BOOL trigger variable is automatically generated under the corresponding channel of Modbus TCP Slave I/O Mapping. When this variable is high level, the master station polls the slave station to send and receive instruction data
Cycle Time		0~65,535	100	The cycle time for the master station to poll the slave station can be set when the trigger is Cycle. The cycle time shall be set to an integer multiple of the task cycle and shall be ≥10 For cycle instruction, the polling interval does not take effect, and the cycle time is used as the interval between sending and receiving data
Read Register	Offset	0~65,535	0	The corresponding starting address offset on the slave station
	Length	Read coil/read discrete input: 1~2000 Read holding register/read input register: 1 - 125 Read/write multiple registers: 1 - 118	1	The length value corresponds to the number of channels of the slave station
	Error handling	Hold, clear	Keep	Keep: Keep the current data after an abnormal response Clear: Clear the current data after an abnormal response
Write Register	Offset	0~65,535	0	The corresponding starting address offset on the slave station
	Length	Write multiple coils: 1 - 1968 Write multiple registers: 1 - 123 Read/write multiple registers: 1 - 118	1	The length value corresponds to the number of channels of the slave station

- For read operation, the data storage area of the master station: I, M.
- For write operation, the data storage area of the master station: Q, M.

- The addition, deletion, or modification of Modbus instructions and channels only affects this module, and does not cause a reassignment of data area addresses for all modules.

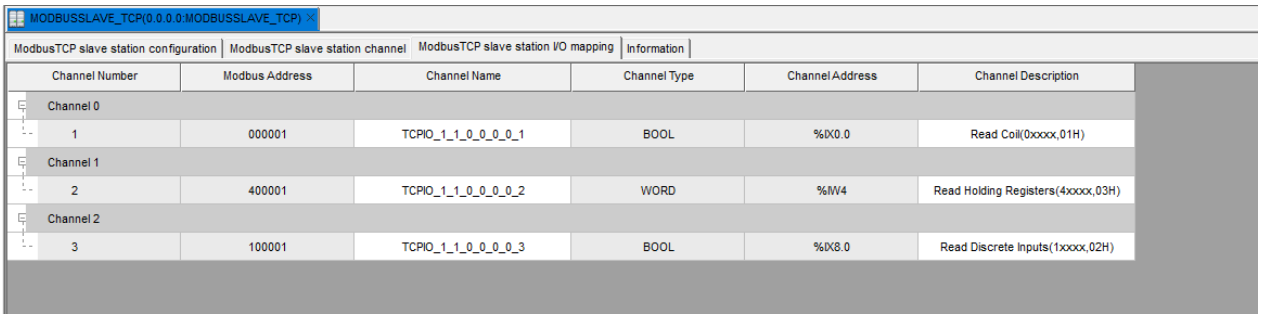
- The starting address of Modbus read/write is very important. Incorrect settings may cause device faults. Please communicate with the third-party device manufacturer to confirm it. (This system starts counting from 0 for the starting address of Modbus, whereas the PLC system generally starts counting from 1.)

(3) Slave station I/O mapping

After the instruction is configured, the corresponding I/O channels are mapped in the Modbus TCP Slave Station I/O Mapping tab.

- Starting address of Modbus = 00001 + offset address;

- Channel name: The initial value is TCPIO_device serial number_protocol serial number_four-bit IP address of slave station_channel number;
- Channel address: By directly accessing this address through the algorithm configuration, access to the data of the Modbus slave station can be realized.



9.2.3 Configure EtherNet/IP slave station

9.2.3.1 Introduction

The controller can communicate with the master station as an EtherNet/IP slave station. Each controller Ethernet port supports up to one EtherNetIP_Adapter. Before communication, users need to add modules under EtherNetIP_Adapter to configure input/output data. A maximum of 15 modules can be added under each EtherNetIP_Adapter.

9.2.3.2 Requirement

The compilation passed.

9.2.3.3 Steps

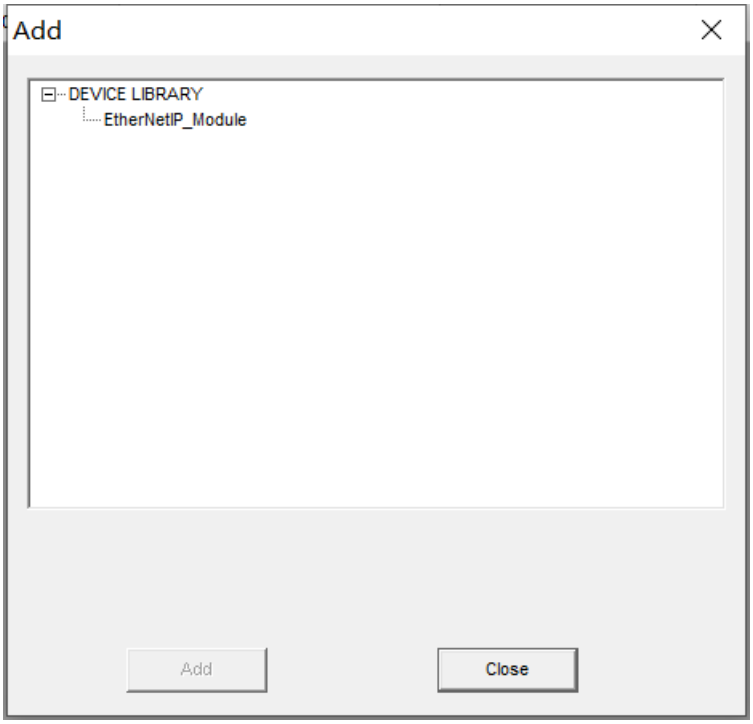
To configure an EtherNetIP slave station, please follow these steps:

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

(2) The Add dialog box is displayed.

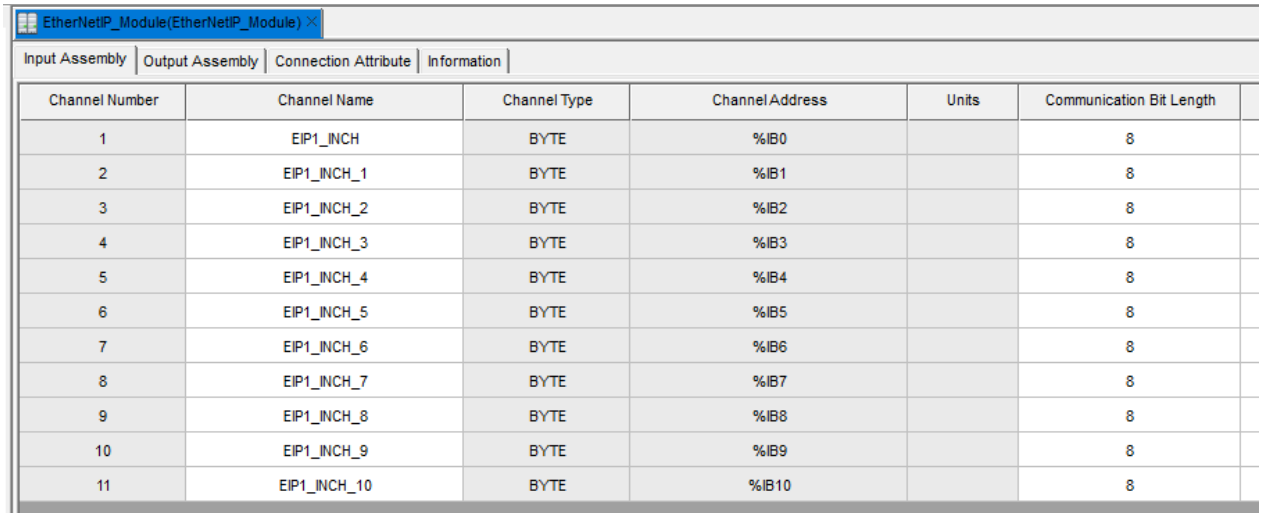
(3) Select the EtherNetIP_Adapter protocol. Click Add.

(4) Right-click the newly added EtherNetIP_Adapter tree node. Select the Add Device command. The Add page is displayed.



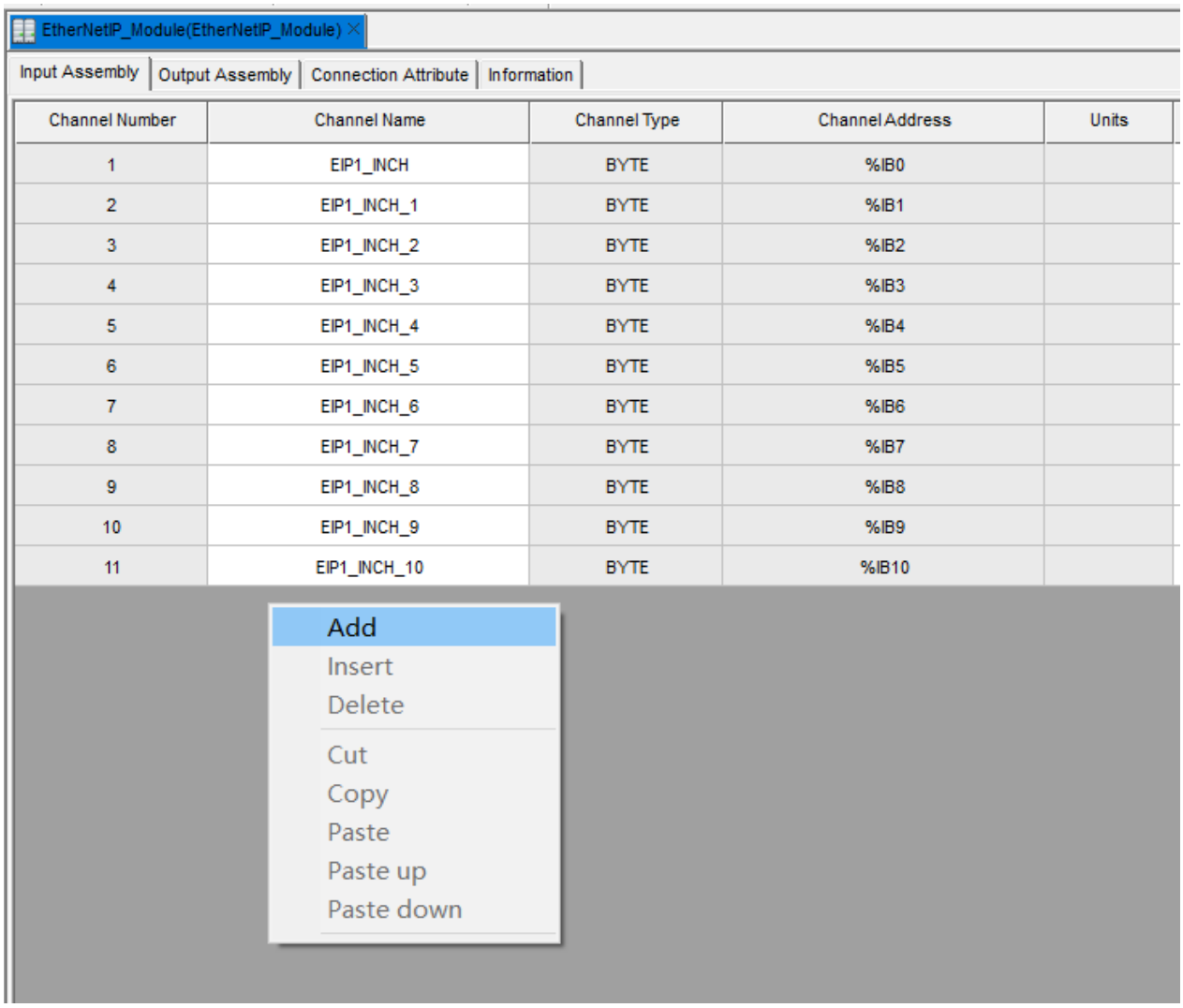
(5) Select EtherNetIP_Module, and click Add. Select EtherNetIP_Module, and click Add continuously for bulk adding.

(6) Double-click the EtherNetIP_Module tree node to open the Module Configuration interface.



Channel Number	Channel Name	Channel Type	Channel Address	Units	Communication Bit Length
1	EIP1_INCH	BYTE	%IB0		8
2	EIP1_INCH_1	BYTE	%IB1		8
3	EIP1_INCH_2	BYTE	%IB2		8
4	EIP1_INCH_3	BYTE	%IB3		8
5	EIP1_INCH_4	BYTE	%IB4		8
6	EIP1_INCH_5	BYTE	%IB5		8
7	EIP1_INCH_6	BYTE	%IB6		8
8	EIP1_INCH_7	BYTE	%IB7		8
9	EIP1_INCH_8	BYTE	%IB8		8
10	EIP1_INCH_9	BYTE	%IB9		8
11	EIP1_INCH_10	BYTE	%IB10		8

(7) Right-click the blank area. Click Add to add the channel data unit.



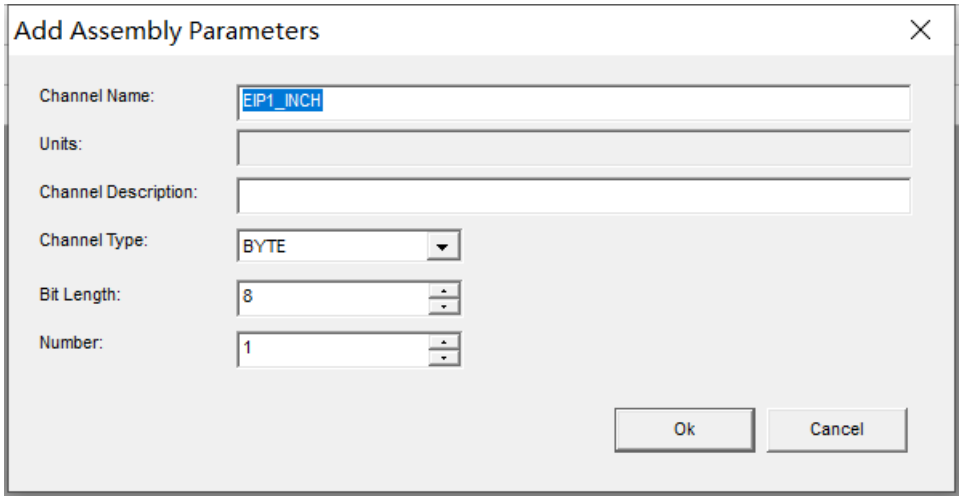
(8) Configure Assembly parameters.

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: It supports customization.

Number: Each module supports adding up to 502 bytes of area I variables and 502 bytes of area Q variables.



(9) Click OK to complete the addition.

9.2.3.4 Export EDS file

9.2.3.4.1 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.

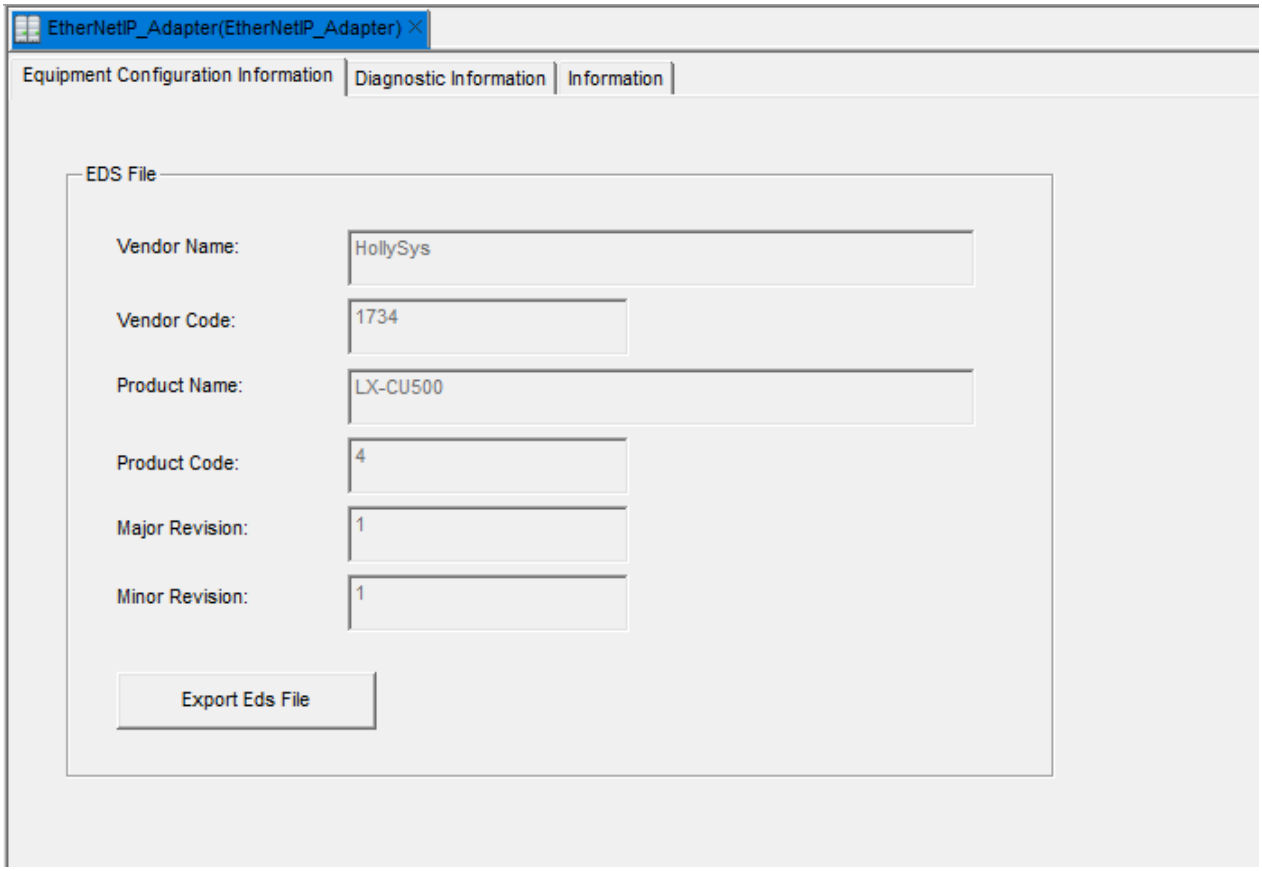
9.2.3.4.2 Requirement

The compilation passed.

9.2.3.4.3 Steps

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

(1) Under the controller's Ethernet port, double-click the EtherNetIP_Adapter tree node to open the Equipment Configuration Information page.



(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.

9.2.4 Configure EtherNet/IP Master Station

9.2.4.1 Import EDS file

9.2.4.1.1 Introduction

When the controller communicates with the master station as an EtherNet/IP master 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. Please prepare the EDS file of the slave station before operation.

9.2.4.1.2 Requirement

The compilation passed.

9.2.4.1.3 Steps

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

(1) Select [Project] - [Import Device Description File].

(2) Select the EDS file from the slave station (the example file is shown below), and the interface will prompt that the EDS file has been successfully imported.

9.2.4.2 Configure EtherNet/IP master station

9.2.4.2.1 Introduction

The controller can communicate with the salve station as an EtherNet/IP master station. Each controller Ethernet port supports up to one EtherNetIP_Scanner. Before communication, users need to add modules under EtherNetIP_Scanner to configure input/output data. A maximum of 64 modules can be added under each EtherNetIP_Scanner.

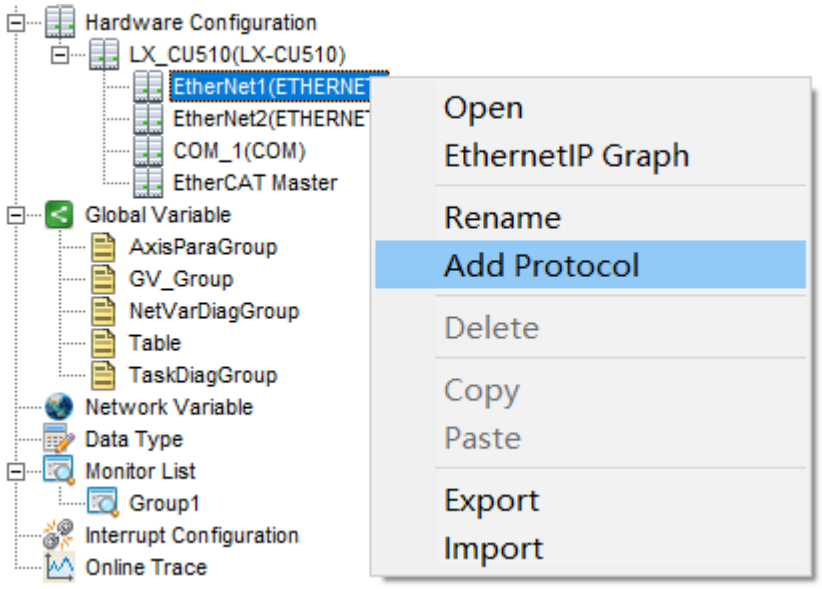
9.2.4.2.2 Requirement

The compilation passed.

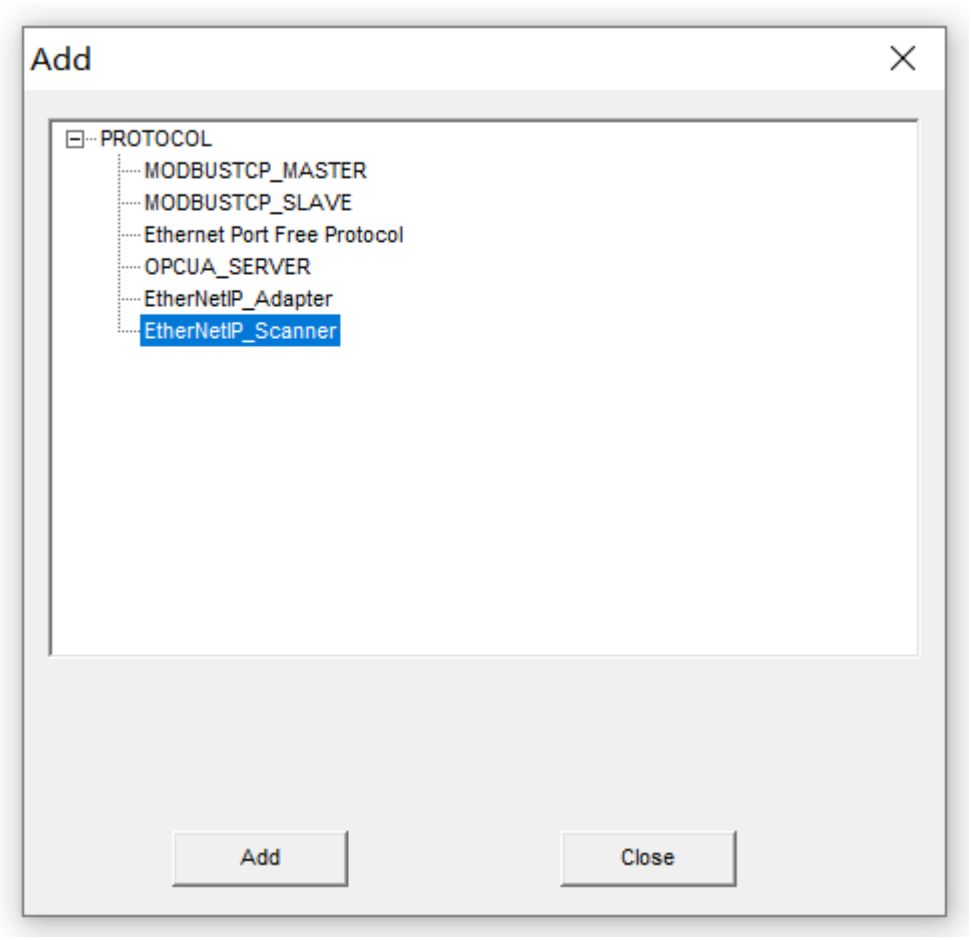
9.2.4.2.3 Steps

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

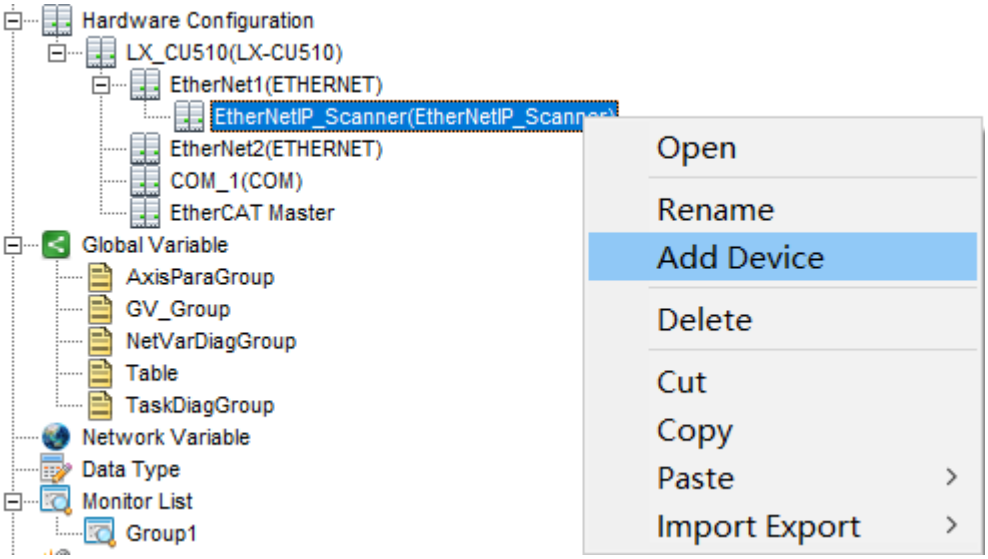
(1) In the right-click menu of the EtherNet1 tree node, select the Add Protocol command.



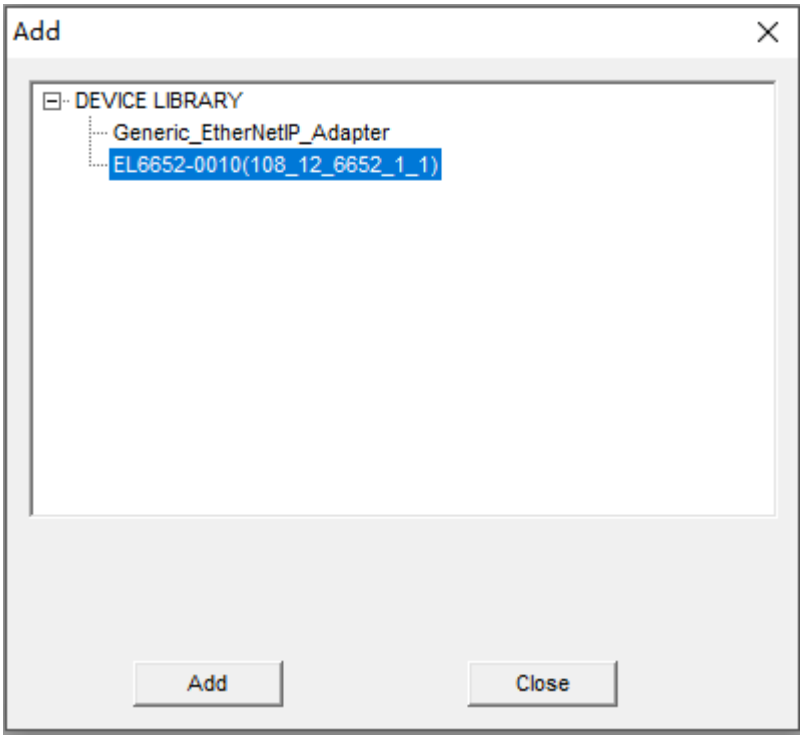
(2) Select the EtherNetIP_Adapter protocol. Click Add.



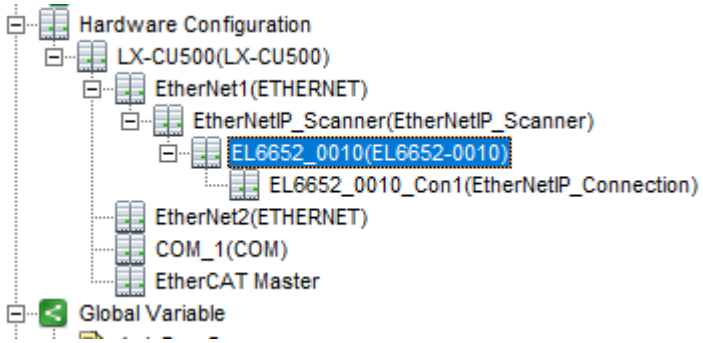
(3) In the right-click menu of the EtherNetIP Scanner tree node, select the Add Device.



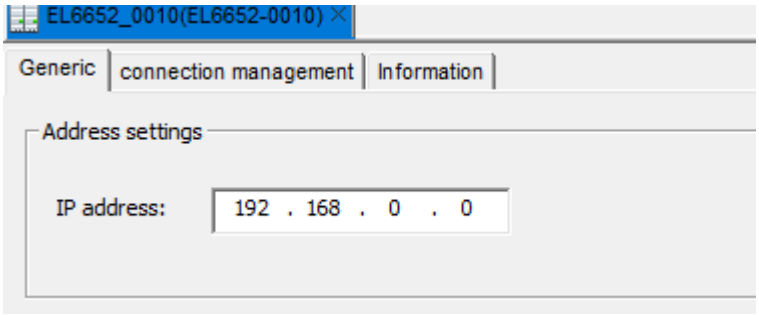
(4) In the pop-up dialog box, select the slave device you want to add and click "Add".



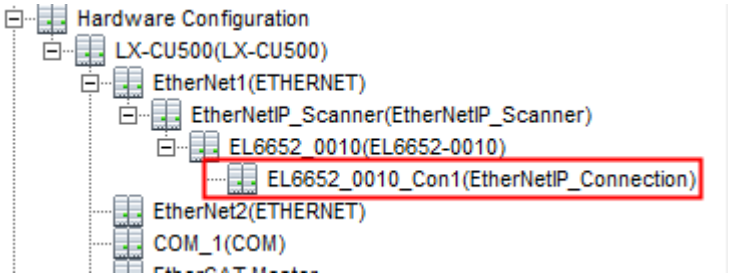
(5) Double click the newly added slave device.



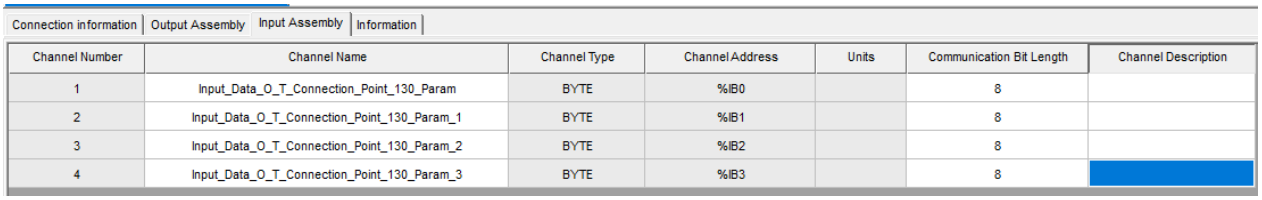
(6) Configure the IP address of the slave station in the "Generic" tab.



(7) Double click on EthernetIP_Connect under the slave device.



(8) 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	

9.2.5 Configure Free Communication Protocol

9.2.5.1 Introduction

When LX PLC communicates with other devices via serial ports, if the protocol supported by these third-party devices is not the standard MODBUS RTU protocol, it can be implemented through free protocol

configuration.

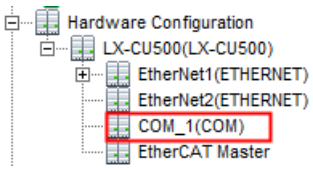
9.2.5.2 Requirement

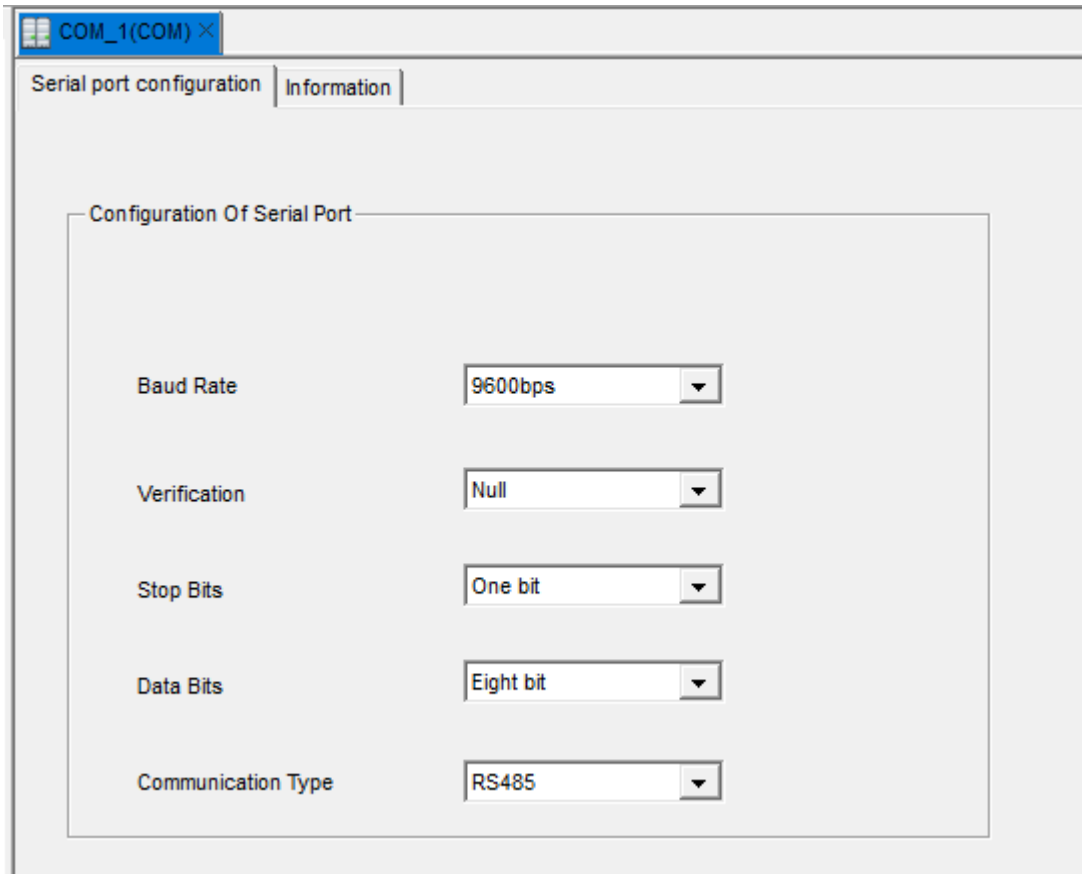
The compilation passed.

9.2.5.3 Steps

1. COM parameter configuration

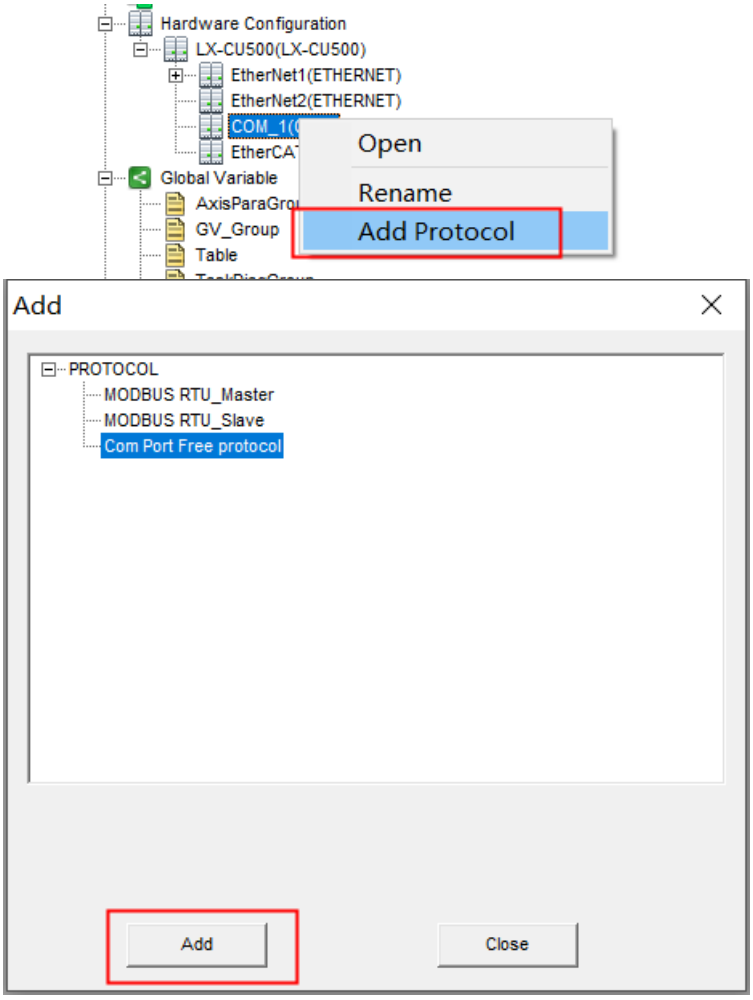
   Double-click on the COM_1 node, and the serial port configuration window is displayed, then modify the serial port parameters.



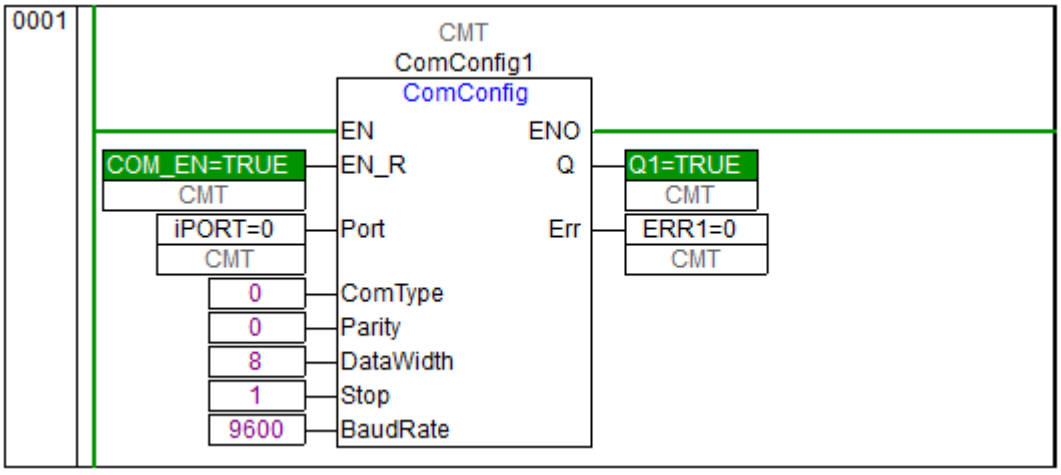


2. Protocol configuration

   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.



3. 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:

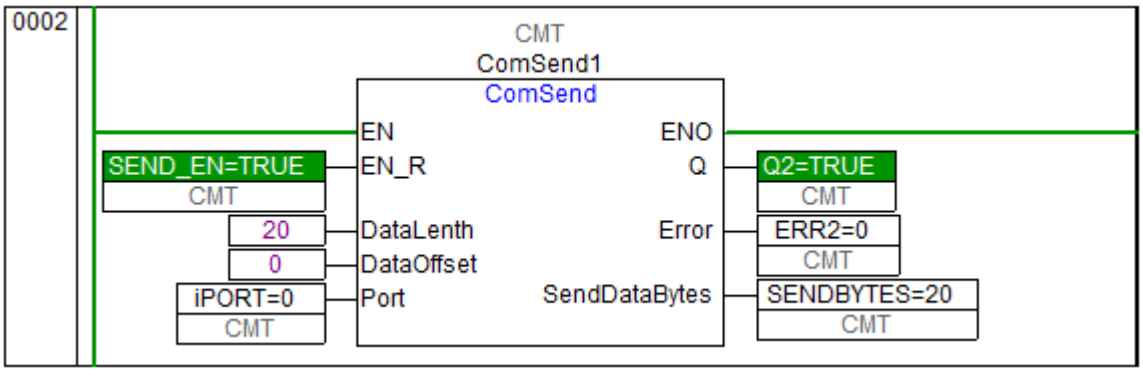


The parameters are described as follows:

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: RS-485 1: RS-422 2: RS-232
	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

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

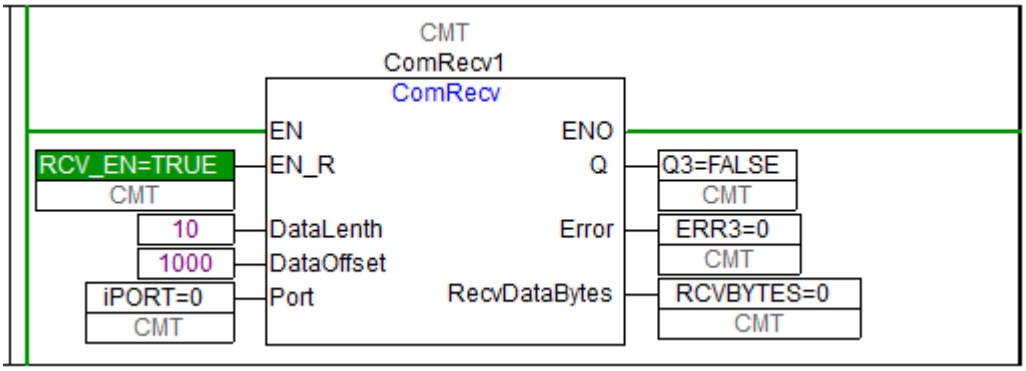


The parameters are described as follows:

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

5. 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

9.3 Configure COM Port

9.3.1 Configuring Modbus RTU Master Station

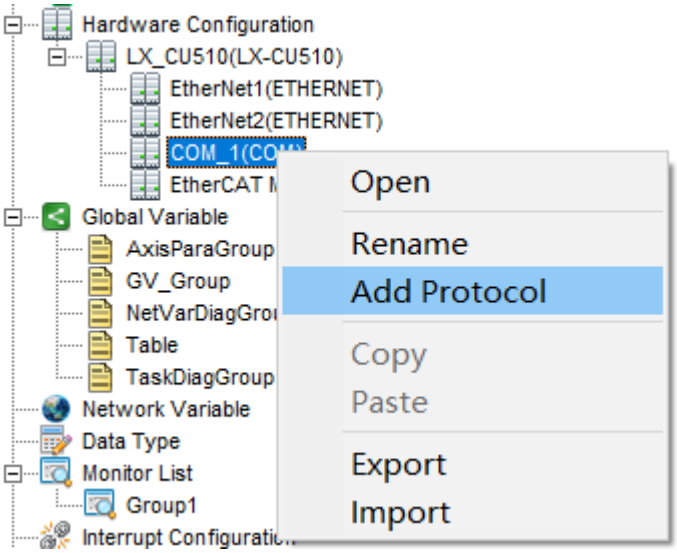
9.3.1.1 Requirement

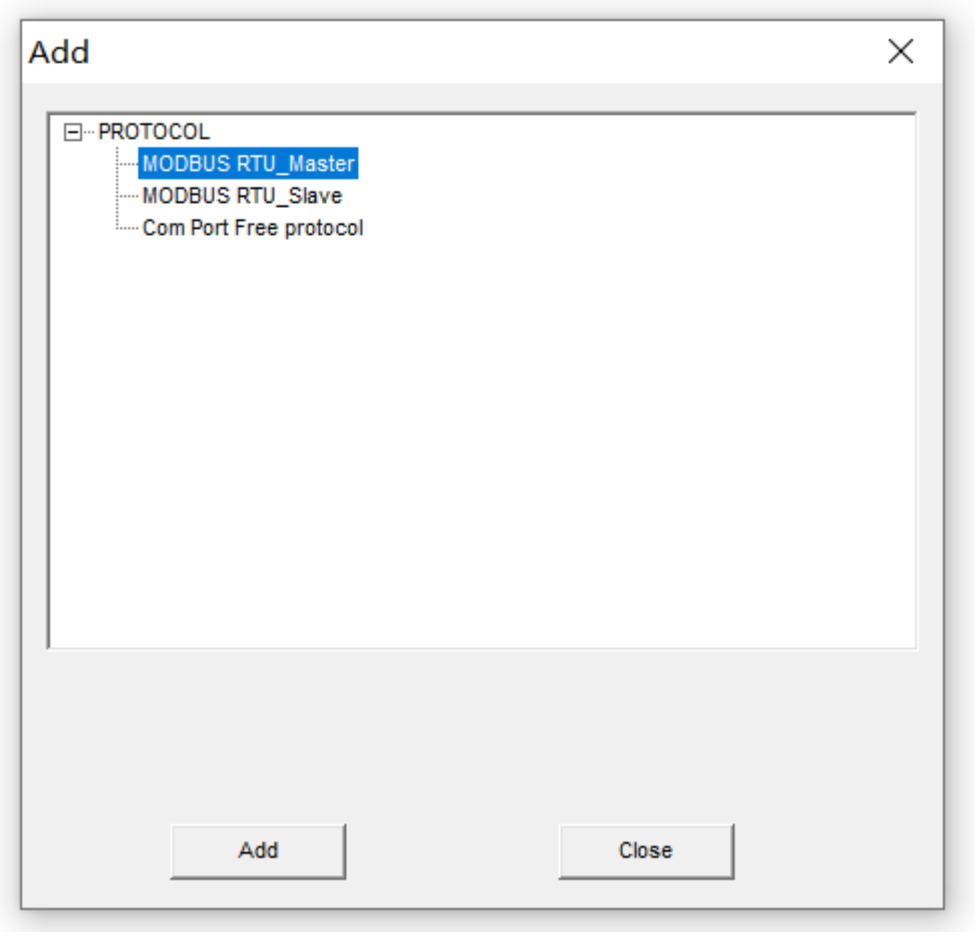
The compilation passed.

9.3.1.2 Steps

(1) Add master station protocol

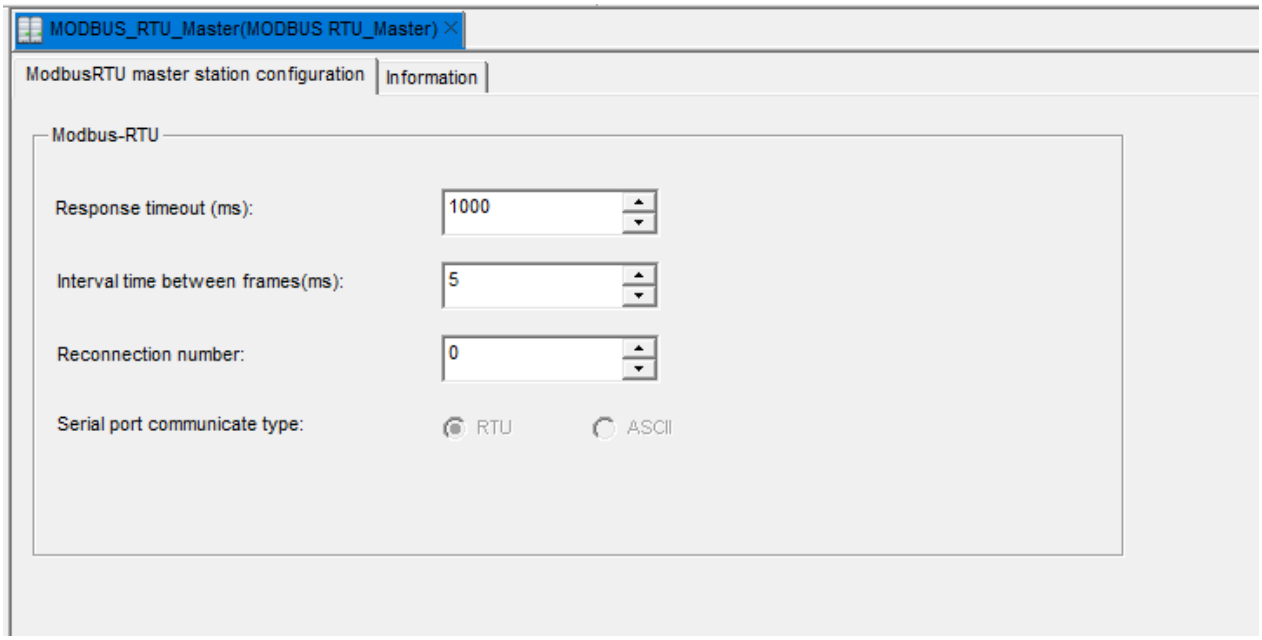
Take COM1 of LXCU500 as an example. Right-click the COM_1 tree node, as shown in the figure.





(2) Configure Modbus RTU master station

Double-click the MODBUS_RTU_Master node. Or select the Open command in the right-click menu. The master station configuration window is displayed, as shown in the figure.



Configuration Parameters for Master Station Communication

Parameter	Parameter Value	Default Value	Description
Response Timeout (ms)	10~65,535	1000	Allowed slave station delay response time after the master station sends the request frame
Frame to frame interval time (ms)	2~65,535	5	The interval between the master station receiving the slave station response frame and sending the next query frame. If the slave station responds to the last frame with a timeout, the master station can ignore the interval and send a query frame directly
Number of retries	0~ 10	0	Number of times the master station resends the request after an abnormal response

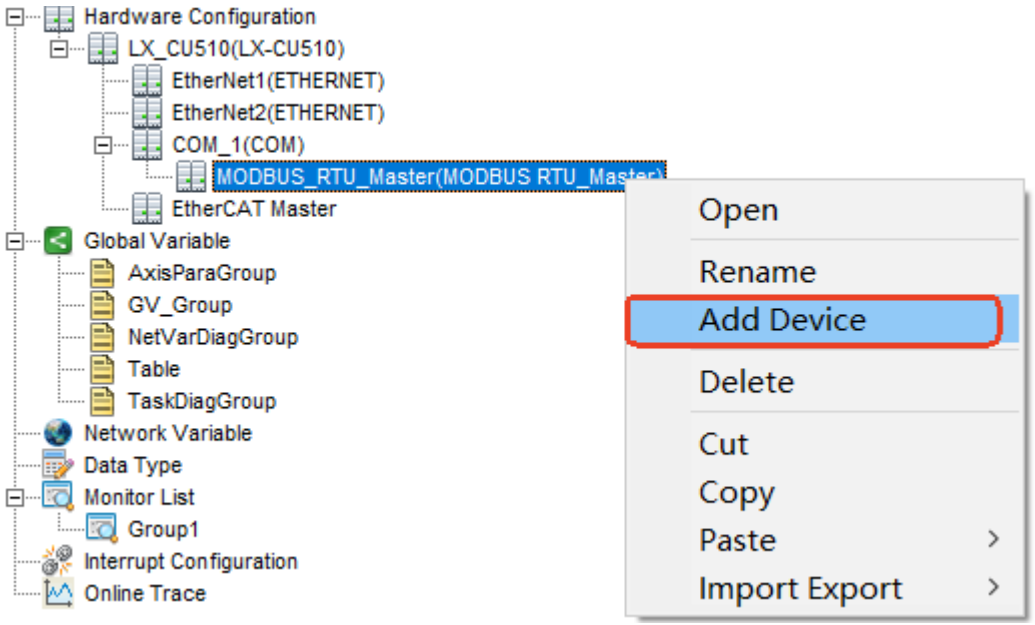
9.3.2 Configuring Modbus RTU Slave Station

9.3.2.1 Requirement

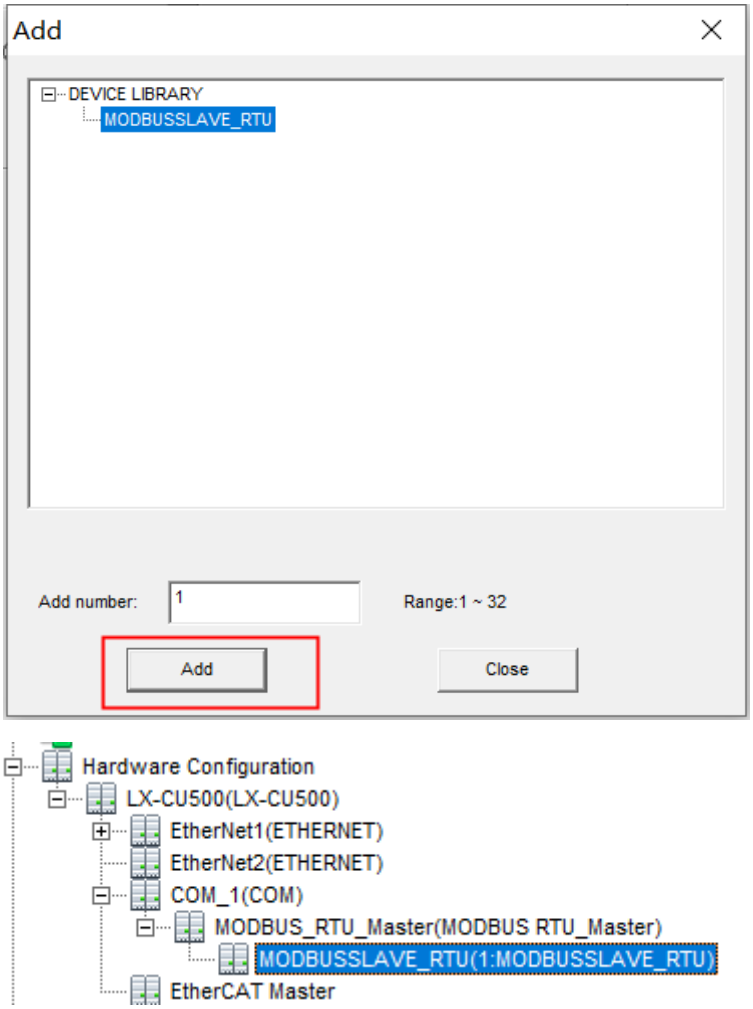
The compilation passed.

9.3.2.2 Steps

(1) In the right-click menu of the MODBUS_RTU_Master node, select the Add Device command, as shown in the figure.



(2) The Add dialog box is displayed. Select the Slave Station Device. Enter the Number of Additions. Click OK to add the Modbus RTU slave station under the master station node, as shown in the figure below.

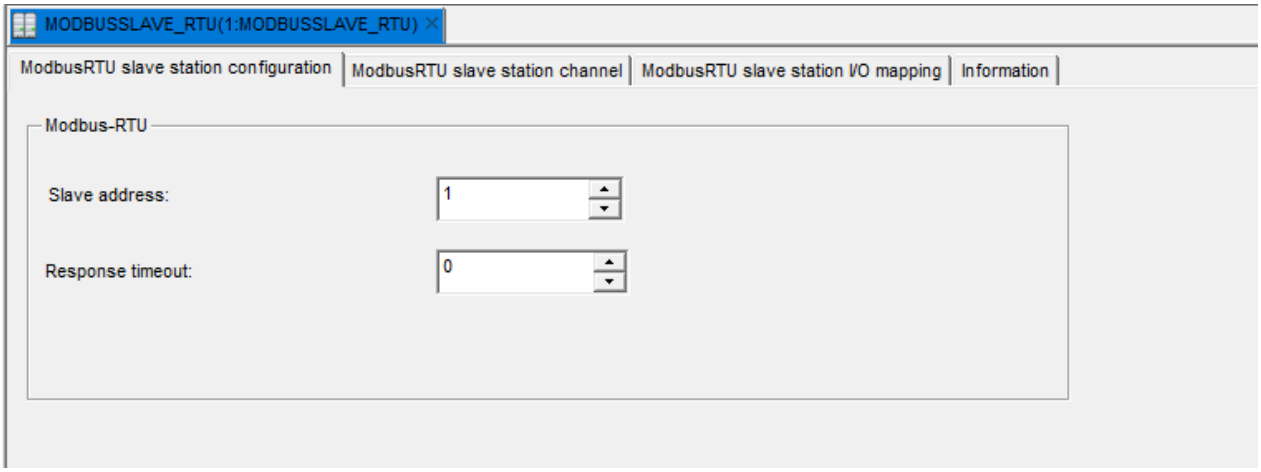


(3) Add Modbus RTU slave station

Up to 32 slave stations can be added to each port. The added slave stations are displayed by default: slave station name (slave station address: device name). Users can modify the slave station address.

(4) Configure Modbus RTU slave station

By double-clicking the MODBUS_RTU_SLAVE node or selecting the Open command in the right-click menu, open the MODBUSSLAVE_RTU window, as shown in the figure.



Configuration Parameters for Slave Station Communication

Parameter	Parameter Value	Default Value	Description
Slave Station Address	1~247	1	Slave station address requested by the master station
Response Timeout (ms)	0~65,535	0	Default: The default response timeout period of the slave station is zero. In this case, the Response Timeout period configured by the master station shall prevail; Users can separately configure the response timeout period for a slave station. If this parameter is configured to be greater than zero, the timeout period for this slave station shall prevail over the current slave station configuration.

The Modbus RTU standard protocol requires a maximum data frame size of 256 bytes.

Data Frame Format

Slave Address	Function Code	Data	CRC
1 byte	1 byte	0 up to 252 byte(s)	2 bytes (CRC Low → CRC High)

- In the master communication mode, the master station can configure up to 32 slave station devices, and each slave station can configure up to 32 instructions.

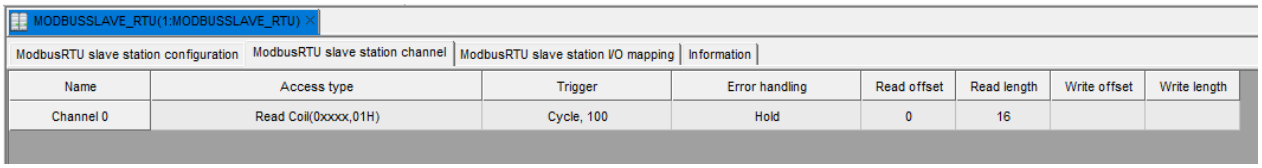
(5) Instruction configuration for slave station

The Modbus RTU slave station instructions are configured in the same way as those for Modbus TCP. See Slave Station Instruction Configuration for Modbus TCP master-slave station.

(6) Slave station I/O mapping

After the instruction is configured, the corresponding I/O channels are mapped in the Modbus RTU Slave Station I/O Mapping tab.

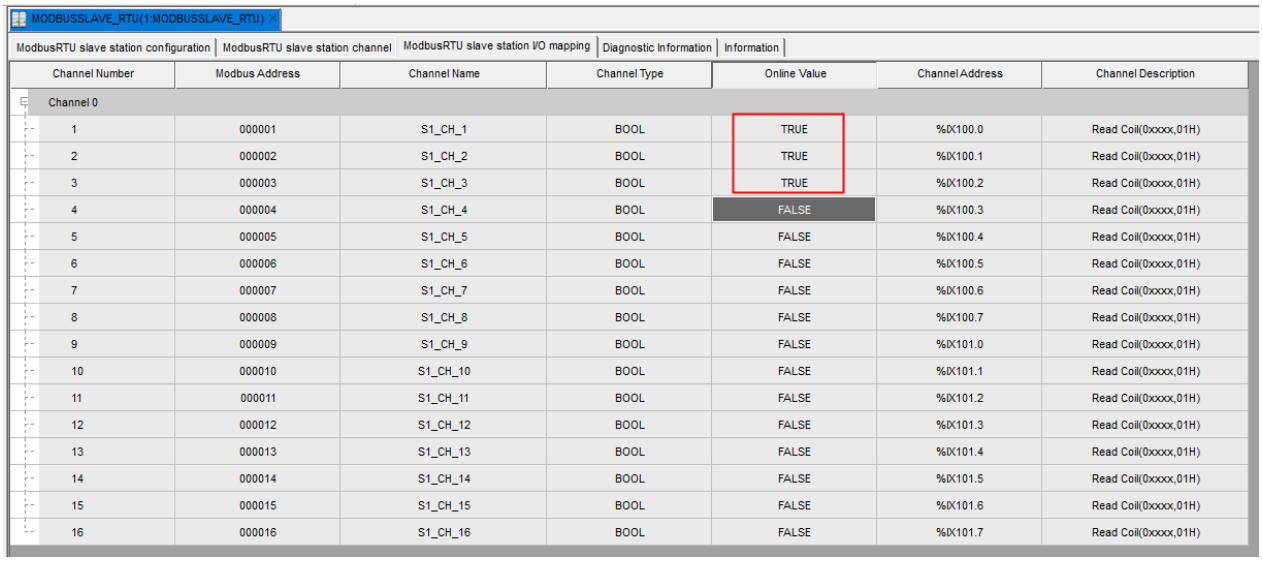
- Starting address of Modbus = 00001 + offset address;
- Channel name: The initial value is Sn_CH_channel number, where n is the serial number of the added slave station;
- Channel address: By directly accessing this address through the algorithm configuration, access to the data of the Modbus slave station can be realized.



(7) Application example

■ Function code 01H (read coil status)

For the Modbus test of the function code 01H (for reading coil status),.choose a suitable serial port debugging tool based on usage habits. In this case, the value of the channel parameter information QX0.0, QX0.1, QX0.2, and QX0.3 in the monitored hardware configuration is TRUE, and the rest are FALSE.



The following example shows how to read the status of DOs QX0.0 - QX0.7 in slave station 1. The 8 bits read out constitute 1 byte. The slave station address is 01; The function code is 01; The high and low bits of the starting address for %QX0.0 (Modbus address 0) are 00 and 00 in hexadecimal; The number of data coils is 8, so enter 00 for the high bit of the data coil number and 08 for the low bit of the data coil number; The checksum CRC is calculated by the serial port commissioning tool.

RTU Query Frame

Slave Station Address	Function Code	High Bit of Starting Address	Low Bit of Starting Address	High Bit of Data Coil Number	Low Bit of Data Coil Number	Checksum CRC
01H	01H	00H	00H	00H	08H	3DH CCH

RTU Response Frame

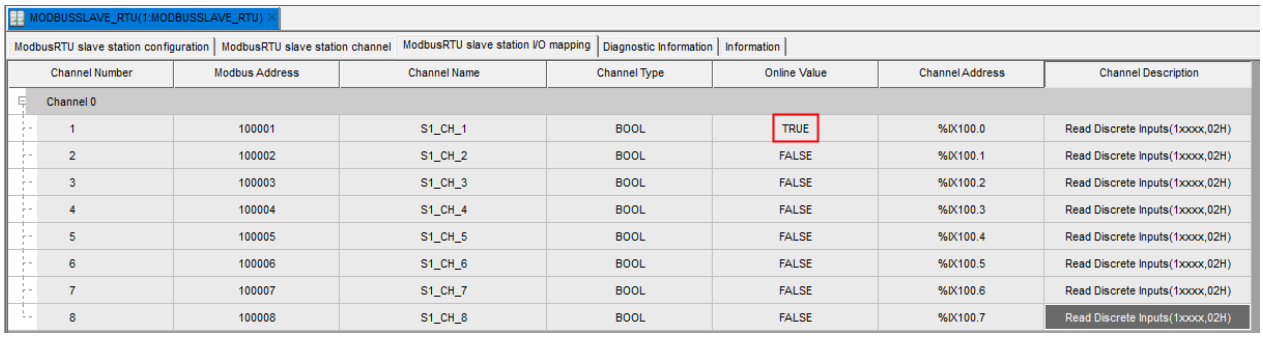
Slave station address	Function Code	Byte Count	Data	Checksum CRC
01H	01H	01H	0FH	11H 8CH

Data bit 0FH, converted to binary is 00001111, indicates that %QX0.0-%QX0.3 states are 1, and the rest of %QX0.4~%QX0.7 states are all 0.

■ Function code 02H (read input status)

For the Modbus test of the function code 02H (for reading input status) test, choose a suitable

serial port debugging tool based on usage habits. In this case, the value of the channel parameter information IX0.0 in the monitored hardware configuration is TRUE, and the rest are FALSE.



Channel Number	Modbus Address	Channel Name	Channel Type	Online Value	Channel Address	Channel Description
Channel 0						
1	100001	S1_CH_1	BOOL	TRUE	%IX100.0	Read Discrete Inputs(1xxxx,02H)
2	100002	S1_CH_2	BOOL	FALSE	%IX100.1	Read Discrete Inputs(1xxxx,02H)
3	100003	S1_CH_3	BOOL	FALSE	%IX100.2	Read Discrete Inputs(1xxxx,02H)
4	100004	S1_CH_4	BOOL	FALSE	%IX100.3	Read Discrete Inputs(1xxxx,02H)
5	100005	S1_CH_5	BOOL	FALSE	%IX100.4	Read Discrete Inputs(1xxxx,02H)
6	100006	S1_CH_6	BOOL	FALSE	%IX100.5	Read Discrete Inputs(1xxxx,02H)
7	100007	S1_CH_7	BOOL	FALSE	%IX100.6	Read Discrete Inputs(1xxxx,02H)
8	100008	S1_CH_8	BOOL	FALSE	%IX100.7	Read Discrete Inputs(1xxxx,02H)

The following example shows how to read the status of DIs IX0.0 - IX0.7 in slave station 1. The 8 bits read out constitute 1 byte. The slave station address is 01; The function code is 02; The high and low bits of the starting address for %IX0.0 (Modbus address 0) are 00 and 00 in hexadecimal; The number of data coils is 8, so enter 00 for the high bit of the data coil number and 08 for the low bit of the data coil number; The checksum CRC is calculated by the serial port commissioning tool.

RTU Query Frame

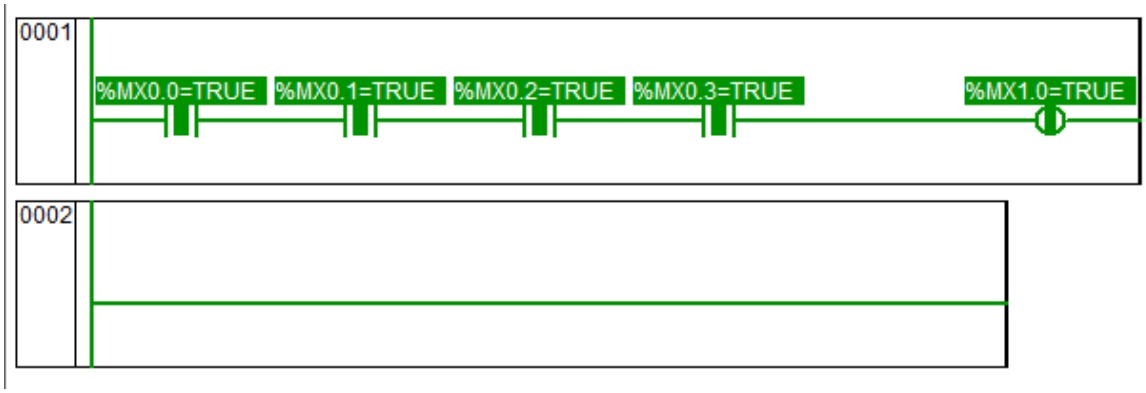
Slave Station Address	Function Code	High Bit of Starting Address	Low Bit of Starting Address	High Bit of Data Coil Number	Low Bit of Data Coil Number	Checksum CRC
01H	02H	00H	00H	00H	08H	79H CCH

RTU Response Frame

Slave Station Address	Function Code	Byte Count	Data	Checksum CRC
01H	02H	01H	01H	60H 48H

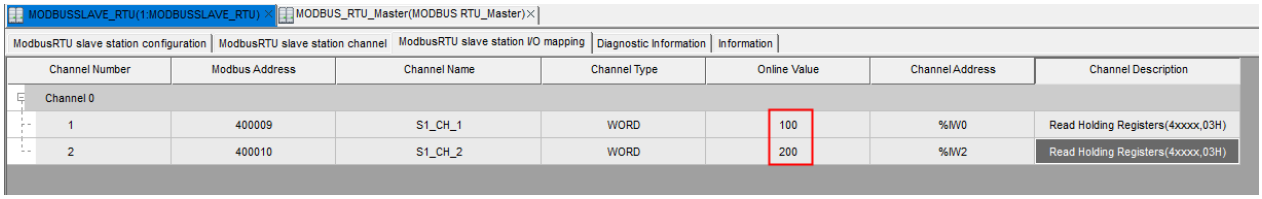
Data bit 01H, converted to binary is 00000001, indicates that %IX0.0 state is 1 and %IX0.1-%IX0.7 states are all 0.

The following example shows how to read the status of DIs MX0.0 - MX0.4 in slave station 1, all of which are TRUE. Both 01H and 02H function codes can be used.



- Function code 03H (read holding registers)

For the Modbus test of the function code 03H (for reading holding registers), choose a suitable serial port debugging tool based on usage habits. In this case, the value of channel parameter information QW8 in the monitored hardware configuration is 100, and the value of QW10 is 200.



The following example shows how to read the status of QW8 - QW10 in slave station 1. The slave station address is 01; The function code is 03; The high and low bits of the starting address for %QW8 (Modbus address 0) are 00 and 04 in hexadecimal; The number of registers is 2, so enter 00 for the high bit of the register number and 02 for the low bit of the register number; The checksum CRC is calculated by the serial port commissioning tool.

RTU Query Frame

Slave Station Address	Function Code	High Bit of Starting Address	Low Bit of Starting Address	High Bit of Register Number	Low Bit of Register Number	Checksum CRC
01H	03H	00H	04H	00H	02H	85H CAH

RTU Response Frame

Slave Station Address	Function Code	Byte Count	Data	Checksum CRC
01H	03H	04H	00H 64H 00H C8H	BAH 7AH

The data bits 00H, 64H, 00H, and C8H, converted to decimal would be a value of 100 for %QW8 and 200 for %QW10.

Function code 04H (read input registers)

For the Modbus test of the function code 04H (for reading input registers), choose a suitable serial port debugging tool based on usage habits. In this case, the value of channel parameter information IW8 in the monitored hardware configuration is 0, and the value of IW10 is 0.

The following example shows how to read the status of IW8 - IW10 in slave station 1. The slave station address is 01; The function code is 04; The high and low bits of the starting address for %QW8 (Modbus address 0) are 00 and 04 in hexadecimal; The number of registers is 2, so enter 00 for the high bit of the register number and 02 for the low bit of the register number; The checksum CRC is calculated by the serial port commissioning tool.

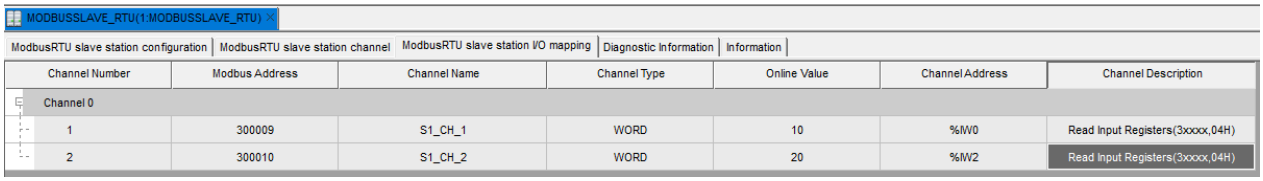
RTU Query Frame

Slave Station Address	Function Code	High Bit of Starting Address	Low Bit of Starting Address	High Bit of Register Number	Low Bit of Register Number	Checksum CRC
01H	04H	00H	04H	00H	02H	30H 0AH

RTU Response Frame

Slave Station Address	Function Code	Byte Count	Data	Checksum CRC
01H	04H	04H	00H 00H 00H 00H	FBH 84H

The data bits 00H, 00H, 00H, and 00H, converted to decimal would be a value of 0 for %IW8 and 0 for %IW10.



The following example shows how to read the values of CPU address %MW300 and %MW302 with slave station address 1. The slave station address is 01; The function code is 03 or 04; The high and low bits of the starting address for 300 (Modbus address 3000 + 300 / 2 = 3150) are 0C and 4E in hexadecimal; The number of registers is 2, so enter 00 for the high bit of the register number and 02 for the low bit of the register number; The checksum CRC is calculated by the serial port commissioning tool.

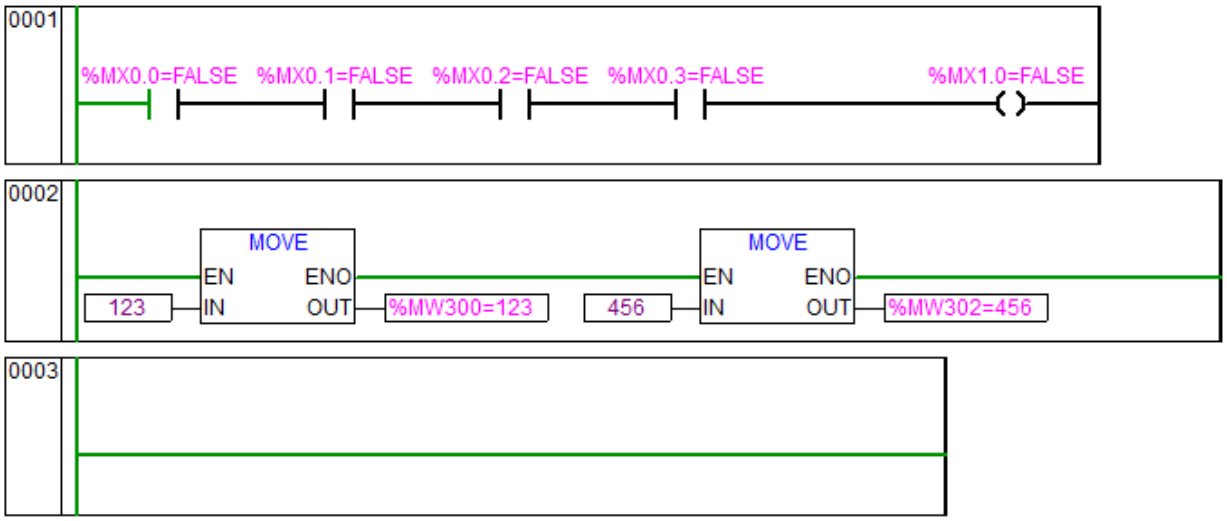
RTU Query Frame

Slave Station Address	Function Code	High Bit of Starting Address	Low Bit of Starting Address	High Bit of Register Number	Low Bit of Register Number	Checksum CRC
01H	03H	0CH	4EH	00H	02H	A7H 4CH

RTU Response Frame

Slave Station Address	Function Code	Byte Count	Data	Checksum CRC
01H	03H	04H	00H 7BH 01H C8H	8AH 2CH

The data bits 00H, 7BH, 01H, and C8H, converted to decimal would be a value of 123 for %MW300 and 456 for %MW302.



■ Function code 05H (force single coil)

For the Modbus test of the function code 05H (for forcing single coil), choose a suitable serial port debugging tool based on usage habits. The following example shows how to write the CPU address %QX0.0 with slave station address 1. The state of this switch value is 1. And then write 0. The slave station address is 01; The function code is 05; The high and low bits of the starting address for %QX0.0 (Modbus address 0) are 00 and 00 in hexadecimal; The data is FF, and the original switch status is 00; The checksum CRC is calculated by the serial port commissioning tool. In this case, the online value of %QX0.0 address in the monitored AT hardware configuration

changes from FALSE to TRUE.

RTU Query Frame

Slave Station Address	Function Code	High Bit of Starting Address	Low Bit of Starting Address	Data	Original Switch Status	Checksum CRC
01H	05H	00H	00H	FFH	00H	8CH 3AH

RTU Response Frame

Slave Station Address	Function Code	High Bit of Starting Address	Low Bit of Starting Address	Data	Original Switch Status	Checksum CRC
01H	05H	00H	00H	FFH	00H	8CH 3AH

And then write 0. The slave station address is 01; The function code is 05; The high and low bits of the starting address for %QX0.0 (Modbus address 0) are 00 and 00 in hexadecimal; The data is 00, and the original switch status is 00; The checksum CRC is calculated by the serial port commissioning tool. In this case, the online value of %QX0.0 address in the monitored AT hardware configuration changes from TRUE to FALSE.

RTU Query Frame

Slave Station Address	Function Code	High Bit of Starting Address	Low Bit of Starting Address	Data	Original Switch Status	Checksum CRC
01H	05H	00H	00H	00H	00H	CDH CAH

RTU Response Frame

Slave Station Address	Function Code	High Bit of Starting Address	Low Bit of Starting Address	Data	Original Switch Status	Checksum CRC
01H	05H	00H	00H	00H	00H	CDH CAH

The following example shows how to write the CPU address %MX150.1 with slave station address 1. The state of this switch value is 1. The slave station address is 01; The function code is 05; The high and low bits of the starting address for %MX150.1 (Modbus address 150 * 8 + 1 + 3000 = 4201) are 10 and 69 in hexadecimal; The data is FF, and the original switch status is 00; The checksum CRC is calculated by the serial port commissioning tool. In this case, the online value of %MX150.1 address in the monitored AT changes from FALSE to TRUE.

RTU Query Frame

Slave Station Address	Function Code	High Bit of Starting Address	Low Bit of Starting Address	Data	Original Switch Status	Checksum CRC
01H	05H	10H	69H	FFH	00H	58H E6H

RTU Response Frame

Slave Station Address	Function Code	High Bit of Starting Address	Low Bit of Starting Address	Data	Original Switch Status	Checksum CRC
01H	05H	10H	69H	FFH	00H	58H E6H

The following example shows how to write the CPU address %MX150.1 with slave station address 1. The state of this analog value is 0. The slave station address is 01; The function code is 05; The high and low bits of the starting address for %MX150.1 (Modbus address 150 * 8 + 1 + 3000 = 4201) are 10 and 69 in hexadecimal; The data is 00, and the original switch status is 00; The checksum CRC is calculated by the serial port commissioning tool. In this case, the online value of %MX150.1 address in the monitored AT changes from TRUE to FALSE.

RTU Query Frame

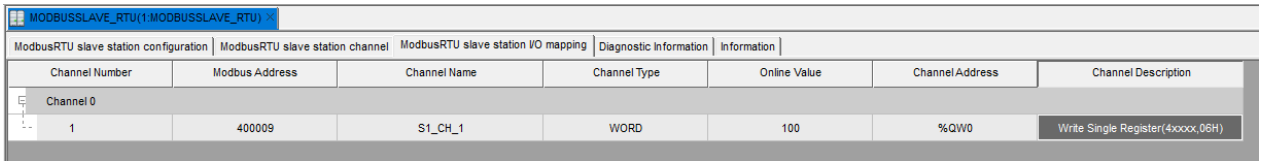
Slave Station Address	Function Code	High Bit of Starting Address	Low Bit of Starting Address	Data	Original Switch Status	Checksum CRC
01H	05H	10H	69H	00H	00H	19H 16H

RTU Response Frame

Slave Station Address	Function Code	High Bit of Starting Address	Low Bit of Starting Address	Data	Original Switch Status	Checksum CRC
01H	05H	10H	69H	00H	00H	19H 16H

- Function code 06H (preset single register)

For the Modbus test of the function code 06H (for preseting single register), choose a suitable serial port debugging tool based on usage habits. The following example shows how to write the CPU address %QW8 with slave station address 1. The state of this analog value is 100. The slave station address is 01; The function code is 06; The high and low bits of the starting address for %QW8 (Modbus address 8 / 2 = 4) are 00 and 04 in hexadecimal; For data 100, the hexadecimal is 00 and 64; The checksum CRC is calculated by the serial port commissioning tool. In this case, the online value of %QW8 address in the monitored AT hardware configuration changes to 100.



RTU Query Frame

Slave Station Address	Function Code	High Bit of Starting Address	Low Bit of Starting Address	High Bit of Data	Low Bit of Data	Checksum CRC
01H	06H	00H	04H	00H	64H	C9H E0H

RTU Response Frame

Slave Station Address	Function Code	High Bit of Starting Address	Low Bit of Starting Address	High Bit of Data	Low Bit of Data	Checksum CRC
01H	06H	00H	04H	00H	64H	C9H E0H

The following example shows how to write the CPU address %MW200 with slave station address 1. The state of this analog value is 3000. The slave station address is 01; The function code is 06; The high and low bits of the starting address for %MW200 (Modbus address 200/2 + 3000 = 3100) are 0C and 1C in hexadecimal; For data 3000, the hexadecimal is 0B and B8; The checksum CRC is calculated by the serial port commissioning tool. In this case, the online value of %MW200 address in the monitored AT changes to 3000.

RTU Query Frame

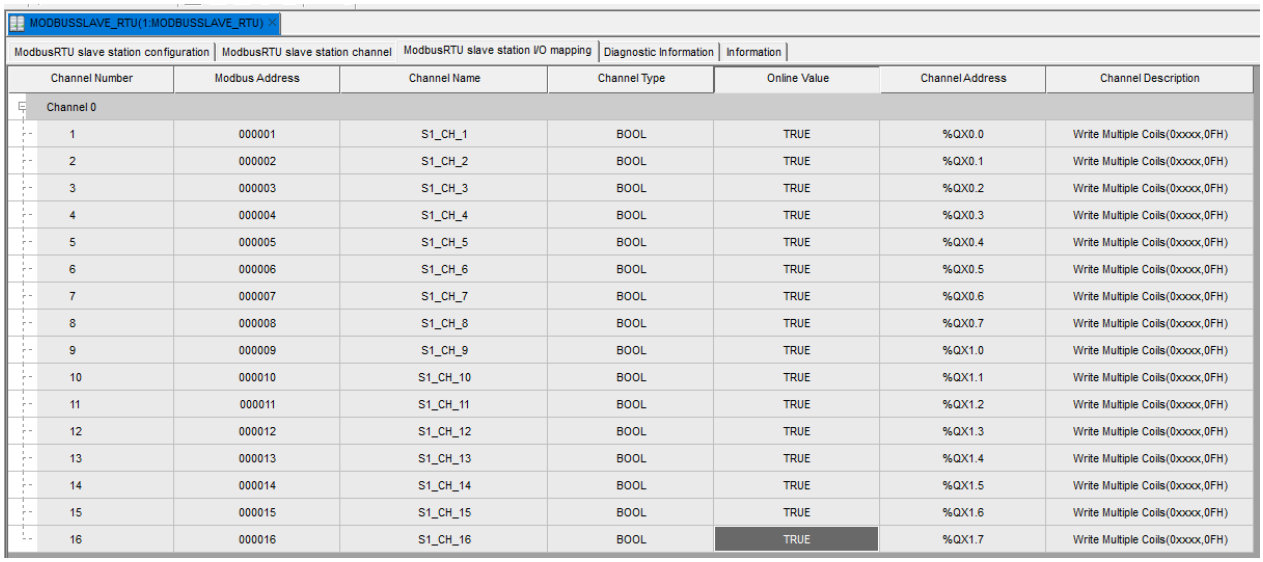
Slave Station Address	Function Code	High Bit of Starting Address	Low Bit of Starting Address	High Bit of Data	Low Bit of Data	Checksum CRC
01H	06H	0CH	1CH	0BH	B8H	4CH 1EH

RTU Response Frame

Slave Station Address	Function Code	High Bit of Starting Address	Low Bit of Starting Address	High Bit of Data	Low Bit of Data	Checksum CRC
01H	06H	0CH	1CH	0BH	B8H	4CH 1EH

- Function code 0FH (force multiple coils)

For the Modbus test of the function code 0FH (for forcing multiple coils), choose a suitable serial port debugging tool based on usage habits. The following example shows how to write the CPU address %QX0.0 - %QX1.7 with slave station address 1. The states of these 16 switch values are FF and FF, which means that all states of the DOs will be forced to ON. The slave station address is 01; The function code is 0F; The high and low bits of the starting address for %QX0.0 (Modbus address 0) are 00 and 00 in hexadecimal; The number of registers is 16, so enter 00 for the high bit of the register number and 10 for the low bit of the register number; The byte count is 02; The data bit are FF and FF; The checksum CRC is calculated by the serial port commissioning tool. In this case, the online values of %QX0.0 - %QX1.7 addresses in the monitored AT hardware configuration change to TRUE.


MODBUSSLAVE_RTU(1:MODBUSSLAVE_RTU)

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

Channel Number	Modbus Address	Channel Name	Channel Type	Online Value	Channel Address	Channel Description
Channel 0						
1	000001	S1_CH_1	BOOL	TRUE	%QX0.0	Write Multiple Coils(0xxxx,0FH)
2	000002	S1_CH_2	BOOL	TRUE	%QX0.1	Write Multiple Coils(0xxxx,0FH)
3	000003	S1_CH_3	BOOL	TRUE	%QX0.2	Write Multiple Coils(0xxxx,0FH)
4	000004	S1_CH_4	BOOL	TRUE	%QX0.3	Write Multiple Coils(0xxxx,0FH)
5	000005	S1_CH_5	BOOL	TRUE	%QX0.4	Write Multiple Coils(0xxxx,0FH)
6	000006	S1_CH_6	BOOL	TRUE	%QX0.5	Write Multiple Coils(0xxxx,0FH)
7	000007	S1_CH_7	BOOL	TRUE	%QX0.6	Write Multiple Coils(0xxxx,0FH)
8	000008	S1_CH_8	BOOL	TRUE	%QX0.7	Write Multiple Coils(0xxxx,0FH)
9	000009	S1_CH_9	BOOL	TRUE	%QX1.0	Write Multiple Coils(0xxxx,0FH)
10	000010	S1_CH_10	BOOL	TRUE	%QX1.1	Write Multiple Coils(0xxxx,0FH)
11	000011	S1_CH_11	BOOL	TRUE	%QX1.2	Write Multiple Coils(0xxxx,0FH)
12	000012	S1_CH_12	BOOL	TRUE	%QX1.3	Write Multiple Coils(0xxxx,0FH)
13	000013	S1_CH_13	BOOL	TRUE	%QX1.4	Write Multiple Coils(0xxxx,0FH)
14	000014	S1_CH_14	BOOL	TRUE	%QX1.5	Write Multiple Coils(0xxxx,0FH)
15	000015	S1_CH_15	BOOL	TRUE	%QX1.6	Write Multiple Coils(0xxxx,0FH)
16	000016	S1_CH_16	BOOL	TRUE	%QX1.7	Write Multiple Coils(0xxxx,0FH)

RTU Query Frame

Slave Station Address	Function Code	High Bit of Starting Address	Low Bit of Starting Address	High Bit of Register Number	Low Bit of Register Number	Byte Count	Data	Checksum CRC
01H	0FH	00H	00H	00H	10H	02H	FFH FFH	E3H 90H

RTU Response Frame

Slave Station Address	Function Code	High Bit of Starting Address	Low Bit of Starting Address	High Bit of Register Number	Low Bit of Register Number	Checksum CRC
01H	0FH	00H	00H	00H	10H	54H 07H

The following example shows how to write the CPU address %MX100.0 - %MX100.7 with slave station address 1. The states of these 8 switch values are 0F, which means that the states of the switch values %MX100.0 - %MX100.3 will be forced to ON. The slave station address is 01; The function code is 0F; The high and low bits of the starting address for %MX100.0 (Modbus address 100 * 8 + 3000 = 3800) are 0E and D8 in hexadecimal; The number of registers is 8, so enter 00 for the high bit of the register number and 08 for the low bit of the register number; The byte count is 01; The data bit is 0F; The checksum CRC is calculated by the serial port commissioning tool. In this case, the online values of %MX100.0 - %MX100.3 addresses in the monitored AT hardware configuration change to TRUE.

RTU Query Frame

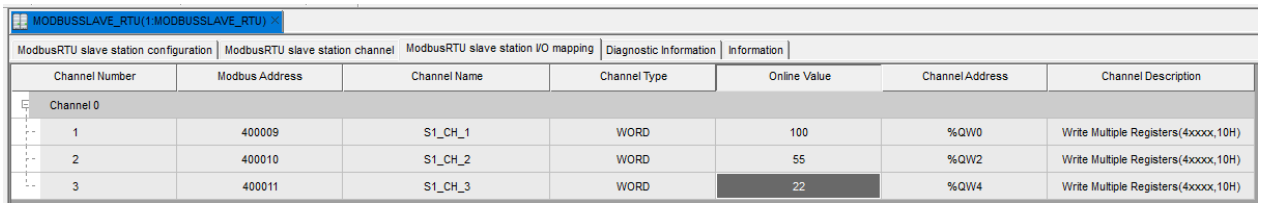
Slave Station Address	Function Code	High Bit of Starting Address	Low Bit of Starting Address	High Bit of Register Number	Low Bit of Register Number	Byte Count	Data	Checksum CRC
01H	0FH	0EH	D8H	00H	08H	01H	0FH	9FH ACH

RTU Response Frame

Slave Station Address	Function Code	High Bit of Starting Address	Low Bit of Starting Address	High Bit of Register Number	Low Bit of Register Number	Checksum CRC
01H	0FH	0EH	D8H	00H	08H	D6H DEH

- Function code 10H (preset multiple registers)

For the Modbus test of the function code 10H (for preseting multiple registers), choose a suitable serial port debugging tool based on usage habits. The following example shows how to write the CPU address %QW8 - %QW12 with slave station address 1. The states of these 3 analog values are 100, 55, and 22. The slave station address is 01; The function code is 10; The high and low bits of the starting address for %QW8 (Modbus address 4) are 00 and 04 in hexadecimal; The number of registers is 3, so enter 00 for the high bit of the register number and 03 for the low bit of the register number; The byte count is 3 words, or 6 bytes; The data is the value to be written to the slave station (e.g. 100, 50, and 22 are written to %QW8, %QW10, and %QW12, respectively), so enter hexadecimal numbers 00H 64H, 00H 32H, and 00H 16H for 100, 50, and 22; The checksum CRC is calculated by the serial port commissioning tool. In this case, the online values of %QW8 - %QW12 in the monitored AT hardware configuration are 100, 50, and 22 respectively.



RTU Query Frame

Slave Station Address	Function Code	High Bit of Starting Address	Low Bit of Starting Address	High Bit of Register Number	Low Bit of Register Number	Byte Count	Data	Checksum CRC
01H	10H	00H	04H	00H	03H	06H	00H 64H 00H 32H 00H 16H	F6H 9CH

RTU Response Frame

Slave Station Address	Function Code	High Bit of Starting Address	Low Bit of Starting Address	High Bit of Register Number	Low Bit of Register Number	Checksum CRC
01H	10H	00H	04H	00H	03H	C1H C9H

The following example shows how to write the values of CPU address %MW400 and %MW402 with slave station address 1. The slave station address is 01; The function code is 10; The high and low bits of the starting address for 400 (Modbus address 400 / 2 + 3000 = 3200) are 0C and 80 in hexadecimal; The number of registers is 2, so enter 00 for the high bit of the register number and 02 for the low bit of the register number; The byte count is 2 words, or 4 bytes; The data is the

value to be written to the slave station (e.g. 100 and 200 are written to %MW400 and %MW402, respectively), so enter hexadecimal numbers 00H 64H and 00H C8H for 100 and 200; The checksum CRC is calculated by the serial port commissioning tool. In this case, the online values of %MW400 and %MW402 in the monitored AT are 100 and 200.

RTU Query Frame

Slave Station Address	Function Code	High Bit of Starting Address	Low Bit of Starting Address	High Bit of Register Number	Low Bit of Register Number	Byte Count	Data	Checksum CRC
01H	10H	0CH	80H	00H	02H	04H	00H 64H 00H C8H	EEH 86H

RTU Response Frame

Slave Station Address	Function Code	High Bit of Starting Address	Low Bit of Starting Address	High Bit of Register Number	Low Bit of Register Number	Checksum CRC
01H	10H	0CH	80H	00H	02H	43H 70H

9.3.3 Configuring COM Port Free Protocol

9.3.3.1 Introduction

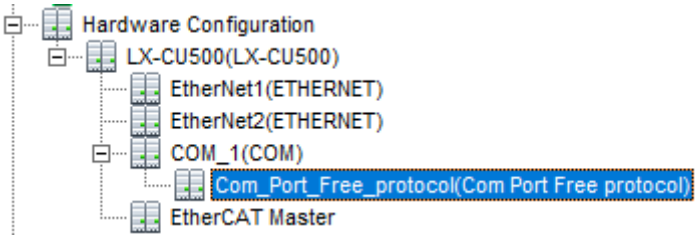
This operation enables to configure the COM port free protocol.

9.3.3.2 Requirement

The compilation passed.

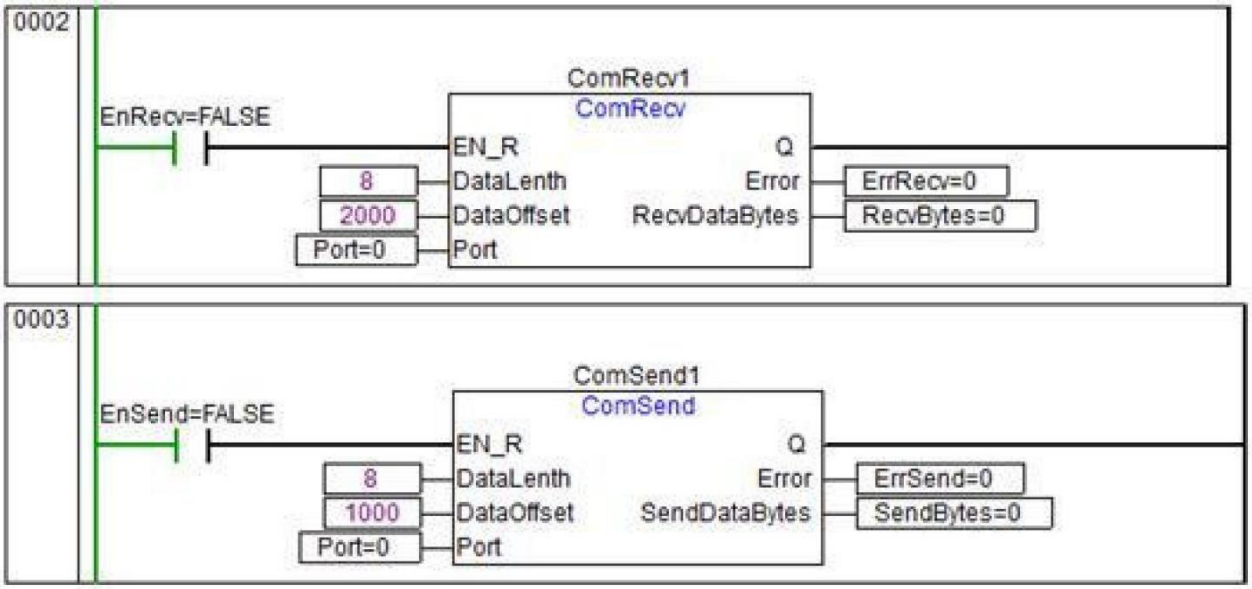
9.3.3.3 Steps

(1) Take COM_1 of LX as an example. The protocol is added in the same way as that of Modbus RTU master station protocol. See Add Master Station Protocol. The added tree node is shown in the figure.



(2) Double-click the COM port to set the communication parameters. See Serial Port Communication Parameter Configuration.

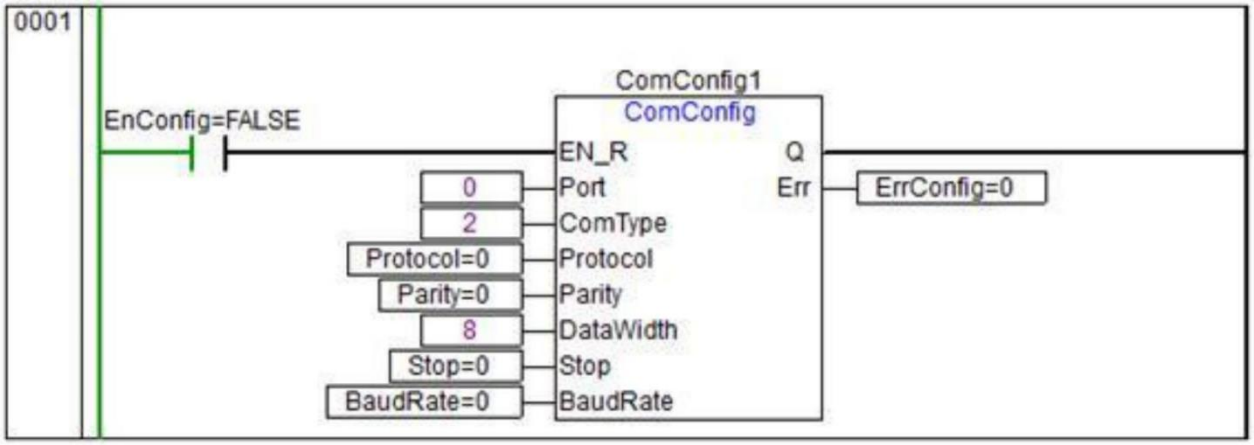
(3) Once the free protocol is configured, users can use the ComRecv and ComSend functional blocks to send and receive data, as shown in the figure.



(4) Online modification

When online, the free protocol configuration can be modified via the ComConfig functional block.

(5) The LX-CU500 is configured as shown in the figure below.



9.4 Configure EtherCAT

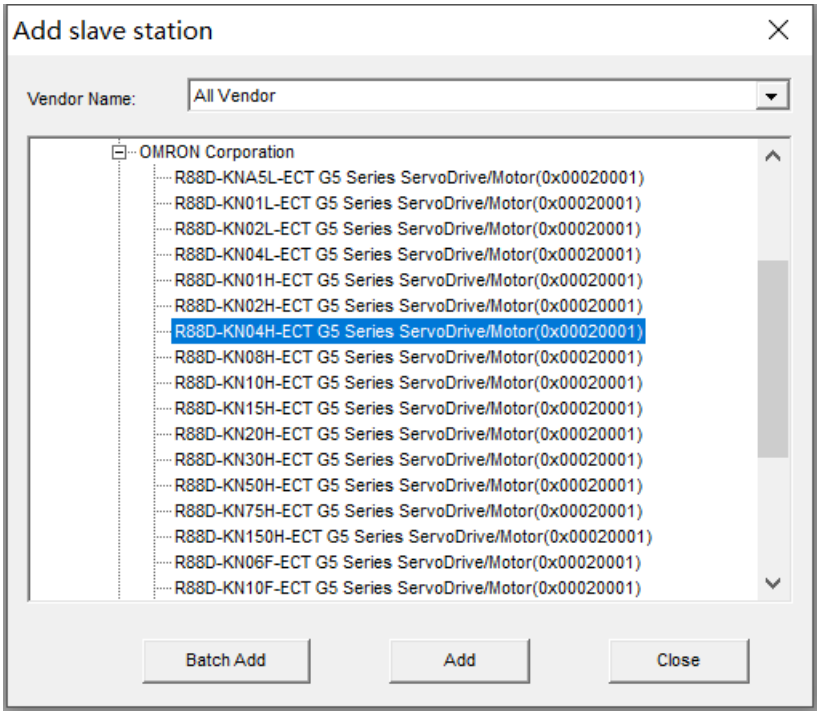
9.4.1 Add Slave Station

Users can add it via:

(1) Project Management Tree: Under the Hardware Configuration node, right-click EtherCAT Master.

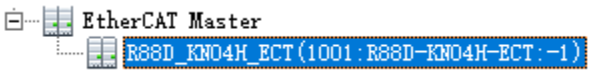
Click Add Slave Station.

(2) The Add Slave Station dialog box is displayed as shown in the figure. The Vendor drop-down box displays the vendor to which all slave station devices belong and can be selected by users. Also, the models of devices supported by this vendor are displayed under the Device Library node. Other third-party devices can be imported manually. See the hardware configuration document for detailed import operations.



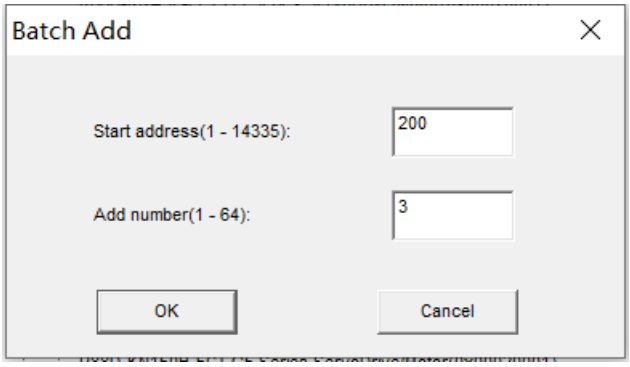
(3) According to the slave station device connected to the actual project, select the device model. Click Add to complete adding the slave station device.

As shown in the figure, a corresponding slave station is generated under the EtherCAT Master node. The slave station name consists of: slave station name_serial number (slave station number: slave station device model: axis number). The definition of the slave station name complies with the variable naming specification. See Variable Names of Variable Declaration Specification. The default slave station number starts from 1001, which can be modified by users in the Slave Station Configuration interface.

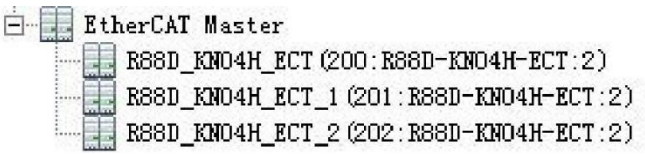


■ Batch adding

(1) A set number of slave stations can be inserted by batch adding. As shown in the figure above (Select slave station devices), select the devices to be added. Click Bulk Add to open the Bulk Add dialog box, as shown in the figure below.



(2) Enter the slave station numbers and the number of slave stations to be added. Click OK. 3 slave stations with slave station numbers starting from 200 are added, as shown in the figure.



If the entered starting address has been occupied, detect the unoccupied starting address as the slave station number in sequence, and add the slave stations in turn.

9.4.1.1 Module adding methods

When adding modules or protocols under the Hardware Configuration node, select the appropriate adding method as needed.

- Single adding

  In the Add dialog box, select the device that needs to be added. Click Add button to add the module.

- Repeat adding

  In the Add dialog box, select the device that needs to be added. Click the Add button successively to add the module repeatedly. After adding, users can select other modules in the dialog box to continue adding them repeatedly.

- Batch adding

  EtherCAT slave stations can be batch added. In the Add dialog box, select the devices that need to be added. Click Bulk Add to bulk-add the devices according to the set number of slave stations.

9.4.1.2 Scan slave station module

9.4.1.2.1 Introduction

After connecting to the controller online, users can scan the number of slave stations and slave station data configured under the current EtherCAT master station. Users can choose to add the corresponding slave station device under the EtherCAT_Master node.

When added, slave station devices are added cascading down according to the actual network topology.

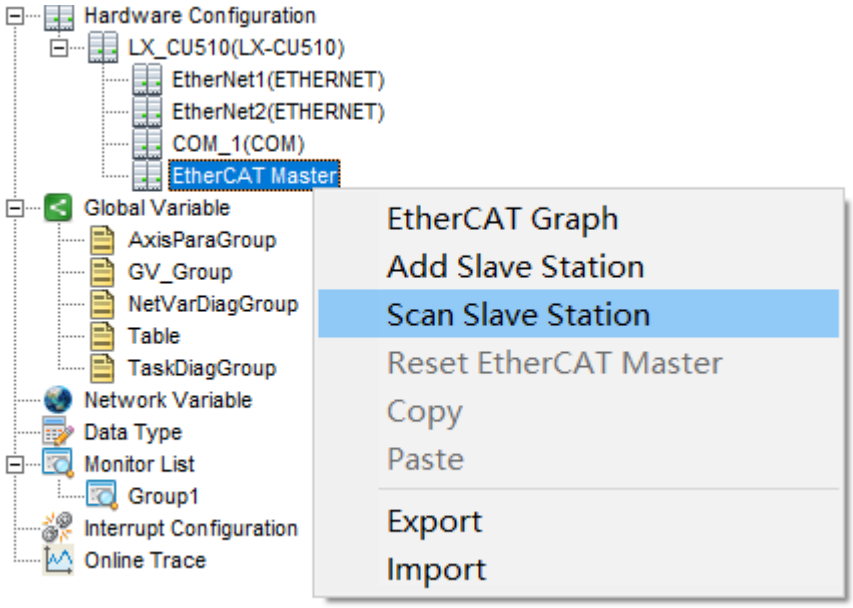
9.4.1.2.2 Requirement

Login to the controller successfully

9.4.1.2.3 Steps

To scan the controller, follow the steps below:

(1) Right-click the EtherCAT_Master node. Select the Scan Slave Station command.



(2) The Login to Controller dialog box is displayed.

(3) Enter the controller account and password. Click OK.

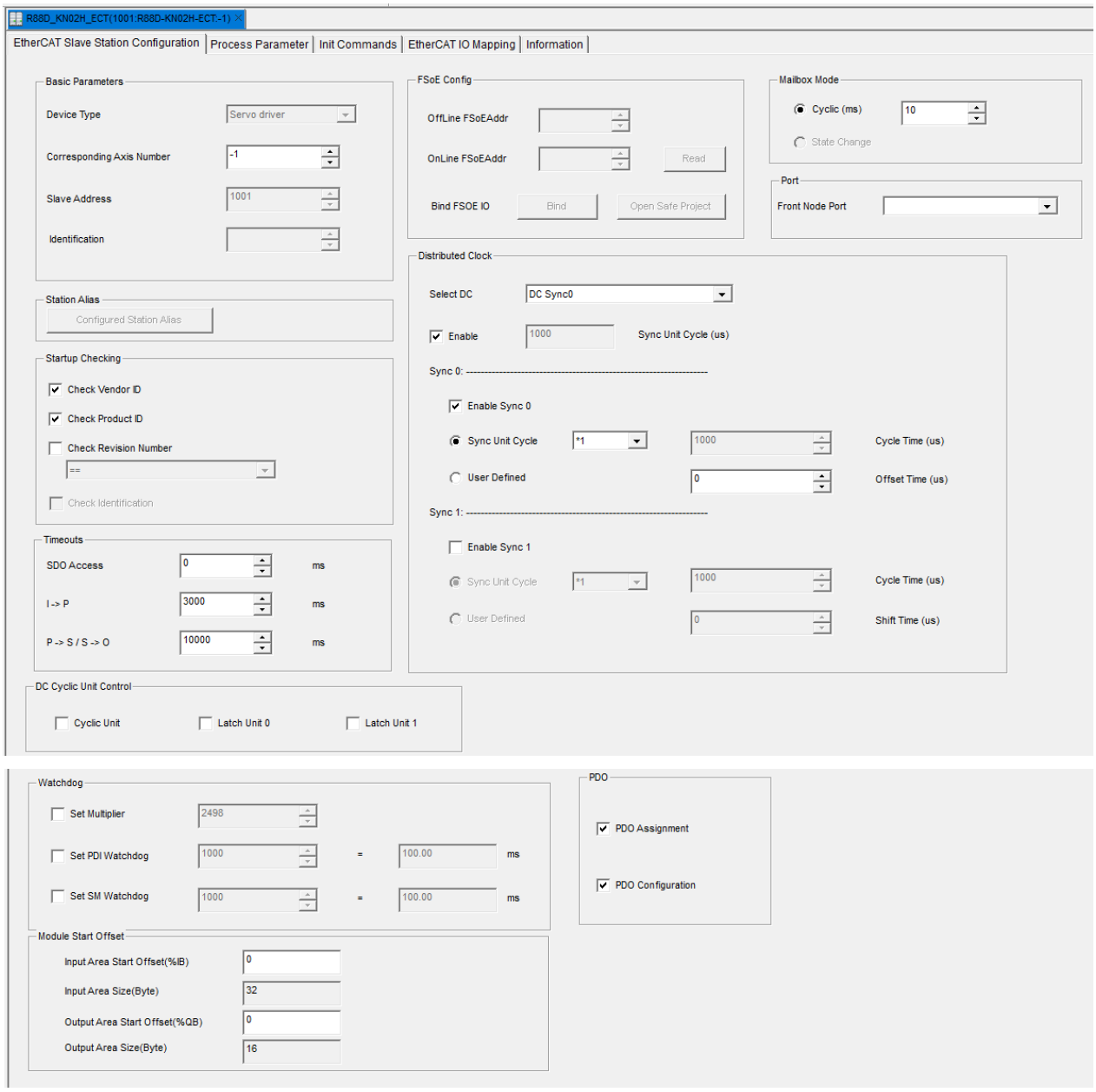
(4) In the pop-up EtherCAT_Master Network Topology dialog box, select the slave station modules that need to be added.

(5) Click Add. The modules are added under the EtherCAT_Master node.

9.4.1.3 Configure slave station address

9.4.1.3.1 Introduction

Right-click the slave station under the EtherCAT_Master node. The slave station's Parameter Configuration interface is displayed in the right area, as shown in the figure. This interface includes five tabs: EtherCAT Slave Station Configuration, Process Parameters, Initialization Command, EtherCAT IO Mapping, and Information. Related parameters can be set through each tab.



9.4.1.3.2 Requirement

The compilation passed.

9.4.1.3.3 EtherCAT slave station configuration

- Basic parameters

  Device type: Servo and general IO supported.

  Axis number: When the device type is a servo, the axis number needs to be set with a range of 0 to 63 and the default value of 2.

Slave address: The default slave station address is 1001, and the user can set the slave address here. Address range: 1 - 65535.

- Mailbox mode

  Cycle: Interval (milliseconds) for reading the input mailbox (polling mode), setting range: 0~99999.

  Status transition: The input mailbox will only be read when the status bit is set.

- Startup check

  By default, when the system starts, it automatically checks the supplier 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.

- Timeout

  By default, the following operations are not defined as timeouts. However, if users want to see if the operation exceeds a specific time, the time can be set here (in microseconds): SDO access: When the system starts, send the SDO list, with the setting range 0 - 100000.

  I -> P: Convert from Initialization to Pre-run mode, with setting range: 0 - 100000.

  P -> S/S -> O: Convert from Pre-run to Safe running mode or from Safe running to Run mode, with 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. Setting range: 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 shall be allocated to the local microprocessor. The control functions have been completed in the registers of the EtherCAT slave station: Possible settings: Cycle unit, Latch unit 0, Latch unit 1.

- Watchdog

  Setting multiplier: The watchdogs PDI and SM 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

  PDO allocation: Check this parameter, and the request command with index 0x1cxx is added to the Initialization 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, and the corresponding PDO configuration is added to the Initialization Command, and then 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, you can modify the start address of the module's EtherCAT IO mapping input channel.

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

  Output Area Start Offse (%QB)t: In offline mode, you can modify the start address of the module's EtherCAT IO mapping output channel.

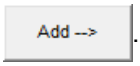
  Output Area Size (Byte): The number of bytes in the output area.

9.4.1.4 Module parameter configuration

9.4.1.4.1 IO configuration

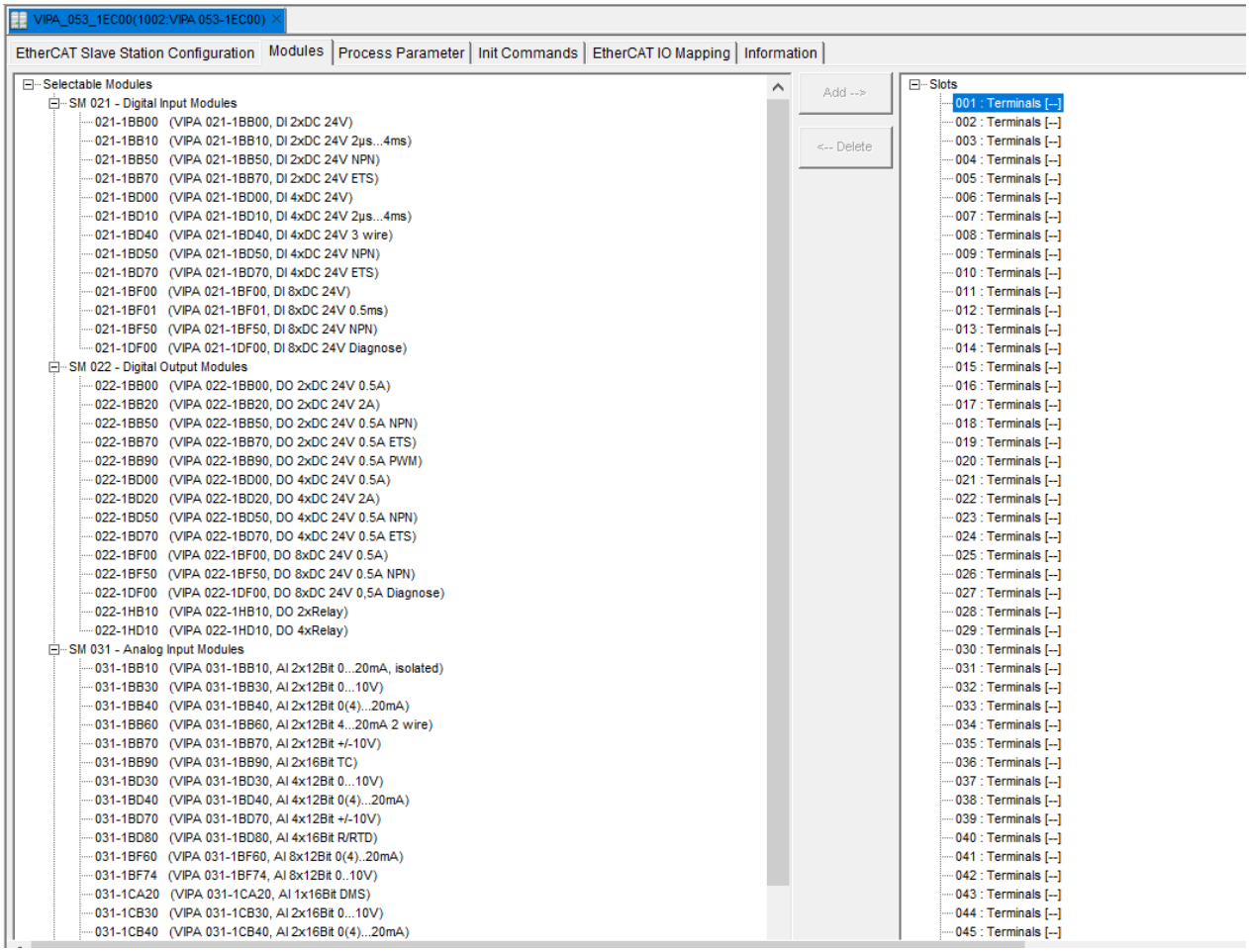
The EtherCAT master station supports VIPA IO modules and Slot multi-axis modules. Users can add the modules in the Modules tab depending on the actual connected IO module type.

1. Add VIPA IO module

   To add a module, first select the position of the IO module to be added in the right list box. Then select the module in the left list box. Click 

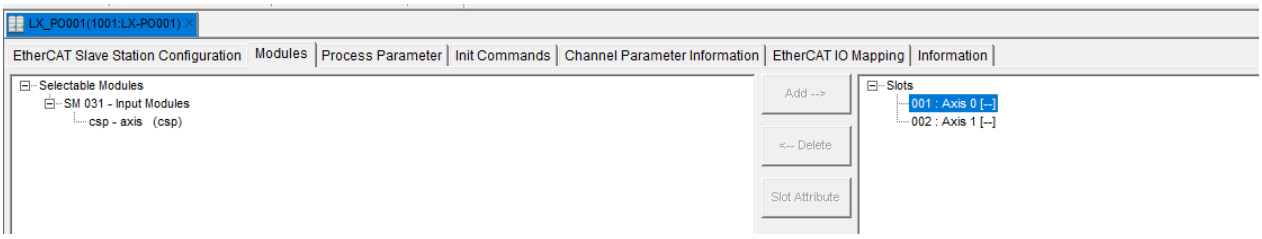
. To delete an added module, select the module, and

click <-- Delete .

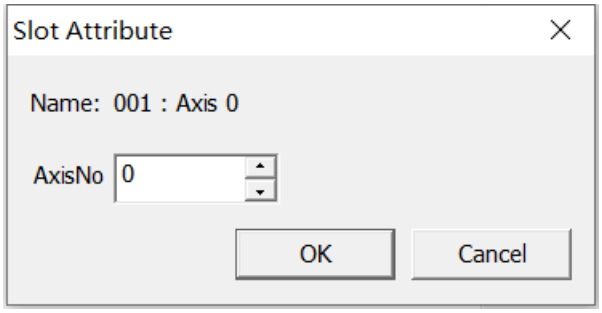


2. Add Slot multi-axis module

To add a Slot multi-axis module, first select an axis position in the right list box. Then select the module in the left list box. Click Add --> . To delete an added module, select the module, and click <-- Delete .



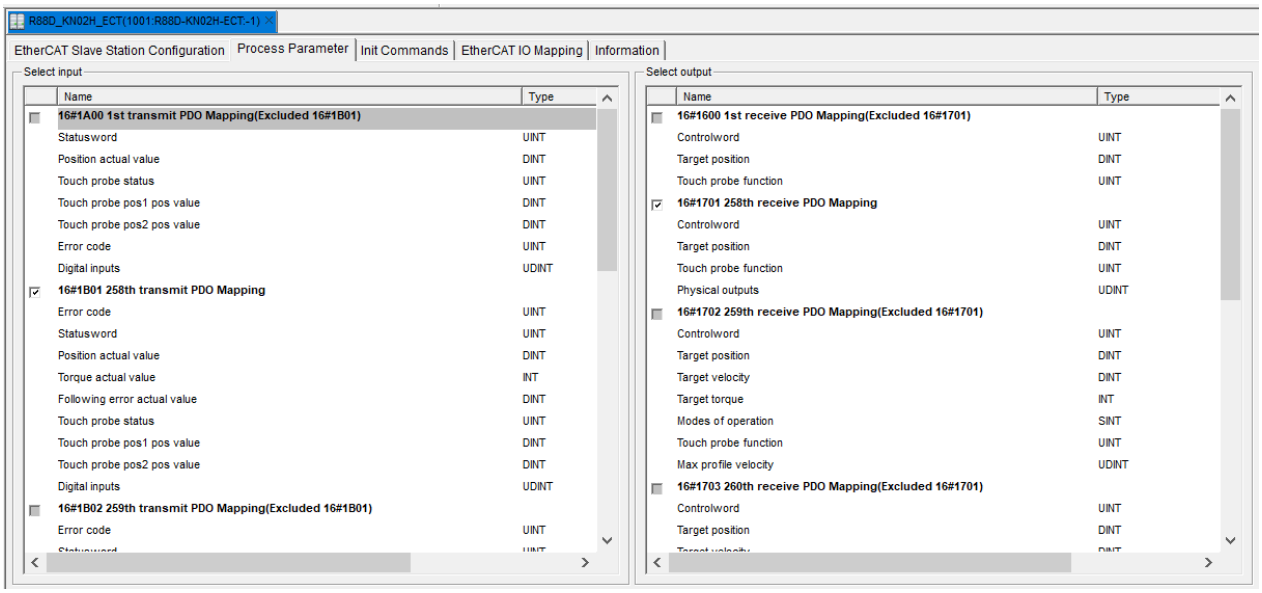
After adding modules, users need to configure an axis number for each module. In the right list box, select the added module. Click Slot Properties to open the Properties interface. And configure the axis number here.



9.4.1.4.2 Process parameters

The Process Parameters tab displays all modules supported by the slave station device, as shown in the figure. Check the desired PDO Mappings to automatically map the channel addresses of the checked modules in the EtherCAT IO Mapping tab. The process parameters of VIPA slave stations and Slot multi-axis modules are automatically mapped by the IO modules added in Modules, without the need for manual configuration.

The process parameters shall be set according to the requirements of the device and field applications. The actual connected slave station shall be of the same type and sequence as the AT configuration. Otherwise, the EtherCAT protocol stack cannot operate normally.



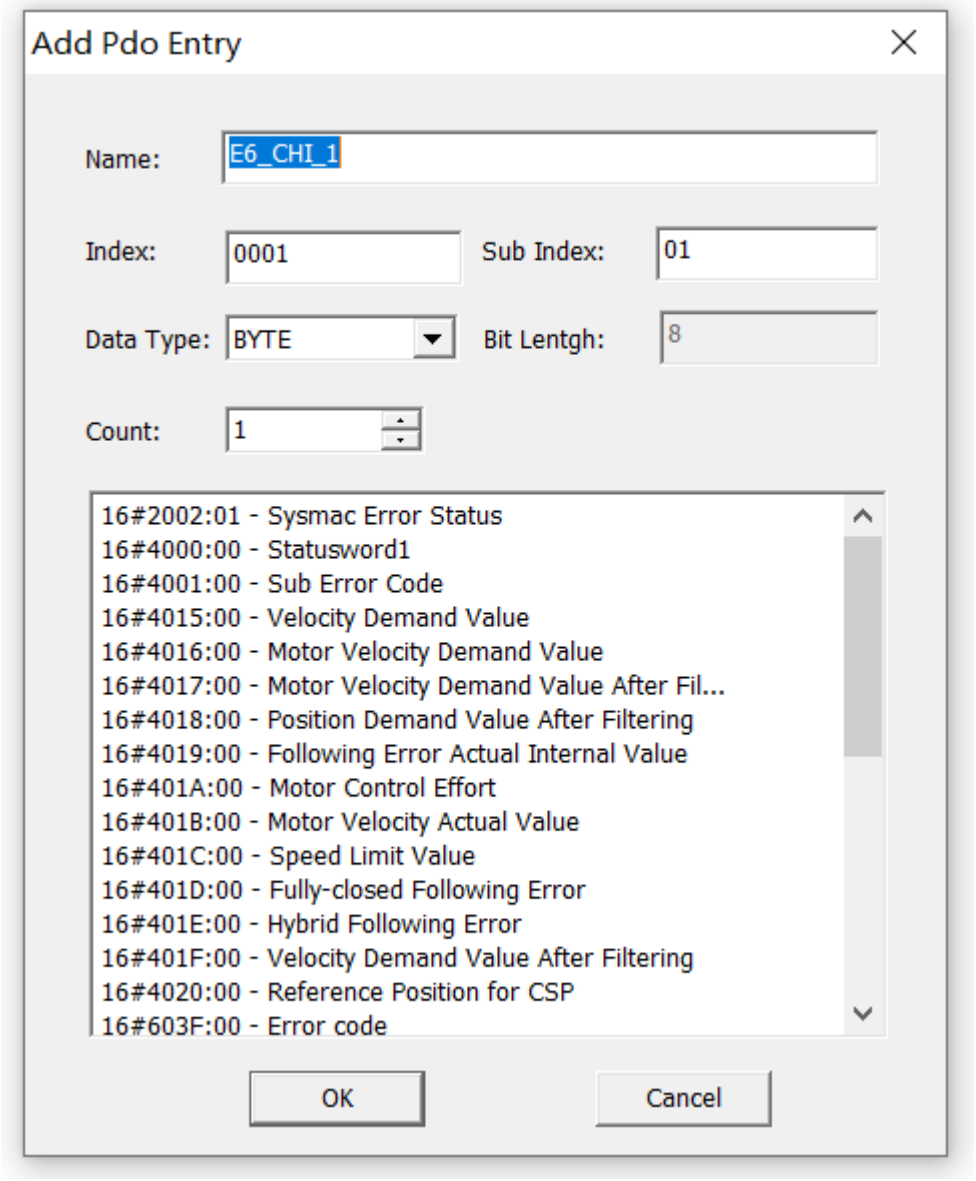
- Add PDO parameter

  Users can add system PDO parameters or custom PDO parameters for input/output process parameters. Right-click the PDO parameter. Select the Add command. The PDO Parameters dialog box is displayed, as shown in the figure. Users can enter custom PDO parameter information, or select a parameter in the list box to add the parameter.

  Only one PDO parameter with the same name under all input/output process parameters can take effect. To add the parameter, please note the following two points:

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

- If the PDO parameter has been added to other process parameters and is checked to take effect, the parameter cannot be added repeatedly.

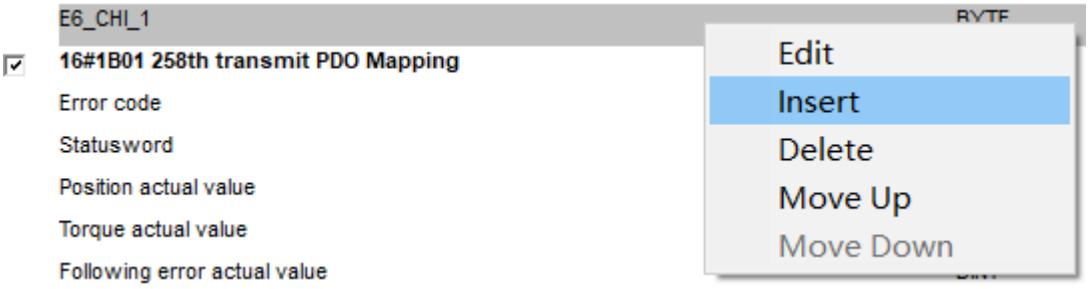


■ Insert PDO parameter

Users can insert a new parameter in front of a PDO parameter. The PDO Parameters dialog box is displayed, as shown in the figure above. Select a parameter to add.

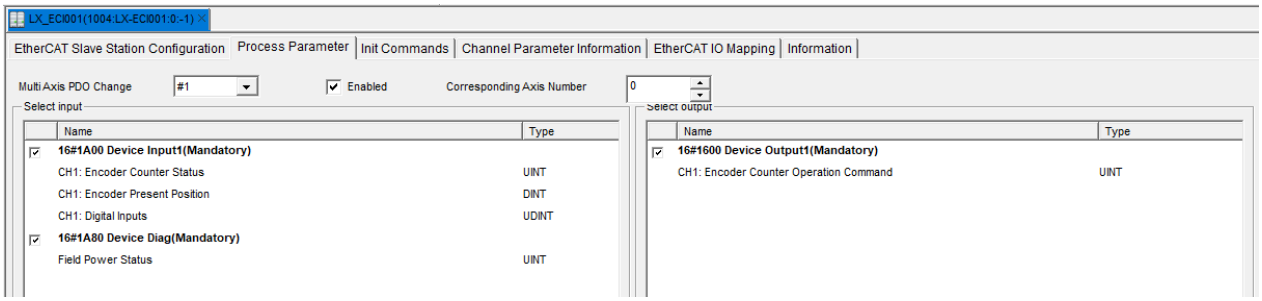
Only one PDO parameter with the same name under all input/output process parameters can take effect. To add the parameter, please note the following two points:

- The same PDO parameter cannot be added to the same process parameter repeatedly.
- If the PDO parameter has been added to other process parameters and is checked to take effect, the parameter cannot be added repeatedly.



- Parameters for PDO type multi-axis modules

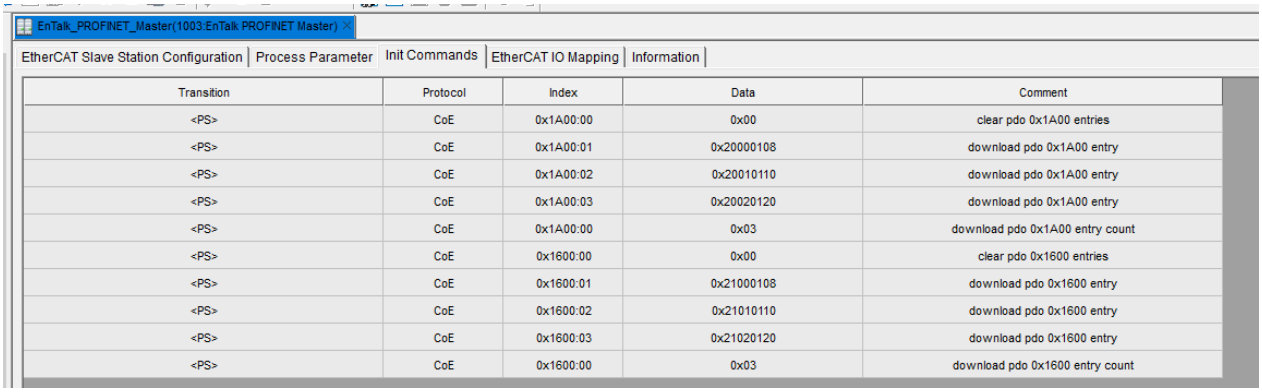
  For the process parameters for PDO type multi-axis modules, the corresponding axis numbers need to be specified. In the Process Parameter Configuration interface, by switching the interface, the process parameters of different axes can be configured separately.



  - Switching of multi-axis process parameters: In the Process Parameter Configuration interface, select different axes. Switch to the corresponding axes to configure the process parameters.
  - Axis number: The process parameter is associated with the specified axis number, which can be set by clicking the up and down buttons.
  - Enable: Enables all process parameters for the axis by default. When unchecked, the process parameters for the axis are disabled.

9.4.1.4.3 Initialization command

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

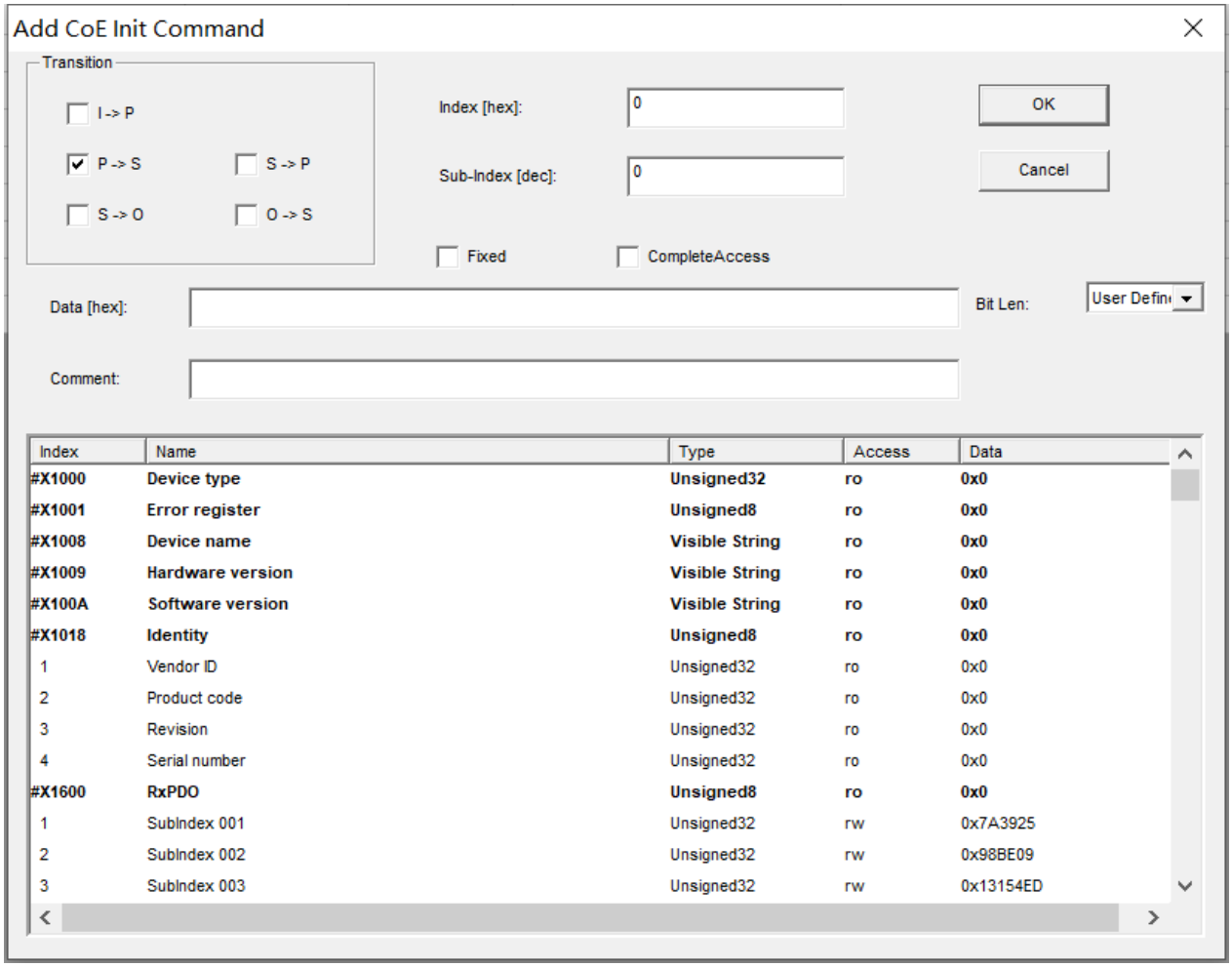


List of Parameter Items for Initialization Command

Parameter	Description
Convert	The conversion commands 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

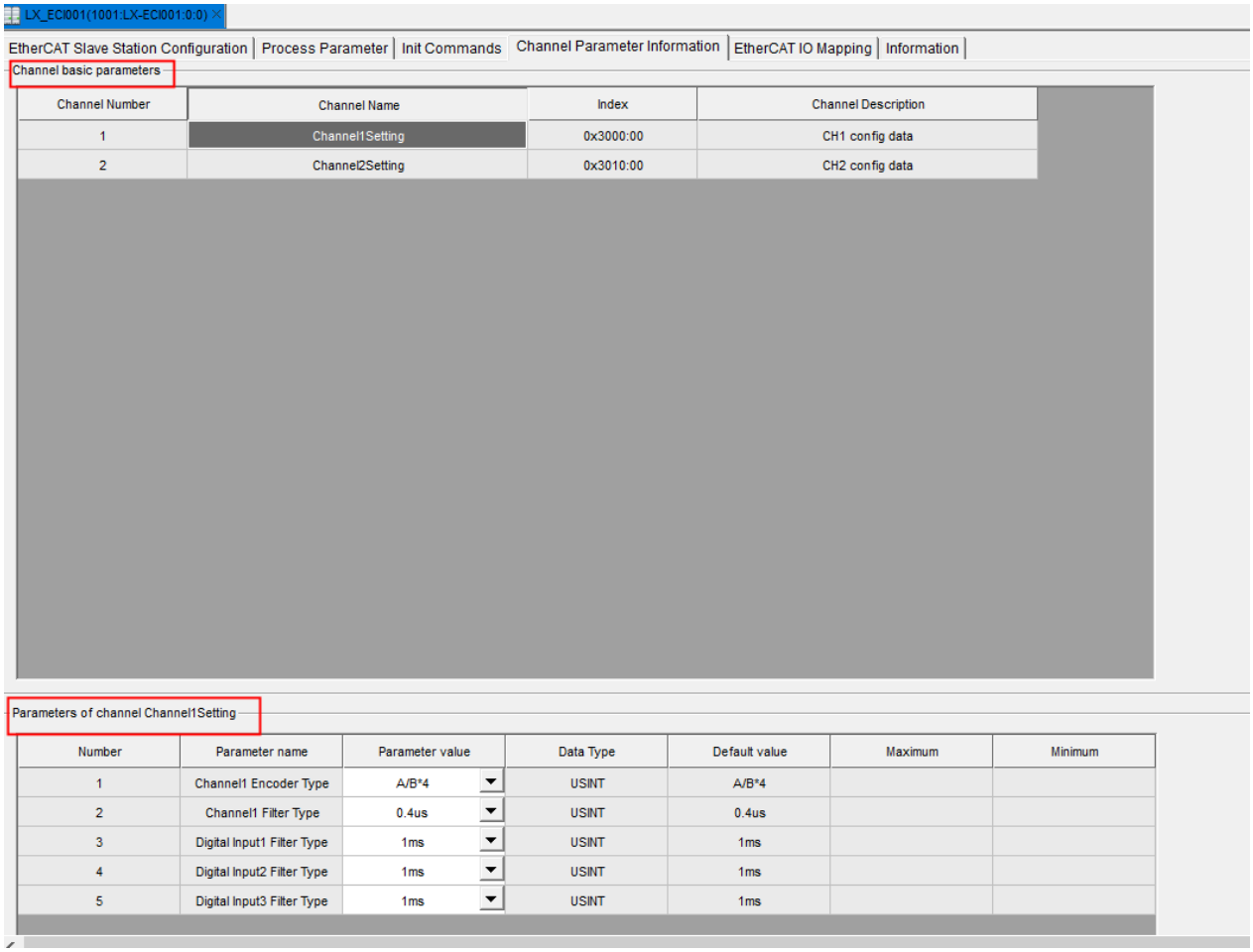
You can add an Initialization Command by right-clicking the Add menu item.

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.



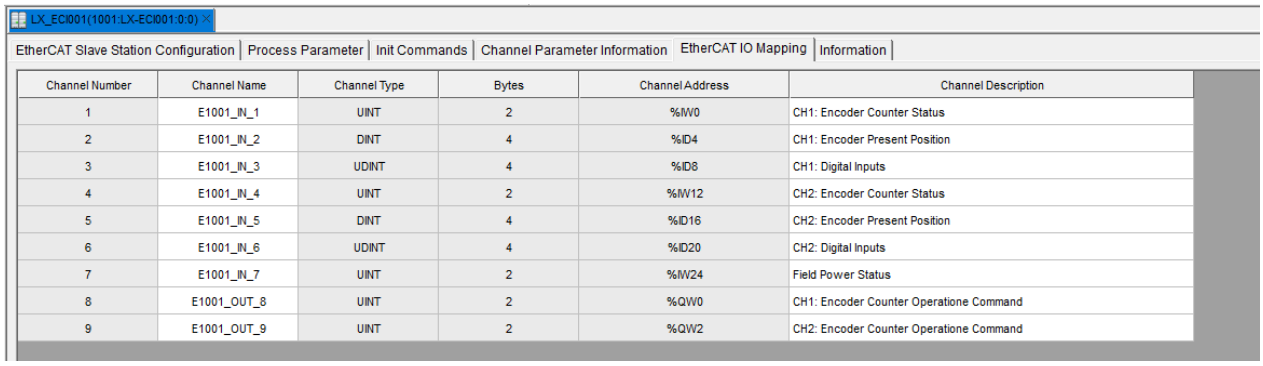
9.4.1.4.4 Channel parameter information

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 module type. Please configure according to the actual configuration module.



9.4.1.4.5 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.



9.4.1.4.6 Module information

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

9.4.1.5 Modify module name

9.4.1.5.1 Introduction

After adding the module, users can modify the module by the Rename command in the right-click menu or by the Alias parameter in the Module Properties page.

9.4.1.5.2 Requirement

The module has been added

9.4.1.5.3 Steps

Rename via the right-click menu:

(1) Under the Hardware Configuration node, right-click the module model.

(2) In the right-click menu, click the Rename command. The Rename dialog box is displayed.

(3) Enter the new name

The naming rules are the same as the variable definition rules. See Variable Names.

(4) Click OK to complete the modification of the module name.

Rename via the Module Properties page (Currently only supports renaming DP master/slave stations):

(1) Under the Hardware Configuration node, double-click the module model. The Module Properties page is displayed.

(2) On the Device Configuration page, double-click the module name of the Alias parameter. The cursor shows an editable state.

(3) Enter the new name. Press Enter to complete the modification of the module name.

9.4.1.6 Delete module

Users can delete it via:

- Hardware Configuration: Right-click the LX module node. Select Delete to delete the selected module.

9.4.1.7 Insert module

9.4.1.7.1 Introduction

Insert a new module in front of the currently selected IO module.

9.4.1.7.2 Requirement

The compilation passed.

9.4.1.7.3 Steps

Users can insert the module via:

- Graphical Configuration Editing Area: Drag the module from the DEVICE LIBRARY to the graphic configuration editing area, release the mouse, and the module will be inserted in front of the selected module;
- Project Management Tree: Right-click the module. Select Insert Slave Station，module will be inserted in front of the selected module.

9.4.2 Configuring DeviceNet Master Station

9.4.2.1 Introduction

LX-CM006 is a master module of DeviceNet protocol, which communicates with instruments or sensors in the field through CAN interface for data operation.

The module acts upwards as an EtherCAT slave station and downwards as a DeviceNet master station. When configured, the module is added under the EtherCAT node or coupler node.

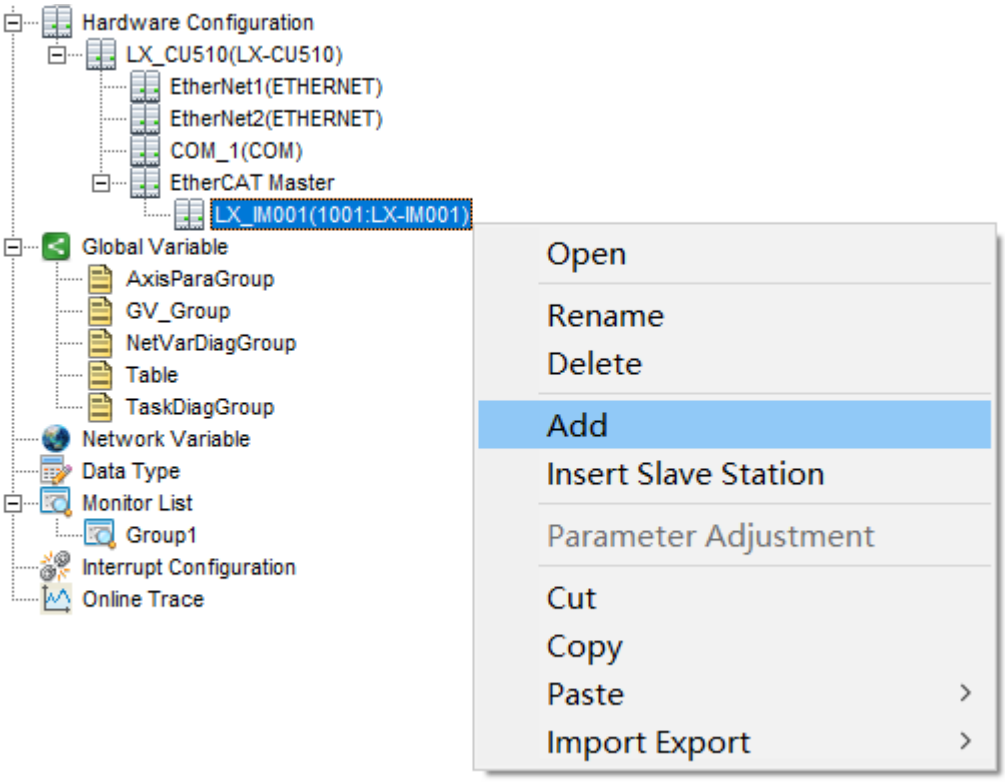
9.4.2.2 Requirement

The compilation passed.

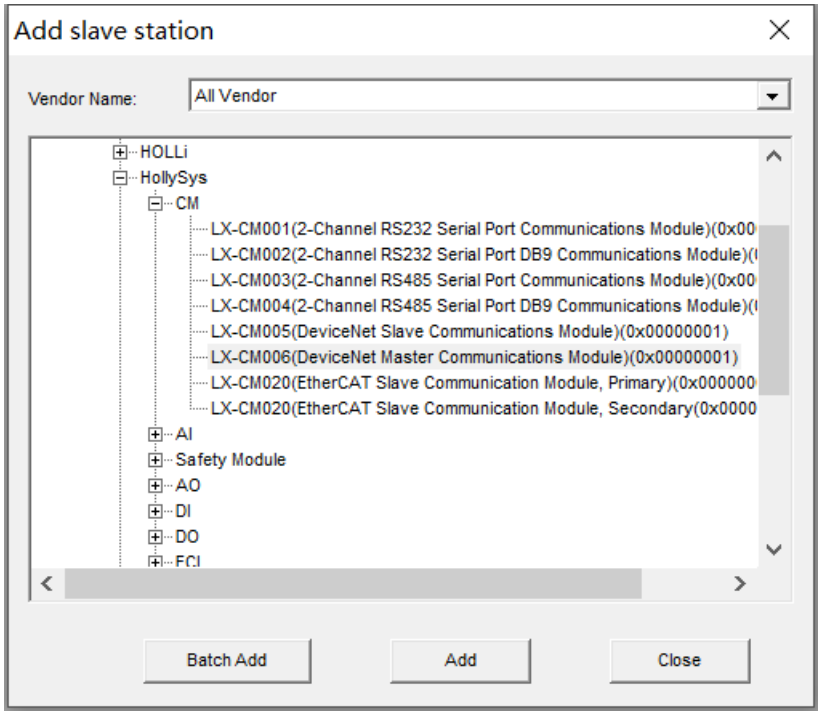
9.4.2.3 Steps

To configure a DeviceNet master station, follow the steps below:

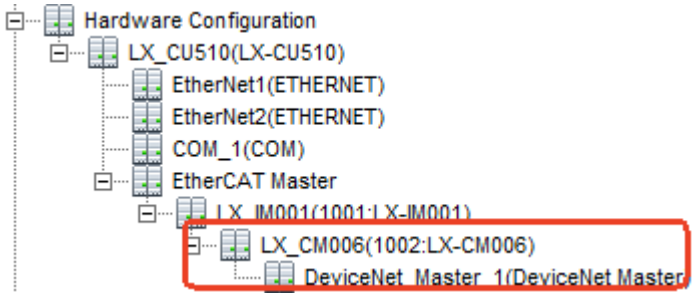
(1) Right-click the EtherCAT node or coupler node LX-IM001. Select the Add command. The Add dialog box is displayed.



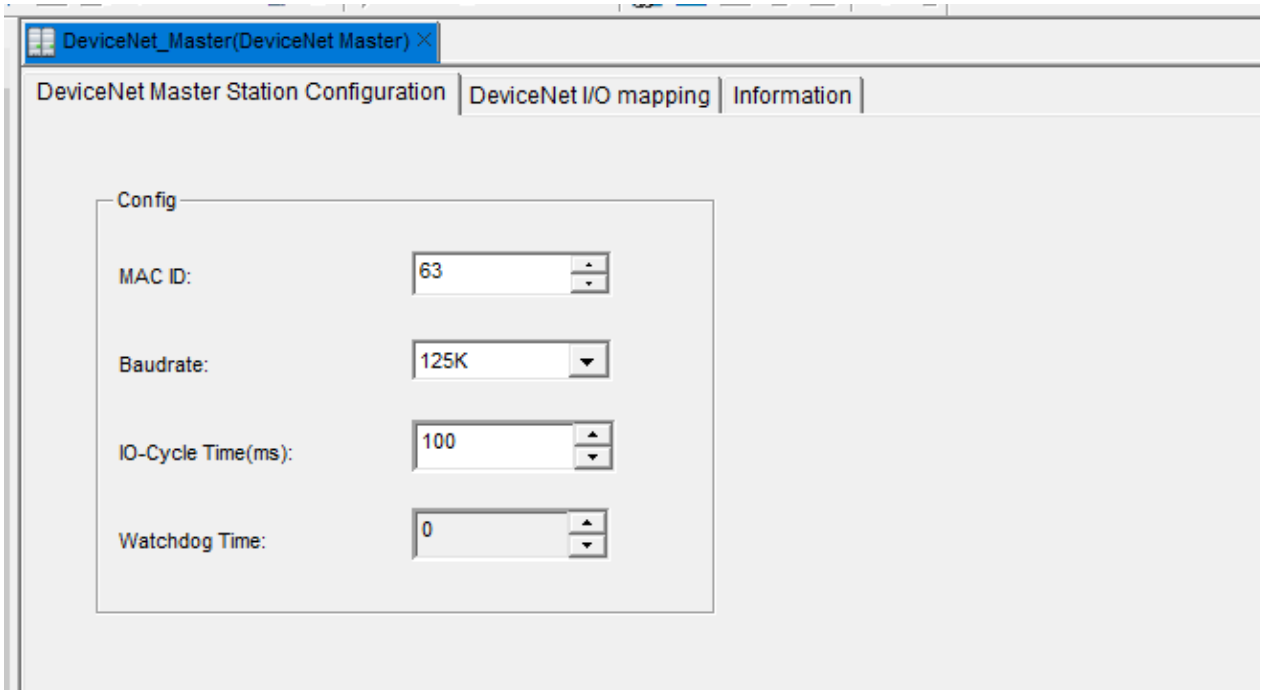
(2) Select the LX-CM006 module. Click Add.



(3) The LX-IM001 node displays the modules and the DeviceNet master station.



(4) Double-click the DeviceNet_Master node to configure the master station parameters.

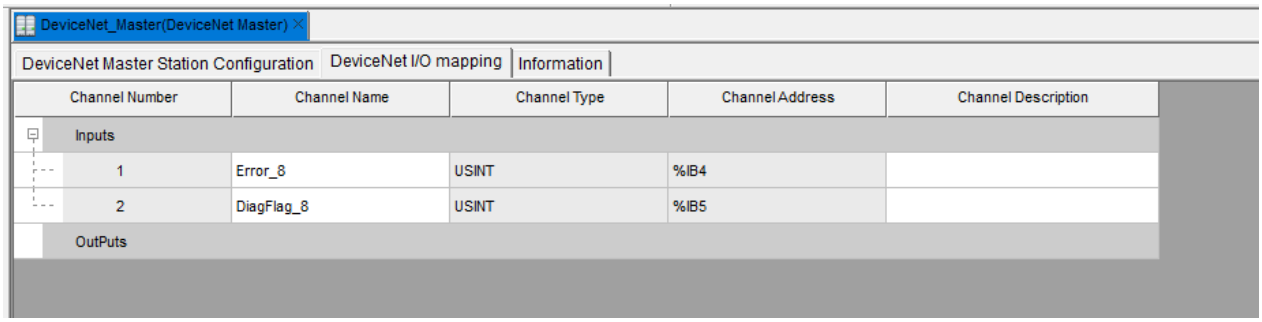


DeviceNet_Master configuration parameters

Parameter	Parameter Value	Default Value	Description
MAC ID	0~63	1	Each master station is assigned a unique ID. That is, the ID shall not be duplicated during the configuration
Baud Rate	125 K, 250 K, 500 K	125 K	Configure baud rate based on end-to-end network distance The baud rate is 125 Kbp/s for 500 m The baud rate is 250 Kbp/s for 250 m The baud rate is 1500 Kbp/s for 100 m
IO-cycle Time (ms)	1~65535	100	
Watchdog time	0~ 100000	0	Not used, no configuration required

■ DeviceNet I/O mapping

The DeviceNet I/O Mapping tab is shown in the figure below.



9.4.3 Configuring DeviceNet Slave Station

9.4.3.1 Introduction

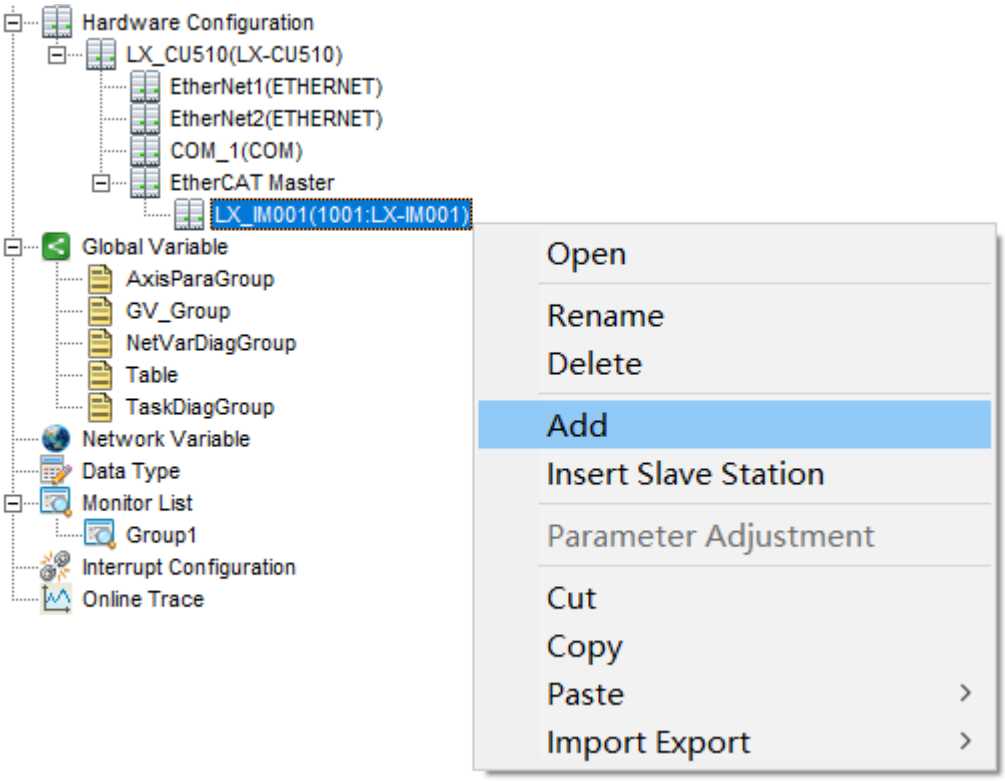
LX-CM005 is a slave module of DeviceNet protocol, which communicates with instruments or sensors in the field through CAN interface for data operation.

The module acts upwards as an EtherCAT slave station and downwards as a DeviceNet slave station. When configured, the module is added under the EtherCAT node or coupler node.

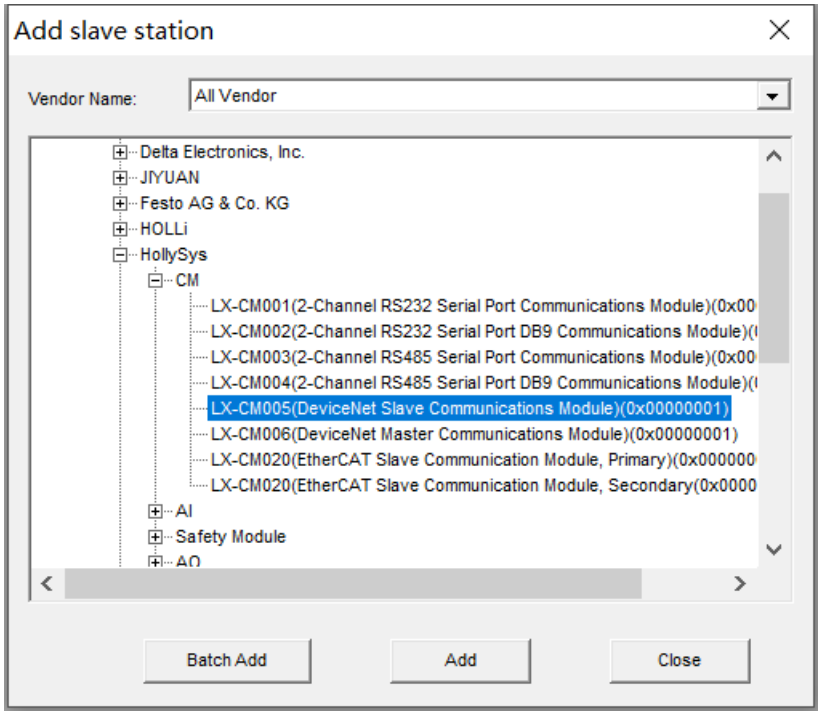
9.4.3.2 Steps

To add a DeviceNet slave station, follow the steps below:

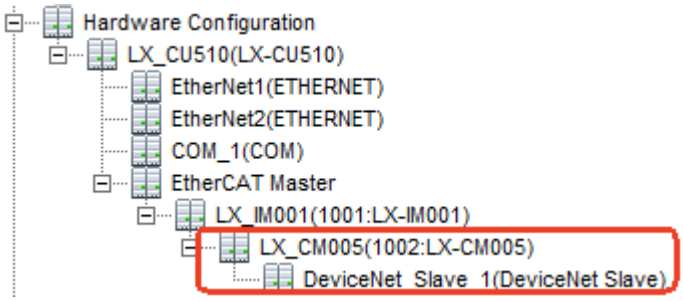
(1) Right-click the EtherCAT node or coupler node LX-IM001. Select the Add command. The Add dialog box is displayed.



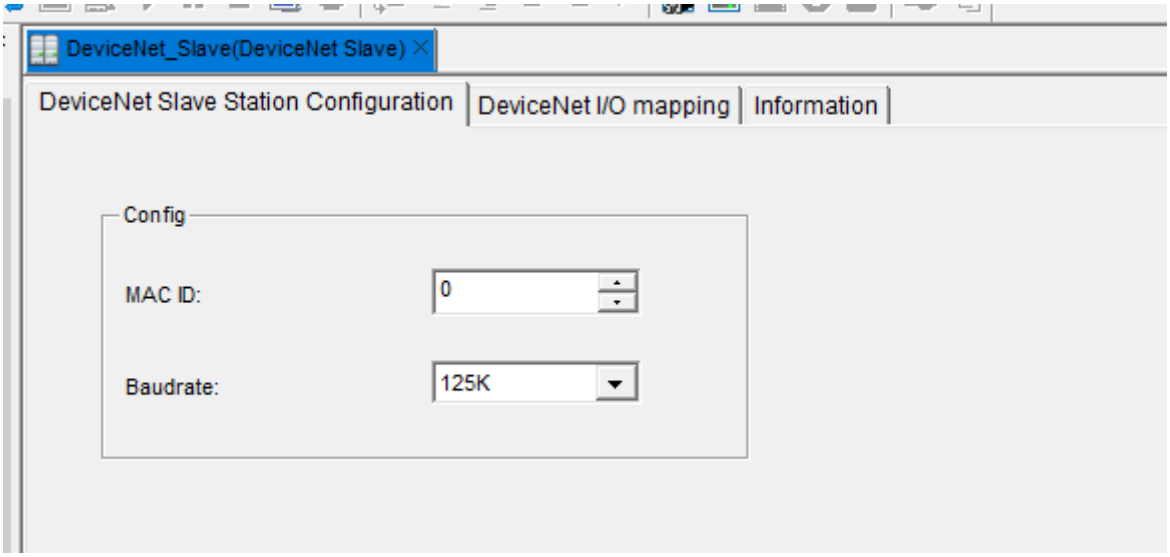
(2) Select the LX-CM005 module. Click Add. The module is displayed under the LX-IM001 node.



(3) The LX-IM001 node displays the modules and the DeviceNet slave station.



(4) Double-click the added DeviceNet_Slave module tree node. The configuration interface is displayed.



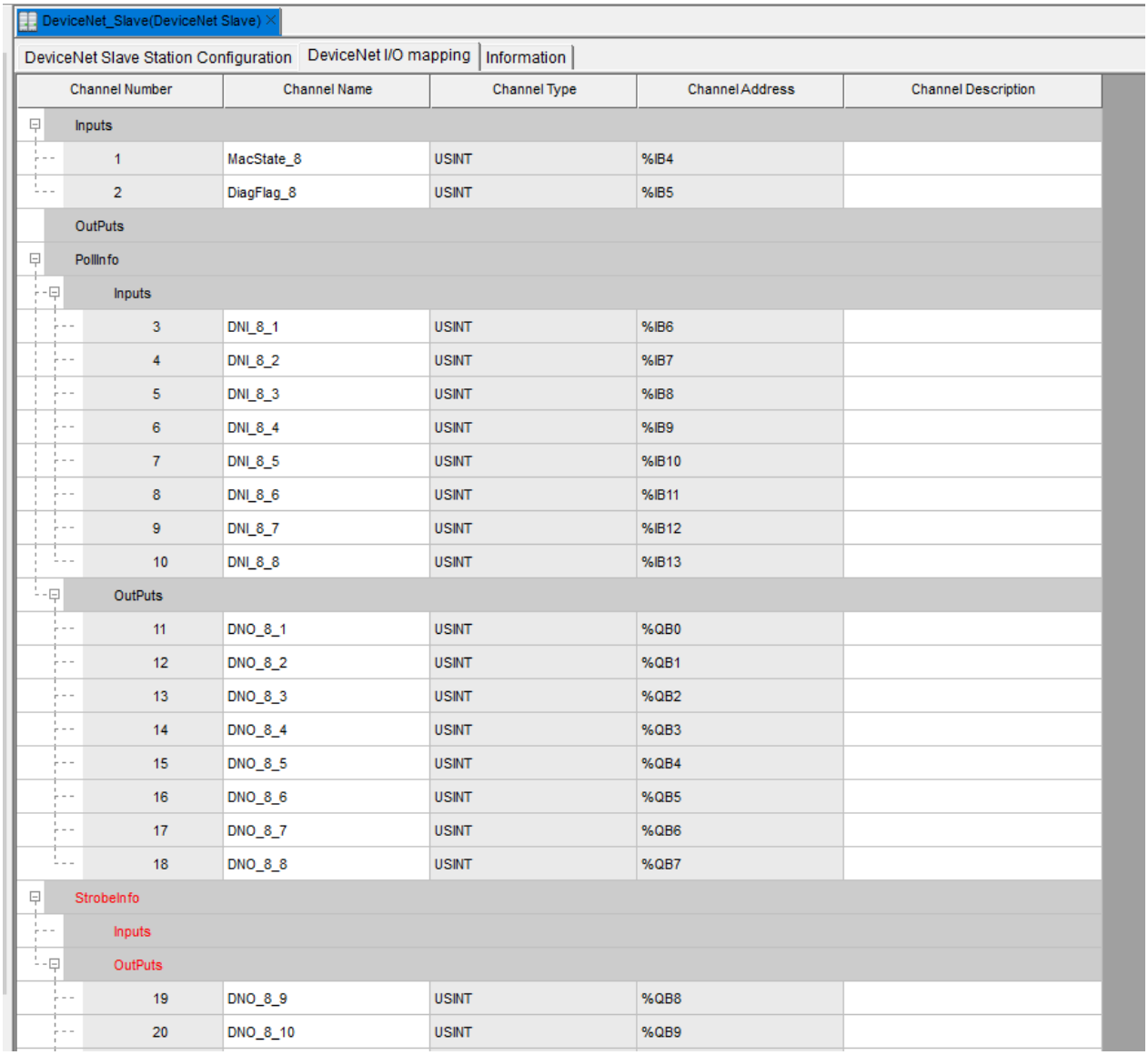
□ DeviceNet configuration parameters

DeviceNet_Slave configuration parameters

Parameter	Parameter Value	Default Value	Description
MAC ID	0~63	0	Each slave station is assigned a unique ID. That is, the ID shall not be duplicated during the configuration
Baud Rate	125 K, 250 K, 500 K	125 K	Configure baud rate based on end-to-end network distance The baud rate is 125 Kbp/s for 500 m The baud rate is 250 Kbp/s for 250 m The baud rate is 1500 Kbp/s for 100 m

□ DeviceNet I/O mapping

The I/O data of the slave station is displayed in this tab, including input/output data, polling information, bit strobe information, status change information, and cycle information. Among them, the information in red (bit strobe 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.



Channel Number	Channel Name	Channel Type	Channel Address	Channel Description
Inputs				
1	MacState_8	USINT	%IB4	
2	DiagFlag_8	USINT	%IB5	
OutPuts				
PollInfo				
Inputs				
3	DNI_8_1	USINT	%IB6	
4	DNI_8_2	USINT	%IB7	
5	DNI_8_3	USINT	%IB8	
6	DNI_8_4	USINT	%IB9	
7	DNI_8_5	USINT	%IB10	
8	DNI_8_6	USINT	%IB11	
9	DNI_8_7	USINT	%IB12	
10	DNI_8_8	USINT	%IB13	
OutPuts				
11	DNO_8_1	USINT	%QB0	
12	DNO_8_2	USINT	%QB1	
13	DNO_8_3	USINT	%QB2	
14	DNO_8_4	USINT	%QB3	
15	DNO_8_5	USINT	%QB4	
16	DNO_8_6	USINT	%QB5	
17	DNO_8_7	USINT	%QB6	
18	DNO_8_8	USINT	%QB7	
StrobeInfo				
Inputs				
OutPuts				
19	DNO_8_9	USINT	%QB8	
20	DNO_8_10	USINT	%QB9	

Chapter 10 PLC Programming

10.1 Data

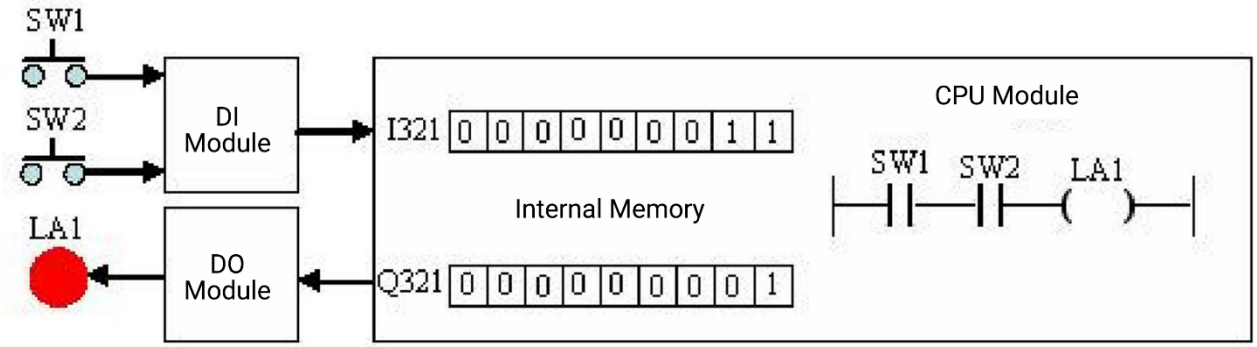
Data is the operand of control and operation in industrial systems and is stored in the internal memory of the CPU module.

10.1.1 Data Meaning

Programmable controllers are used to control production processes or equipment. Their control requirements shall be realized by means of a program. The fundamental element of a program is an instruction, which consists of operators and operands. The operands are divided into two categories. One is the actual data from the I/O module, such as the ON or OFF of the travel switch, the temperature level and the flow rate, etc.; And the other is derived from the operation results or intermediate results of the internal function modules and programs.

The external representation of data is the variable name. For example, the variable name of a travel switch is SW1. When the travel switch is turned on, SW1 = ON (or 1 or TRUE); And when it is turned off, SW1 = OFF (or 0 or FALSE). The internal storage of data takes the form of a change in the state of a bit, byte or word in the memory of the CPU module.

For example, when the travel switch is turned on, the corresponding memory bit is 1; And when it is turned off, the corresponding memory bit is 0. When the travel switches SW1 and SW2 are turned on, their status is sent to the memory of the CPU module through the DI module. For example, bit 0 = 1 and bit 1 = 1 of the I321 byte in the input area; Then it is operated through the logic ladder diagram (LD), whose result LA1 is stored in the memory of the CPU module. For example, bit 0 = 1 of the Q321 byte in the output area; Finally, it is output through the DO module, which makes the light LA1 light up.



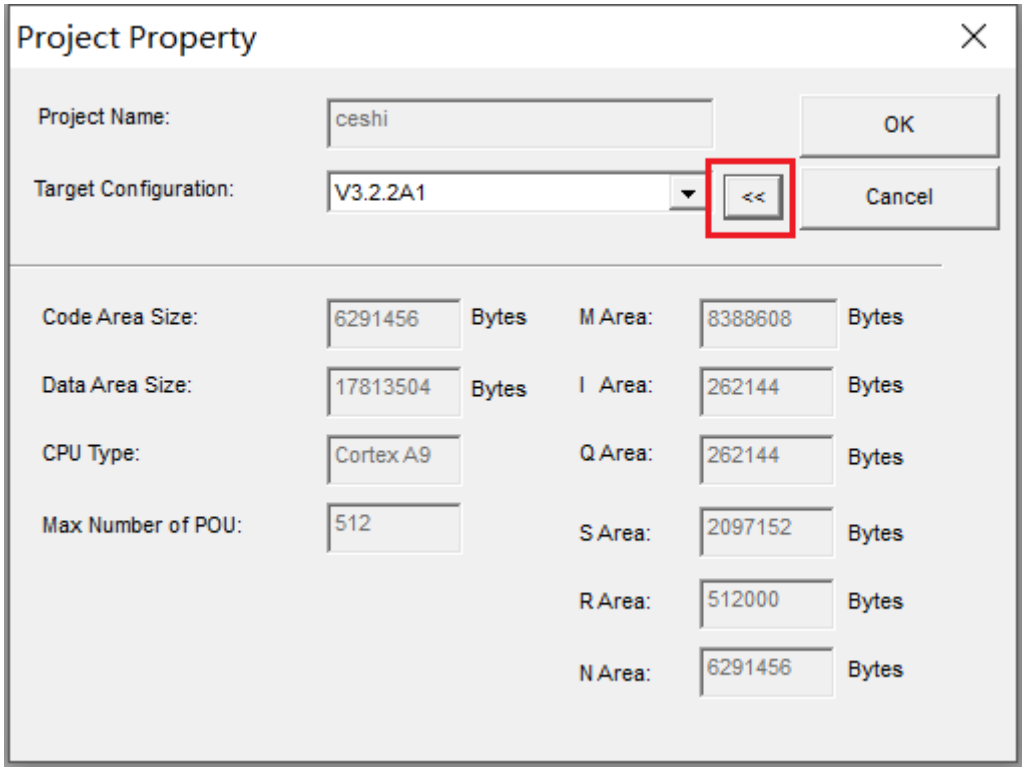
10.1.2 Data Storage Area

Data is stored in the memory of the PLC's CPU module, which is divided into a program area, which is used to store the program, and a data area, which is used to store the data.

The data area in the memory is used to store the data, which is divided into:

- Input area (I)
- Output area (Q)
- Intermediate area (M)
- Special register area (S)
- Random area (N)
- Protection area (R)

Right-click the root node of Project Management. Select the Properties command. In the pop-up Project Properties dialog box, click the area circled in red to view the size limitations of the various areas that can be used in the project. Let's take the LX project as an example for illustration.



The main functions of each area are as follows:

(1) Input area (I)

At the beginning of each scan cycle, the CPU samples the input points and stores the sample values in the input area of the internal memory. The input area can be accessed by bit, byte, word, or double word.

(2) Output area (Q)

At the end of each scan cycle, the CPU transfers data from the output area of the internal memory to the physical output point. The output area can be accessed by bit (BIT), byte (BYTE), word (WORD), or double word (DWORD).

(3) Intermediate area (M)

The intermediate area is used to store intermediate results, operating status, or other control information of the program. The intermediate area can be accessed by bit (BIT), byte (BYTE), word (WORD), or double word (DWORD).

Data area with LX power failure protection

Power Failure Protection Area	LX-CU500
Area M	%MB0~%MB4095

(4) Random area (N)

The random area is used to store user-defined variables with unspecified addresses.

The N storage area also belongs to the intermediate register area of the PLC, which is used to store and manage the data and status generated by the intermediate process. Unlike the M storage area, the N storage area can only be accessed and called through variables.

The variable addresses in the N storage area are automatically assigned by the system and cannot be specified by users. The variable data types in the N area are not only bit, byte, word, and double word, but also REAL, TIME, INT and many other data types. Also, in addition to data variables, defined functional block variables are stored in the N storage area.

The N storage area is readable and writable, and can be entered and forced. However, the data in the N storage area cannot be power-failure protected.

(5) Protection area (R)

The Protection Area stores the power-failure protection variables. If the data or variables are defined as hold type (the power-failure protection value is TRUE), the system automatically saves the data when the CPU module is power-failed, and then automatically restores the data when the power is restored.

This area is called in the same way as the N area. That means it is also accessed through variables, and cannot be specified with an address.

The variables in R storage area are readable and writable and can be entered and forced.

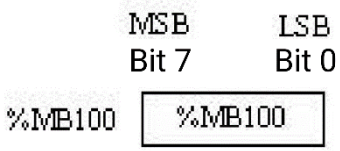
When a variable is defined, if the hold function is not selected, the variable is stored in the N area; And if the hold function is selected, the variable is stored in the R area with a power failure protection function.

10.1.3 Data Storage Format

The data storage format is divided into 4 types: byte, word, double word, and bit. Let's take the intermediate area (M) as an example to describe the data storage format.

(1) Byte data

The storage format of byte data is shown in the figure below.



Where MSB indicates the highest bit (bit 7) of the data, and LSB indicates the lowest bit (bit 0) of the data.

% indicates the address, M indicates the memory identifier, and B indicates the byte data, which occupies 8 bits.

(2) Word data

The storage format of word data is shown in the figure below.

	MSB Bit 15	LSB Bit 0
%MW100	%MB101	%MB100
%MW102	%MB103	%MB102

Where MSB indicates the highest bit (bit 15) of the data, and LSB indicates the lowest bit (bit 0) of the data.

% indicates the address, M indicates the memory identifier, and W indicates the word data, which occupies 16 bits.

(3) Dword data

The storage format of dword data is shown in the figure below.

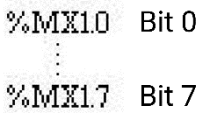
	MSB Bit 31			LSB Bit 0
%MD100	%MB103	%MB102	%MB101	%MB100
%MD104	%MB107	%MB106	%MB105	%MB104

Where MSB indicates the highest bit (bit 31) of the data, and LSB indicates the lowest bit (bit 0) of the data.

% indicates the address, M indicates the memory identifier, and D indicates the double-word data, which occupies 32 bits.

(4) Bit data

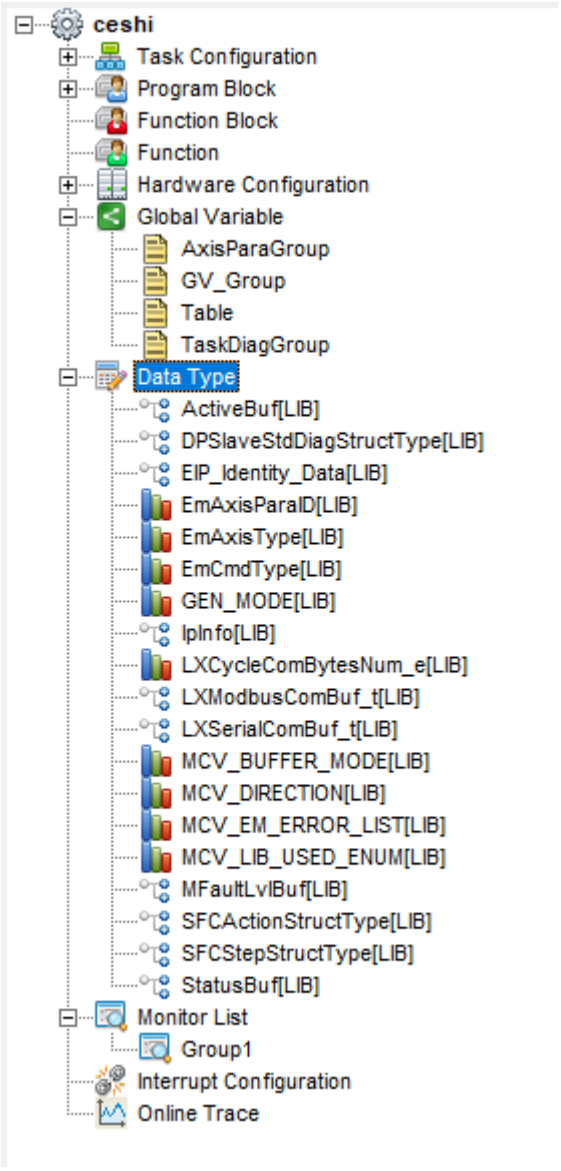
The storage format of bit data is shown in the figure below.



% indicates the address, M indicates the memory identifier, and X indicates a single bit, which occupies 1 bit.

10.2 Data Type

Under the Data Types node, the system-defined data types and custom data types are displayed. The data types are managed in a tree diagram, and custom data types can be added.



10.2.1 Standard Data Types

The software supports the standard data types shown in the table below, in addition to real, time, array, pointer, long real, and string.

Description Table of Data Types

Type	Type Name	Lower Limit of Data	Upper Limit of Data	Number of Storage Bits
BOOL	Bool	0	1	1 bit
BYTE	Byte	0	255	8 bit
WORD	Word	0	65,535	16 bit
DWORD	Double word	0	4,294,967,295	32 bit
SINT	Short integer	- 128	127	8 bit
USINT	Unsigned short integer	0	255	8 bit
INT	Integer	-32,768	32,767	16 bit

UINT	Unsigned integer	0	65,535	16 bit
DINT	32-bit integer	-2, 147,483,648	2, 147,483,647	32 bit
UDINT	32-bit unsigned integer	0	4294967295	32 bit
REAL	Real	-3.402823466E+38	- 1. 175494351E-38	32 bit
		1. 175494351E-38	3.402823466E+38	
LREAL	Long real	- 1.7976931348623158e+308	-2.2250738585072014e-308	64 bit
		2.2250738585072014e-308	1.7976931348623158 e+308	

10.2.1.1 Bool (BOOL) data

BOOL data is a logical quantity, occupying 1 bit of memory, with status 0 or 1, and TRUE or FALSE.

10.2.1.2 Integer (INT) data

Integer data can be divided into BYTE, WORD, DWORD, SINT, USINT, INT, UINT, DINT, and UDINT.

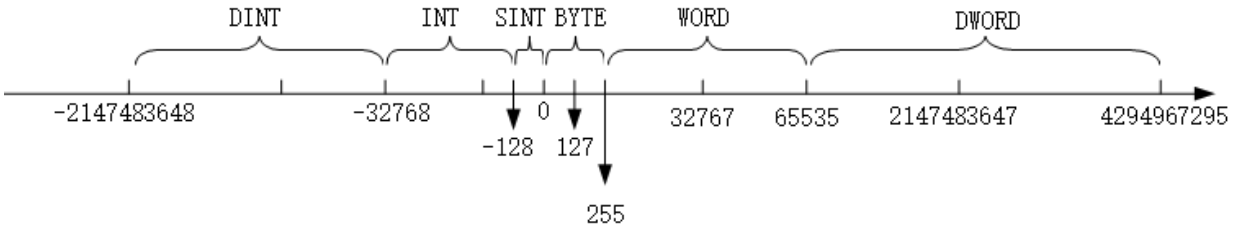
Integer data is characterized by the absence of decimal places.

- Rule for identifying integer constants

  The sequence in which integer constants are recognized in AutoThink is as follows:

  BYTE (0 - 255) -> SINT (-128 - 127) -> USINT (0 - 255) -> WORD (0 - 65535) -> INT (-32768 - 32767) -> UINT (0 - 65535) -> DWORD (0 - 4294967295) -> DINT (-2147483648 - 2147483647) -> UDINT (0 - 4294967295)

  That is, positive numbers are preferentially recognized as unsigned data, as illustrated more graphically in the figure below.



- Type priority

  The type priority of integer data in AutoThink is as follows, from lowest to highest:

  SINT (4) -> BYTE (5) -> USINT (6) -> INT (7) - > WORD (8) -> UINT (9) -> DINT (11) -> DWORD (12) -> UDINT (13)

- Rules for type escalation

  The information may be lost when a type with a larger range is converted to one with a smaller range.

  It is not recommended to use signed and unsigned numbers at the same time for operations, which cannot achieve the expected results in some cases.

- 

- When converting from a larger type to a smaller type, information may be lost.
- It is not recommended to perform operations using both signed and unsigned numbers simultaneously, as this can sometimes lead to unexpected results.

- 

- Users shall note the following points during configuration:
- Positive integer constants in AutoThink are recognized as unsigned data. The unsigned type has a higher priority.
- Do not use signed and unsigned numbers together in a basic operation unless the result is as expected.
- For basic instructions with multiple inputs, the inputs shall not all be constants unless the result is as expected.
- For basic operations where the result may be negative, it is preferred to use an INT variable if a 16-bit integer is used as input, or a DINT variable if a 32-bit integer is used as input.

10.2.1.3 Real (REAL) data

Real data is characterized by the presence of decimal places. That means it can be expressed in both integer and decimal form, as well as in exponential form. Data expressed as REAL.

Example: R1:REAL := 23.34, R2:REAL := 1.64e+009.

- 

- When there is real data involved in the operation, the values 1#INF and 1#QNAN0 sometimes appear with the following meanings:
- 1#INF indicates an infinite number, which is divided into positive and negative infinity. It is obtained by dividing a finite number that is not 0 by 0, or exceeds the upper and lower limits of the real data. No error is reported for real division by zero. .
- 1#QNAN0 indicates not a number. It is caused by some meaningless operations.

10.2.1.4 Time (TIME) data

Time data is used to define time (TIME), time of day (TOD), date (DATE), and date and time (DT).

No.	Variable Name	Address	Variable Description	Variable Type	Initial Value	Power Fail Safeguard
0001	TimeVar1		time	TIME	T#0MS	FALSE
0002	TimeVar2		date	DATE	D#1970-01-01	FALSE
0003	TimeVar3		Date and time	DT	DT#1970-01-01-00:00:00	FALSE
0004	TimeVar4		time moment	TOD	TOD#00:00:00	FALSE

- Time (TIME)

  The identifier for the time data is TIME, which is used to define the operation time. There are 4 formats for defining time: TIME#, time#, T#, and t#, with the format character # followed by the time data, in order of hour, minute, second, and millisecond. As shown in TimeVar1 in the figure.

  Example:

  Correct definition:

  TIME1 := T#14ms;

  TIME1 := T#100S12ms; (*The highest bit is allowed to exceed its data upper limit*)

  TIME1 := t#13d12h34m15s;

  Incorrect definition:

  TIME1 := t#5m68s; (*The lower bit exceeds the limit*)

  TIME1 := 15ms; (*Lack of prompt T#*)

  TIME1 := t#4ms13d; (*Entered in the wrong sequence*)

- The highest bit of the defined time is allowed to exceed its upper limit, while the lower bit is not allowed to exceed its upper limit.

- Date (DATE)

  The identifier for the date is DATE, which is used to store the date. There are 4 formats for defining date: D#, d#, DATE#, and date#, with the format character # followed by the date data, in order of year, month, and day. As shown in TimeVar2 in the figure.

- Date and time (DT)

  The identifier for the date and time is DT, which is used to store the date and time. There are 4 formats for defining date and time: DT#, dt#, DATE_AND_TIME#, and date_and_time#, with the format character # followed by the date and time data, in order of year, month, day, hour, minute, and second. As shown in TimeVar3 in the figure.

- Time of day (TOD)

  The identifier for the time of day is TOD, which is used to store the time. There are 4 formats for defining time of day: TOD#, tod#, TIME_OF_DAY#, and time of day#, with the format character # followed by the time-of-day data, in order of hour, minute, and second. Where the second is a real number, that is, the second can have decimals. As shown in TimeVar4 in the figure.

-

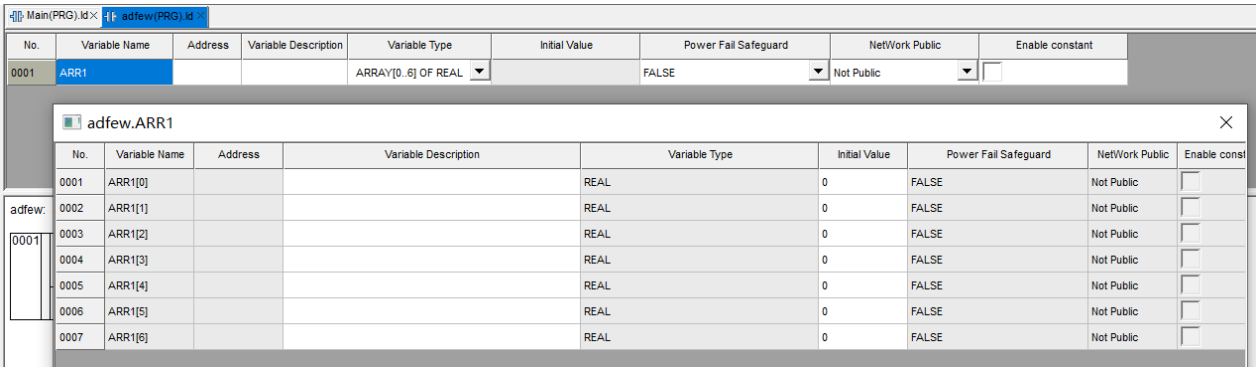
- The highest bit of the defined time is allowed to exceed its upper limit, while the lower bit is not allowed to exceed its upper limit.
- The second of time defined is a real number, that is, the second can have decimals.

10.2.1.5 Array (ARRAY)

Sometimes it is necessary to have an ordered and integrated combination of data elements with the same type for referencing. An array (ARRAY) serves this purpose.

We can define one- and two-dimensional arrays based on standard data types, and use these arrays to combine data with the same type in an orderly manner. Each element within the array has a fixed number (or index). For example, the numbers (or indexes) of the 4 elements within a 2 × 2 two-dimensional array are 11, 12, 21, and 22, respectively.

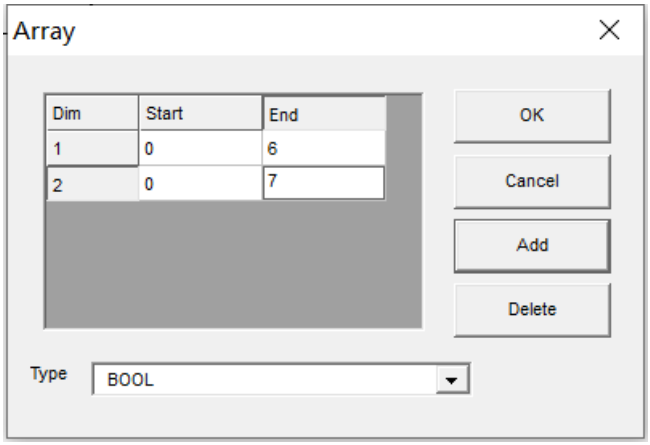
For example, to assign an initial value to a one-dimensional array ARR1. The ARR1 array is indexed from 0 - 6, and the variable type is REAL, as shown in the figure below.



To assign values to the above array elements in the ST language editor, with the format ARR1[1]:= 10;. Or to define another array variable ARR2 and to ensure that the array dimensions (indexes) and data types are exactly the same as those of ARR1. In this case, the format that can be used is ARR1:= ARR2;.

■ Define array

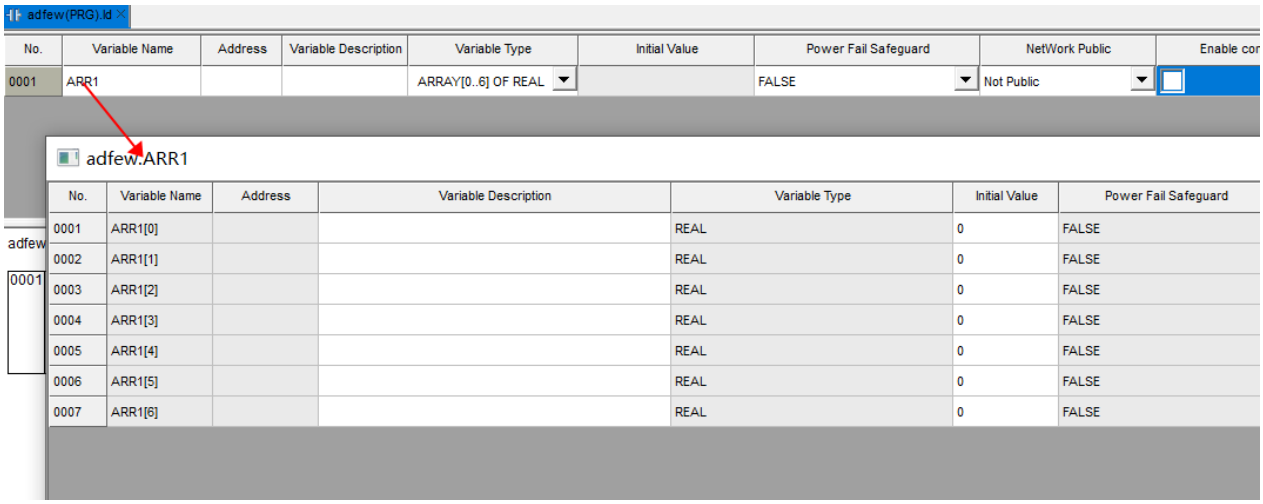
In the Variable Declaration dialog box, we can define one- and two-dimensional array variables. After adding the variable, in the variable row, select the variable type unit. Click the drop-down arrow . In the drop-down list of data types, select the ARRAY item. The Array definition dialog box is displayed, as shown in the figure below.



The Dim column indicates the number of defined array dimensions, and our software supports multi-dimensional arrays; The Start column indicates the starting index of the array; And the End column indicates the ending index. In the Type drop-down menu, users can select the type of the array element.

- The default array index starts from 0. To define an array, the Start index value shall be smaller than or equal to the End index value. Otherwise, the array cannot be defined.

Check the row where the array variable point is located in the variable table. Double-click the serial number or select the Detail command in the right-click menu. The definition window of each sub-item of the variable is displayed, including Variable Name, Variable Description, Variable Type, Initial Value, Power Failure Protection, and other properties. In this window, users can modify the details of each sub-item.



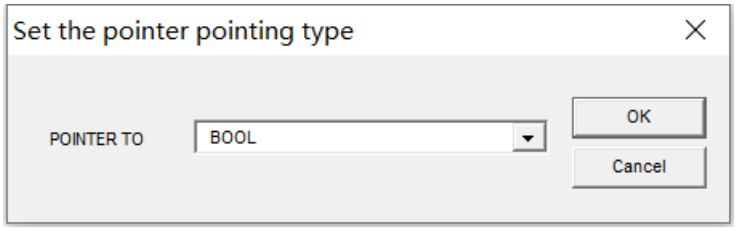
■ Reference array element

Array elements can be referenced in the program. For example, to reference the two-dimensional array element ArrayVar[1,2]:

It is Var1 := ArrayVar[1,2] (*Two-dimensional array element [1,2]*) in ST language

10.2.1.6 Pointer

A pointer is a variable used to indicate a computer memory address. When defining a variable, select the variable type as POINTER. The Set Pointer To Type dialog box is displayed. In the POINTER TO drop-down list, select the pointer to type.



Here, the pointer supports the following two operations:

- P := ADR(VAR1); (*P is a pointer variable, and VAR1 is an ordinary variable. Here, the address of VAR1 is assigned to variable P*)
- VAR1 := VAL(P); (*P is a pointer variable, and VAR1 is an ordinary variable. Here, the value of variable P is assigned to VAR1*)

Description:

- Values to odd addresses are not supported, such as p1:pointer to word, p1: = 12548637. The value to p1, that is, the VAL operation, may cause the controller to reset.
- The address of a variable or functional block is stored in a pointer while the program is running.
- A pointer can point to any data type or functional block, and can even be used for custom types. The address operator ADR functions to assign the address of a variable or functional block to a pointer.
- The value operator VAL functions to value the memory pointed to by the pointer.

- The pointer is counted in bytes. It can be counted as in C_Compiler by using the statement p = p + SIZEOF (the variable pointed to by p).

10.2.1.7 Long real (LREAL) data

The negative LREAL values range from -1.7976931348623158e+308 to -2.2250738585072014e-308, and the positive values range from 2.2250738585072014e-308 to 1.7976931348623158e+308.

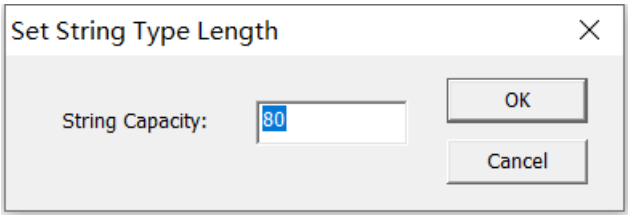
10.2.1.8 String (STRING) data

- String variables

  String variables can contain any string of characters. The length of the variable when declared determines how much storage space is reserved for the variable.

  When selecting the data type of the variable as STRING, users need to set the length of the string data,

as shown in the figure below.



The length range of a string variable is 1 - 1000, which means that users can enter 1 - 1000 characters, each of which occupies one byte (8 bit) of the address space. If it is not specified, the system defaults to a length of 80 characters.

When setting the initial value of a string variable, users need to add English single quotes at the beginning and end of the content, such as 'this is a string'.

- String constants

The length range of a string constant is 1 - 1000 characters, enclosed in single quotes. Users can also insert spaces and special characters, which are defined as follows:

Special Character Definitions

Input String	Meaning
$R	Enter
$L or $N	Line feed
$T	Horizontal tab
$’	Single quote
$$	$ symbol
$P	Form feed

10.2.2 Custom Data Types

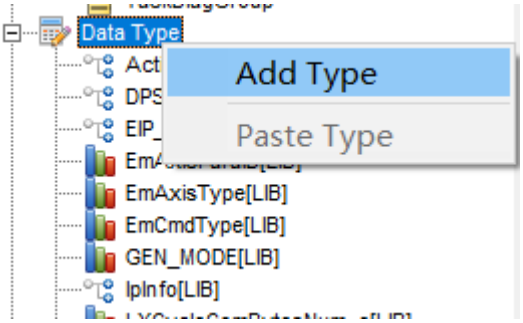
The custom data types supported by AutoThink software are divided into structure (STRUCT), enumeration (ENUM), and union (UNION).

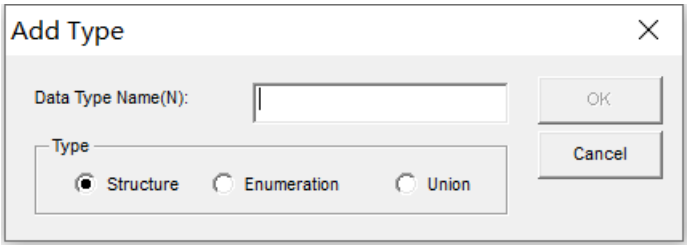
- Structure (STRUCT)

Sometimes it is necessary to have an integrated combination of data with the different types for referencing. A structure (STRUCT) serves this purpose. A structure consists of multiple members (or components). Each member has a member name and a type, which can be different or the same. To add a structure under the Data Type node, follow the steps below:

- Add type

Right-click the Data Type node. The tree structure is displayed as shown in the figure.

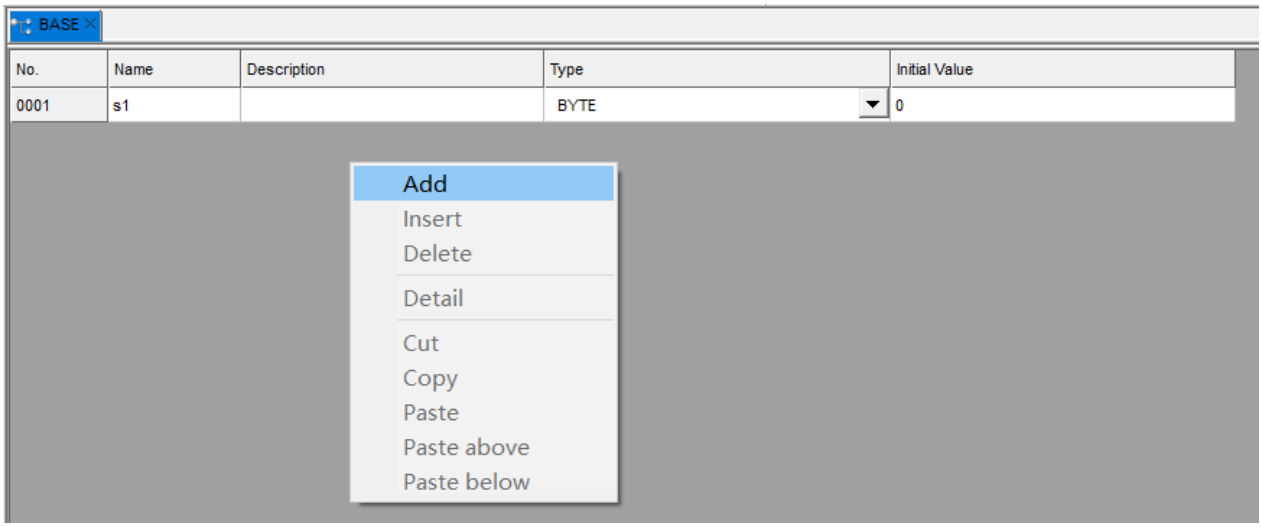




In the Data type name (N) box, enter the name of the structure. In the Type item, select the data type. Click OK to complete the addition of the structure.

■ Add member

Right-click in the structure window. Select the Add Member command to add a row in the defined data table. The default structure member name is s*. Users can modify the structure's Member Name, Member Description, Member Type, and Initial Value.



If the Member Type is pointer, array, string, enumeration, structure, or functional block, users can view the member variables in the detailed Menu.

The rules for member type configuration are as follows:

□ It shall not be configured directly or indirectly as its own structure type.

□ When a member type is configured as an array, the array element type shall not be configured as its own structure type.

□ After a structure is declared as an instance, the members of the structure can be called by

Instance name. Member name.

- Enumeration (ENUM)

  The method of adding an enumeration variable is consistent with that of adding a structural variable, except that the enumeration type is selected for the Type item.

  Multiple enumerations can be defined in a project. The member names of enumerations shall be unique. That is, no rename is allowed in the member names of different enumerations.

- The members of the enumeration can be referenced directly in the program without the need to declare them in the variable area.
- The range of enumeration values is -32768 - 32767, and the upper limit of the number of enumeration members is 65536.

- Union (UNION)

  Sometimes it is necessary to store several different types of variables into the same section of memory units. For example, an INT variable, a BYTE variable, and a DWORD variable can be stored in the memory unit starting at the same address, that is, from the same address 16#100.

- Copy data type

  The created data types can be copied as follows:

  - Right-click the enumeration or structure data type. Select the Copy Type command.
  - Right-click the Data Type node. Select the Paste Type command to copy the data type. The default name of the data type is NewType1.

10.2.2.1 Import data type

10.2.2.1.1 Introduction

The system provides an import function for user-defined structure types in the project. This meets the needs of users who export structure types, edit them in third-party software, and then import them into the project.

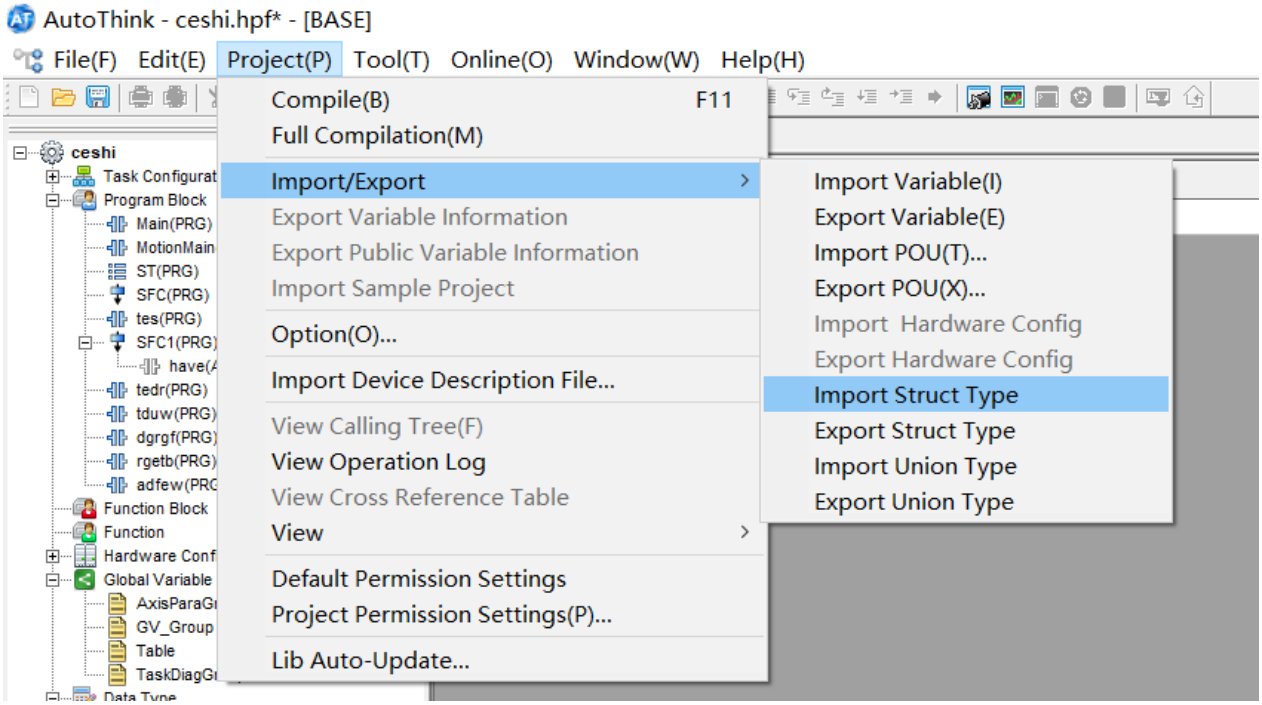
The import format is a .csv file, and the one-time import limit is 100.

10.2.2.1.2 Steps

To import structure types, follow the steps below:

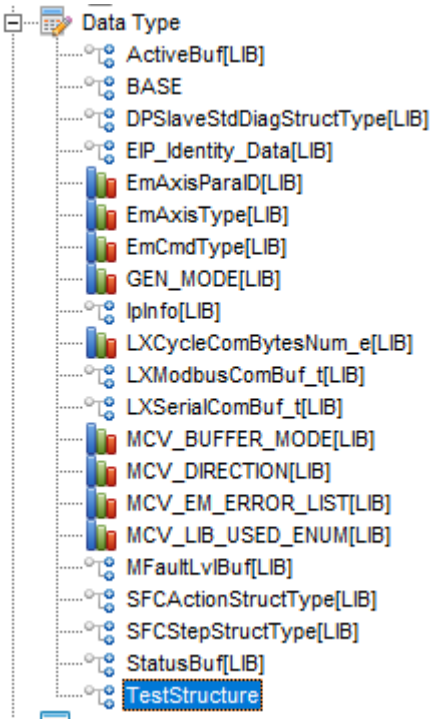
(1) Click the Project menu.

(2) Select the Import Structure Type command.



(3) The File window is displayed. The file type is limited to .csv. Users can select the file for importing.

(4) The structure types imported successfully are displayed in the Data Type node of the Project Management Tree.



10.2.2.2 Export data type

10.2.2.2.1 Introduction

The system provides an export function for user-defined structure types in the project. This meets the needs of users who export structure types from the project in .csv file format.

The exported content is the elements of the first layer of the structure. For the complex type, no sub-element export is performed. For the initial value format of the complex type, refer to the initial value format of the variable export.

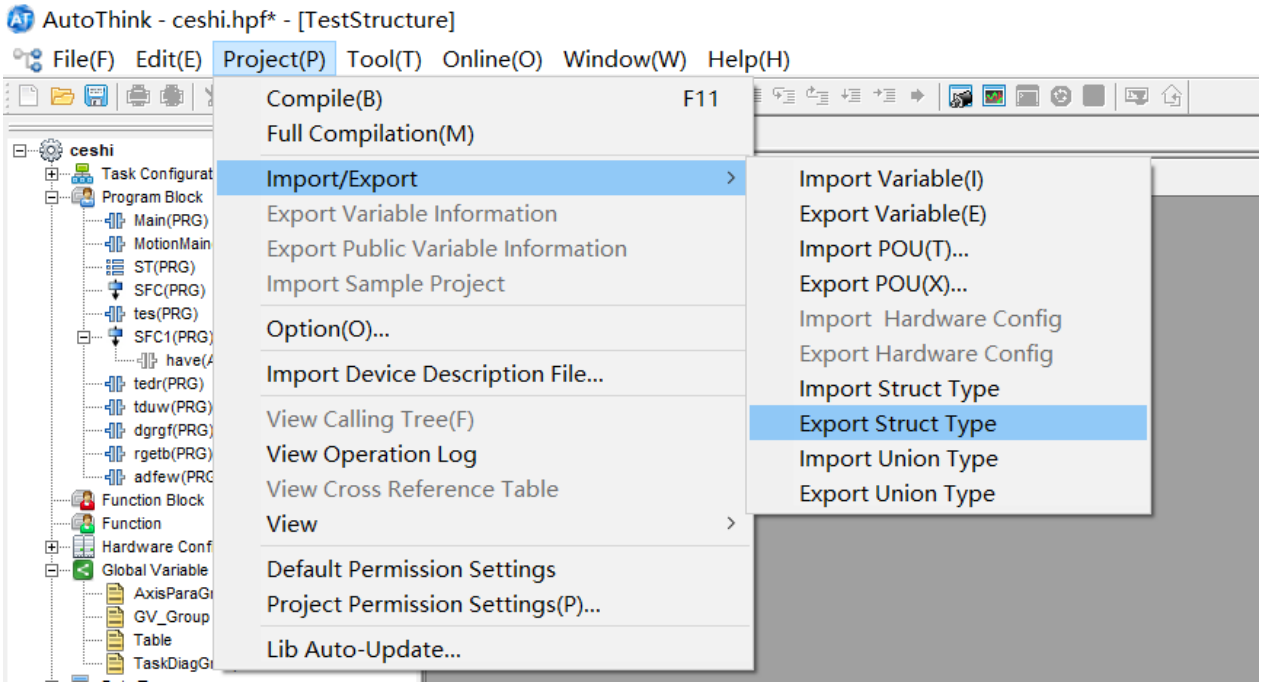
When structures are nested, the preconditioned .csv files shall be generated for all exported structures, which describe the dependencies between the exported structures. These files contain only the current exported type, and the unexported types are not shown in these files.

10.2.2.2.2 Steps

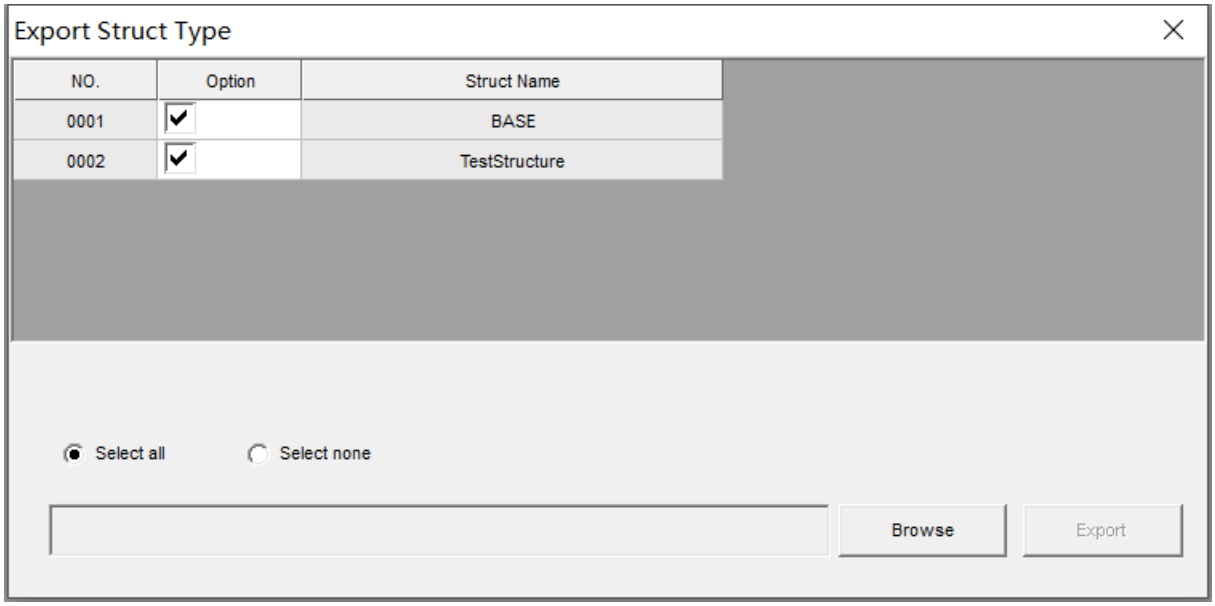
To export structure types, follow the steps below:

(1) Click the Project menu.

(2) Select the Export Structure Type command.



(3) The Structure Export dialog box is displayed. Check the structures that need to be exported. Click Browse. The Select Path dialog box is displayed. After the folder is selected, the Export button is enabled. Click Export to generate the structure export file under the directory selected by the user. Each structure corresponds to a .csv file, with the name of the structure as the name of the file.



10.3 Variables

10.3.1 Variable Types

1. According to the variable data types, variable types are divided into standard types and custom types.

   - Standard types

     Include bool (BOOL), byte (BYTE), word (WORD), double word (DWORD), short integer (SINT), unsigned short integer (USINT), integer (INT), unsigned integer (UINT), 32-bit integer (DINT) and unsigned 32-bit integer (UDINT); also real (REAL), long real (LREAL), string (STRING), array (ARRAY) and time. Where the time includes time (TIME), time of day (TOD), date (DATE), date and time (DT).

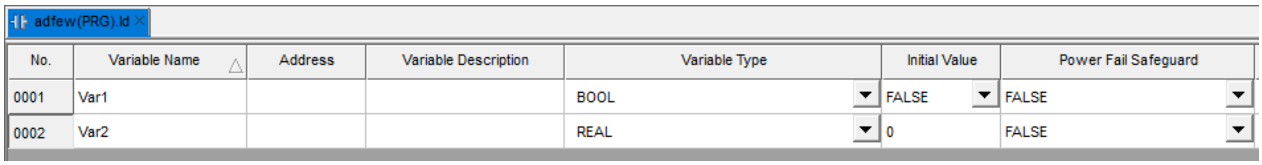
   - Custom types

     Divided into structure (STRUCT) and enumeration (ENUM).

2. Divided into two types in terms of structural form, simple variable and functional block instance.

   - Simple variable

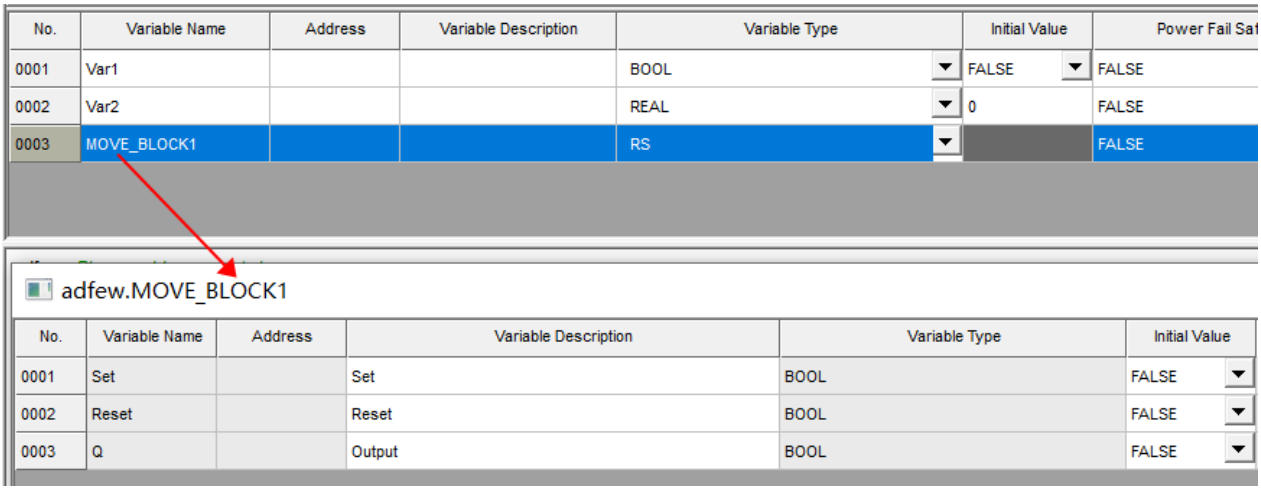
     A simple variable refers to a single variable quantity that can be assigned a well-defined value. A simple variable represents only one meaning, such as BOOL and REAL variables.



   - Functional block instance

A functional block instance consists of a specific set of variables, which depends on the functional block type of the functional block instance. Therefore, the functional block instance is conceptually close to a data structure.

The format of a functional block instance is shown in the figure below. Double-click the information row of the functional block instance definition to expand the window of each internal item window of the functional block.



- When adding the instance name of a functional block type provided by the system, make sure that the library where the functional block of this type is located has been added to the current project. Otherwise, the name of the functional block of the corresponding type cannot be found in the Variable Type drop-down list. In particular, since the functional block type is not supported in the input variable area and local variable area of the function, the functional block type name is not displayed in its corresponding drop-down list.

3. By effective variable scope (scope of use)
   - Global variable (VAR_GLOBAL): It is the most widely used variable. A global variable can be used in several places in a project at the same time. However, global variables with the same name are not allowed in a project.
   - Local variable: Compared to the global variable, it has a limited scope of use and is only valid for use in the POU to which it belongs; Local variables can share the same name in a project.
4. Local variables can be divided into input variable, output variable, input/output variable and local variable by the effective scope of their use.
   - Input variable (Input_Variable): The variable is input from the outside. This functional block (or function) can only read the variable.
   - Output variable (Output_Variable): The variable is output to the outside. This functional block can read/write the variable; And other functional blocks can only read it.
   - Input/output variable (Input/Output_Variable): The variable has the characteristics of both an input variable and an output variable. Other functional blocks and this functional block can read/write this variable.

- Local variable (Local_Variable): The variable, for a specific functional block (or function), is a temporary variable required by the block to implement a logical operation; This variable is not connected to the outside and is valid only in the block to which it belongs. It is called a parameter (VAR_TEMP) in the functional block viewing information.

-  If local variables need to be referenced globally, the format POU name.local variable name shall be used.

10.3.2 Variable Declaration Specification

Variables shall be declared in accordance with the syntax specification of AutoThink software. Otherwise, it may cause a compilation error. There are specific specifications on the defined variable names, variable types, and declarations.

1. Variable Name

   - The name of a defined variable can only consist of letters, numbers, and underscores. The maximum length of the name shall not exceed 32 bytes.
   - The naming of variable names follows the following rules:
   - The variable name shall start with a letter or an underscore, not with a number.
   - The variable name is identified based on the underscore. For example, AB_CD and ABC_D are considered to be two different variable names.
   - The variable name is not case sensitive. For example, VAR1, Var1 and var1 represent the same variable.
   - The variable name shall not be null and shall not contain spaces. For example, AB CD is a wrong variable name.
   - The variable name shall not contain special characters such as the strikethrough - and the plus sign +. For example, AB - CD and AD + CD are wrong variable names.
   - The variable name shall not share the same name as a type name (including custom types), POU name, enumeration name, task name, or type transition function name, and shall not start with AT_.
   - The variable name shall not be the same as the keyword.

2. Variable address

   Specifying a direct address (variable address) for a variable refers to connecting a variable to a defined internal memory address. That is, it is a mapping of the addresses of the input area (I), output area (Q), and intermediate area (M).

   Rules for the use of direct address:

   - The I-area, Q-area, and M-area variables support user-defined direct addresses without the need to associate them with hardware channel addresses;
   - Only the last 128 bytes of Area I are readable/writable;

- When a direct address is defined, the principle of byte alignment is used. That is, the entered direct address (first address) shall be divisible by both the data length of the variable type (counted in bytes) and the length of the addressed byte. For example, if a BYTE variable is defined with a data length of 1 byte, addressed as a double word (4 bytes), the direct address shall be divisible by both 1 and 4, such as %ID8, and %MD12.;
- The addressing modes X, B, W, and D determine only the first address alignment; And the actual length of data addressed is determined by the data type. For example, if a REAL variable with a data length of 4 bytes and a direct address of %IW12, the actual read data is 4 bytes of data starting from the first address of %IW12;
- The use of direct addresses is not supported for TIME and ARRAY variables whose elements are of complex types (enumeration, pointer, string, structure, array, functional block, and time).
- For WORD and BYTE channel data or Modbus communication data, the corresponding data can be read by bit or byte via direct address. For example, channel %IW8 reads 16-bit data by accessing direct addresses %IX8.0 - %IX8.7 and %IX9.0 - %IX9.7, respectively.

Example: For the variable -OUT_SV2 defined in the output area (Q), if the variable address is set to: %QX2.1; (*Q output area, X bit addressing, bit variable*), the meaning of the input format character for variable access is as follows:

Meaning		**Identification**
Address Identifier		%
Storage Area Location		Input area-I, output area-Q, intermediate area-M
Addressing Modes		X - bit addressing (1 bit)
		B - byte addressing (1 byte)
		W - word Addressing (2 bytes)
		D - double word addressing (4 bytes)
First Address	Word number	Word number of the storage area
	Bit number	Bit number of the storage area. Depending on the data addressing mode, the bit number can be omitted. If there is a bit number, the separator "." shall be added before the bit number. The bit number is 0 - 7

It is recommended to associate a variable name when using an M-area address. Otherwise, it is not processed during compilation, and the address data is not synchronized with the data. For example, if the address of the Area M in the variable definition area has been used to %MB100, and the user directly uses the address %MB101 when configuring without associating a variable name with it, the address used in the M area is displayed as %MB100 during compilation, and at the same time, the data in %MB101 is not synchronized redundantly.

10.3.3 Variable Declaration Methods

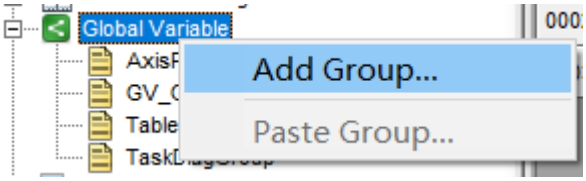
10.3.3.1 Declare global variables

10.3.3.1.1 Introduction

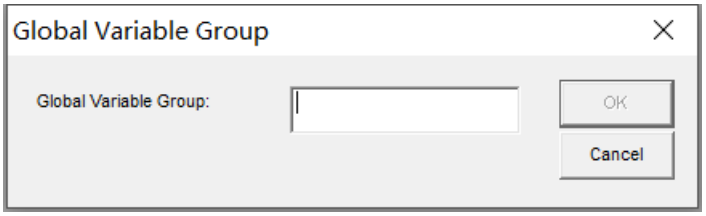
A global variable is valid in the whole project and can be referenced by any of the programs.

10.3.3.1.2 Steps

(1) Project Management Tree: Right-click the Global Variables node, and click Add Group.



(2) In the pop-up New Global Variable Group dialog box, enter the name of the variable group.



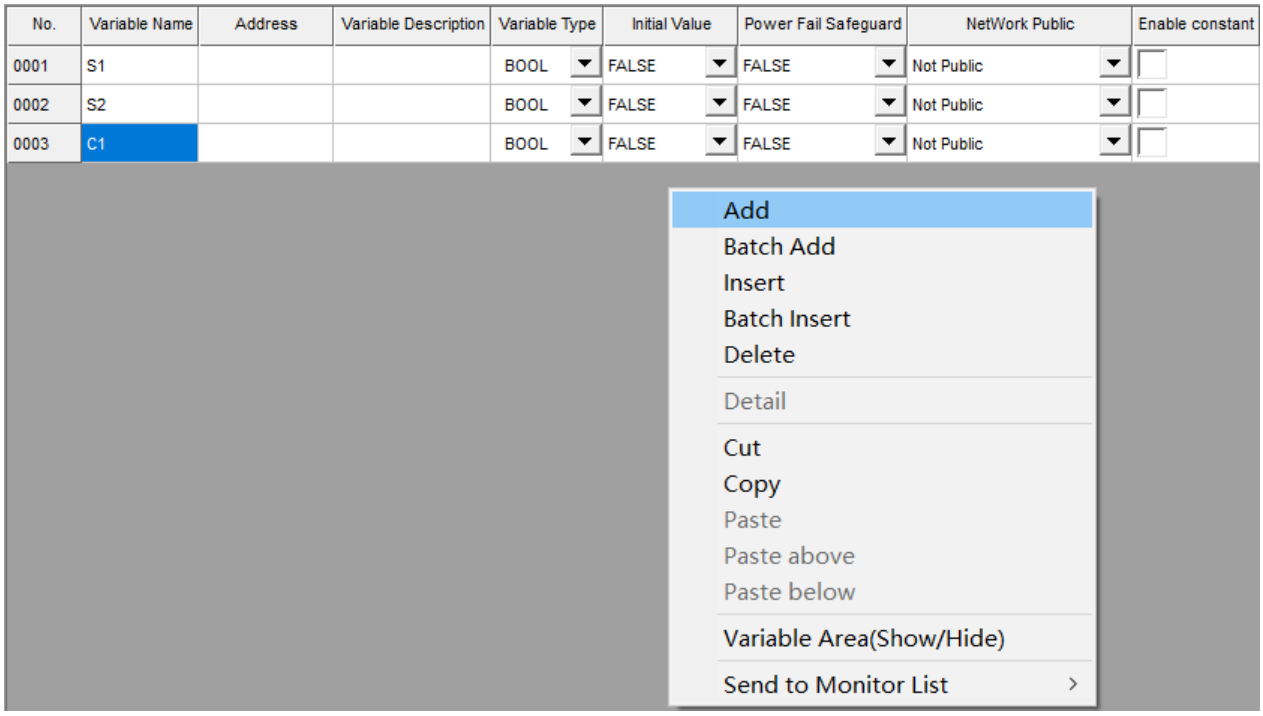
The variable group name can be letters, Chinese characters, numbers, or the underscore _. However, it shall start with letters, Chinese characters, or the underscore, and follow the following principles:

- The variable group name shall not be the same as the POU name, task name, keyword, instruction library name, or functional block name.
- The variable group name shall not be null. When it is null, the OK button is grayed out and unavailable.
- The length of the variable group name shall not exceed 32 bytes, and the part that exceeds the range in invalid.
- The naming rules for variables are the same as those for variable groups and will not be repeated.

(3) Declare variables manually

Enter the variable group name in accordance with the naming rules, such as var_1. Click OK button to complete the addition. Under the common variable node, the newly added variable group name branch is generated. Double-click the branch to open the editing window of the variable group in the workspace. In this window, right-click the corresponding menu command to add or delete

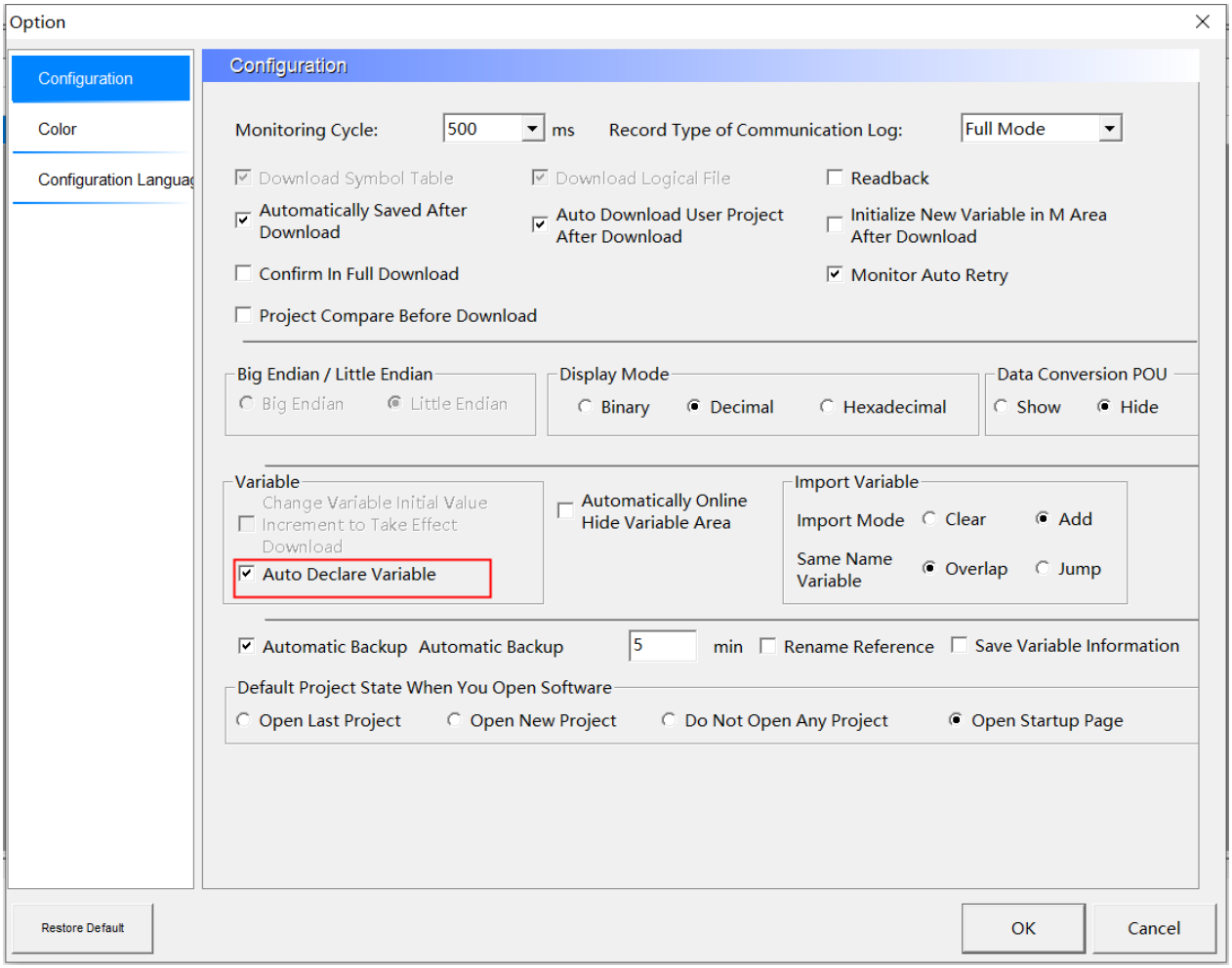
variables. The defined variables can be copied and pasted.



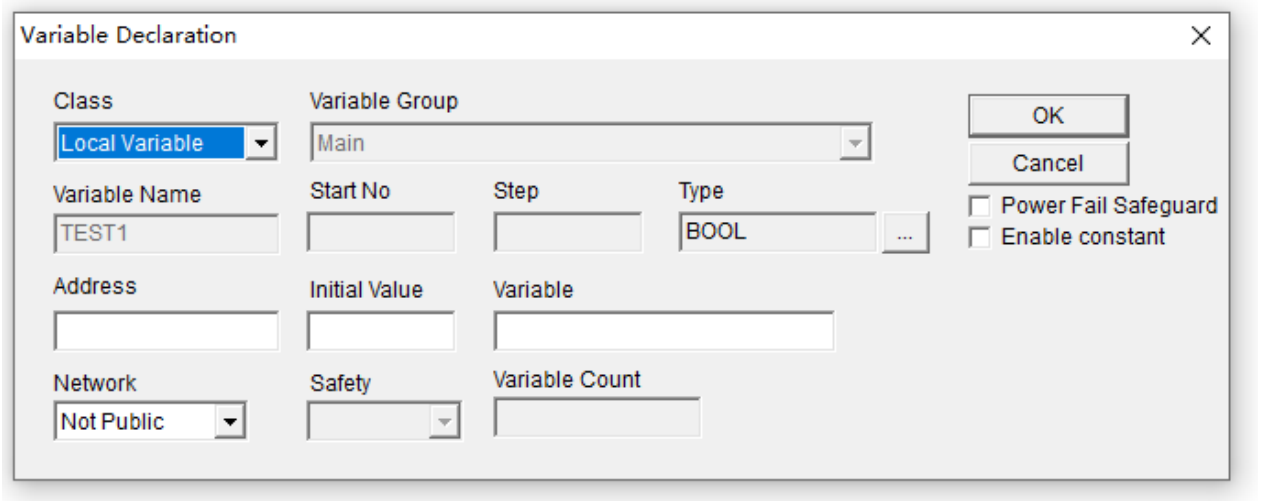
By clicking the variable properties item in the first row, sort the variables by their properties. If the number is not ascending, the Insert and Delete commands are not available in the right-click menu, Edit menu, toolbar, and shortcut keys for variables.

(4) Declare variables automatically

Uses reference variables in the programming window, and the system automatically prompts them to declare undefined variables, thereby adding new variables. Note that it is important here to make sure that the settings for automatic variable declaration are enabled. To enable it, select the menu [Project] - [Options]. In the Settings window, check Automatic variable declaration, as shown in the figure.

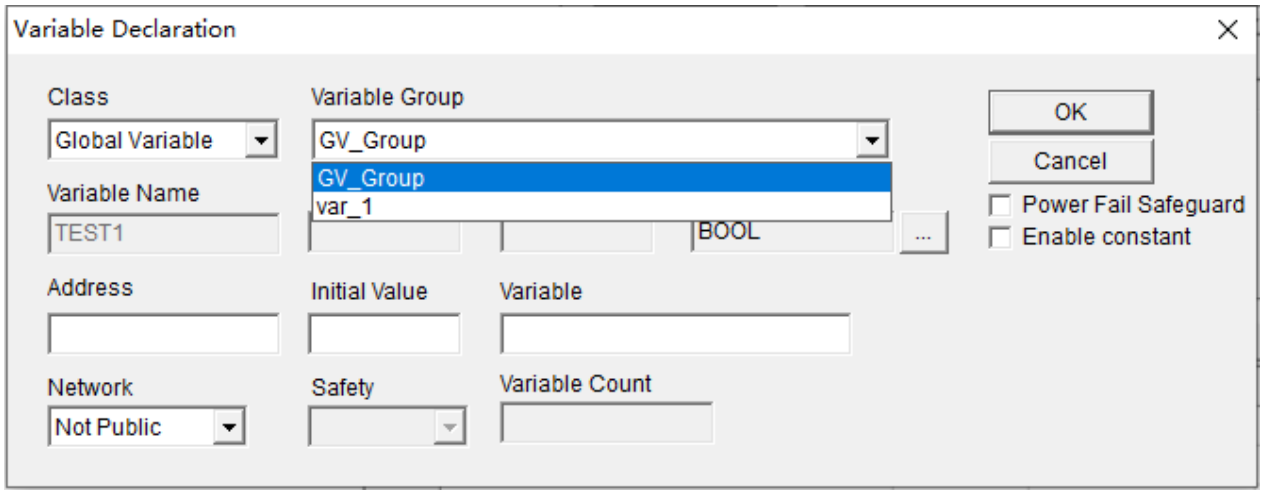


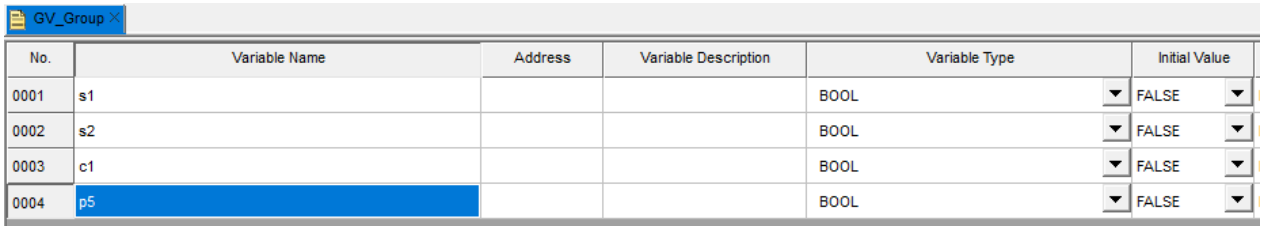
In the program area of the programming interface, if users write an undefined variable and press Enter, the Variable Declaration window is automatically displayed and the variable can be defined.



In the Category drop-down menu, users can select whether to define the variable as a local or global variable. If users select to define the variable as a global variable, the name of the variable

group to which the variable is saved can be selected in the Variable group list drop-down menu, as shown in the figure below. Enter the Direct address and Variable description. Set the power failure protection property. Click OK to add the variable to the variable group.





- When defining the variable category in the project, make sure to select the Global variable category. The global variables can only be stored in the corresponding variable group under the Global Variable node.

10.3.3.2 Declare local variables

10.3.3.2.1 Introduction

Currently, variables defined in the variable definition area of a POU program, as well as in functional block (FB) and function (Function) POUs are local variables, which are only valid in the current POU and cannot be directly referenced by other programs. However, they can be globally referenced in the format of POU program name.variable name.

10.3.3.2.2 Requirement

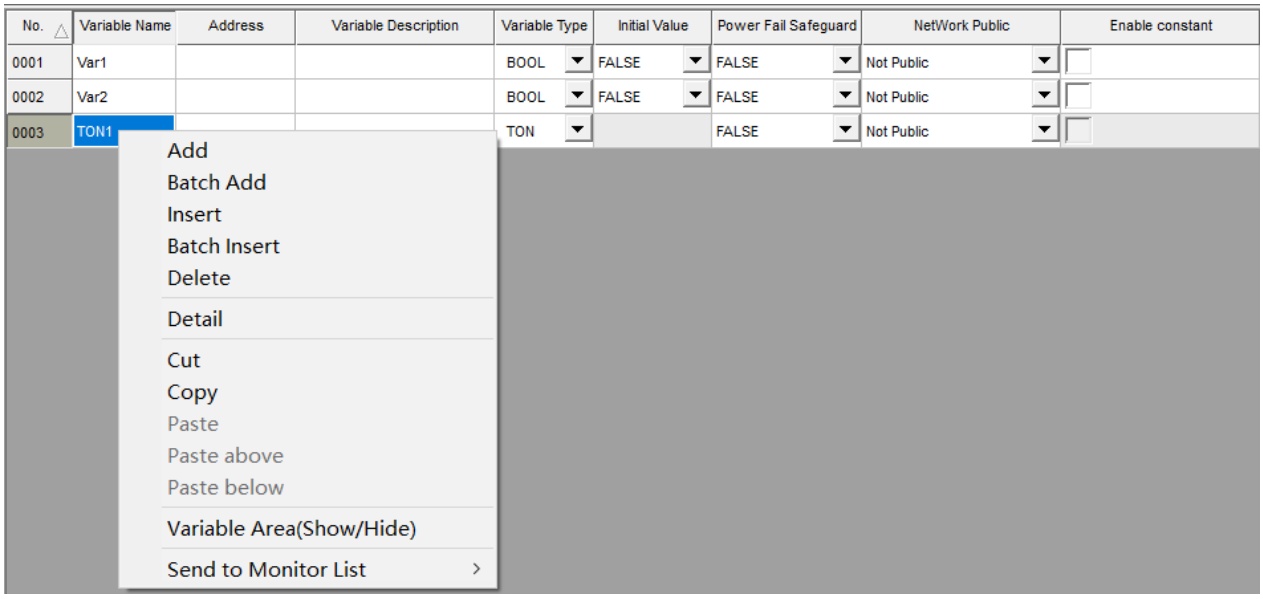
Local variables can only be declared in the corresponding variable area of the respective program.

10.3.3.2.3 Steps

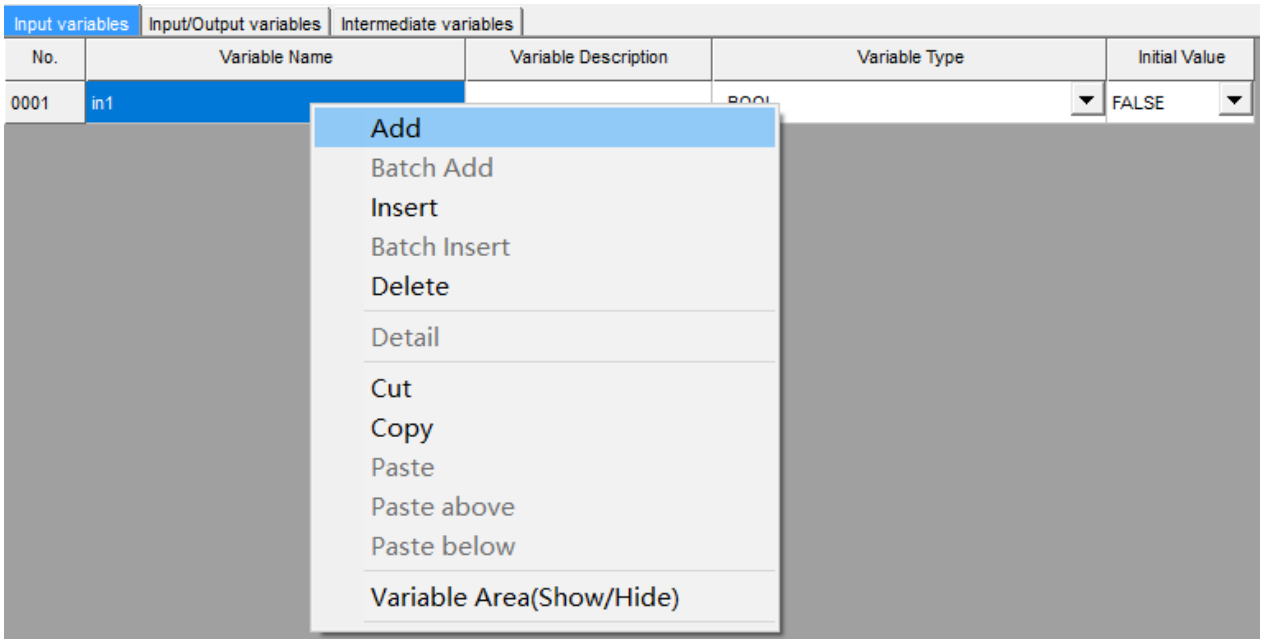
(1) Declare variables manually

When editing function and functional block types of POUs, declare variables or define variables through the variable area of the programming interface.

Right-click the corresponding menu command to add or delete variables. Every time a variable is added, a record is added to the variable table. The number is incremented according to the sequence of addition. Users can modify the properties of Variable Name, Variable Description, Variable Type, Initial Value, Power Failure Protection, Network Disclosure, and Constant or Not.Batch variable declaration allows modifying the following properties: variable name prefix, starting number, step, variable type, direct address, initial value, variable description, power failure protection, network publication, and whether constant.

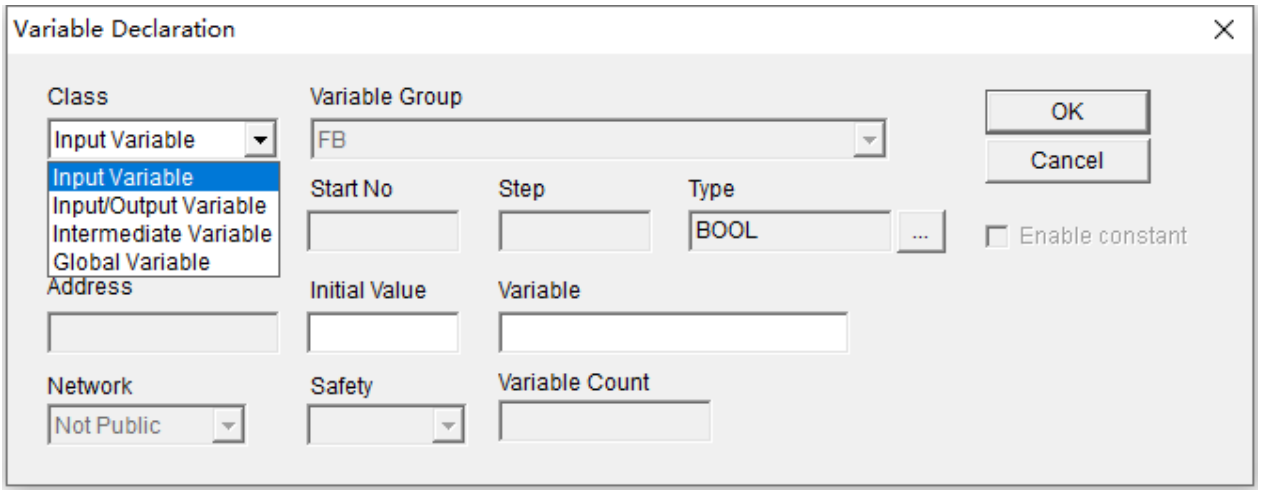


When editing a function POU, determine each variable to be defined to which of the three tabs Input Variables, Input/Output Variables, and Intermediate Variables belongs to. Click the name of the tab to enter the definition area of each type of variable. Variables are operated in the same way as in the functional blocks.



(2) Declare variables automatically

See Declare Variables Automatically in the Declare Global Variables for the declaration method. Only the category options are different, as shown in the figure below.



Global and local variables can be copied and pasted into each other. For different properties of a variable, the system automatically defaults to the value of the variable property in the pasted area.

10.3.3.3 Modify variable name

10.3.3.3.1 Introduction

After a variable is created, you can modify the variable name. If the variable has been used by the program, you can choose whether to update the program where the variable is used.

- Rules:
  - If the global variable name is modified, all programs where the variables with the same name are used in the project are retrieved.
  - If the local variable name is modified, all programs where the variable is used in the current POU are retrieved.
  - If the channel variable name is modified, the program where the variable is used cannot be updated synchronously.
  - If the new modified variable name already exists, no modification is made.

10.3.3.3.2 Requirement

The Rename variable references has been checked in the project Options menu.

10.3.3.3.3 Steps

To modify the variable name, follow the steps below:

(1) In the variable definition area, double-click the variable name. The variable name is displayed as an edit box.

(2) Enter a new variable name.

(3) Press Enter, or click other areas of the edit box. Then the Rename dialog box is displayed, showing all the programs where the variables with the same name are used.

(4) Check the program where the modification needs to be synchronized.

(5) Click OK to modify the variable name synchronously in the variable definition area and in the checked program where the variable is used; Click Cancel to not rename the program where the variable is used.

10.3.4 Modify Variable Type

Variables are created as BOOL by default. You can modify the variable type as needed. The variable type can be selected via a drop-down box, or entered directly.

Array variables can have their indexes modified directly in the Variable Type box.

10.3.4.1 Requirement

The variable has been created

10.3.4.2 Steps

To modify the variable type, follow the steps below:

(1) Double-click the variable data type. A drop-down list box is displayed.

(2) Select the data type or enter it directly.

10.3.5 Global Variable

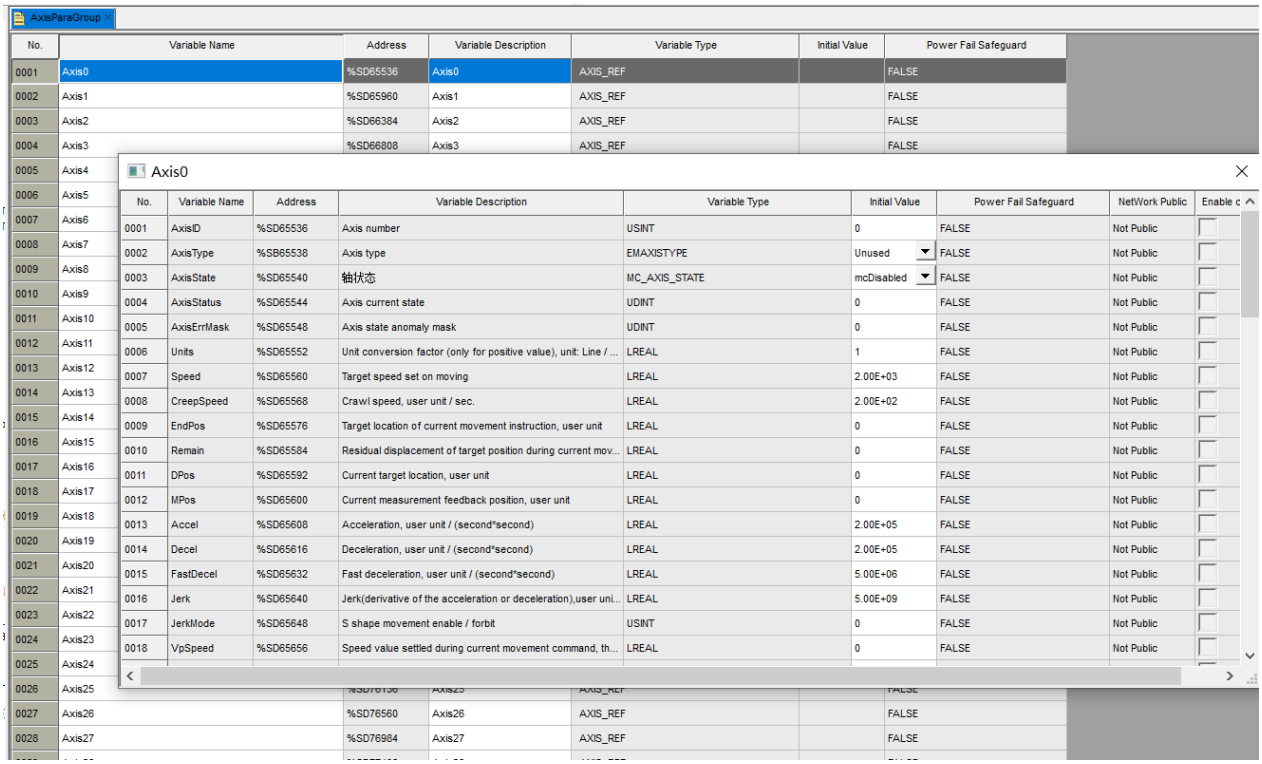
The node of [Global Variable] contains follows variable group, as shown in Table.

Variable Group List

Variable Group Name	Variable Group Description	Content
AxisParaGroup	Axis Parameter Group	Refer to AxisParaGroup
GV_Group	Global Variable Group	Refer to GV_Group
NetVarDiagGroup	Network Variable Group	Refer to NetVarDiagGroup
Table	Table Data Area Management	Refer to Table
TaskDiagGroup	Task Diagnostic Information Group	Refer to TaskDiagGroup

10.3.5.1 AxisParaGroup

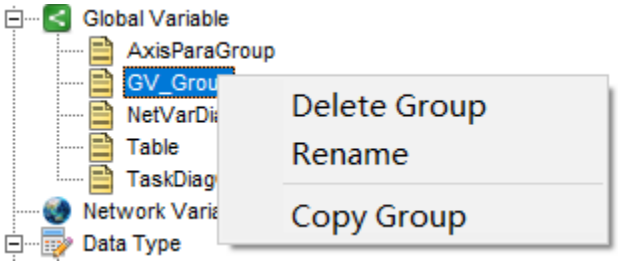
Double-click the [AxisParaGroup] node under the [Global Variables] node to open the axis parameter variable table, which contains 64 axis parameters. Users can modify the variable names and descriptions. Double-click the sequence number in the variable list row to open the detailed parameter settings dialog box, where you can edit the values in the white editable area of the initial value column. As shown in the figure below.



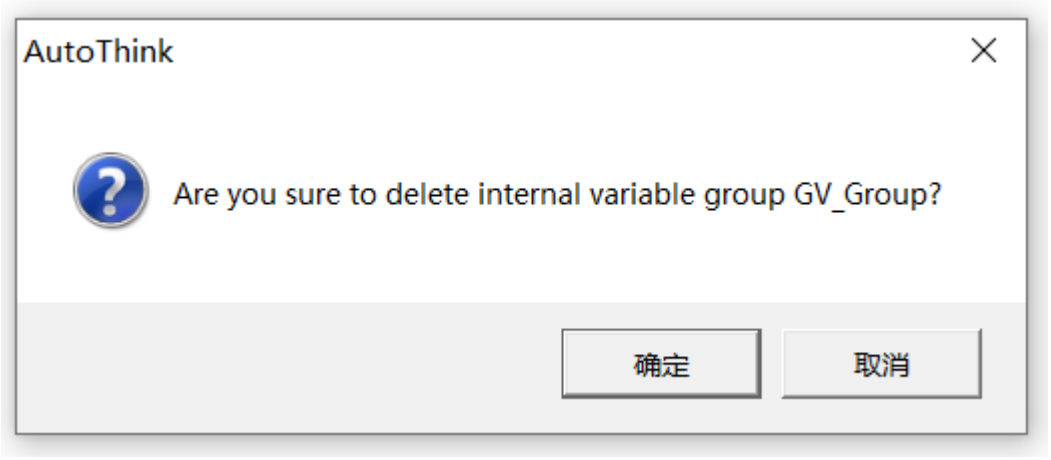
10.3.5.2 GV_Group

In the variable group of GV_Group, global variable is defined to be referenced when users configures logic. See[Variable Declaration Methods](#) for details.

The user can also delete, rename and copy the global variable group, as shown in figure.

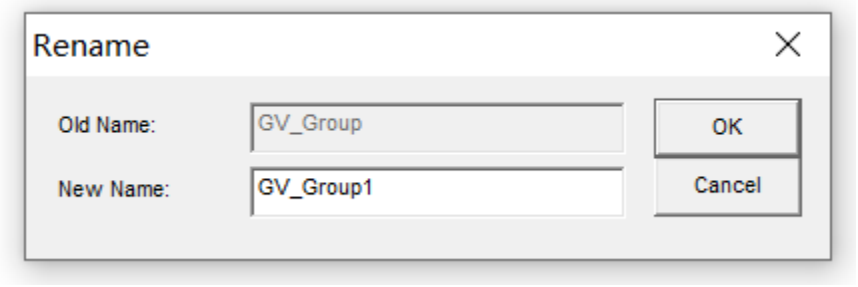


Click the order of [Delete Group] to pop up the prompt box as shown in figure, click the button of **OK** to delete the variable group.



- Please confirm whether to delete the variable group for it cannot be restored.

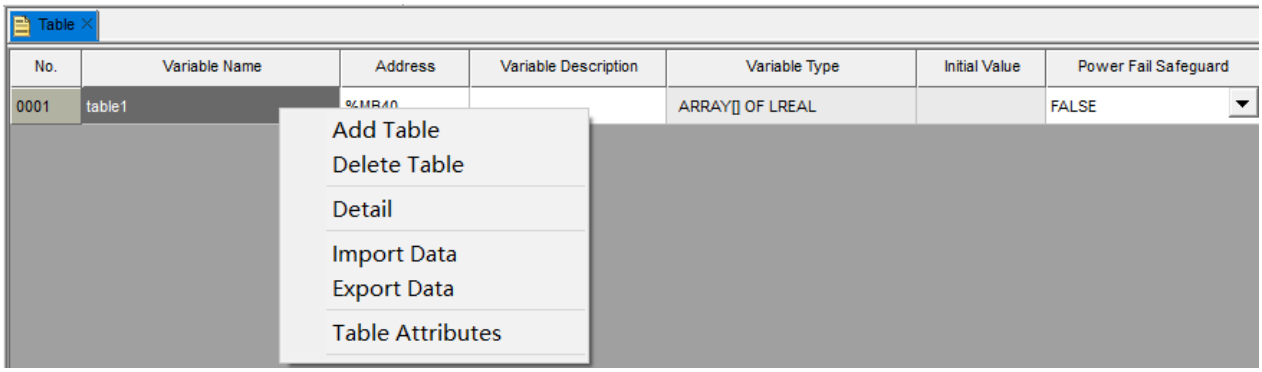
Click [Rename…] to pop up the dialog box of **Rename**, as shown in figure. Enter the variable group name conforming to requirements in the box of **New Name,** and click the button of **OK** to complete.



Click the [Copy Group], then right-click the "Global Variable" node and select the [Paste Group], variable group is copied with the default name "NewGroup1".

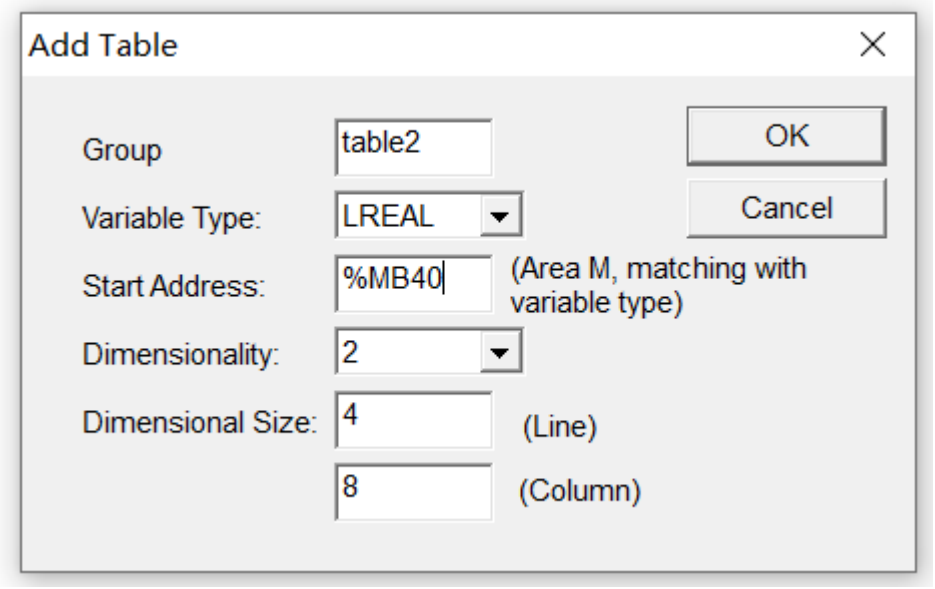
10.3.5.3 Table

In the Table variable group view, users can create, delete, import, and export Table data, among other functionalities. The following sections introduce each supported feature of the Table:



1. Add Table

   Right-click in the view and select Add Table from the context menu.



The naming requirements for Simulation Group names are the same as those for Global Variable names. The variable type is selected from the list, and the starting address can only be in the M area. Dimensions can be one-dimensional or two-dimensional, and the dimension size defines the number of rows and columns in the Table. After definition, the newly added Table variable can be viewed in the Table variable group as shown in the figure below.

Table

No.	Variable Name	Address	Variable Description	Variable Type	Initial Value	Power Fail Safeguard
0001	table1	%MB40		ARRAY[] OF LREAL		FALSE
0002	table2	%MB40		ARRAY[] OF LREAL		FALSE

2. Deleting a Table

Select one or more Table variables, press the DEL key or use the Delete Table command from the right-click context menu to delete the selected Tables.

3. Viewing Table Elements

Select a Table variable. Double-click the variable's sequence number or use the Details command from the right-click context menu. The Table elements will be displayed in a tabular format, with the first row showing row numbers and the first column showing column numbers (as shown in the figure below for table3). You can modify the initial values (offline) and online values (online) of the data elements here.

table2

序号	0	1	2	3	4	5	6	7
0000	0	0	0	0	0	0	0	0
0001	0	0	0	0	0	0	0	0
0002	0	0	0	0	0	0	0	0
0003	0	0	0	0	0	0	0	0

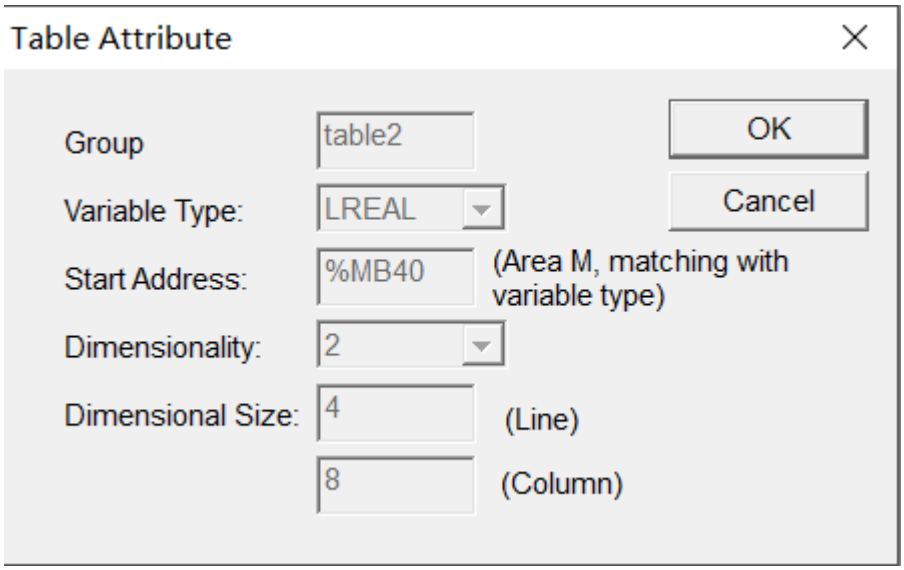
4. Importing/Exporting Table Data

In the Table variable group, after selecting a Table variable, use the Import Data or Export Data command from the right-click context menu to perform the respective operation.When choosing to export data, the exported file will open in the format shown in the figure below. The Table variable is exported row by row, with each row containing the values of the elements in that row.

	A	B	C	D	E	F	G	H	I
1	0	1	2	3	4	5	6	7	
2	0	0	22	33	0	0	0	0	
3	0	111	0	0	0	0	0	0	
4	0	0	0	0	4444	0	0	0	
5									
6									
7									

5. Table Attributes

Select a Table variable. Use the Table Properties command from the right-click context menu. In the dialog box that opens, you can only view the properties of the Table; modifications are not allowed.



10.3.5.4 TaskDiagGroup

Double-click the TaskDiagGroup node under the Global Variables node to open the Task Diagnostic Information Table. Each task has a total of 12 variables, including operating status, operating mode, number of operations, and cycle time (unit: μs).Diagnostic information can be sent to the Monitoring List or OPC UA List as shown in the figure below.

TaskDiagGroup

No.	Variable Name	Address	Variable Description	Variable Type	Initial Value	Power Fail Safeguard
0001	TaskOrder1_First_Scan_On	%SX201608.0	Task1 first cycle running sta...	BOOL	FALSE	FALSE
0002	TaskOrder1_RunStatus	%SB201609	Task1 running status-0:Unk...	BYTE	0	FALSE
0003	TaskOrder1_RunMod	%SB201610	Task1 operating mode-0:Sin...	USINT	0	FALSE
0004	TaskOrder1_RunCount	%SD201612	Task1 run count	UDINT	0	FALSE
0005	TaskOrder1_RunTime	%SD201616	Task1 last run time	UDINT	0	FALSE
0006	TaskOrder1_RunTimeMin	%SD201620	Task1 minimum value of hist...	UDINT	0	FALSE
0007	TaskOrder1_RunTimeMax	%SD201624	Task1 maximum value of his...	UDINT	0	FALSE
0008	TaskOrder1_RunTimeAverage	%SD201628	Task1 average value of hist...	UDINT	0	FALSE
0009	TaskOrder1_RunCycleTime	%SD201632	Task1 last operation cycle	UDINT	0	FALSE
0010	TaskOrder1_RunCycleTimeMin	%SD201636	Task1 minimum value of hist...	UDINT	0	FALSE
0011	TaskOrder1_RunCycleTimeMax	%SD201640	Task1 maximum value of his...	UDINT	0	FALSE
0012	TaskOrder1_RunCycleTimeAverage	%SD201644	Task1 average value of hist...	UDINT	0	FALSE

10.3.6 Network Variable

10.3.6.1 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.

10.3.6.2 Precautions

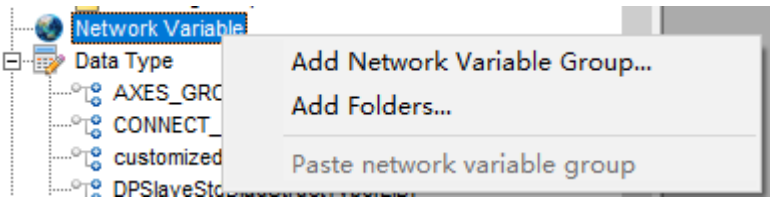
- To ensure data consistency, the IEC task only updates the network variable receiver data before the start of each cycle, and updates the network variable transmitter data after the end of each cycle. Therefore, the actual transmission and reception cycle of network variable data is determined by the protocol configuration and the IEC task cycle.
- The minimum transmission interval for the network variable group associated with full speed tasks is 1ms.
- 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.

10.3.6.3 Adding Network Variable Group

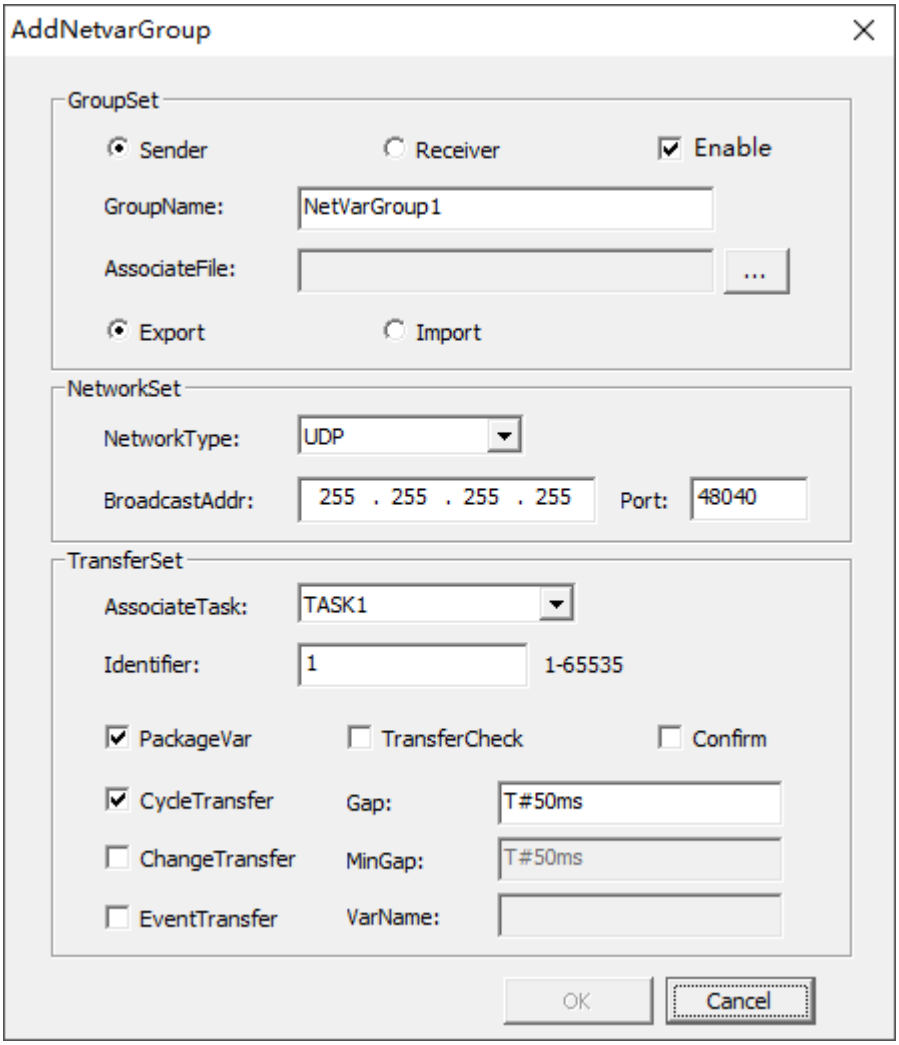
10.3.6.3.1 Adding sender network variable group

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 task of the current application that controls the variables to be sent. The LX controller processes received variable data at the beginning of the IEC task cycle and sends variable data at the end of the IEC task cycle. 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

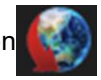
interval' can be used to define the shortest time between two transmissions, for example: 'T # 50ms'.

- 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.

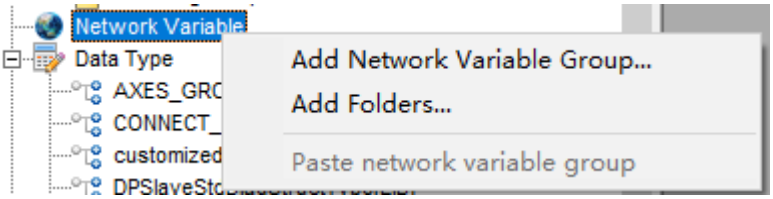
(4) Click 'OK' to successfully add the sender network variable group.

The sender icon  is , if the group is not enabled, the icon will be grayed out.

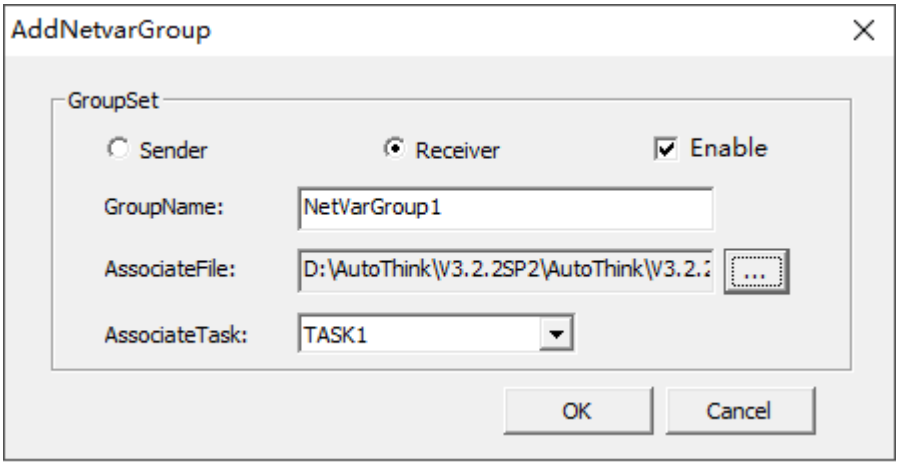
10.3.6.3.2 Adding receiver network variable group

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.

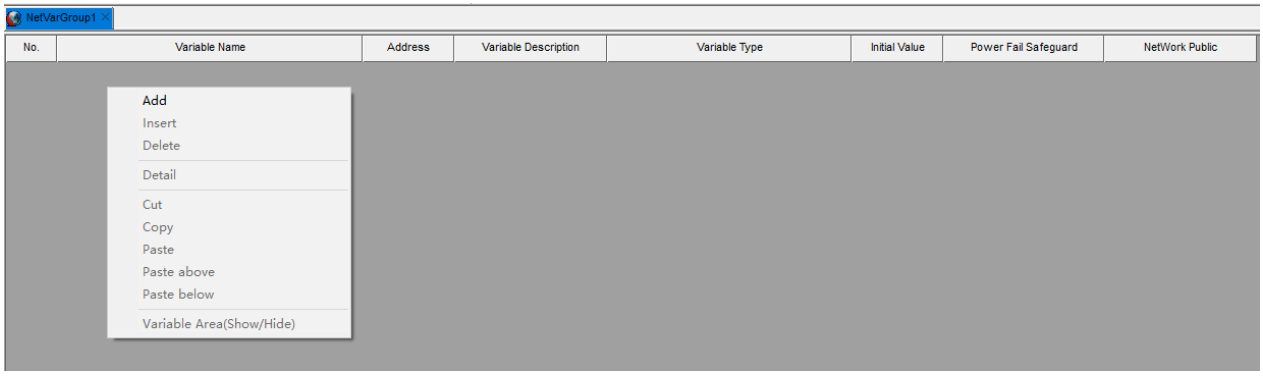
- The GroupName and AssociateTask in the receiver interface are consistent with those of the sender.
- AssociateFile: The system creates a recipient network variable group based on the associated file, and imports the configuration information of the variable group during creation. Editing is not supported 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.

10.3.6.4 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.

10.3.6.5 Network Variable Management

10.3.6.5.1 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.

10.3.6.5.2 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.

10.3.6.5.3 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.

10.3.6.5.4 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.

10.3.6.5.5 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

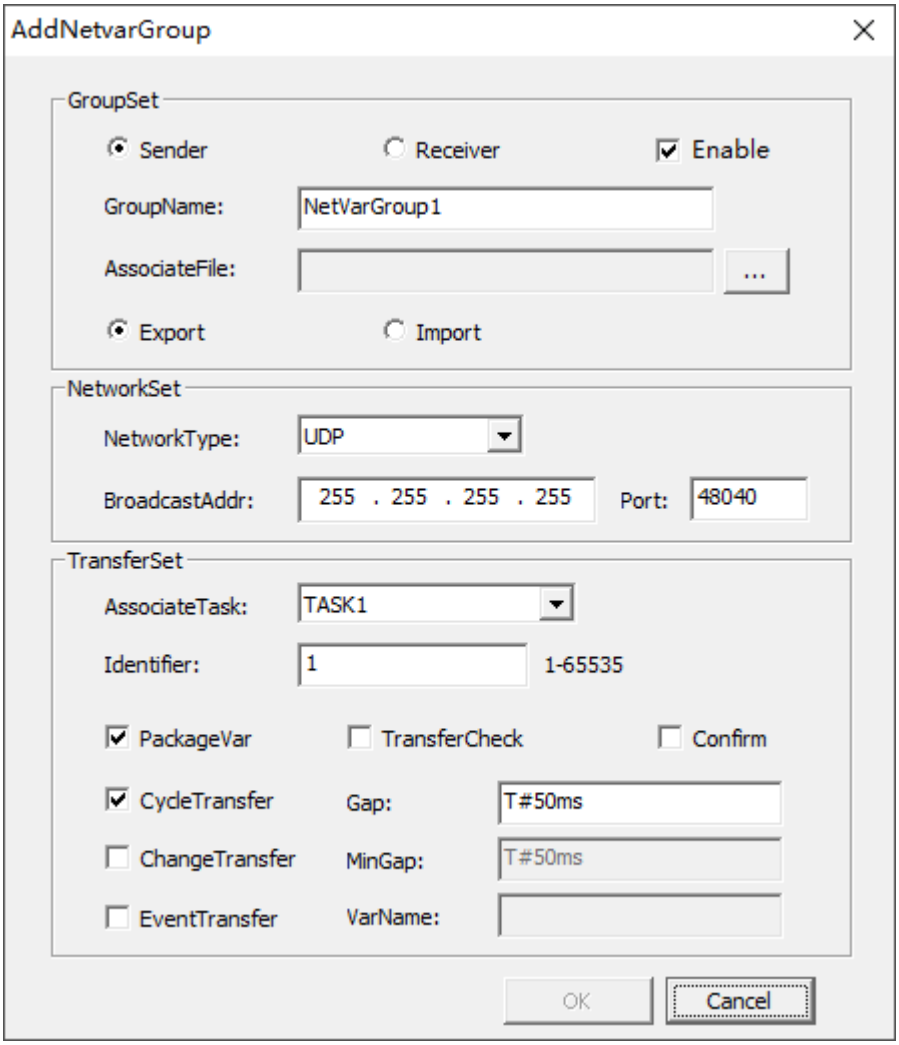
   Automatic synchronization refers to the mandatory synchronization of network variable groups and nvg file contents during compilation, ensuring strict consistency of downloaded files.

10.3.6.5.6 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.



10.3.6.6 Diagnostic instructions

Explanation of Network Variable Group Diagnostic Information:

Diagnostic Name	Diagnostic Description	Explanation of Diagnostic Information
NetVarGroupN_SendSuccessCounter	Number N network variable group successfully sent count	
NetVarGroupN_SendErrorCounter	Number N network variable group sending failure count	
NetVarGroupN_SendLastErrorNumber	Nimber N network variable group failed to send error code	0: Sent successfully 100: Socket sending failed
NetVarGroupN_ReceiveSuccessCounter	Number N network variable group received successful count	
NetVarGroupN_ReceiveErrorCounter	Number N network variable group reception failure count	
NetVarGroupN_ReceiveLastErrorNumber	Number N network variable group failed to receive error code	0: Received successfully 1: Reserved, not yet used 2: Abnormal length of received message 3: Received message identity error

		4: Received message packaging mode mismatch 5: Received non REQ message 6: Received message GroupID error 7: The number of variables in the received message is incorrect 8: Received message CRC check error 9: Receiving unpackaged message with incomplete sub packets 10: Received message out of order 11: Receiving unpackaged packets with sub packets out of order 12: Received duplicate serial number message 13: Inter core communication queue full and packet loss

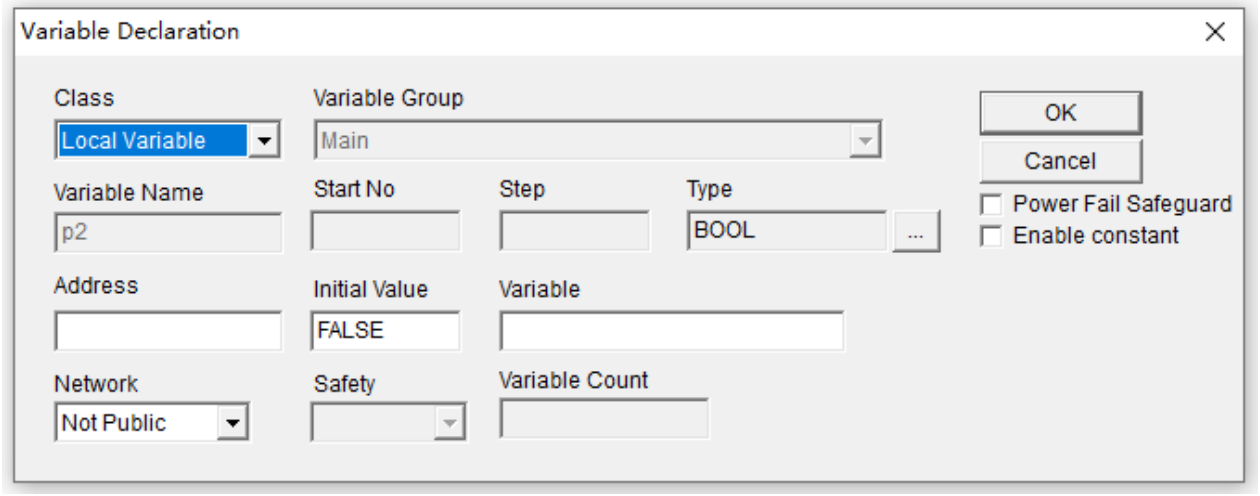
10.3.7 Variable Property

By selecting the Variable Properties menu item, you can quickly display the properties window of the currently selected variable.

In the program editing area, select a variable element, variable name, or a transition in SFC language, and view the variable properties through the following methods:

- Program Area: Right-click the element or variable name, then click [Variable Property].
- Menu Bar: Click [Edit] - [Variable Property].
- Shortcut Key: Press **Shift + F2**.

This will open the Variable Declaration dialog box corresponding to the variable, as shown in the figure. You can modify the Class, Type, Address, Initial value, Variable, Network, and for global variables, you can also modify the Variable Group.



10.3.8 Monitor variables

10.3.8.1 Create watch list

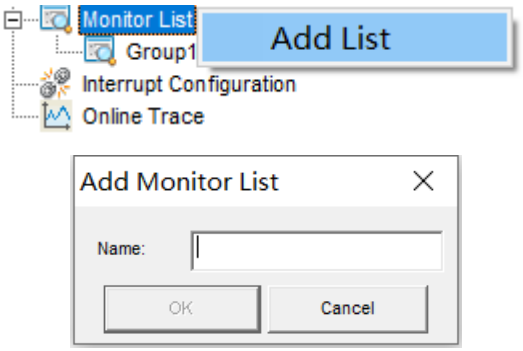
10.3.8.1.1 Introduction

It is used to add important variables to each watch list so that changes in the value of each variable can be tracked and observed in monitoring mode

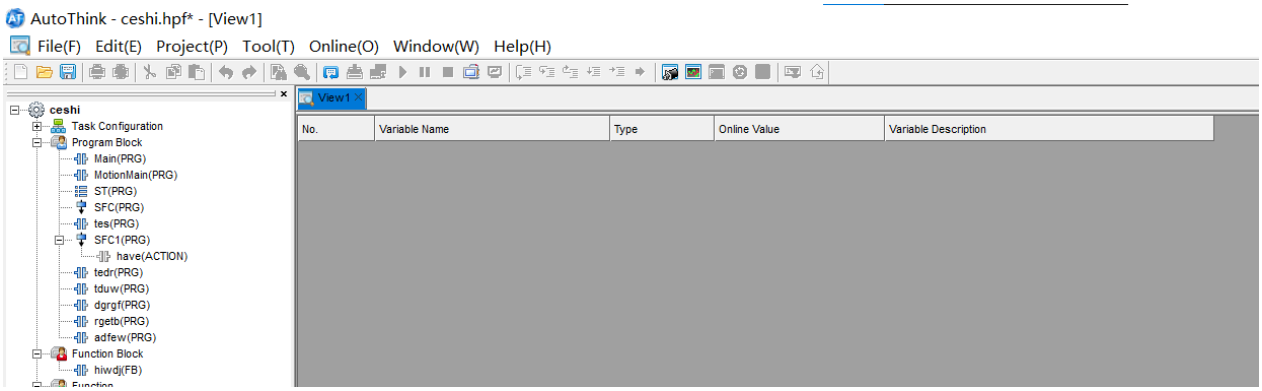
10.3.8.1.2 Steps

1. Create watch list

   Project Management Tree: Right-click the Watch List, and click Add List.



   In the Name box, enter a name for the watchlist. Click OK to add a branch to the list under the Watch List node and to automatically load the editing window for the list in the workspace. As shown in the figure.



The sub-nodes of the watch list can be deleted or renamed.

2. Edit variables in the watch list

   In the editing window of the watch list, the monitored variables can be added, modified, or deleted. Right-click in the editing window of the watch list. Select the Add Variable command. In the Variable Name, enter the variable name or variable address defined in the program. The Variable Type, Variable Description, and Online Value (the real-time value of the variable) of the variable are read and displayed automatically in monitoring mode.

10.3.8.2 Add variables to watch list

10.3.8.2.1 Introduction

Global variables, PRG variables, channel variables, or diagnostic variables can be added to the watch list.

10.3.8.2.2 Requirement

The watch list has been created.

10.3.8.2.3 Steps

(1) In the variable definition area, right-click the Variables.

(2) In the right-click menu, click the Send to Watch List command to display all watch list names.

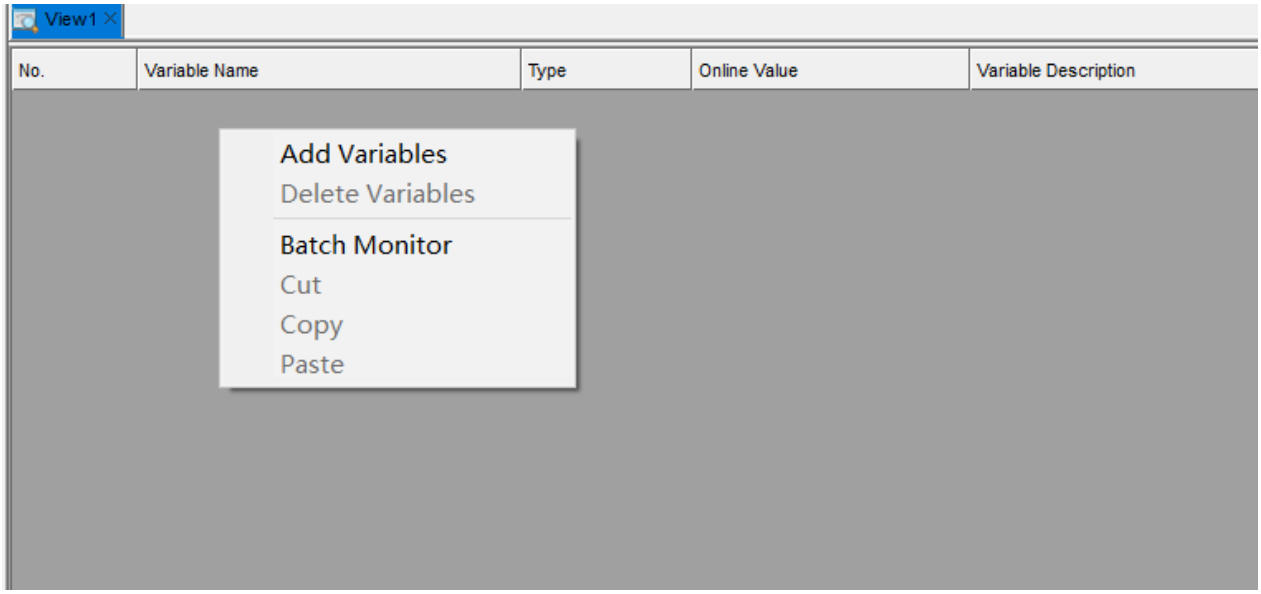
(3) Select the watch list to be added. The variables are added to the watch list.

10.3.8.3 Bulk monitoring

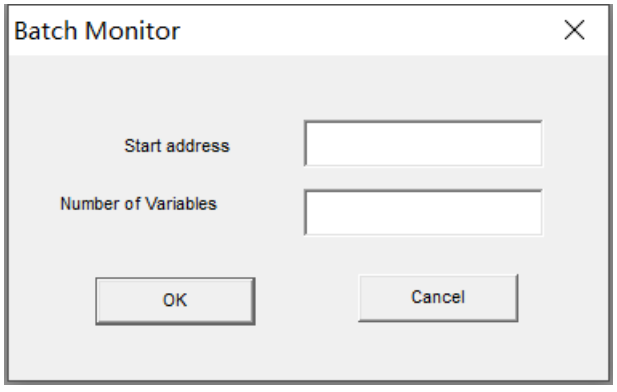
10.3.8.3.1 Introduction

Multiple watch lists can be created to classify and manage important variables, making it easy to grasp and track changes in their values in monitoring mode.

10.3.8.3.2 Steps

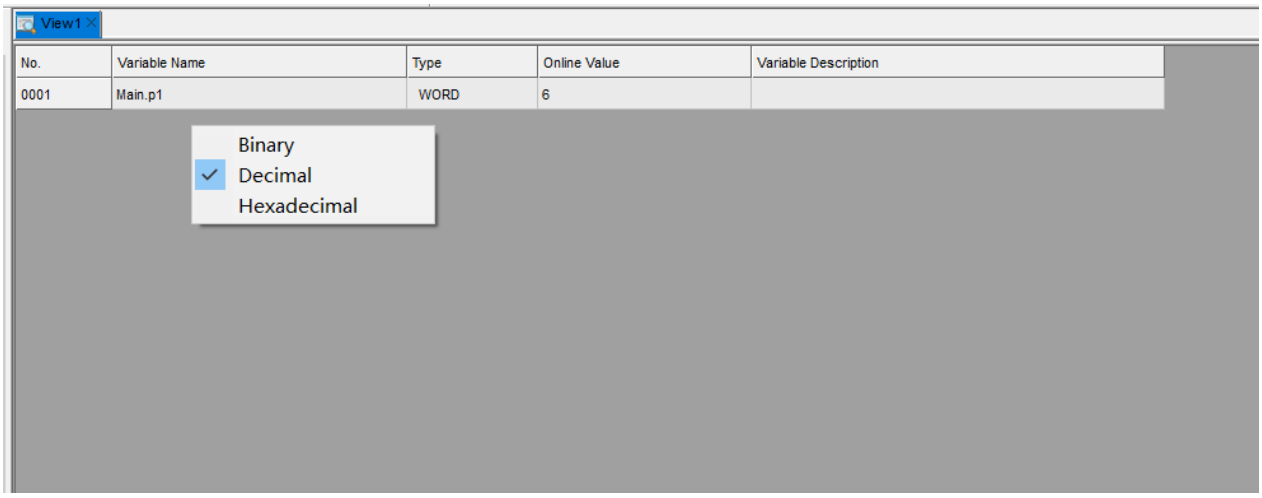


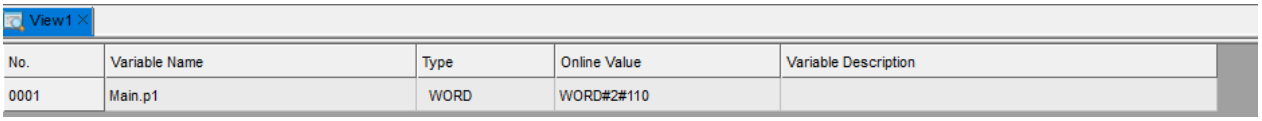
By right-clicking in the editing window of the watch list in offline, users can select the Batch Monitor menu item to bulk monitor the contents of a consecutive address (up to 100 characters). As shown in the figure below, bulk monitoring is completed by determining the memory area to be monitored, entering the starting address and the number of monitored variables. Where the starting address contains the following information: the area (I/Q/M/S) of the variable to be monitored, the type (bit/byte/word/double word), and the starting address. The type of all variables in a bulk monitoring setup shall be the same.

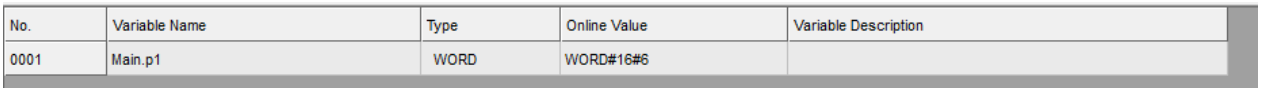


After users click OK, the software checks whether the starting address and the number of variables entered are legal and whether the length of the consecutive address is more than 100 characters.

In the online monitoring state, users can select the number system of online values of the monitored variables via the right-click menu, which indicates the display of the number system of all online values of the monitored variables in the watch list.







- global variables can be added to the watch list, or the local variables can be added to the program block (which shall be of PRG). To add the local variables, the format shall be Program name.variable name, such as TEST1.VAR1.

10.3.9 Variable access

10.3.9.1 Introduction

Access means use. Access to variables during operations includes reading the value of a variable and assigning a value to a variable.

10.3.9.2 Steps

1. To access a simplex variable, just write its variable name.

   For example: AM01 := AM02 + AM03.

   This example means to read the values of the variables AM02 and AM03, add them, and assign the result of the addition to the variable AM01.

2. To access the data bits of an integer variable, write it in the format Variable name.bit index, where the index bit shall be smaller than the number of data storage bits. Otherwise, a compilation error is reported.

For example, variable X is an INT data, and Y is a BOOL data. Y: =X.3 means to assign the 4th bit of variable X to variable Y.

3. Breakpoint commissioning does not support bit data configuration.

   Data types that support data bit access include BYTE, INT, DINT, SINT, UINT, UDINT, USINT, WORD, DWORD, and ARRAYOF.

4. To access an item of a functional block instance, write it in the format Functional block variable name.item name.

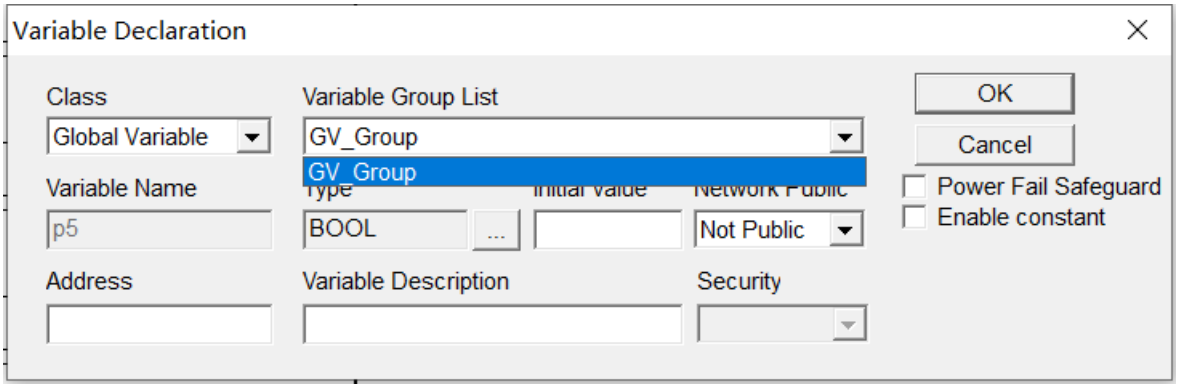
   For example: PID01.SP (To take the set value item of PID01, provided that the type of PID01 has been declared as HSPID).

5. To access a local variable (referenced by other programs), write it in the format Program name.variable name.

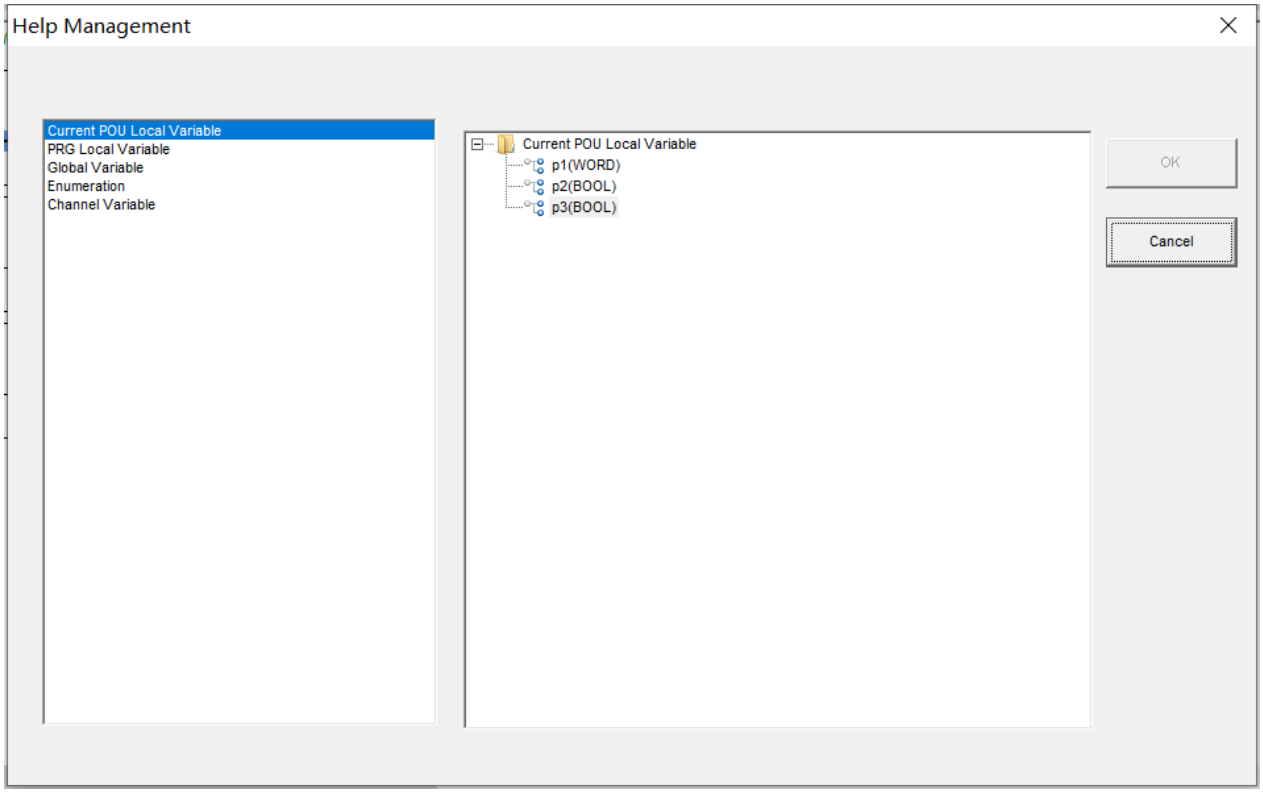
   For example: VAR1 (local variable defined in POU1), referenced in POU2 in the format POU1.VAR1.

6. To have quick access to the variables

   In the program editing area, select a variable name. Press Shift + F2 to bring up the corresponding Variable Declaration dialog box, which quickly displays the properties window of the currently selected variable, as shown in the figure below. The edit box with white background can be modified. See the contents of Variable Properties under each editing window for detailed operations.



In the program editing area, select a variable name. Press F2 to bring up the Help Management dialog box, as shown in the figure below, in which users can select the variables they want to use.



-

- The variables defined in the variable definition area at the top of the editing window for function, functional block, and program POUs are local variables. It is allowed to define global and local variables with the same name. However, note that local variables have higher priority and overwrite global variables.
- When using variables in a program, be sure for consistency in data types. That is, the data types of the variables at both ends of a signaling link shall definitely be the same.

10.3.10 Set variable display

10.3.10.1 Introduction

For the variables defined in the I area, Q area, M area, and S area, when the logic is configured in the POU scheme page, if the direct address, such as %IX0.0, is written, and if there is a variable with the same direct address, it is displayed according to the following priority: global variable > PRG local variable > channel variable. That is, if a global variable g1:BOOL:%IX0.0 is defined, it is prioritized to associate with g1, and so on.

Example: A global variable g1 is defined for the configuration, as shown in the figure below.

GV_Group

No.	Variable Name	Address	Variable Description	Variable Type	Initial Value	Power Fail Safeguard	NetWork Public
0001	g1	%IX0.0		BOOL	FALSE	FALSE	Not Public

A PRG local variable g2 is defined, as shown in the figure below.

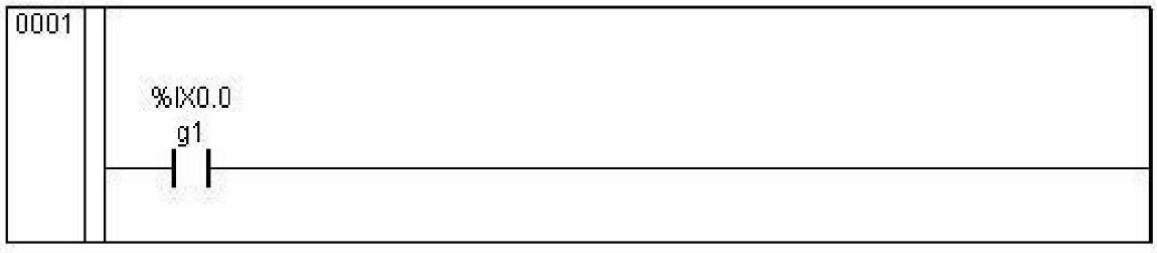
adfew(PRG).ld

No.	Variable Name	Address	Variable Description	Variable Type	Initial Value	Power Fail Safeguard	NetWork Public
0001	g2	%IX0.0		BOOL	FALSE	FALSE	Not Public

The channel variable is shown in the figure below.

Channel Number	Channel Name	GroupName	Channel Type	Bytes	Channel Address	Channel Description
1	E8_IN_1		BOOL	0.1	%IX176.0	Input
2	E8_IN_2		BOOL	0.1	%IX176.1	Input

When configuring logic, in the menu [Project] - [Options] - [Configuration Language] - [Display], check the entered address. The address is replaced by a variable. In the coil or contact, enter %IX0.0. The variable g1 with higher priority is displayed as shown in the figure below.



- Except for variables with the same name.

10.3.11 Import Variables

■ Steps:

The format of the data table is shown in the figure below.

	A	B	C	D	E	F
1	GV_Group(COMMON)					
2	Variable	Address	variable Description	variable Type	Initial value	Power Fail Safeguard
3	P1			DWORD	0	FALSE
4	P2			DWORD	0	FALSE
5						
6						
7						
8						
9						
10						

The data that can be imported is divided into two categories: variables in the global variable group and

variables in the program (PRG).

- Variables in the global variable group

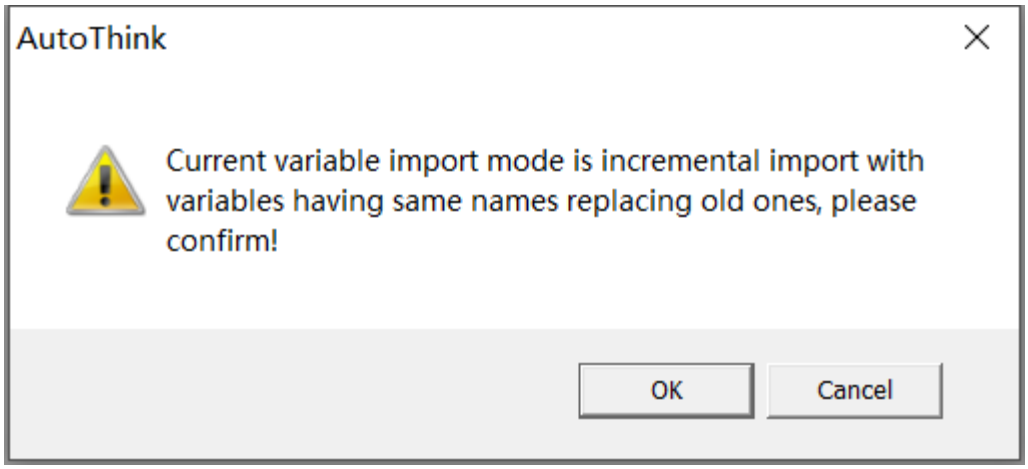
  There is no requirement for the name of this form. The first row of the form shall be formatted as Variable group name + (COMMON). When the import operation is executed, if there is no corresponding variable group in the project, the project automatically creates a variable group with the same name as the import form and adds the corresponding data.

- Local variables in the program (PRG)

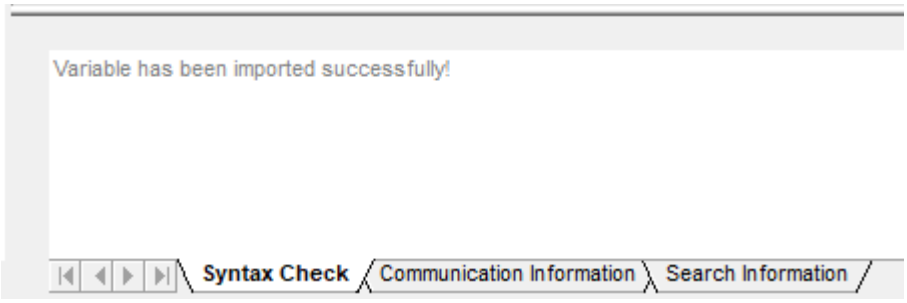
  There is no requirement for the name of this form. The first row of the form shall be formatted as Program name + (COMMON). When the import operation is executed, if there is no corresponding program in the project, the variables in the corresponding form cannot be imported, and the message PRG local variable group does not exist: **Import failed is displayed in the message window.

  Before importing variables, users can select the import mode of the variables. The default is incremental overwrite import. See Configuration for import mode selection.

  The value definitions of the individual terms of the variable shall be consistent with those in the project. Execute this command. The import mode confirmation prompt box is displayed, as shown in the figure.



Click OK. The data in the selected table is imported into the corresponding program or global variable group. The Message Output Window prompts for import information. As shown in the figure.



- When users bulk import variables, the width of string variables shall not exceed 1000 characters. Otherwise, the import may fail.

- Except for the variable columns that need to be modified, the rest of the columns in the form shall not be modified, including adjusting the column width, etc. Otherwise, the import may fail. If fails, please re-export the form for use.
- When the import operation is executed, the hidden worksheets in the form are imported. Please delete the worksheets that are not in use.

10.3.12 Export Variables

10.3.12.1 Introduction

To make bulk edits to variables in a project, or to import them into another project, it is necessary to export the variables in the project first.

The exportable variables include the variables in the global variable group and the program (PRG). You can select the corresponding variable group to export as needed.

The variables are exported as .xls variable table in variable groups, which are classified and displayed by tabs in the variable table. All properties supported by the variable are exported to the variable table, such as power failure protection, force, and SOE enable.

10.3.12.2 Requirement

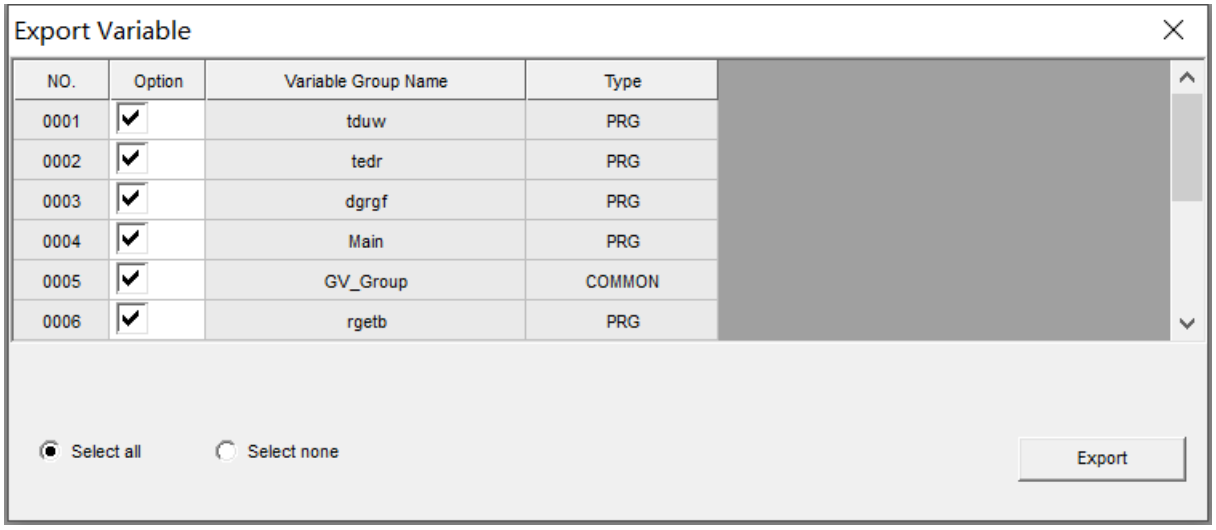
The project is open

10.3.12.3 Steps

To export variables, follow the steps below:

(1) Click the Project menu.

(2) Select the Export Variables command. The Export Variables dialog box is displayed.



(3) Check the variable groups that need to be exported.

(4) Click Export. The Save As dialog box is displayed.

(5) Enter a filename and select a save path. The default is to save with the project name.

(6) Click Save. The variable table is exported.

	A	B	C	D	E	F	G	H	I
1	Main(PRG)								
2	Variable Name	Address	Variable Description	Variable Type	Initial Value	Power Fail Safeguard	Enable to force	NetWork Public	Enable constant
3	p1			BOOL	0	FALSE	TRUE	Not Public	FALSE
4	p2			BOOL	0	FALSE	TRUE	Not Public	FALSE
5	p3			BOOL	0	FALSE	TRUE	Not Public	FALSE
6	p4			BOOL	0	FALSE	TRUE	Not Public	FALSE
7	p5			BOOL	0	FALSE	TRUE	Not Public	FALSE
8	p6			STRING(80)	''VAR'	FALSE	TRUE	Not Public	FALSE
9									

When users edit the initial value of the STRING in the table, 3 single quotes appear. The first quote is Excel's escape character for the English quotes, such as the E8 cell in the figure. That means users can only edit the initial value of the last two single quotes between them.

10.3.13 Export Variable Table

10.3.13.1 Introduction

The point table in the project can be exported for review and bulk editing. There are two ways to export the point table, either automatically or manually.

The point table contains information about variables in the global variable group, program (PRG), and diagnostic variable group.

The point table is exported as a .csv file and includes point table information such as Variable Name, Variable Group Name, Variable Description, Variable Type, Data Area, Variable Offset, and Variable Size.

The description of the point table information:

Point Table Information	Description
ProjectName	Project name
DateTime	Export time, displayed as year-month-day hour-minute-second
Variable Name	Variable name
Variable Group Name	Name of the POU, global variable group, or diagnostic variable group in which the variable is located
Variable Description	Variable description information
Variable Type	Variable type
Data Area	The storage area assigned to variables, which defaults to G area for variables in the POU and global variable group, and to S area for variables in the diagnostic variable group
Variable Offset	Refers to the offset of the variable in the storage area. A variable associated with a direct address has an offset that maps to the direct address as follows. Variables that are not associated with a direct address have their offsets randomly assigned by the system 1 For BOOL variables, the mapping relationship between offset and direct address is: If the direct address of a BOOL variable is %IXm.n/%QXm.n/%MXm.n, offset = m*8+n 2 For variables of other data types, offset = direct address
Variable Size	Refers to the number of bytes stored in the variable, which is determined by the variable's data type For BOOL variables, the variables associated with direct addresses have a variable size of 0. The variables that are not associated with direct addresses have a variable size of 1 byte by default
Modbus Address	All variables in the I, Q, and M areas, within the corresponding address ranges (as shown in the table below), are automatically mapped with a unique Modbus address

Data Area		Type	Variable Address Range	Modbus Address
Area I	%IX	BOOL	%IX0.0~%IX374.7	X0000~X2999
	%IW	WORD	%IW0~%IW5998	X0000~X2999
Area Q	%QX	BOOL	%QX0.0~%QX374.7	X0000~X2999
	%QW	WORD	%QW0~%QW5998	X0000~X2999
Area M	%MX	BOOL	%MX0.0~%MX7816.7	X3000~X65535
	%MW	WORD	%MW0~%MW125070	X3000~X65535

	A	B	C	D	E	F	G	H
1	ProjectName: project							
2	DateTime: 2024-11-19 16-34-17							
3								
4	VarName	GroupName	Comment	Type	DataArea	Offset	Size	ModBusAddress
5	AT_BatteryAlarm	AT_D_LX_CU500	Battery alarm status-0:No battery 1:Normal 2:Low battery	BYTE	S	196610	1	
6	AT_CPUFatalErr	AT_D_LX_CU500	Controller Fatal Failure	DWORD	S	196616	4	
7	AT_CPUTimePulse_100ms	AT_D_LX_CU500	100ms pulse	BOOL	S	1572995	0	
8	AT_CPUTimePulse_200ms	AT_D_LX_CU500	200ms pulse	BOOL	S	1572994	0	
9	AT_CPUTimePulse_1min	AT_D_LX_CU500	1 minute pulse	BOOL	S	1572992	0	
10	AT_CPUTimePulse_10ms	AT_D_LX_CU500	10ms pulse	BOOL	S	1572997	0	
11	AT_CPUTimePulse_20ms	AT_D_LX_CU500	20ms pulse	BOOL	S	1572996	0	
12	AT_CPURun	AT_D_LX_CU500	Project operation status-0:Unknown 1:Running 2:Stop	BYTE	S	196608	1	
13	AT_CPUKey	AT_D_LX_CU500	Keyswitch status-0:Unknown 1:RUN 2:STOP	BYTE	S	196609	1	
14	AT_CPUMemUsage	AT_D_LX_CU500	Data area memory usage	BYTE	S	196613	1	
15	AT_TimeOfLastSystemPowerFault	AT_D_LX_CU500	Time of the last system power failure	STRING	S	196628	32	
16	AT_CPUTimePulse_1s	AT_D_LX_CU500	1 second pulse	BOOL	S	1572993	0	
17	AT_SDState	AT_D_LX_CU500	SD Cord status-0:Non-existent 1:Existence	BYTE	S	196611	1	
18	AT_CPUErr	AT_D_LX_CU500	General failure of controller	DWORD	S	196620	4	
19	AT_CPUTemp	AT_D_LX_CU500	Controller CPU temperature	BYTE	S	196612	1	
20	TaskOrder2_RunTimeAverage	TaskDiagGroup	Task2 average value of historical run time	UDINT	S	201668	4	
21	TaskOrder2_RunTimeMax	TaskDiagGroup	Task2 maximum value of historical run time	UDINT	S	201664	4	
22	TaskOrder2_RunTimeMin	TaskDiagGroup	Task2 minimum value of historical run time	UDINT	S	201660	4	

10.3.13.2 Automatically export point table

When users set the automatic generation of the variable table in the project options, save the project and compile it, the point table is generated in the Project directory. The point table is a .csv file, with VarInfo (project name) as the file name by default. The point table is updated each time a compile operation is

executed thereafter.

Note the following two points:

- For the first time to generate the point table, users need to Save the project first, and then Compile. Otherwise, the compiler always prompts to save the point table afterward.
- A previously generated point table shall not be opened during compilation. Otherwise, the point table cannot be automatically updated.

10.3.13.3 Manually export point table

10.3.13.3.1 Requirement

The project shall be compiled successfully

10.3.13.3.2 Steps

To export a point table, follow the steps below:

(1) In the Menu Bar, click [Project] - [Export Variable Table] command. The Save As dialog box is displayed.

(2) Select the save path. Click Save. The point table is saved to the corresponding path with the name of VarInfo (project name) by default.

10.3.14 Export Public Variable Table

10.3.14.1 Introduction

The public variables are communication media for data interaction with the touch screen. After setting the network disclosure properties of global variables/local variables, users can select the variables with public properties to be exported in .xml format for touch screen import.

If the project is opened and compiled successfully by AutoThink, the Export Public Variables Table is enabled. It is grayed out by default if the project is not opened or not successfully compiled.

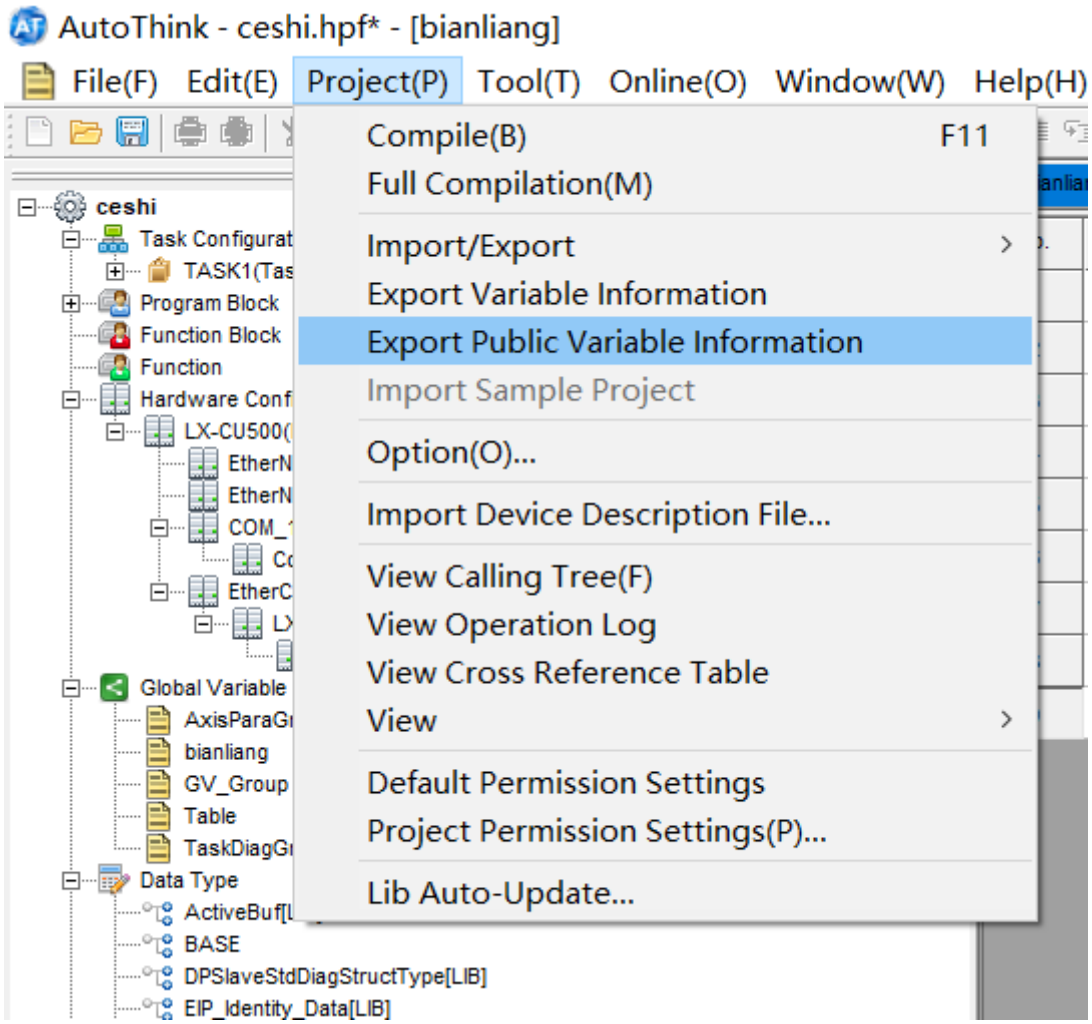
After the project has been compiled, users can export all the public variables in the project to an .xml file via the Export Public Variables Table menu. The variables include program block variables and global variables. The variable types include basic and custom types.

10.3.14.2 Steps

To export the public variable table, follow the steps below:

(1) In the Menu Bar, click [Project] - [Export Public Variable Table] command. The Export Public

Variable Table dialog box is displayed.



(2) Select the save path. Click Save. The default name is VarInfo(Untitled).xml. The public variable table is saved under the corresponding path.

10.3.15 Recycle Unused Variables

10.3.15.1 Introduction

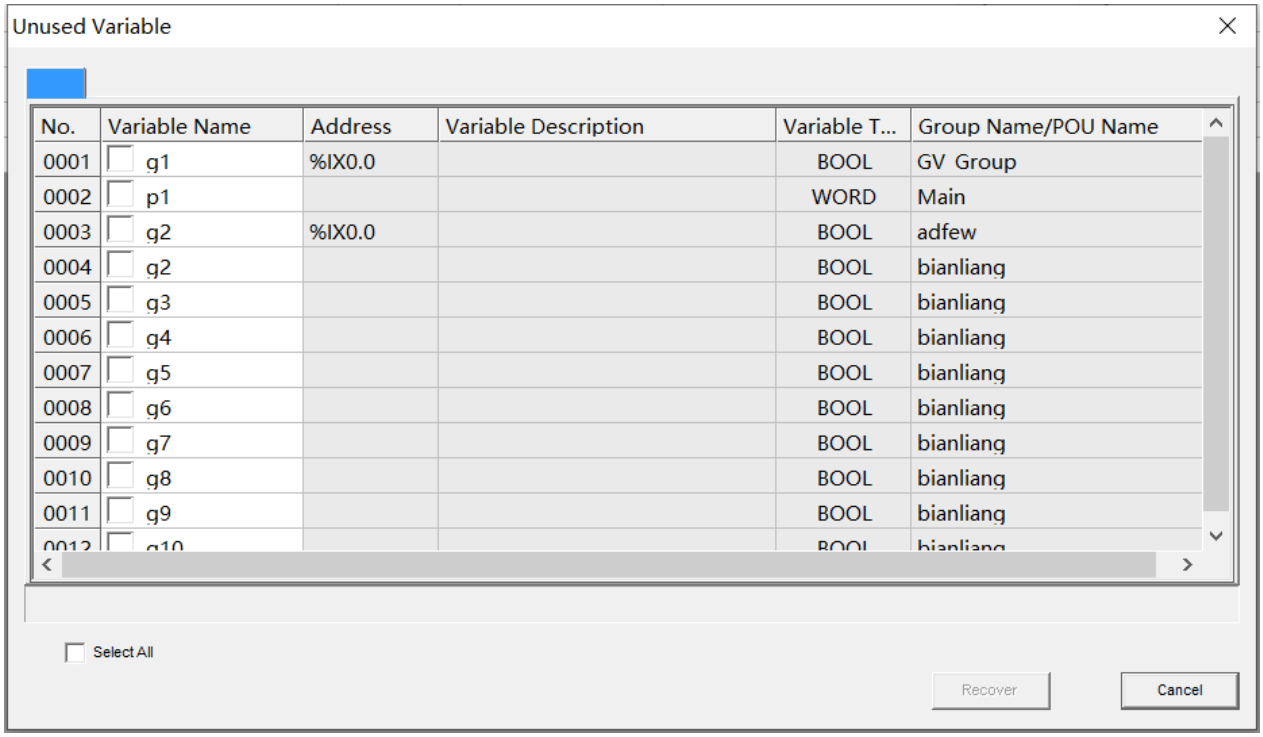
This command allows you to recycle unused variables in the current project.

10.3.15.2 Steps

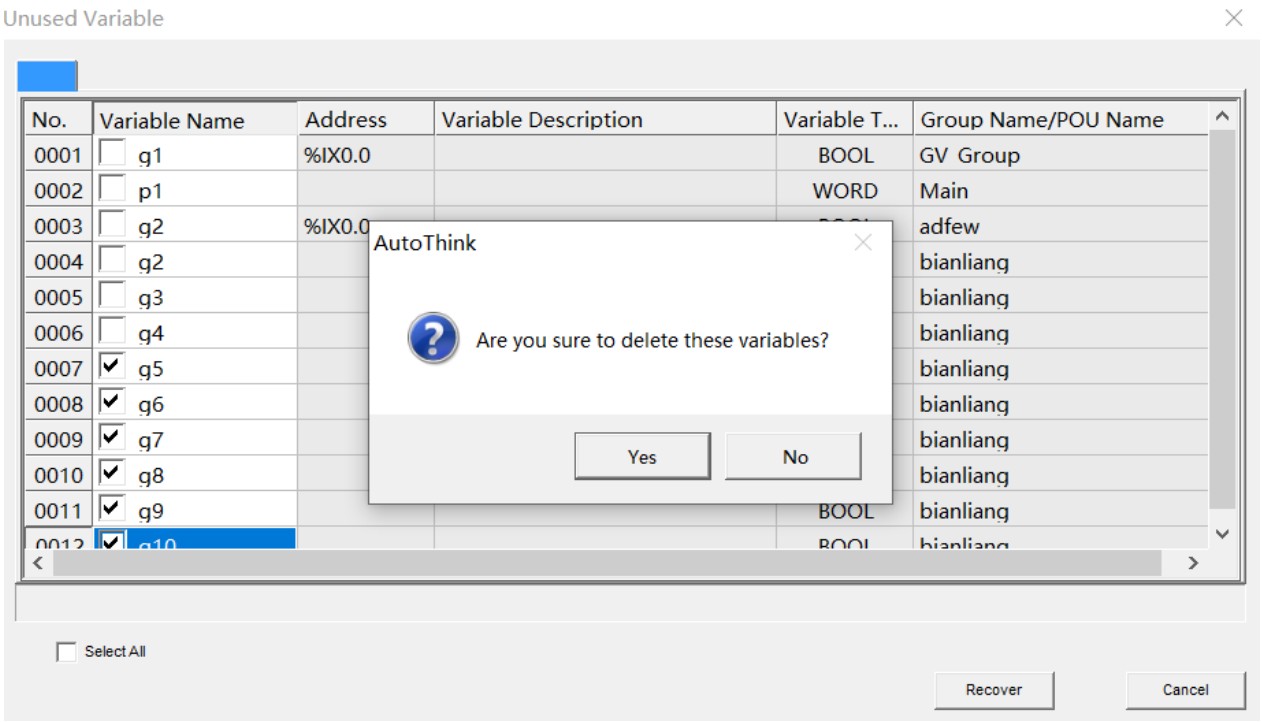
(1) Menu Bar: Click [Project] - [View] - [Unused Variables].

As shown in the figure below. The window displays the variables that have been defined but not used in the global variables group and the PRG POU. The window records the names of unused

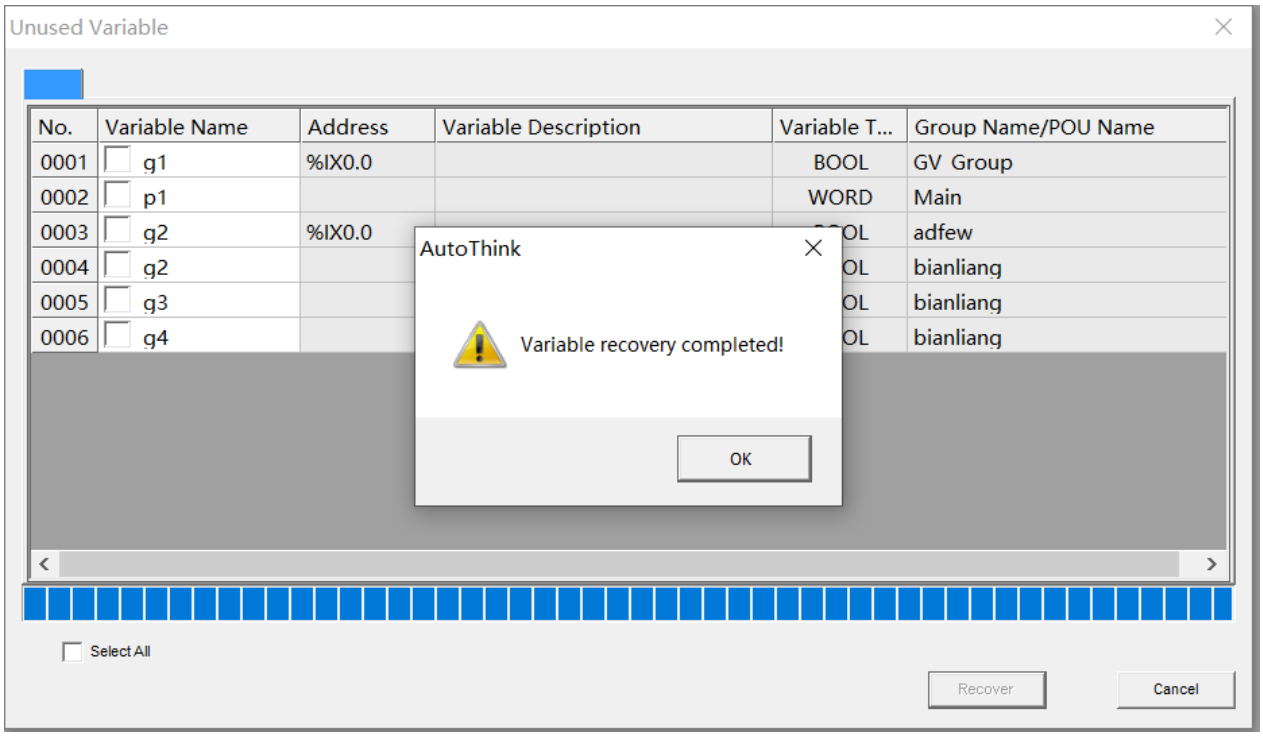
variables and the names of the global variable groups or POUs in which the variables are located. Users can select the variable names for recycling.



(2) Manually check the variables that need to be deleted, or click the Check All box to quickly select them all. Click Recycle. The confirmation dialog box is displayed, as shown in the figure below.



(3) Click Yes. The confirmation box for the completion of variable recycling is displayed, as shown in the figure below.



- 
- This function is only available after the project has been successfully compiled.

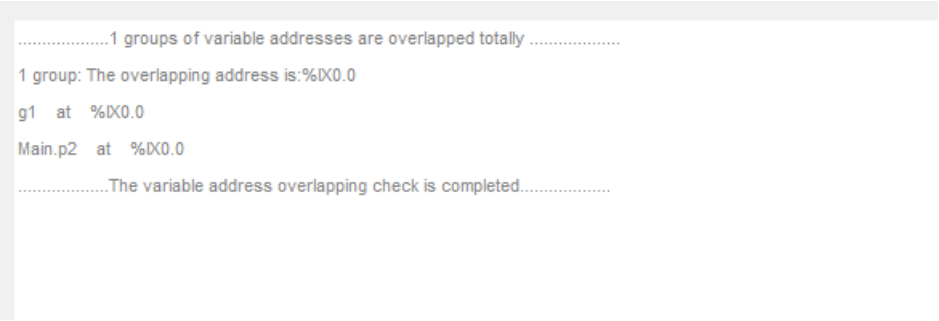
- If the POU property is read-only or the POU has a password but has not been opened, the POU variable cannot be recycled.

10.3.16 View Address Overlap

Menu Bar: Click [Project] - [View] - [Address Overlap].

Select this command to check the overlap of memory allocation in the project. The scope of the check includes variables for which direct addresses have been defined in global variable groups and PRG POUs, storage areas allocated by allocation methods, and channel addresses in the hardware configuration. In addition to the allocation methods, variables in the global variable groups and PRG POUs shall be referenced in the configuration logic to be detected.

The results of the check can be viewed in the Syntax Check tab of the message window, as shown in the figure below. The results of the check are displayed in groups, listing information about each memory overlap, including the address of the memory overlap location and the starting address of the memory where each overlapping variable is located in each group.



- This function is only available after the project has passed the compilation.

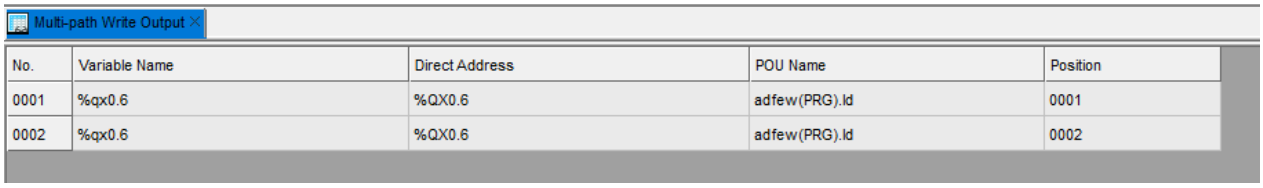
10.3.17 View Multiple Write Outputs

Menu Bar: [Click Project] - [View] - [Multiple Write Outputs].

Select this command to display the Multiple Write Outputs window. View the global variables or variables in PRG POUs that have been used as outputs multiple times, as shown in the figure.

The result information includes the variable names of the multiple write outputs, the names of the POUs that use the variables, and the addresses where the variables are referenced in the POUs.

Multiple Write Outputs can only detect the direct address variable of the Q area.



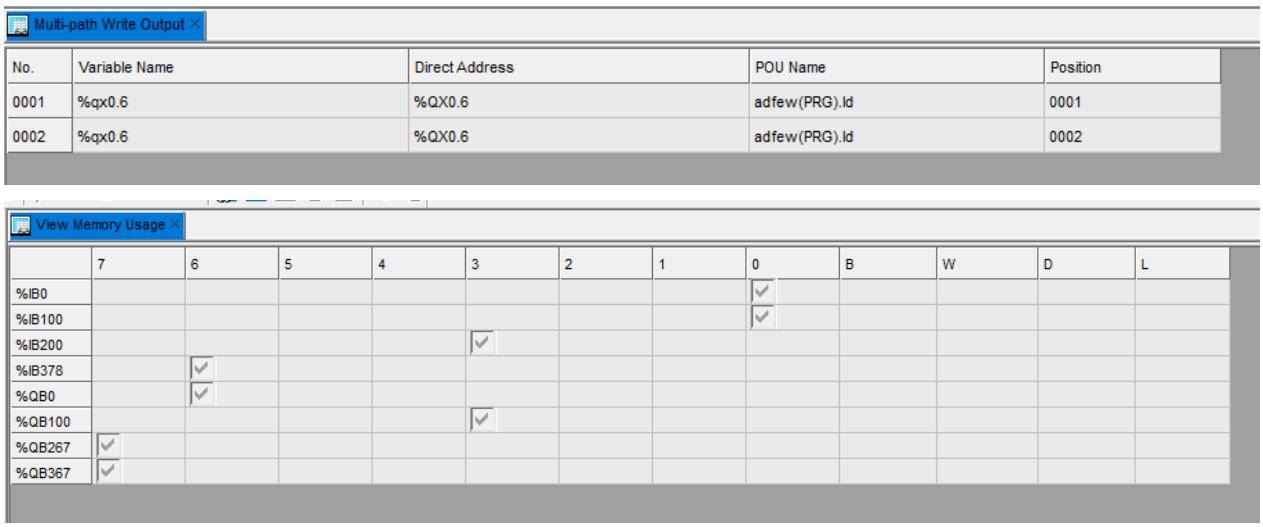
- This function is only available after the project has passed the compilation.

10.3.18 View memory usage

Menu Bar: [Click Project] - [View] - [Memory Usage].

Users can select this command to view the controller memory usage after compiling the configuration project. The software follows the I/Q/M/S sequence in bytes. If a byte of memory has been configured, it displays to the users the usage of that byte, such as which bit is used, or whether the byte has been configured as a byte, word, double word, or long.

As shown in the window below, the result information shows in what form the memory has been configured, and the corresponding form is marked with a check mark.



- This function is only available after the project has passed the compilation.

10.3.19 View cross-reference of variables/addresses

- This function is only available after the project has passed the compilation.

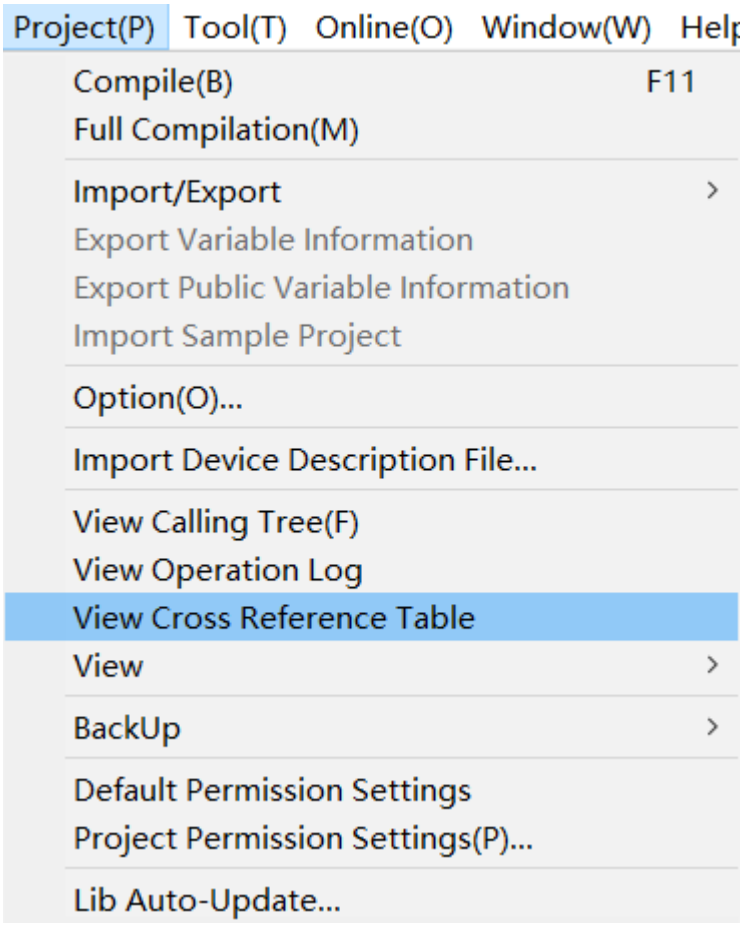
This software supports viewing cross-reference of variables/addresses and allows for quick navigation to

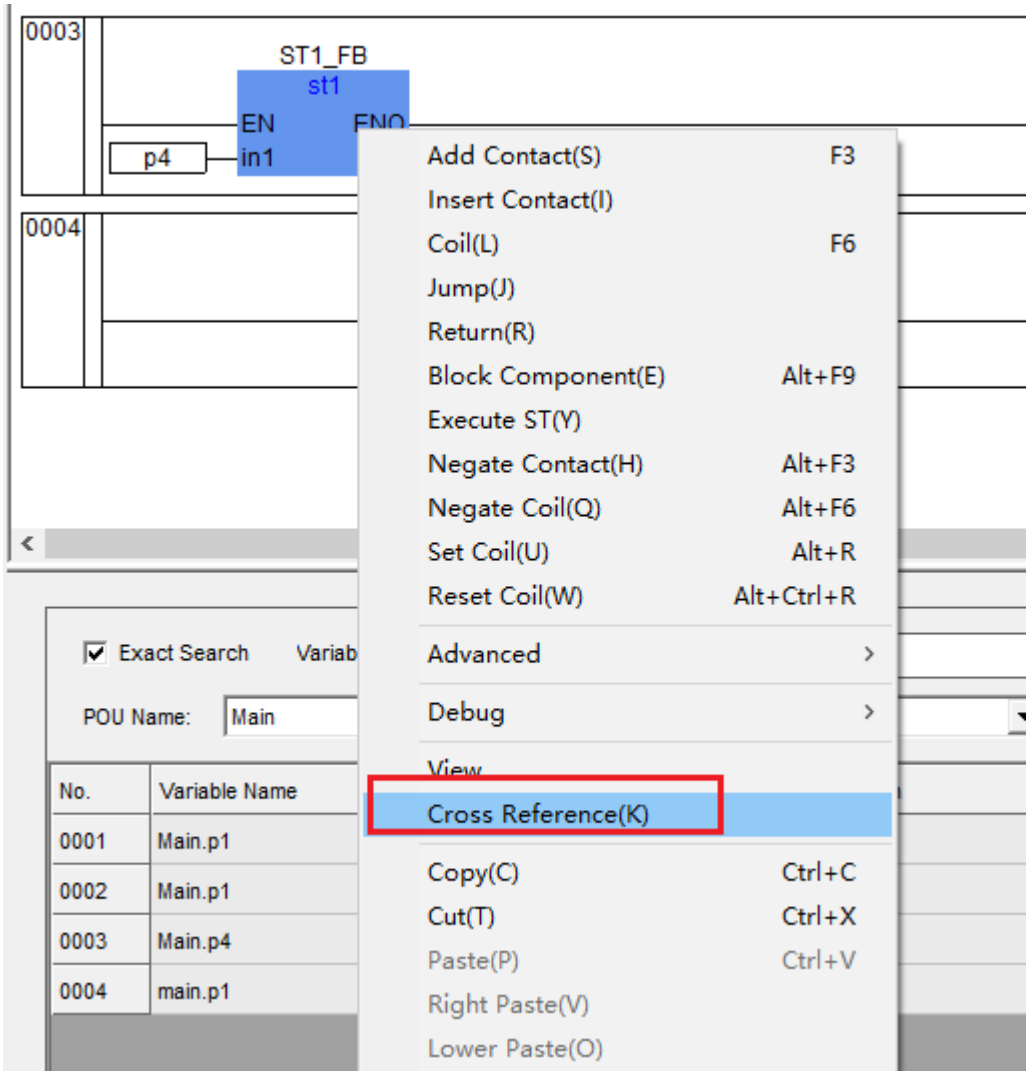
specific reference locations by double clicking in search results. During finding and jumping, the position of the located information is displayed according to the following principles:

- POU name: Display search results in the order of POU nodes from top to bottom in the Project Management Tree.
- Position in the POU: The sequence of variables within the same POU is in accordance with top-to-bottom and left-to-right, and can be accessed and displayed directly from the data in the cross-reference table. Since CFC language does not specify the sequence of top to bottom and left to right, the sequence of variables in CFC is determined by the sequence of component IDs (that is, the sequence in which the CFC components are added).

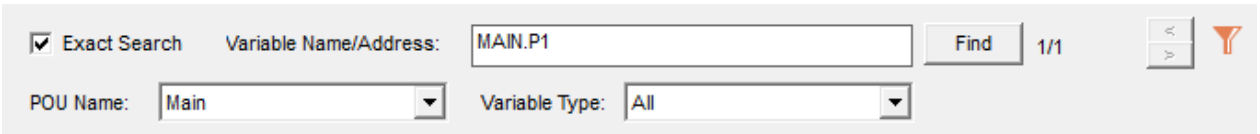
10.3.19.1 Steps

1. View the Cross Reference Table. This software supports opening the Cross Reference Table through the following two methods:
   - After successful compilation, select [Project] - [View Cross Reference Table].
   - After successful compilation, right click on the element in the program that is associated with a variable and select Cross Reference.


Project(P)
Tool(T)
Online(O)
Window(W)
Help
Compile(B) F11
Full Compilation(M)
Import/Export
Export Variable Information
Export Public Variable Information
Import Sample Project
Option(O)...
Import Device Description File...
View Calling Tree(F)
View Operation Log
View Cross Reference Table
View
BackUp
Default Permission Settings
Project Permission Settings(P)...
Lib Auto-Update...



2. In the "Cross Reference Table", use to expand or close the filtering criteria for "POU Name" and "Variable Type".



3. According to the actual situation, the difference between checking or unchecking "Exact Search" and whether it is checked is as follows:

   - checking "Exact Search": When searching for variable names or addresses, it is a full match search, meaning that the variable name must be the keyword being searched for in order to be considered a match; The first address is considered a match only when it overlaps with the searched address.

     For example, if searching for variable p1, only the variable named p1 is considered a match; Search for address% IW0, both address% IW0 and% IX0.0 are considered a match.

   - unchecking "Exact Search": When searching for variable names or addresses, it is a fuzzy search,

which means that if the variable name contains the search keyword, it is considered a match; If there is any overlap between the actual occupied address and the searched address, it is considered a match.

For example, if the variable p1 is searched, both the variable names p1 and p10 are considered to match; Search for address% IW0, address% IW0,% IX0.0,% IX0.1 are all considered matches.

4. Enter the variable name or address you want to view, select the POU name and variable type you want to search for from the drop-down menu, and click "Find".

   The variable name is not case sensitive

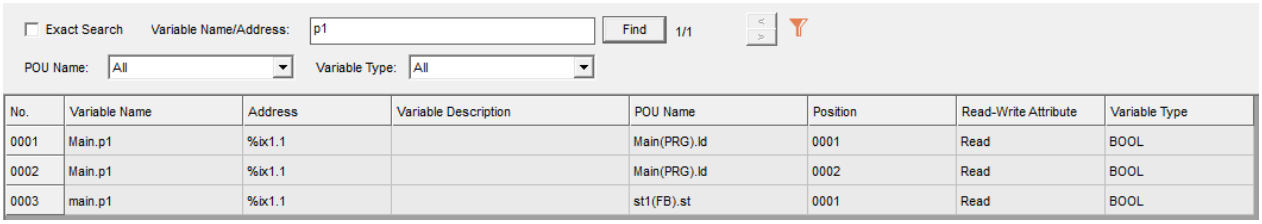
   Finds only the names of variables that have been successfully compiled

   Finds only variables that have been referenced by the user program

5. The software displays the query results as shown in the following figure.

   Click on the title names of each column to arrange them in ascending or descending order.

   When there are many search results, you can use  to flip through the pages, and a maximum of 1000 pieces of information can be displayed on a single page.



No.	Variable Name	Address	Variable Description	POU Name	Position	Read-Write Attribute	Variable Type
0001	Main.p1	%ix1.1		Main(PRG).ld	0001	Read	BOOL
0002	Main.p1	%ix1.1		Main(PRG).ld	0002	Read	BOOL
0003	main.p1	%ix1.1		st1(FB).st	0001	Read	BOOL

6. Double click on any query result to quickly navigate to the specific reference location. After jumping over, the system highlights the specific reference location.

10.3.20 Input Assistant

10.3.20.1 Introduction

Using the Input Assistant allows for the quick selection of required variable names or function block names to associate with elements or function blocks, as well as the import of operators and keywords of the ST language.

- Menu Bar: Click [Edi]t - [Input Assistant];
- Shortcut key: F2.

To use the assistant, in the editing area or watch list, select a variable or functional block name. Press F2. In the pop-up Help Manager, select the target variable or functional block. Import it to replace the original variable or functional block.

If a variable from one of the POUs in the PRG local variables is selected, return the text: POU name.local

variable name.

Users can also select the block component name AND, or select the variable name in the watch list to use the Input Assistant in the same way as above.

To use the Input Assistant in ST editing area, when users press F2, if the text is not selected, the text selected in the Help Manager window is appended where the current cursor is located; And if the text is selected, the selected text is replaced with the text selected in the Help Manager window.

List of Returned Texts

Reference Type	Returned Text
To reference a function defined in a library or a custom function in a project (library)	Returns as ???:=FUN(???, ???, ???), where FUN is the function name, ??? indicates the input parameters and return value of the function
To define the program in the project	Returns as PRG_POU()
To reference a functional block defined in a library or a custom functional block in a project (library)	Returns as FB1(in1:= ???, in2:= ???), where FB1 is the instance name, and in1 and in2 are the input and input/output pin names

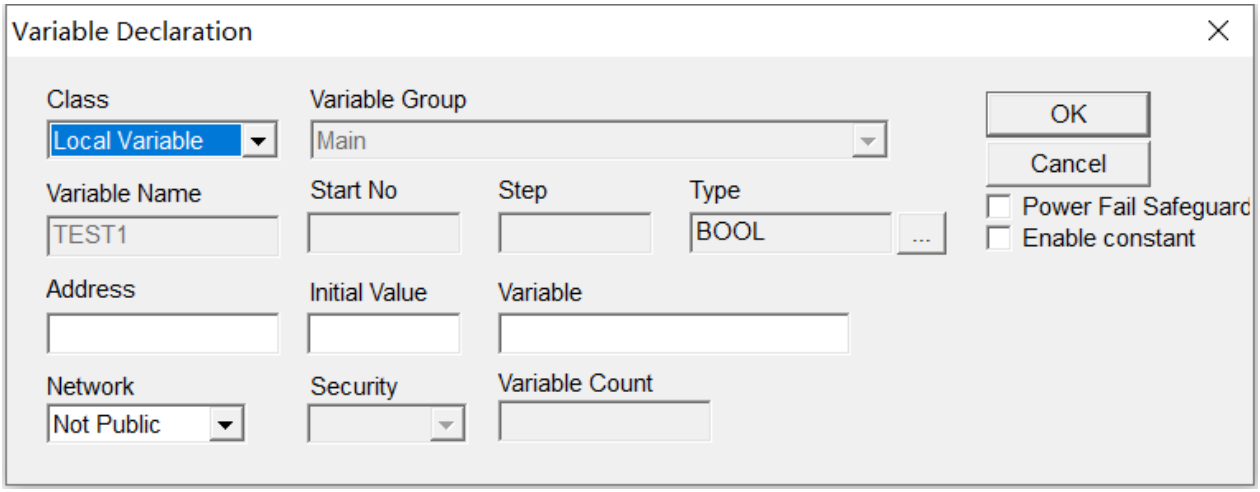
10.3.20.2 Variable properties

The Variable Properties menu item allows you to quickly display the properties window of the currently selected variable.

In the program editing area, select Variable Element or Variable Name, or select Convert in SFC. Users can view the variable properties via:

- Program area: Right-click the element or variable name. Click Variable Properties;
- Menu Bar: Click Edit > Variable Properties;
- Shortcut keys: Shift + F2.

The Variable Declaration dialog box corresponding to this variable is displayed, as shown in the figure. Users can modify Class, Type, Initial value, Network Public, Address, Variable description. When the variable category is Global Variable, the Variable Group List can be modified.



- Category: used to select local or global variable
- Variable group list: Used to select the variable group list
- Variable name: Used to enter the variable name
- Type: Used to select the data type
- Initial value: Used to enter the initial value
- Network disclosure: Used to select whether the variable is public or private
- Direct address: Used to enter the direct address of the variable
- Variable description: Used to enter the comment
- Security: Used to select the variable security property
- Power failure protection: Used to check the variable to enable power failure protection
- Constant or not: Used to check the variable as constant

10.3.21 Associate IEC Variables

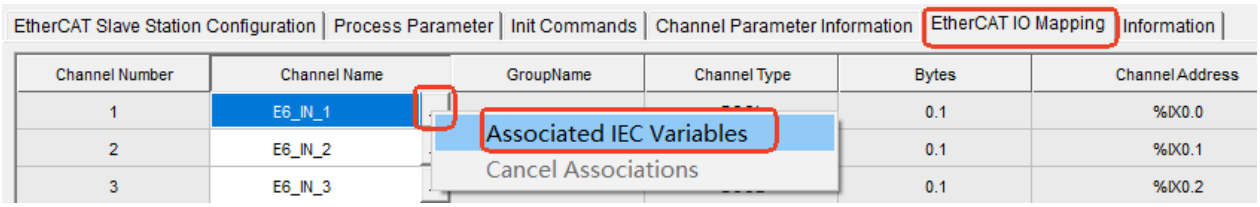
To associate IEC variables, you can map hardware configuration channel variables to PRG variables, network variables and global variables, including servo motors, frequency converters, IO, and other hardware modules. Only simple type variables are supported for mapping.

10.3.21.1 Steps

- To Associate IEC Variables

(1) Open the EtherCAT Slave Device - [EtherCAT IO Mapping] tab.

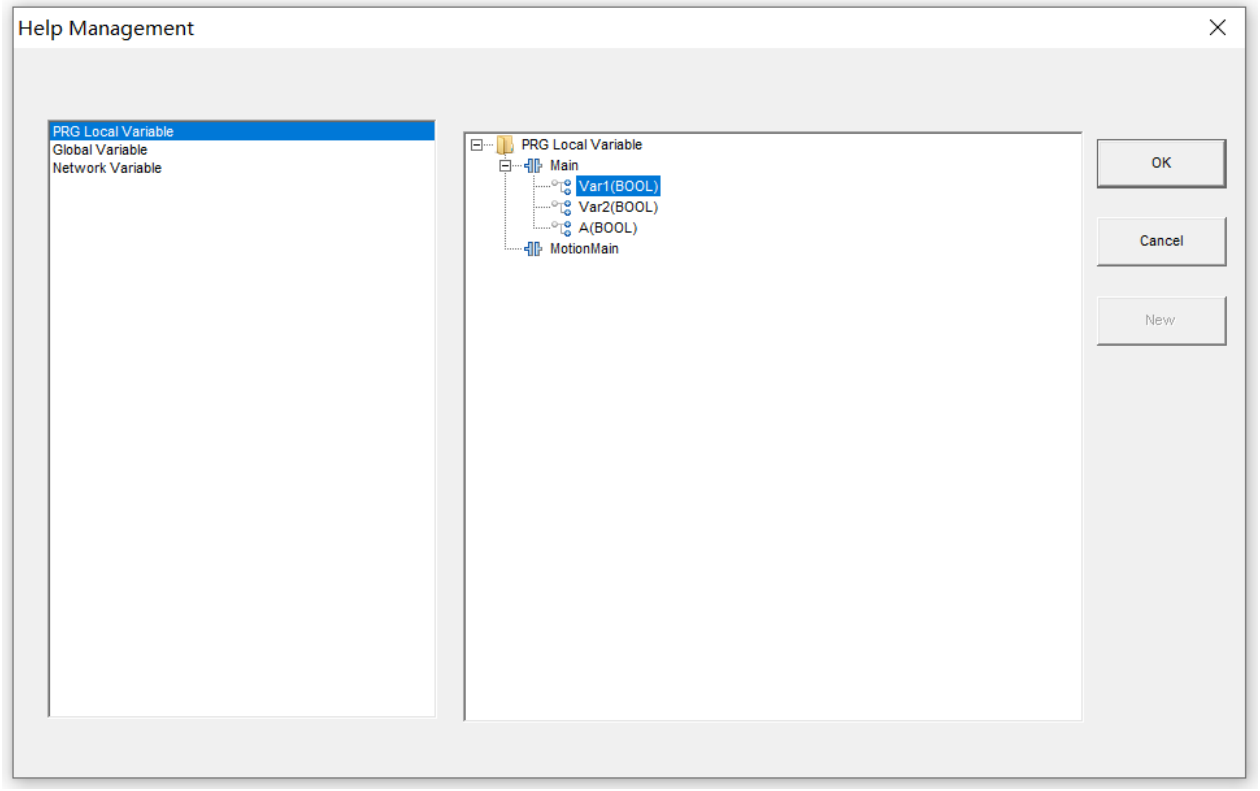
(2) In the "Channel Name" column, click the button ... and select "Associate IEC Variable." As shown in the figure below.



A "Help Management" dialog box will pop up.

(3) Select the variable to be associated and click OK.

The "Associate IEC Variables" dialog displays all available variables for association. Associated variables do not support direct address variables, and the variable type must match the type of the channel variable to be associated. You can also create new variables directly for association.



■ To Cancle Association:

□ In the "Channel Name" column, click the button ... and select "Cancle Association".

This will dissociate the IEC variable from the hardware configuration channel variable.

10.4 Instruction

In a programmable controller, commands and combinations of commands that enable the CPU to complete an operation or implement a function are called instructions. The set of instructions is called the instruction system. The instruction system is the bridge between programmable controller hardware and software, and is the basis for programmable controller programming.

AutoThink provides a wide range of instructions, which can be divided into transition instruction, comparison instruction, type conversion instruction, logic operation instruction, analog application, and other types according to their functions. For easy understanding and memorizing, we divide the algorithm library into six types: standard library, basic application library, system library, industry application library, fieldbus library, and product extension library. All three platforms contain the standard library, basic application library, and system library. For the industry application library and product extension library, the corresponding library files are loaded according to the selected target platform.

PLC instructions are implemented in the programming software in two ways: function and functional block. Both function and functional block are the program organization unit of AutoThink software, which are pre-programmed to implement a certain function. The functional block output can be one or more results. Each functional block instance has an associated identifier (that is, the instance name); While the function does not need an identifier and has only one output result (that is, the return value of the function). See POU types for specific concepts of function and functional block.

10.4.1 Instruction Library

The libraries are divided into six types according to the functions of the instruction codes in the libraries:

- Standard library: The instruction set specified in the IEC61131-3 standard, and the instruction set common to compilers.
- Basic application library: The general-purpose application library.
- Industry application library: The specific industry-related library, such as the motion control and chemical industry libraries.
- System Library: The general-purpose system interface instructions provided by OS, and the general-purpose PLC system interface instructions provided by RTS.
- Fieldbus library: The interfaces provided by various fieldbuses. When a controller that supports fieldbus is added, the instruction library is automatically displayed.
- Product extension library: The custom interface instructions provided by a specific product.

10.4.2 Call Instruction

10.4.2.1 Introduction

The instructions in the Library Manager are all enabled by default and can be manually configured by users as needed.

10.4.2.2 Requirement

No instruction implemented as a function needs to be declared when used.

Any instruction implemented as a functional block requires an instance name to be declared when used.

10.4.2.3 Steps

(1) Library Manager: Right-click an empty area, and click Library Configuration.

(2) In the pop-up Library Configuration dialog box, check the library files that need to be added. Click OK to add them to the Library Manager window.

10.5 Task Configuration

10.5.1 Create Task

10.5.1.1 Introduction

Under the "Task Configuration" node on the LX platform, there is a default "TASK1 (Task 1)." A program block is already called by default under this task, and additional program blocks can be called as needed.

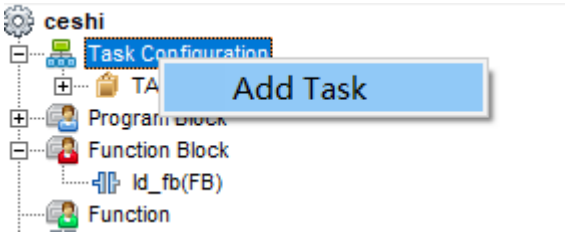
10.5.1.2 Requirement

The project is open.

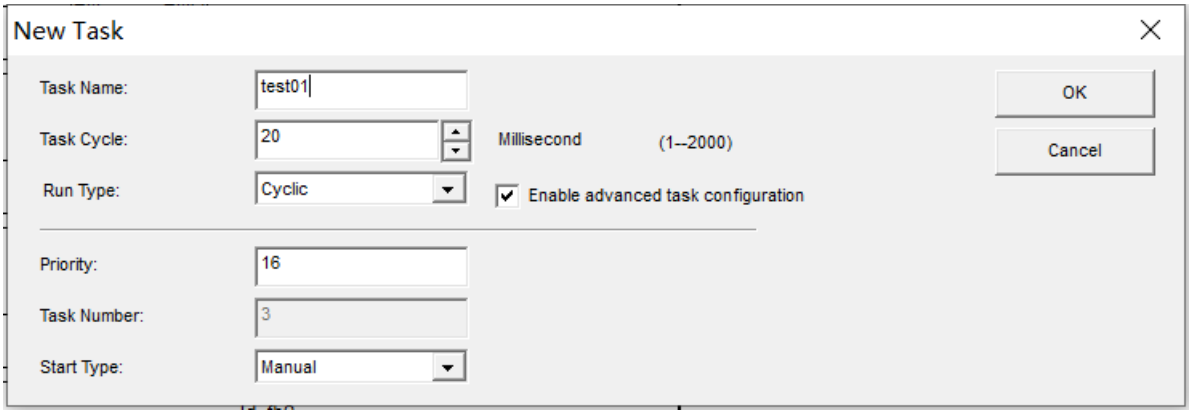
10.5.1.3 Steps

To create a new task, follow the steps below:

(1) In the Project Management Tree, right-click the Task Configuration node, and click Add Task.



(2) The Add Task dialog box is displayed as shown in the figure. Check the Launch advanced task configuration to configure the settings.

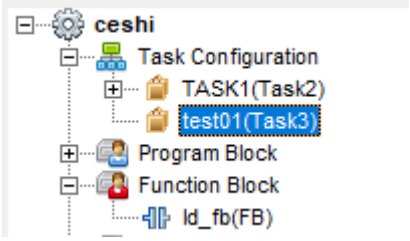


Add Task dialog box

- Task name: The name consists of numbers, letters, and underscores, and shall not start with a number. Its length shall not exceed 32 characters.
- Task cycle: The time interval at which tasks are executed periodically. This can be set when the Run type is selected as Cycle.
- Run type: Single, Cycle and Full Speed are available as options.
- Single: The task is executed only once.
- Cycle: By selecting Cycle as Run type, the LX controller executes tasks at regular intervals according to the set task cycle and priority.
- Full Speed: At full speed, the controller cycles through tasks according to priority.
- Priority: 1 - 16. The smaller the number, the higher the priority, supported by LX controller only.
- Task number: Task ID. Tasks are executed in the sequence of the task number. It cannot be edited and is automatically assigned when a new task is created.
- Startup type: Manual by default. Manual and Auto can be selected. If Manual is selected, the project enters the monitoring state after it is installed. Since the task is stopped, it is necessary to click Run manually to run the task. If Auto is selected, the project enters the monitoring state after it is installed, and the task runs automatically.

(3) After configuring the task parameters, click OK.

Add a task with the name TEST_01. Set the time slice to 3 and the startup mode to Auto. Run the task in a 10 ms cycle as shown in the figure.



10.5.2 Add Program Call

10.5.2.1 Introduction

All POUs need to be called under tasks to be executed. For the LX platform, the program MotionMain is added by default to the MotionTask task, and the program Main is added by default under the TASK1 task.

Instructions for the use of MotionTask tasks:

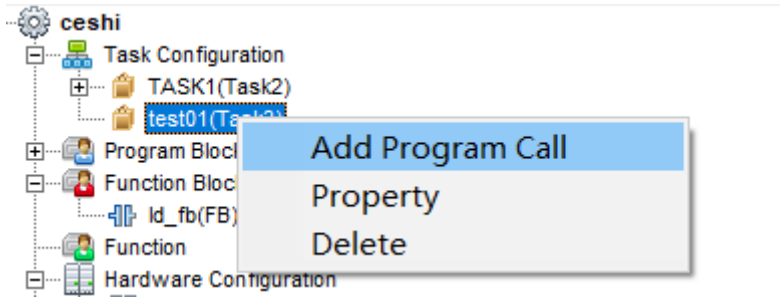
- It is not recommended to add programs under the MotionTask task, especially time-consuming programs or endless loop programs.
- Breakpoint commissioning and online tracking functions shall not be used for this task.
- It is not recommended to call blocking instructions, such as system library, industry application library, or fieldbus library, in this task.
- If the POU name under the Program Block node is modified, the POUs with the same name under the task nodes are updated synchronously.
- If the called POU under the Program Block node is deleted, the POUs with the same name under the task nodes are deleted at the same time.
- The POUs called in the task are executed in top-to-bottom sequence.

10.5.2.2 Requirement

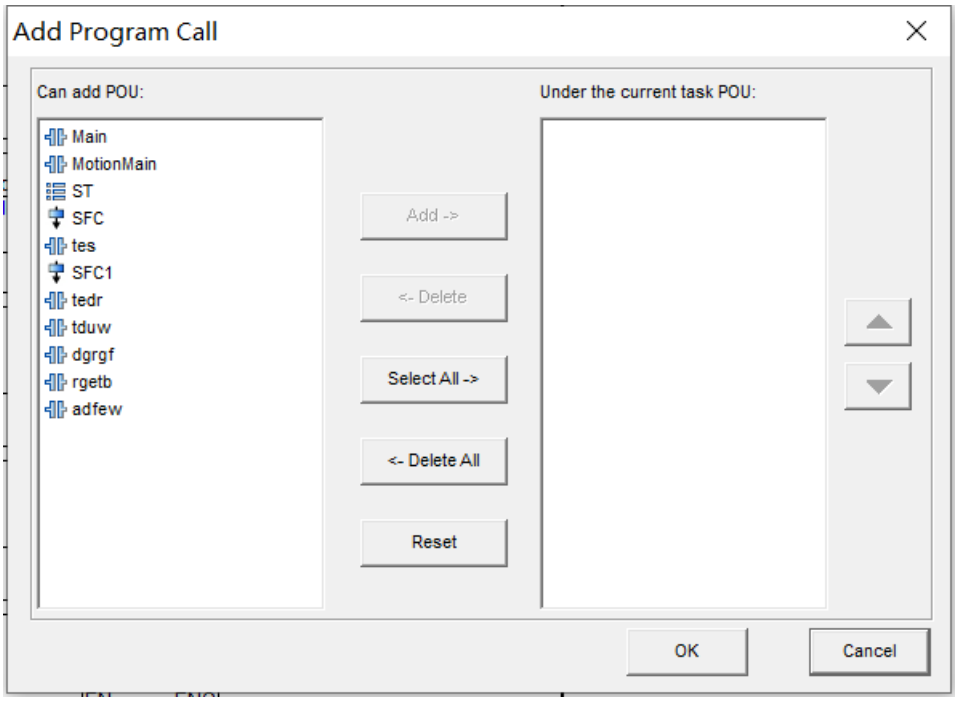
The POU that needs to be task-called has been configured.

10.5.2.3 Call via the task right-click menu

(1) Select the task node. In the right-click menu, select the Add Program Call command, as shown in the figure.



(2) The Add Program Call dialog box is displayed. All the POUs under the Program Block node are displayed in the addable POU list box. The POUs that have been called under the current task are displayed in the POU list box. As shown in the figure.

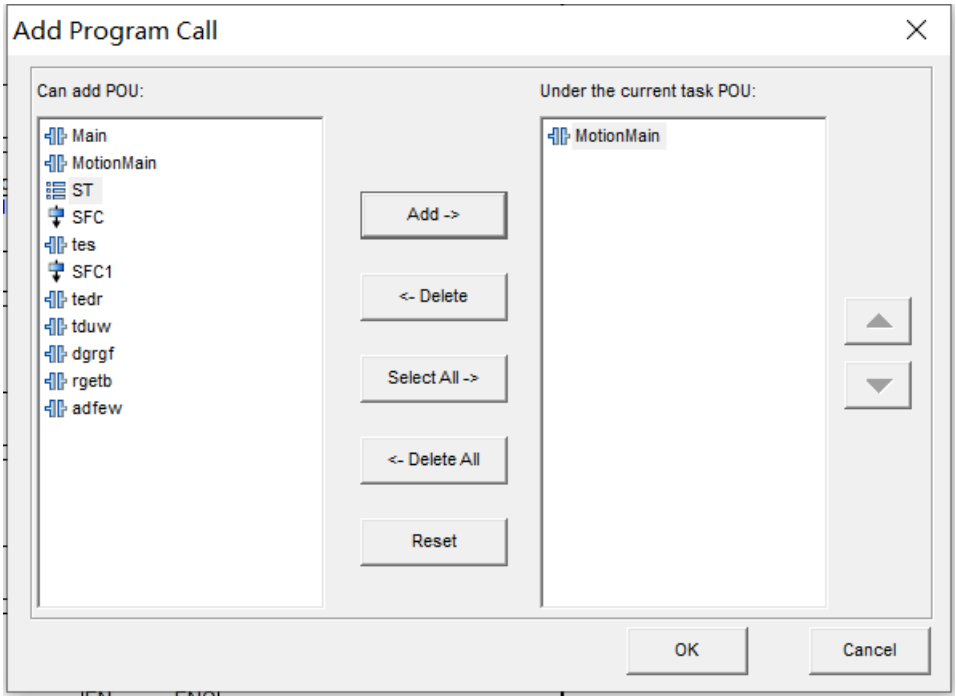


(3) Select the POU that needs to be called. Click Add to add the POU to the current task for calling.

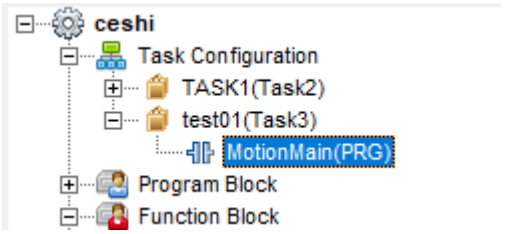
- After each addition, the cursor in the left list box automatically moves to the next POU position.
- In the right list box, select the POU. Click Delete to delete the call to the POU. When adding and deleting, users can use Shift or Ctrl to make multiple selections.
- Click Select All to add all addable POUs to the task for calling.
- Click Delete All to delete all called POUs under the task.
- Click Reset to restore the POU call to the state when the window was opened.
- Click ▲ and ▼ to adjust the sequence in which POUs are called in the right list box.

(4) Once editing is complete, click OK to close the Add Program Call dialog box.

(5) When there is no change in the POU called under the task, click OK, and a prompt box is displayed.

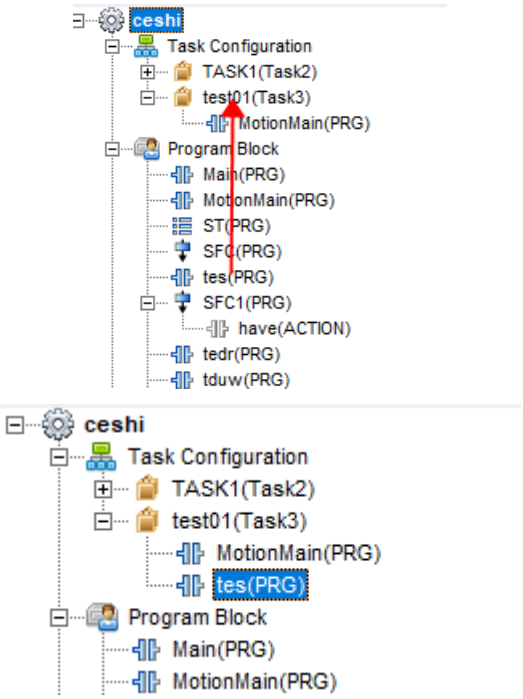


The called POU is generated under the task node, as shown in the figure.



10.5.2.4 Call via drag and drop

Under the Program Block node, select the POU LDprg. Drag and drop it to the task node for calling. A POU with the same name is generated under the task node, as shown in the figure. The called POU can be dragged and dropped to adjust the calling order.



10.5.3 Modify Task Properties

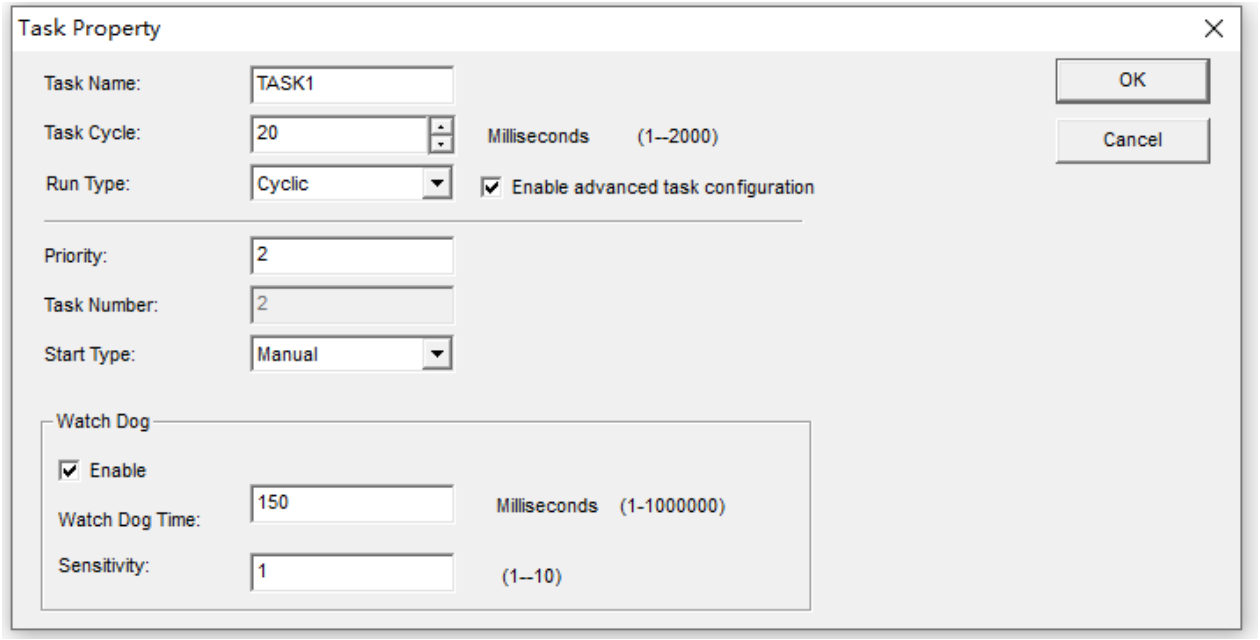
10.5.3.1 Introduction

The Task name, Run type, Priority, and Startup type can be modified in the Task Properties.

10.5.3.2 Steps

To modify the task properties, follow the steps below:

In the Project Management Tree, right-click the task node TEST_01 (Task 2), and click Properties.



See Create Task for various information on modifying task properties.

10.5.4 Running Task

10.5.4.1 Introduction

After the task is installed to the controller, the controller operation can be initiated by a key switch or a software button.

10.5.4.2 Requirement

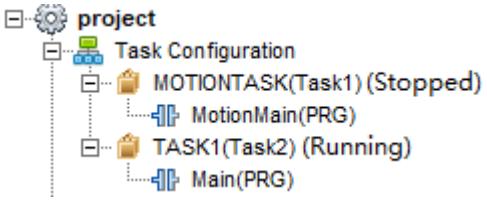
The project has been installed.

10.5.4.3 Steps

To run a task, follow the steps below:

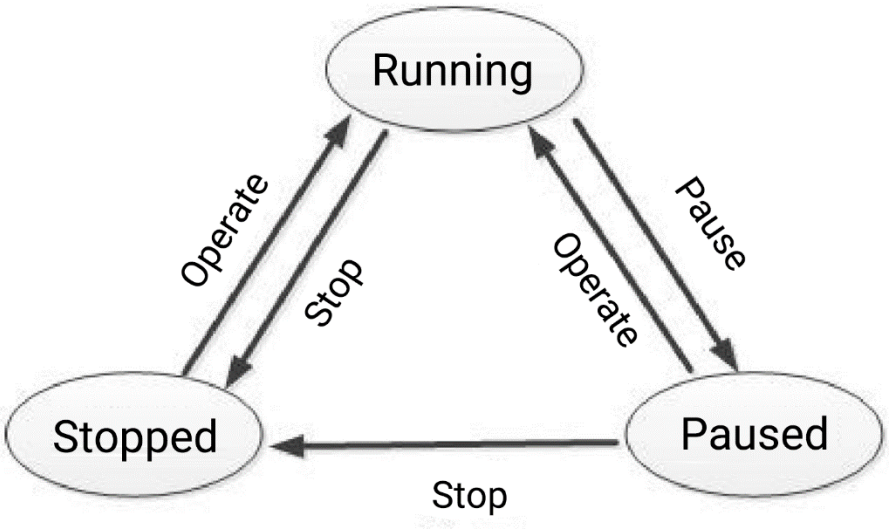
- Click the [Online] - [Monitor] command to enter the online state.

  If the task startup type is configured as Auto, the task runs automatically after entering the online state. The task status is displayed in real time on the task POU, as shown in the figure.



It can also be operated via the online operation buttons of the task, which are Run, Pause and Stop from left to right, valid for all tasks.

Click the Run, Stop, and Pause commands to put the task into the corresponding running state. The relationship between the commands and task status is shown in the figure.



10.5.5 Delete Task

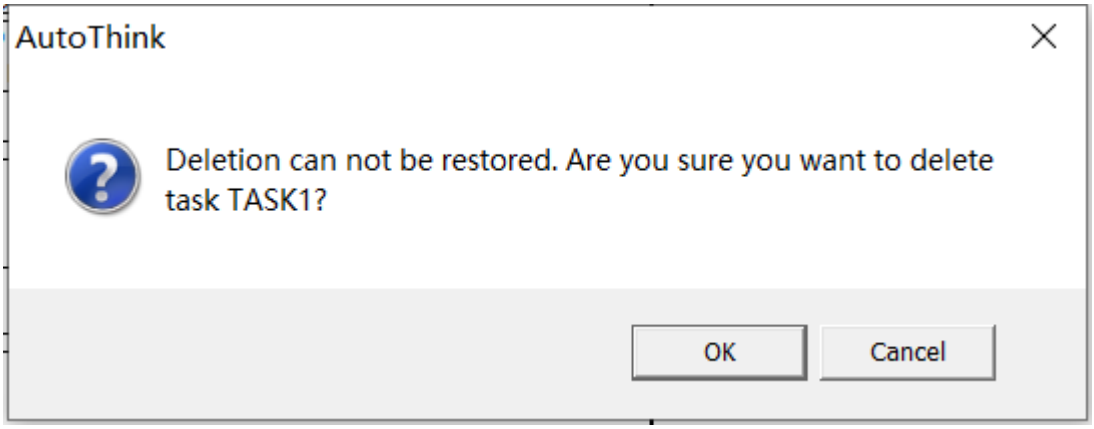
10.5.5.1 Introduction

It allows you to delete user-defined tasks.

10.5.5.2 Steps

To delete a task, follow the steps below:

(1) In the Project Management Tree, right-click the task node. Click Delete. A confirmation dialog box for deletion is displayed.



(2) Click OK to delete the task node TEST_01 and the POUs under it.

10.6 Program Organization Unit

10.6.1 POU Types

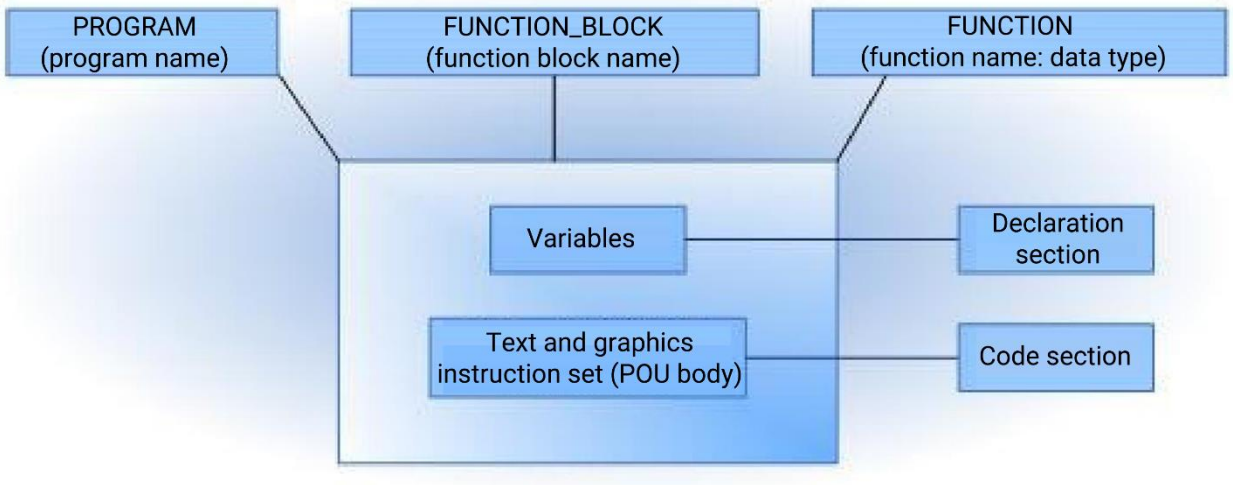
There are three types of POUs: program, functional block, and function.

Each POU consists of two parts: variable declaration and algorithmic programming.

Declaration part: The local variable declaration is made in the variable area of the POU editor for the definition of the variables.

Algorithm part: It is the main body of the POU, where four different languages, LD, ST, CFC, and SFC, are selected in the program area of the editor to perform algorithmic programming operations.

The schematic diagram of the POU structure is shown in the figure.



- Program

  A program is a sequence of statements, or a set of instructions, written to accomplish a task.

AutoThink software divides a program into subprograms (also called program blocks) and main programs (also called tasks). The former is used to implement the control logic in the field; And the latter is used to call the former and make it run.

- Functional block

  The functional block can produce one or more numerically executable logic elements. The functional block retains values for the next operation. After a set of data is input, different output data may be obtained.

  The functional block cannot run on its own; Instead, it shall be called by the program before it runs in the program.

- Function

  The function is the logical element that produces exact results. Unlike the functional block, the function value cannot be retained for the next evaluation.

  When a function is defined, the function shall receive a data type as a return value (to return data type). The calculation result of the function is assigned to the function itself. That is, the output variable of the function is the function name itself.

  The function is executed to produce an output result of a unique data type for a specific set of inputs. Compared to the functional block, the function has only one output result, without any internal conditions. In other words, given the same input parameters, a call to the function must yield the same operation result. Various mathematical operations that are normally used, such as SIN(X), are typical types of functions.

  In the following descriptions, the program POUs are referred to as programs, the functional block POUs are referred to as functional blocks, and the function POUs are referred to as functions.

10.6.2 Create POU

10.6.2.1 Introduction

AutoThink software divides POUs according to their types. However, the dialog box that opens when users add POUs is the same; And the operable contents in the dialog box may be different according to the types of POU. Up to 512 POUs can be added to a project.

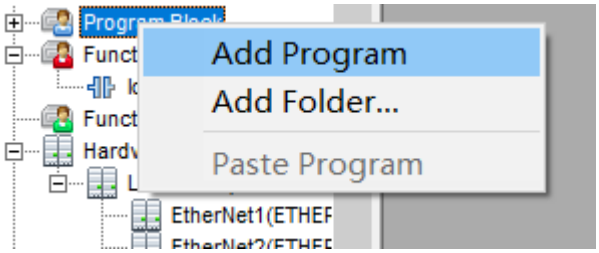
10.6.2.2 Requirement

The project is open.

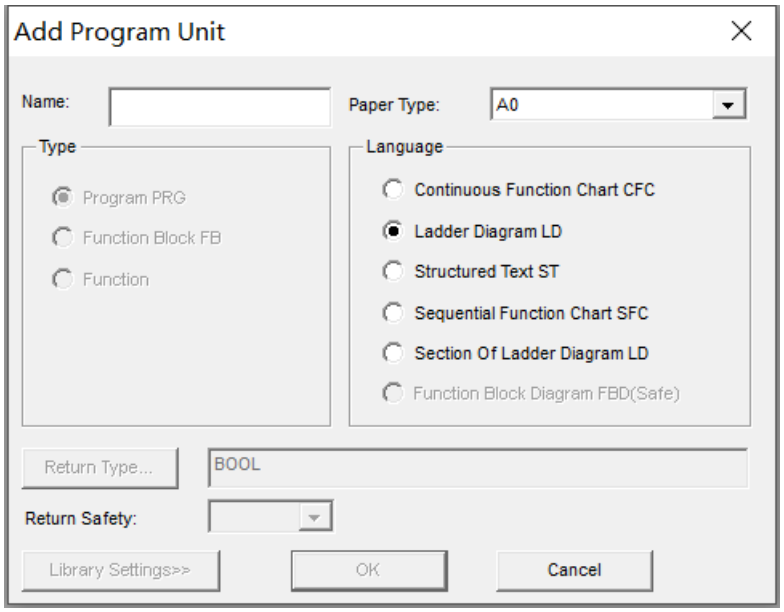
10.6.2.3 Steps

To add a POU, follow the steps below:

(1) In the Project Management Tree, right-click the Program Block node, and click Add Program.



(2) The Add Program Unit dialog box is displayed.



(3) Set POU-related properties.

- Name: Used to enter the POU name. if the name is not legal, the OK button is not available.
- The POU name shall only contain letters, numbers, and the underscore _. The first character shall be either a letter or an underscore, and shall follow the following principles:
- They shall not share the same name with variable name, variable group name, POU folder name, task name, project name, data type (custom or system default), keyword, instruction library name, or functional block name.
- The POU name shall not be null.
- The length of the POU name shall not exceed 32 bytes, and the part that exceeds the range cannot be entered.

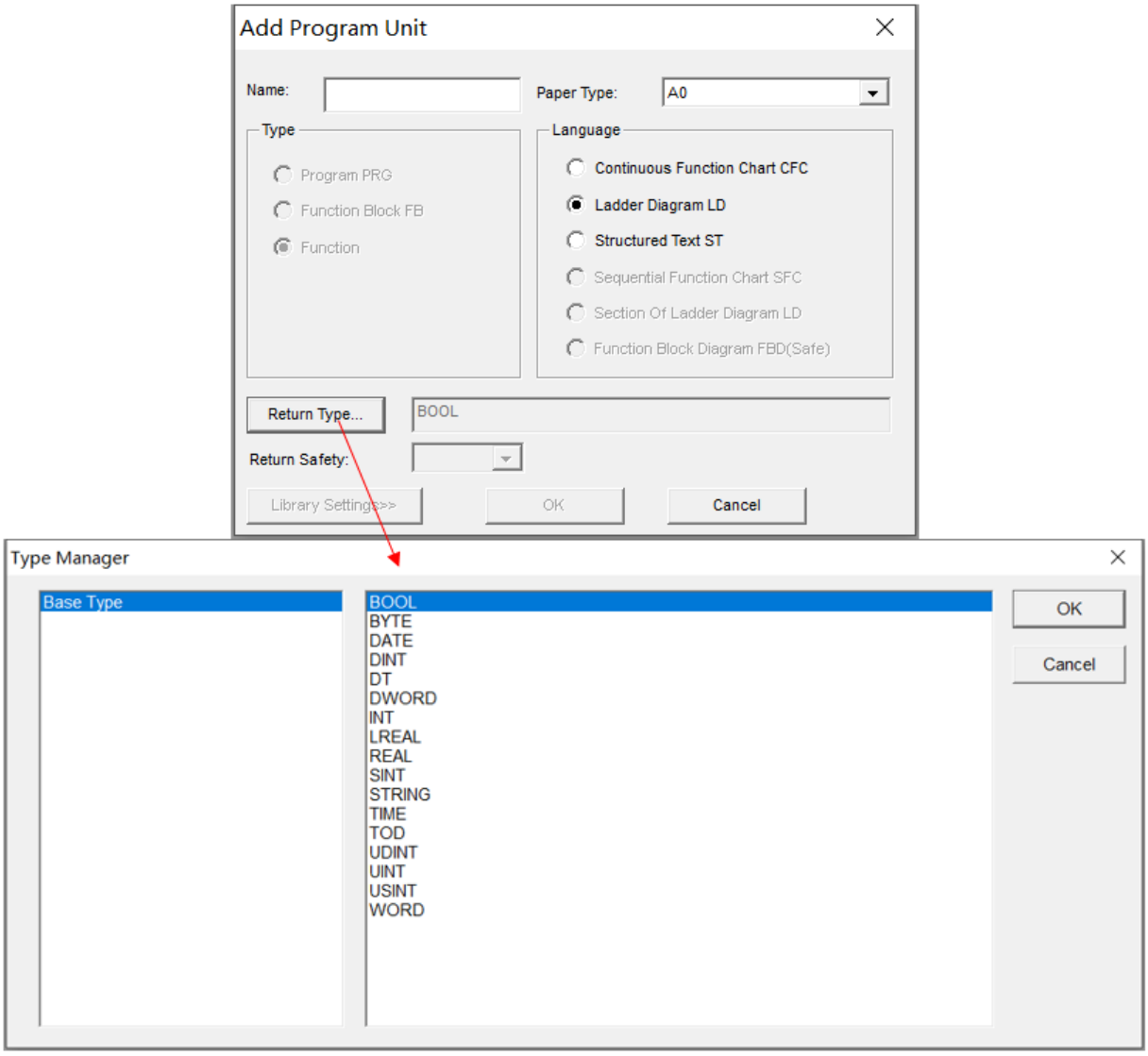
The POU name shall not be defined as a keyword reserved by Windows. The keywords include CON, PRN, AUX, NUL, COM0, COM1, COM2, COM3, COM4, COM5, COM6, COM7, COM8, COM9, LPT0, LPT1, LPT2, LPT3, LPT4, LPT5, LPT6, LPT7, LPT8, and LPT9.

- Paper type: Used to select the paper type, that is, the size of the open programming window. The default is A0, and A3, A4, B5, Ax, and A0 can be selected.
- Type: This option cannot be modified. The POU type is automatically defaulted according to

the POU node selected by users. See POU Types.

- Language: Used to select the programming language, with CFC (Continuous Function Chart), LD (Ladder Diagram), ST (Structured Text), SFC (Sequential Function Chart), and LDS (Ladder Diagram Segmented) available.
- Return type: The button is operable when the function POU is added.

(4) In the pop-up Type Manager dialog box, select the type of data returned by the function. Click OK to complete the type setting.



(5) Click OK to complete the addition of the POU. The POU is displayed as a sub-node under the corresponding node in the Project Management Tree.

10.6.3 Copy and Paste POU

10.6.3.1 Introduction

Copy a POU and paste it into the current project or another project.

There are custom functional blocks or function POU1s with the same name in the source and target projects. If the POU1 in the target project is deleted, and the POU1 in the source project is copied and pasted into the target project, the program where the POU1 of the target project is called is automatically updated. This function is only available for POUs in LD and CFC languages.

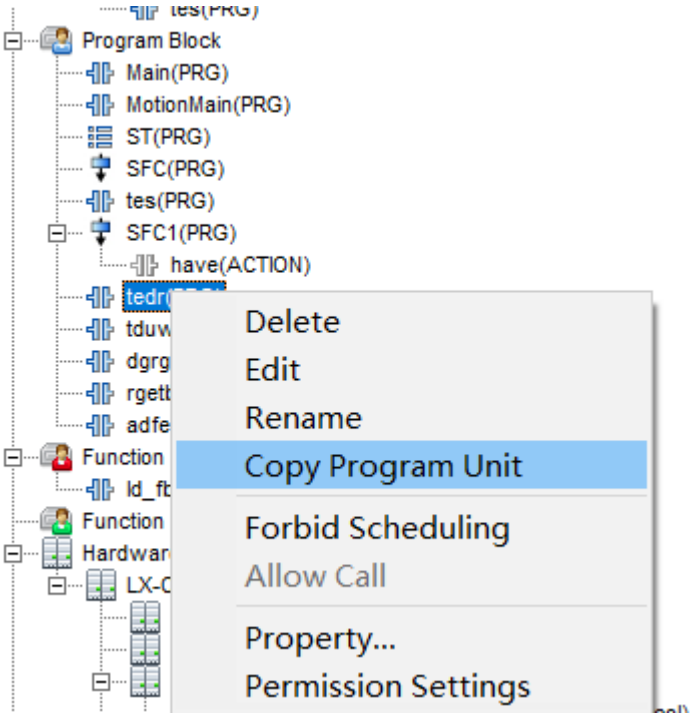
10.6.3.2 Requirement

The POU has been created.

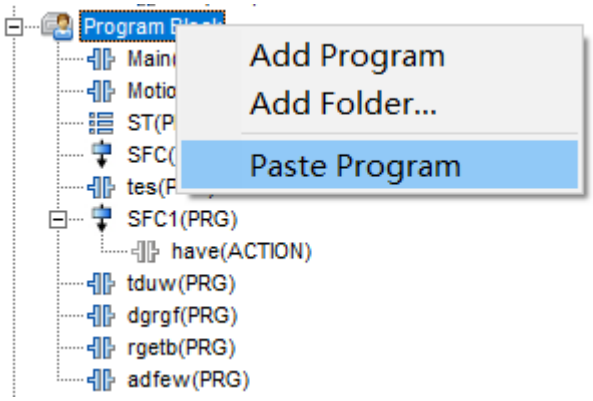
10.6.3.3 Steps

To copy and paste a POU, follow the steps below:

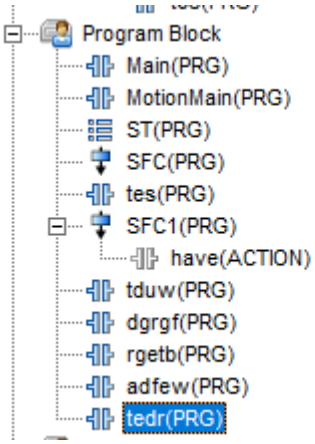
(1) In the Project Management Tree, right-click the POU name, and click Copy Program Unit.



(2) Right-click the target program block, and select Paste Program.



(3) The program is pasted under the selected program block.



10.6.4 Delete POU

10.6.4.1 Introduction

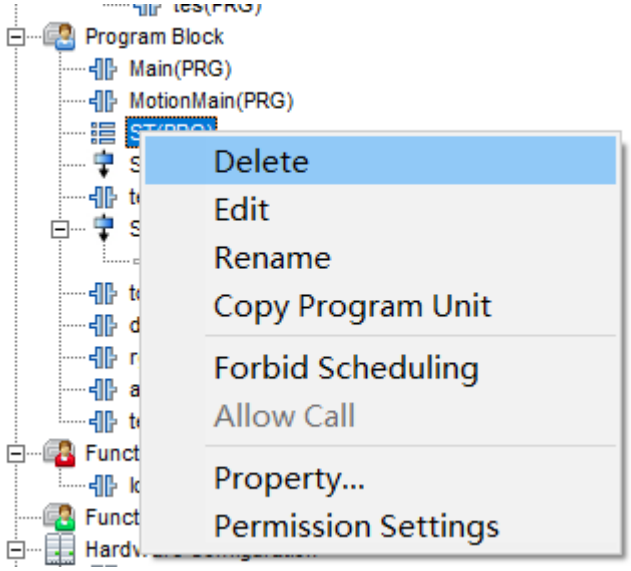
After the delete POU operation is executed, the POUs called under the task node are deleted together.

POU cannot be restored after deletion. Please proceed with caution.

10.6.4.2 Steps

To delete a POU, follow the steps below:

(1) In the Project Management Tree, right-click the POU name, and click Delete.



10.6.5 Rename POU

10.6.5.1 Introduction

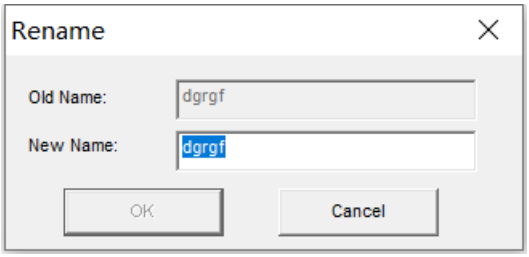
Rename the POU.

10.6.5.2 Steps

To rename a POU, follow the steps below:

(1) In the Project Management Tree, right-click the POU name, and click Rename.

(2) The Rename dialog box is displayed, as shown in the figure below.



(3) In the New name box, enter the new POU name. Click OK.

See the corresponding section of Create POU for naming rules.

10.6.6 Import POU

10.6.6.1 Introduction

If there is a need to reuse the POU program of other projects during project configuration, import the POU to the current project and open it for editing.

When importing, if there are POUs with the same name in the current project, the system prompts for overwriting.

The imported POU file is exported and generated by Export POU(X)....

Note the following points when importing POUs. Otherwise, the import may fail.

- The project supports a maximum of 512 POUs. Beyond this value, no POU can be imported.
- A POU has a maximum of 999 network nodes. Beyond this value, no network node can be imported.
- Only 100 POUs can be imported at a time.
- The POU with more than 32 elements in series or 16 elements in parallel on a network node cannot be imported.
- The imported POU name shall not be the same as the instruction name in the algorithm library, the folder name in the Project Management Tree, or the automatically generated POU name by the system (such as the name of a read/write subprogram for allocation methods).

10.6.6.2 Requirement

There is at least one importable POU.

10.6.6.3 Steps

To import a POU, follow the steps below:

(1) Click the Project menu. Select Import POU(T).... The Open window is displayed.

(2) Select the POU file that needs to be imported. Click Open to import it.

If there are POUs with the same name in the current project, the system prompts whether to perform the replacement of the POUs with the same name.

If the POUs with the same name being replaced have read-only or read/write permissions set, enter the password and replace them.

10.6.7 Export POU

10.6.7.1 Introduction

During project configuration, if there is a need to reuse the POU of another project, export the POU from the other project, and then import it into the current project.

The file is exported in .xml or .exp format. For the .exp format, only LD POUs can be exported.

Note the following points when exporting:

- Only POUs in ST and LD languages can be exported.
- If there are POUs with the same name under the export path, the system prompts whether to overwrite the POUs with the same name.
- By Import POU(T)... menu item, the exported POU can be re-imported into a project.
- The exported POU can be opened directly for editing.

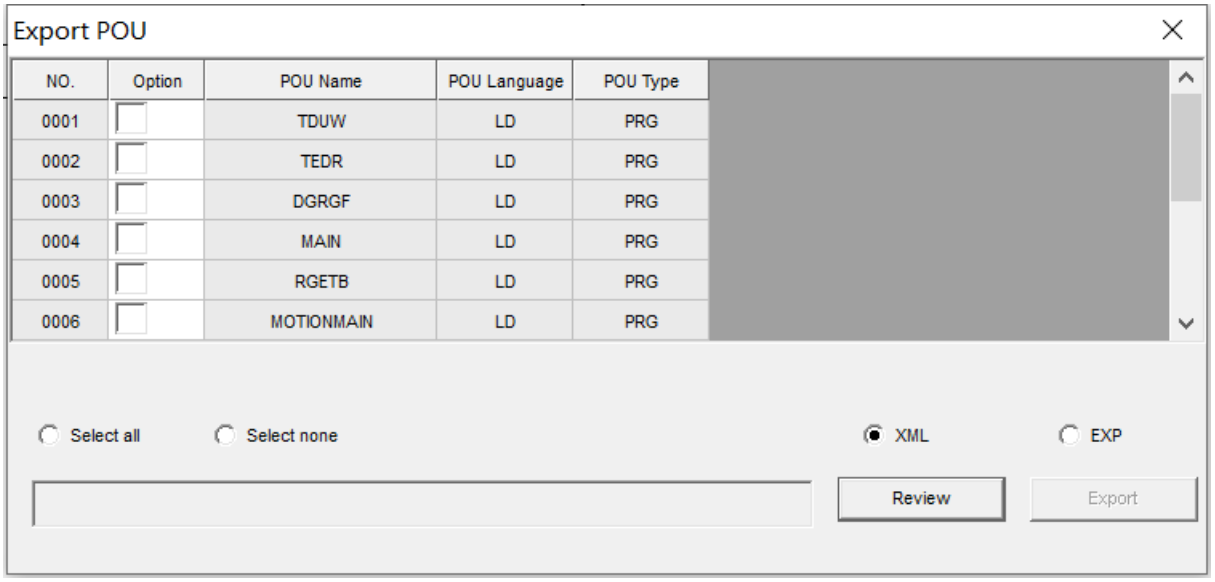
10.6.7.2 Requirement

At least one POU has been created.

10.6.7.3 Steps

To export a POU, follow the steps below:

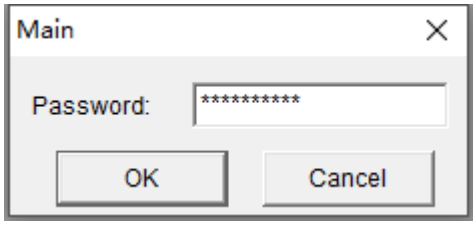
(1) Click the Project menu. Select Export POU(X).... The Export POU dialog box is displayed, as shown in the figure.



(2) Check the name of the POU that needs to be exported. Select the export format.

(3) Click Review. Select the path where the exported file is saved. Click Export. The POU is saved in the target path.

When the POU is set up with read-only or read/write permission, the correct permission password shall be entered before exporting.

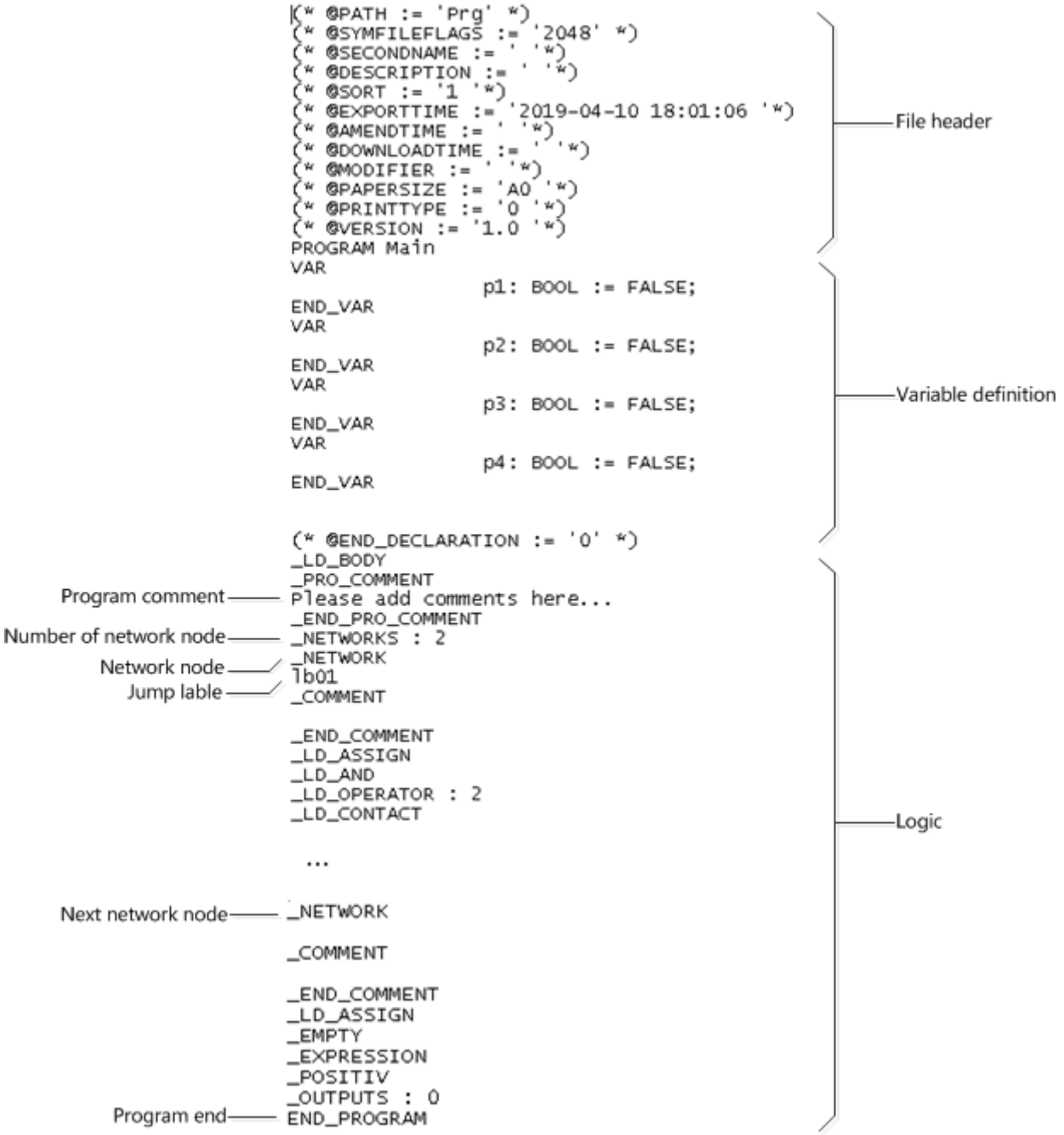


10.6.7.4 Instruction for importing and exporting in .exp format

Export and import the LD language POU in AutoThink.

Import the LD language POU in Codesys (2.3.4.1).

Instruction for the .exp Format File:



Instruction for the export structure of each element:

(1) Contact

_LD_CONTACT --Contact mark

p1 --Variable name

_EXPRESSION --Whether to negate

_POSITIV

(2) Out

- Coil

  _OUTPUT

  _POSITIV --Whether to negate and set

  _NO_SET

  out1

- Jump

  _OUTPUT

  _POSITIV

  _NO_SET

  _JUMP

  out2

- Return

  _OUTPUT

  _NEGATIV

  _SET

  _RET

  Return

- Negate, set, reset

  □ Negate

  _NEGATIV

  _NO_SET

  □ Set

  _POSITIV

  _SET

  □ Reset

  _NEGATIV

  _SET

(3) Block

- Standard library block component

  _OPERATOR --Block mark

  _BOX_EXPR : 2 -- Number of inputs

_ENABLED : 0 --EN is only visual

_OPERAND --Input mark

_EXPRESSION --Whether to negate

_POSITIV

p1 --Variable associated with pin

IN0 --Pin name

_OPERAND

_EXPRESSION

_POSITIV

p2

IN1

_EXPRESSION

_POSITIV

AND --Block name

_OUTPUTS : 1 --Number of outputs

_OUTPUT --Output pin

_POSITIV --Whether to negate and set

_NO_SET

p1 --Variable associated with pin

OUT --Pin name

■ Function

_FUNCTION --Function mark

_BOX_EXPR : 2

_ENABLED : 2

_OPERAND

_EXPRESSION

_POSITIV

p1

in1

_OPERAND

_EXPRESSION

_POSITIV

p2

in2

_EXPRESSION

_POSITIV

funTest --Function name

_OUTPUTS : 1

_OUTPUT

_POSITIV

_NO_SET

p1

funTest

- Function block

_FUNCTIONBLOCK --Function block mark

instanceFB1 --Function block instance

_BOX_EXPR : 1

_ENABLED : 2

_OPERAND

_EXPRESSION

_POSITIV

p1

in1

_EXPRESSION

_POSITIV

fb1 --Function block name

_OUTPUTS : 1

_OUTPUT

_POSITIV

_NO_SET

p2

out1

(4) Logic relationship

- Series logic

_LD_AND --AND

_LD_OPERATOR : 2 --2 operators

_LD_CONTACT

p1

_EXPRESSION

_POSITIV

_LD_CONTACT

p2

_EXPRESSION

_POSITIV

_EXPRESSION

_POSITIV

_EXPRESSION

_POSITIV

- Parallel logic

_LD_OR --OR

_LD_OPERATOR : 2 --2 operators

_LD_CONTACT

p1

_EXPRESSION

_POSITIV

_LD_CONTACT

p2

_EXPRESSION

_POSITIV

_EXPRESSION

_POSITIV

_EXPRESSION

_POSITIV

- Only one output element

_NETWORK

_COMMENT

_END_COMMENT

_LD_ASSIGN

_EXPRESSION

_POSITIV

_OUTPUTS : 1 --1 output

_OUTPUT

_POSITIV

_NO_SET

_JUMP

p1

END_PROGRAM

- Multiple output element

ENABLELIST : 3 --3 outputs

_LD_ASSIGN

_OUTPUT

_POSITIV

_NO_SET

p1

_EXPRESSION

_POSITIV

_LD_ASSIGN

_OUTPUT

_NEGATIV

_NO_SET

_JUMP

p2

_EXPRESSION

_POSITIV

_LD_ASSIGN

_OUTPUT

_NEGATIV

_SET

_RET

Return

_EXPRESSION

_POSITIV

ENABLELIST_END

(5) Instructions for .exp import

To import an .exp file generated by Codesys, note the following points:

1.AutoThink does not support the configuration that the output pin of the enable operator is inverted and set or reset. When the relevant configuration is imported, only the inverted state is retained, and the set or reset state does not take effect.

2.After Codesys enable operator is imported into AutoThink, the enable input and enable output pins are displayed.

3.Codesys output pin cannot be assigned. While the input pin is on the energy line and is cascaded or assigned.

4.The functional block input/output pin is imported as an input pin.

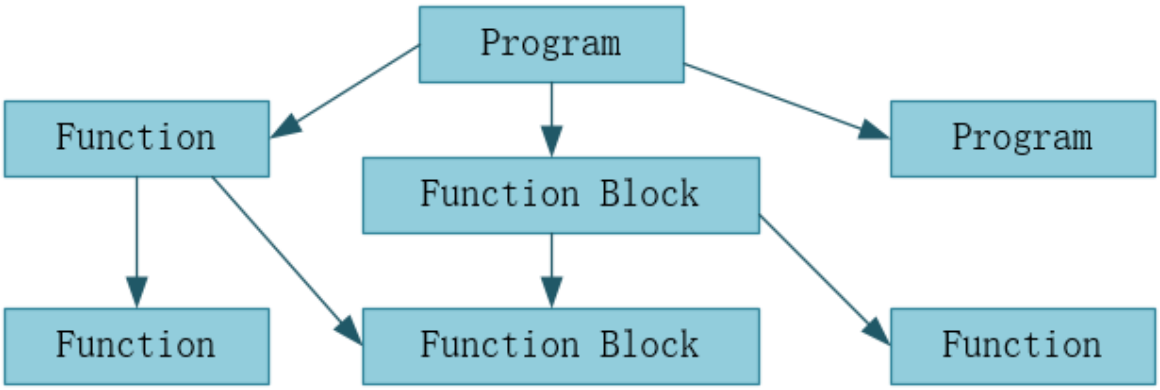
5.When there are POUs with the same name in the current project, if the import of those POUs fails, the original POUs are retained.

10.6.8 Call POU

10.6.8.1 POU calling principles

To call the POU, follow the following principles, as shown in the figure below.

- A program can call function, functional block, and other programs.
- A functional block can call function and other functional blocks.
- A function can call functional block or function.



10.6.8.2 POU calling methods

POU call: A POU (caller) calls another POU (callee) (recursive call is not allowed). The caller POU here can be any user-defined POU.

By default, when there are multiple called POUs under a caller POU, they are called in order from top to bottom.

Interrupt call: Used to call an interrupt program, enabling the system to respond to special internal or external events. When the system responds to an interrupt, it automatically saves the execution status of the current subprogram, and executes the interrupt program; And when the interrupt processing is completed, it automatically restores to the status before the interrupt of the subprogram.

10.6.8.3 Program calls another program

10.6.8.3.1 Introduction

Used to call another POU in a POU program.

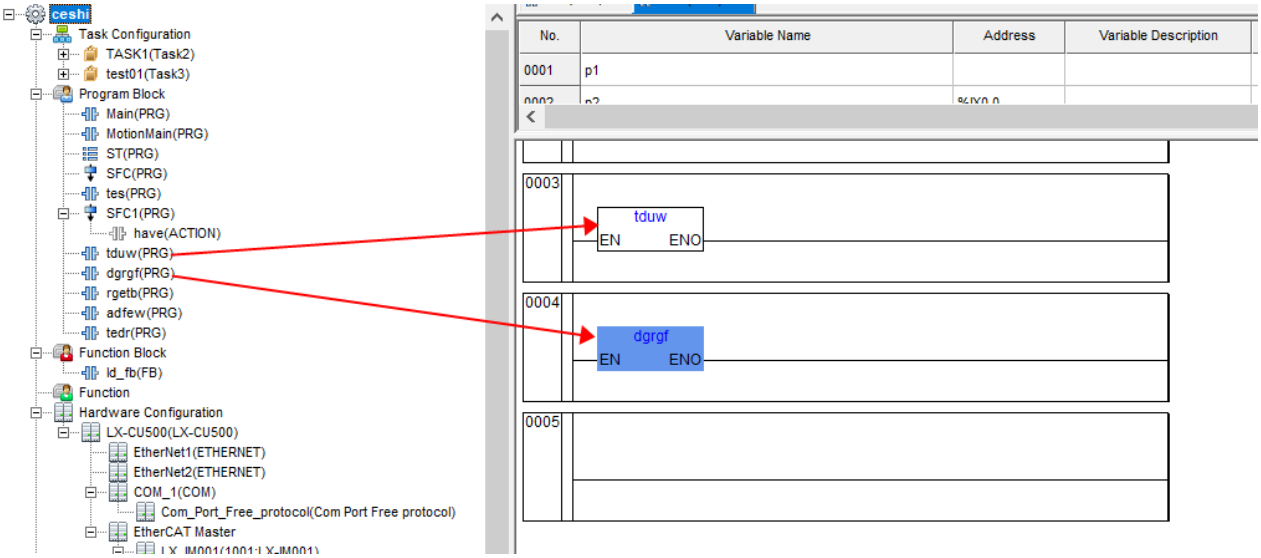
10.6.8.3.2 Requirement

There are two available POUs.

10.6.8.3.3 Steps

To make the program call another program, follow the steps below:

- Open the POU. Select the called POU, and drag and drop it to the program area of the POU.



10.6.8.4 Program calls functional block

10.6.8.4.1 Introduction

The program calls a functional block created by users themselves in the same way as it calls the standard functional block instruction provided by the system itself.

When a program calls a functional block, the instance declaration of the functional block shall be made first. After the instance declaration of the functional block, users can call the parameters in the functional block by Instance name.parameter name.

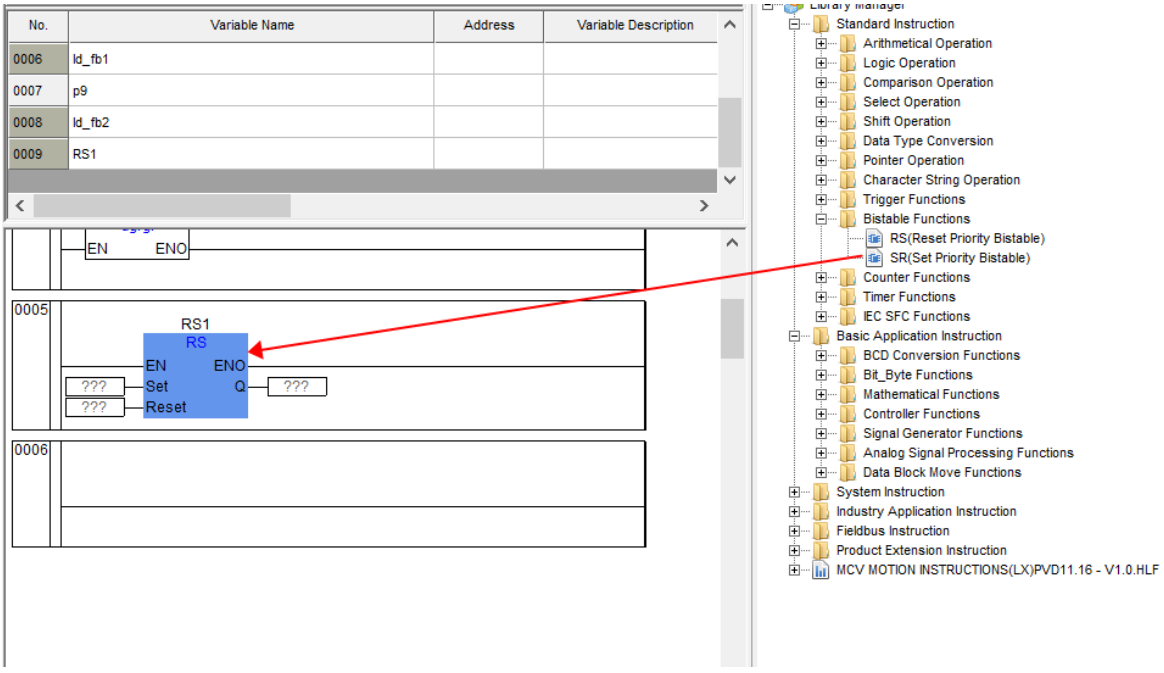
An instance is specific to a functional block. Each functional block instance is a separate, active object that accomplishes a specific logical function. Different programs and tasks can define and call application instances of functional blocks. Each calling instance occupies separate memory and retains separate logical state.

The functional blocks are called in the same way in LD, CFC, and SFC actions. Here is an example of LD.

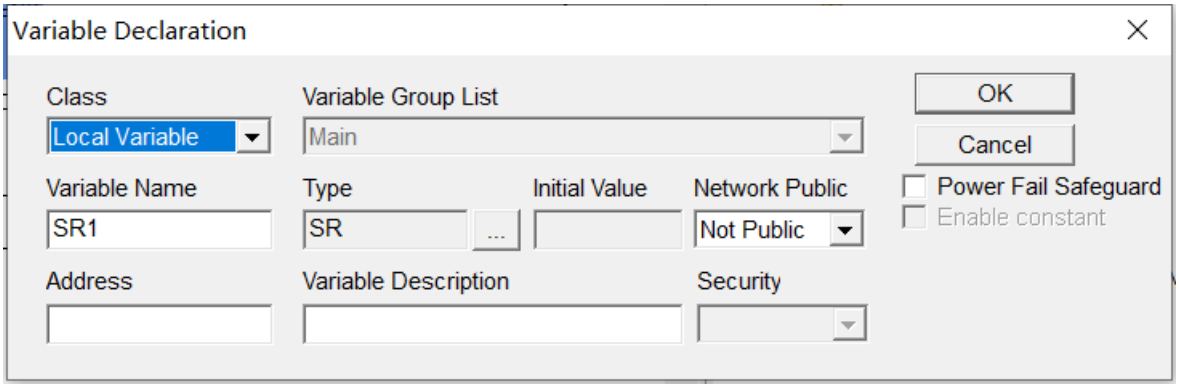
10.6.8.4.2 Call by drag and drop

1. Method 1:

   (1) Drag and drop a functional block from the Library Manager onto the program area node. As shown in the figure.

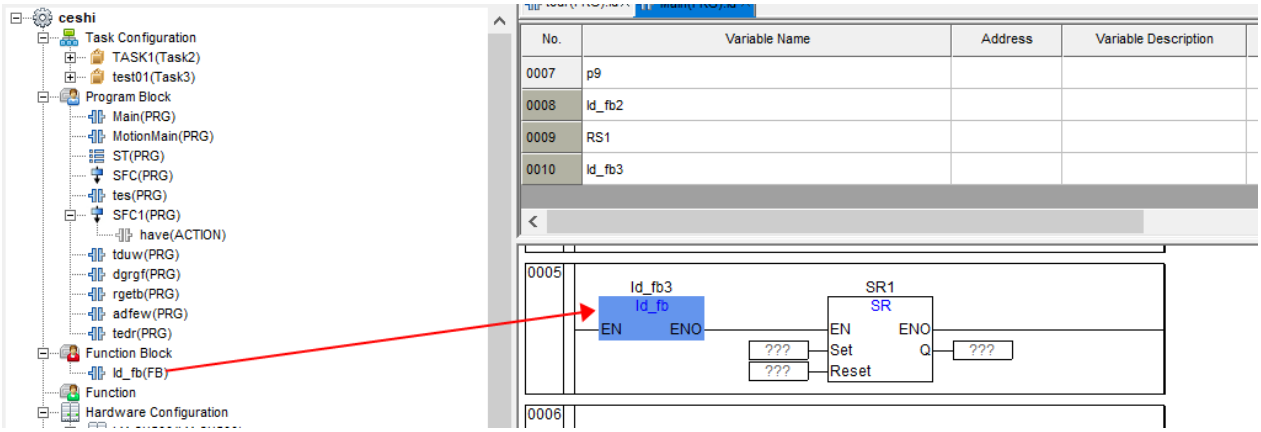


(2) The Variable Declaration dialog box is displayed. In the Variable Name, enter the variable name. Click OK to complete the instance declaration.



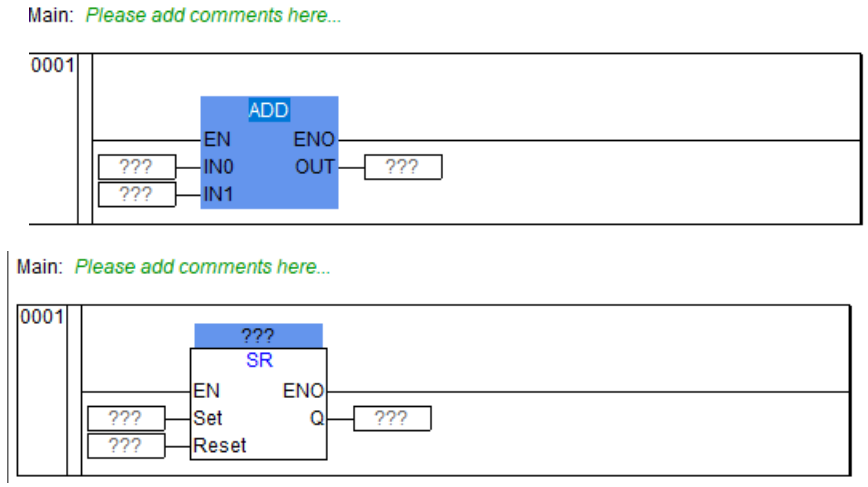
2. Method 2:

(1) Or directly drag the custom functional block FB1 under the Functional Block node to the selected section. The process is the same as above. As shown in the figure.



10.6.8.4.3 Method of adding a block element

(1) Right-click the program area. Select the Block Element. Modify the default AND to the type of functional block to be added, as shown in the figure.

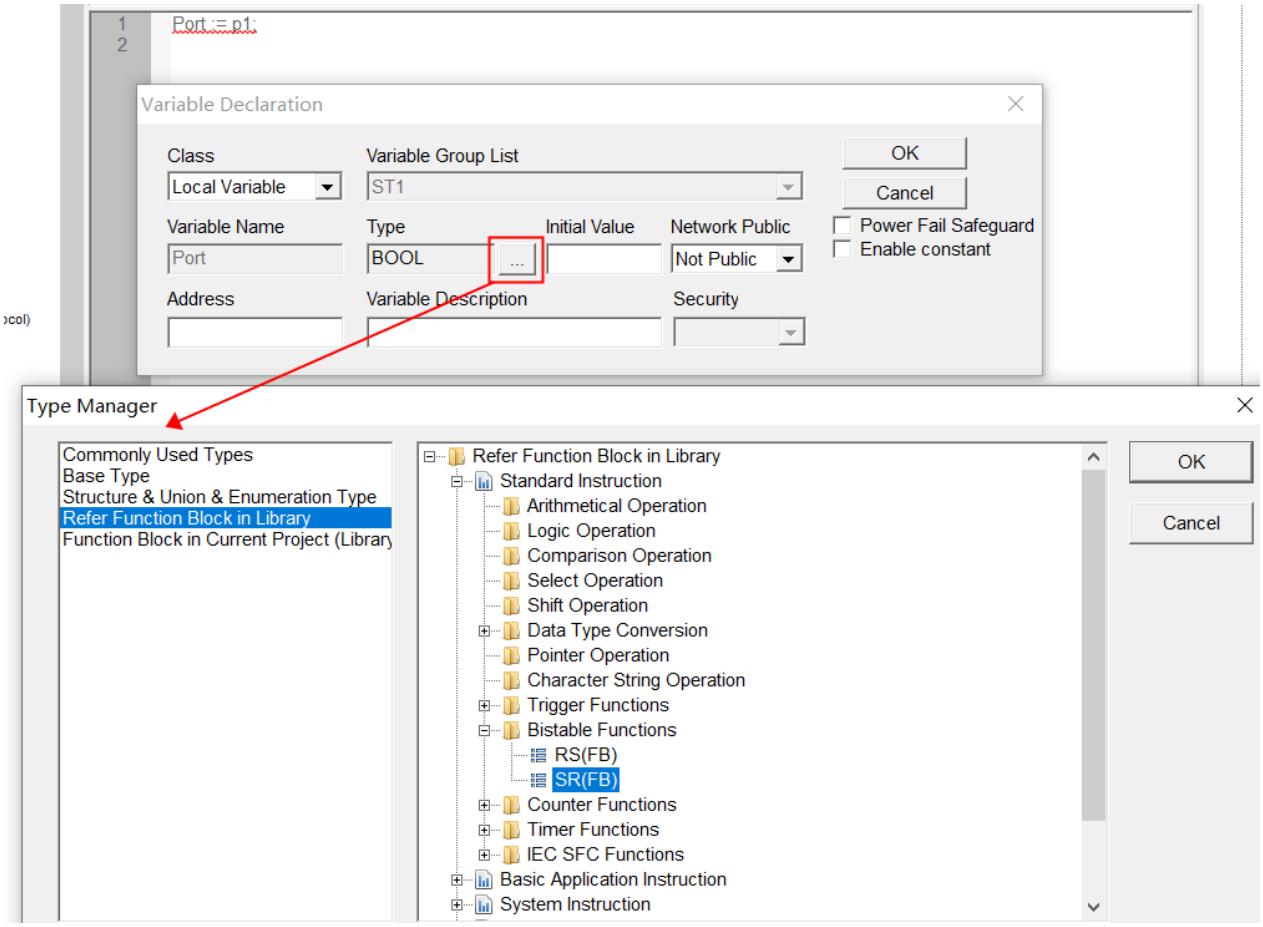


(2) Select the text ???. Enter the instance name. The Variable Declaration dialog box is displayed. Click OK to complete the instance declaration.

10.6.8.4.4 ST program calls functional block

The functional block can be called in ST via dragging and dropping, and adding by entering. Dragging and dropping is the same as LD, which drags the functional block in the Library Manager to the current cursor position in the ST Editor. Only the input addition method is described here.

- In the programming area, write the instance name of the functional block. Press Enter. The Variable Declaration dialog box is displayed.
- Click Type. In the Type Manager, select the functional block to be called. As shown in the figure.
- To add a functional block by manual entry, add a pair of parentheses after the functional block instance name, such as SRinst();.



Click OK.

10.6.8.5 Program calls function

10.6.8.5.1 Introduction

A program calls a function in the same way it calls a functional block, except that the function does not have an instance name. See the Program Calls Functional Block for the calling method.

10.6.8.6 View called program

10.6.8.6.1 Introduction

When a program or functional block is called in the POU, users can view the internal logic of the called program via the right-click menu.

Note that the functional block here refers to the user-defined functional block POU. For the functional blocks in the Library Manager, users can view the basic information about the functional block instructions.

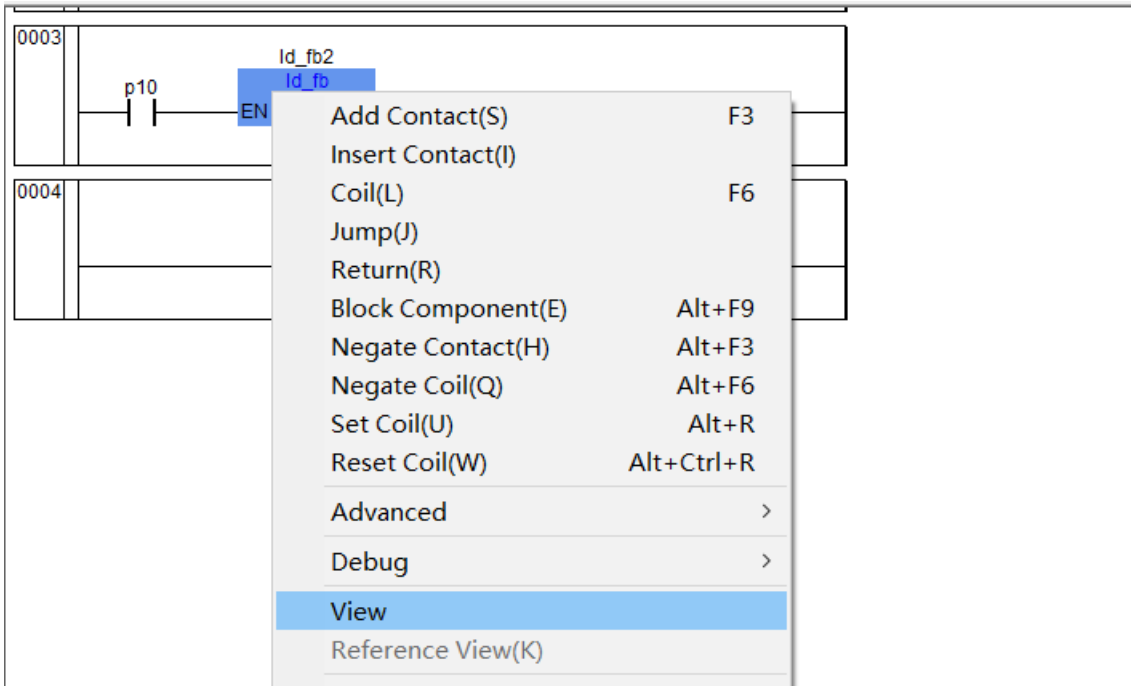
10.6.8.6.2 Requirement

The program or functional block has been called

10.6.8.6.3 Steps

To view a called program, follow the steps below:

(1) In the POU, right-click the called program. In the right-click menu, select the View command to open the program.



10.6.8.7 View called tree

10.6.8.7.1 Introduction

The called tree allows you to view the calling relationships in the current project, that is, the calling relationships between tasks and POUs, POUs, and library instructions, which can be expanded or closed layer by layer. The called tree can only be viewed after the project has compiled successfully.

10.6.8.7.2 Requirement

The project has passed the compilation

10.6.8.7.3 Steps

To view a called tree, follow the steps below:

(1) Click the Project menu. Select the View Called Tree command.

(2) By clicking ⊞ node, users can expand the called tree to display the called tree relationships.

10.6.9 Folder Management

10.6.9.1 Add folder

10.6.9.1.1 Introduction

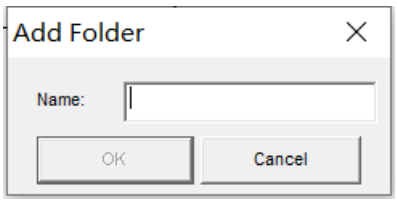
Users can classify and manage the added POUs in hierarchical folders for easy finding and operations.

A folder can be created under a program block, functional block and function node, or under a folder.

10.6.9.1.2 Steps

To add a folder, follow the steps below:

(1) In the Project Management Tree, right-click the POU node or folder. Click Add Folder.



(2) Enter the folder name, and click OK.

The folder name shall only contain letters, numbers, Chinese characters, and the underscore _. Its length shall not exceed 32 bytes.

10.6.9.2 Delete folder

10.6.9.2.1 Introduction

When a folder is deleted, all POUs and folders under the folder are deleted.

10.6.9.2.2 Steps

To delete a folder, follow the steps below:

(1) In the Project Management Tree, right-click the folder name, and click Delete.

10.6.9.3 Rename folder

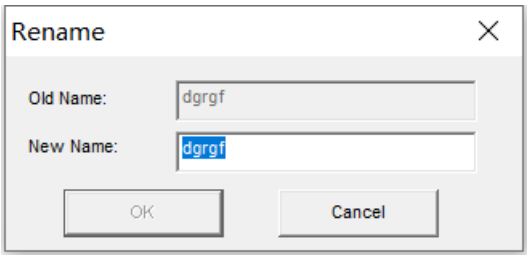
10.6.9.3.1 Introduction

Rename the folder.

10.6.9.3.2 Steps

To rename a folder, follow the steps below:

(1) In the Project Management Tree, right-click the folder name. Click Rename. The Rename dialog box is displayed.



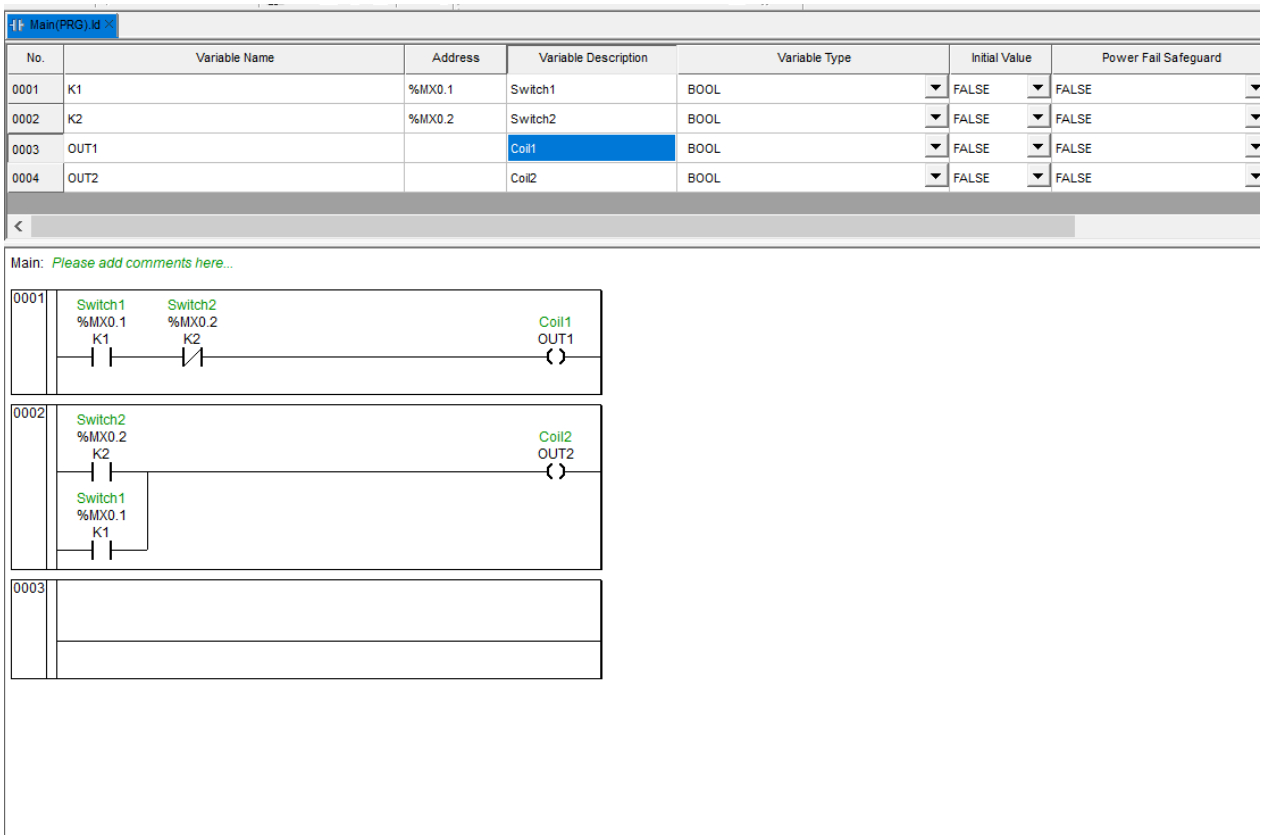
(2) Enter the new folder name. Click OK to modify the folder name.

10.7 Create LD Program

10.7.1 LD Programming Overview

LD is short for Ladder Diagram, a graphical programming language.

The LD consists mainly of programming elements such as contacts, coils, functional blocks, and connecting lines, which are connected by horizontal and vertical lines to form a planar mesh diagram, as shown in the figure below.



The leftmost vertical line of the section is generally called the Energy Line. And its state is always TRUE. Each programming element is connected to each other by certain rules, and finally connected to this energy line, forming a section, segment, or network, which accomplishes a specific logic operation. The line on each section is called a busbar, where contact, coil, and functional block can be added.

10.7.2 LD Elements

10.7.2.1 Section

A section is the basic unit of an LD program. Each POU written in LD language consists of sections. The section number defaults from 0001. Up to 999 sections can be added to each POU.

- No more than 64 columns horizontally and 64 rows vertically shall be included in the network of each section of the LD language programming area.

10.7.2.2 Contact

Each contact represents a BOOL variable, which can be a channel input variable such as a switch or button, or an internal variable. The contact passes the value from left to right.

┤ ├: Normally open contact. When the variable value is TRUE, the contact is turned on and the energy flow is transmitted to the right. Otherwise, the energy flow is broken at the contact.

10.7.2.3 Coil

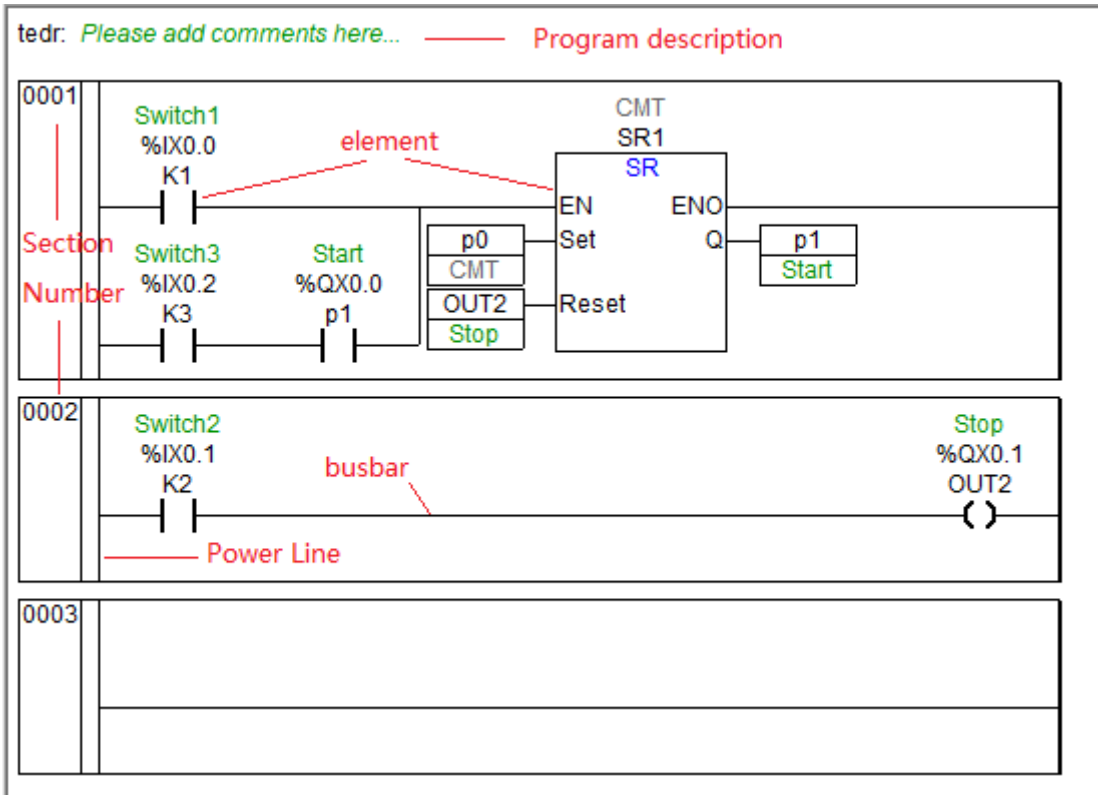
The coil represents the output of a logic operation, denoted by (). It transmits the energy flow from left to right, which can be connected to the channel output variable to control the start and stop of equipment, the start, stop and shutdown of the motor, etc.

10.7.2.4 Block elements

The block elements include three types: functional block, function, and program. See POU Types for a detailed description.

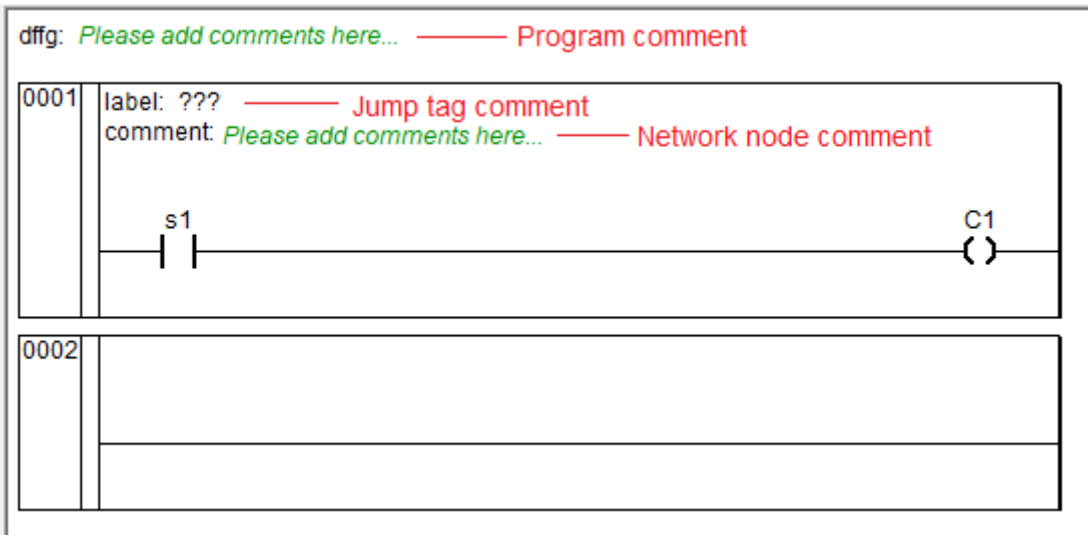
10.7.2.5 Legend description

The LD program consists of a series of sections. The power lines on the left and right, and the elements on the middle busbar form the entire program network.



10.7.2.6 Comment description

- Program comment: Used to describe the POU program, such as Pump start/stop.
- Jump tag comment: Used to identify the destination program segment of a jump. It is not shown by default. Users need to add Jump Codename to show it. Up to 32 characters can be displayed.
- Network node comment: Used to describe the current section of the program. It is not shown by default. Users need to select the Modify Comment command in the right-click menu of the node to show the comment area.

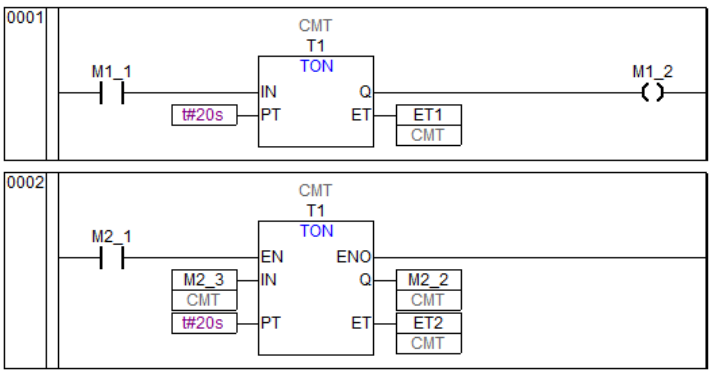


10.7.2.7 Functional block and enable operator

The fundamental difference between a functional block and an enable operator is the way in which they are executed. For a functional block, it is executed when the program is running, regardless of whether it is enabled or not. However, for an enable operator, it is executed only when the enable terminal EN is valid. A simple program written in LD language is described here.

No.	Variable Name	Address	Variable Description	Variable Type	Initial Value
0003	T1			TON	
0004	ET1			TIME	T#0MS
0005	M2_1			BOOL	FALSE
0006	M2_3			BOOL	FALSE
0007	M2_2			BOOL	FALSE
0008	TON1			TON	
0009	ET2			TIME	T#0MS

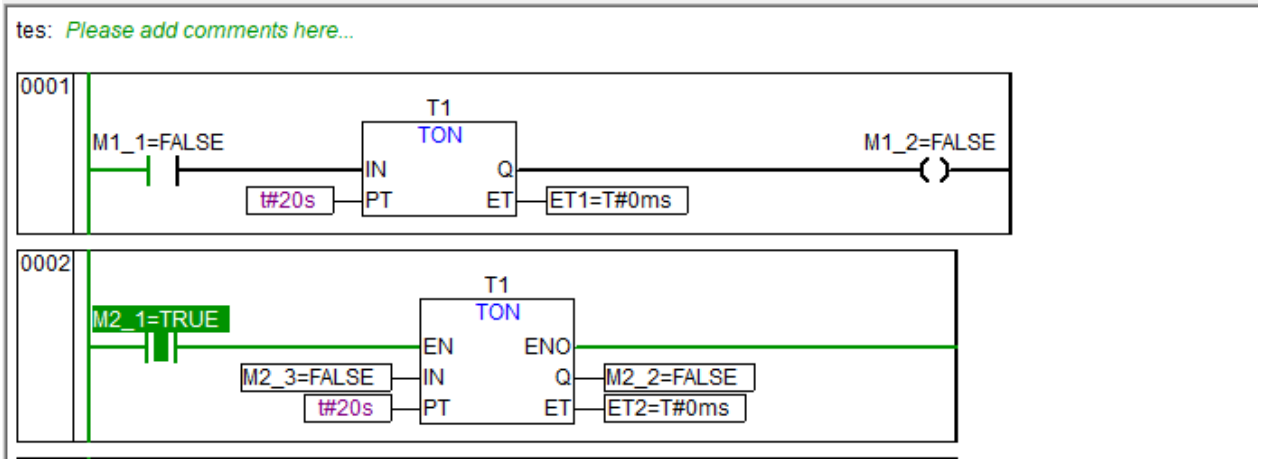
tes: Please add comments here...



As shown in the figure, the power-on delay timers are called in sections 0001 and 0002 with a functional block and an enable operator, respectively.

When starting to run the program, first turn on M1_1, M2_1, and M2_3 at the same time. At this time, the two timers are running in the same state. Disconnect M1_1 and M2_1, and keep M2_3 turned on. At this time, ET1 is cleared, while ET2 is not cleared.

Then disconnect M2_3. The ET2 time is still not cleared. Only when M2_1 is kept turned on and M2_3 is detected as disconnected, ET2 is cleared. This is because only when the enable terminal is valid, the code inside the TON is executed.



To use the enable operator to call the subprogram, when the enable terminal is invalid, the subprogram is not executed. To use the functional block to call the subprogram, when the functional block is enabled, the enabled part is run; And when the functional block is not enabled, the non-enabled branch is run.

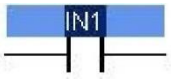
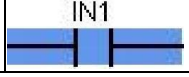
10.7.3 Select Element

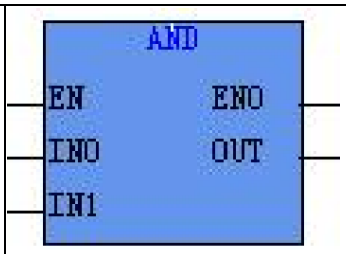
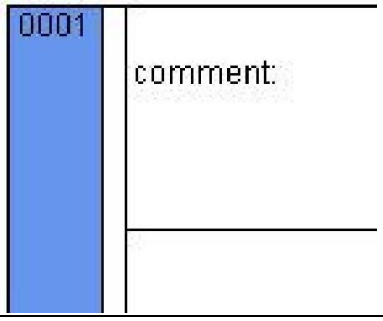
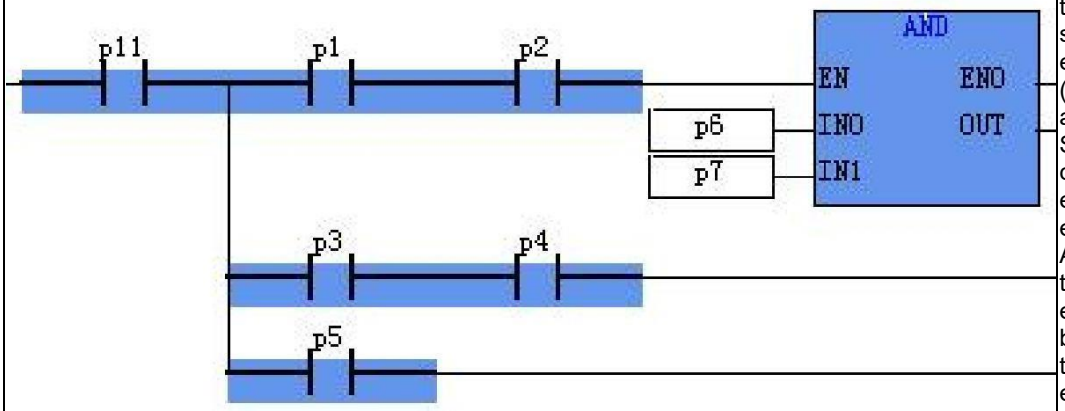
When configuring in LD, users need to select elements to execute all operations. Users can select a single element or multiple elements to execute related operations.

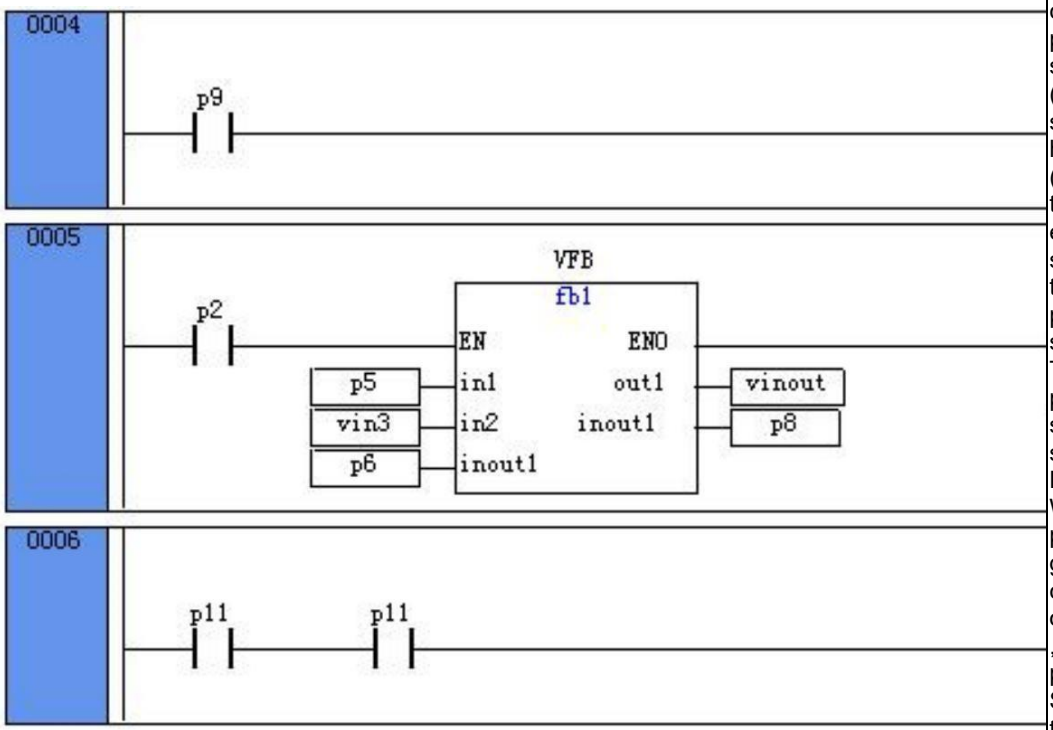
The commands that can be executed vary depending on the selected position.

The element is highlighted in blue when it is selected by the cursor.

See the following table for details of the selected positions and operations:

Selected Position	Legend	How to Select
Text Area		Click the variable name or block name of the element
Contact		Click the contact

Block Elements		Click the block element
Coil		Click the coil
Section		Click the section
Comment for Network Node		Click the Please add a comment here...
Multiple Elements		Steps: (1) Select an element on the section with the cursor as the starting element. (2) Press and hold Shift, and click the ending element. At this time, all elements between the two elements are selected. Note: (3) The starting and ending elements

		here refer to the positions where the elements are located, counted sequentially from top to bottom and from left to right. 4 When selecting multiple elements to execute an operation, press and hold Shift until the operation command is executed.
Program Segment		Steps: (1) Select the program section (2) Press and hold Shift (3) Click the ending section of the program segment This program section is selected. Note: When performing a right-click operation, keep pressing Shift until the operation command

		is executed.

- In the LD language programming area, when the block elements are added, the AND functional blocks are added by default, with enable input terminal EN and enable output terminal ENO. The ENO output value is equal to the input value of the EN terminal.

10.7.4 Insert in Front of the Section

10.7.4.1 Introduction

Insert a new section in front of the section where the cursor is located.

10.7.4.2 Steps

To insert a new section in front of the selected section, follow the steps below:

(1) Select the section. Add the entry action via:

- Menu bar: Click [Insert] - [Network(Before)];
- Programming area: Right-click the section. Click Network(Before).

(2) Right-click the section. Select the Previous Section command. At this point, a new section is inserted in front of the current section and is highlighted in blue, indicating it is selected.

10.7.5 Insert After the Section

10.7.5.1 Introduction

Insert a new section after the section where the cursor is located.

10.7.5.2 Steps

To insert a new section after the selected section, follow the steps below:

(1) Select the current section with the cursor.

- Menu bar: Click [Insert] - [Network(After)];
- Programming area: Right-click the section. Click Network(After).

(2) Right-click the section. Select the Next Section command. At this point, a new section is inserted

after the current section, and is highlighted in blue, indicating it is selected.

10.7.6 Insert Contact

10.7.6.1 Insert contact forward

10.7.6.1.1 Introduction

Connect a contact in series in front of a contact, block element, coil element, or parallel logic.

10.7.6.1.2 Requirement

The element is selected.

10.7.6.1.3 Steps

To connect a contact in series forward, follow the steps below:

(1) Right-click the element.

- Note: If the currently selected element is parallel logic, users need to press and hold Shift key, and right-click the blue highlighted area. If users right-click other blank areas, the selected status is invalid.

(2) In the right-click menu, select Insert Contact command. When the right-click menu pops up, release the Shift key. At this time, a contact is inserted and is highlighted in blue, indicating it is selected.

10.7.6.1.4 Reference message

Select Element.

10.7.6.2 Insert contact backward

10.7.6.2.1 Introduction

Connect a contact in series behind a contact, block element, or parallel logic.

10.7.6.2.2 Requirement

The element is selected.

10.7.6.2.3 Steps

To connect a contact in series backward, follow the steps below:

(1) Right-click the element.

- Note: If the currently selected element is parallel logic, users need to press and hold Shift key, and right-click the blue highlighted area. If users right-click other blank areas, the selected status is invalid.

(2) Select the Append Contact command in the right-click menu or click the Connect Contact in Series icon in the toolbar. When the right-click menu is displayed, release Shift key.

(3) At this time, a contact is inserted and is highlighted in blue, indicating it is selected.

10.7.6.2.4 Reference message

Select Element.

10.7.6.3 Insert contact at the beginning of the section

10.7.6.3.1 Introduction

Insert a contact at the beginning of the current section.

10.7.6.3.2 Requirement

The current section is selected.

10.7.6.3.3 Steps

To insert a contact at the beginning of the section, follow the steps below:

(1) Right-click the section. In the right-click menu, select the Insert Contact command.

(2) At this time, a contact is inserted and is highlighted in blue, indicating it is selected.

10.7.6.3.4 Reference message

Select Element.

10.7.6.4 Connect contact in parallel

10.7.6.4.1 Introduction

Connect a contact in parallel to a contact, block element or multiple elements.

10.7.6.4.2 Requirement

The element is selected.

10.7.6.4.3 Steps

To connect a contact in parallel, follow the steps below:

(1) Click the menu bar [Insert] - [Parallel Contact], or click the “Parallel Contact” icon on the toolbar.

(2) At this time, a contact is connected in parallel to the currently selected element and is highlighted in blue, indicating it is selected.

10.7.6.4.4 Reference message

Select Element.

10.7.6.5 Connect Negated contact in series

10.7.6.5.1 Introduction

Insert a negated contact behind a contact, block element, or in front of a coil. Or connect an negated contact in series for parallel logic.

10.7.6.5.2 Requirement

The element is selected.

10.7.6.5.3 Steps

To insert an negate contact, follow the steps below:

(1) Right-click the element.

- Note: If the currently selected element is parallel logic, users need to press and hold Shift key,

and right-click the blue highlighted area. If users right-click other blank areas, the selected status is invalid.

(2) Select the [Insert] - [Negate Contact] command from the menu bar, or choose the “Negate Contact” command from the right-click menu, or click the toolbar icon  .

(3) When the right-click menu is displayed, release Shift key.

(4) At this time, a negate contact is inserted and is highlighted in blue, indicating it is selected.

10.7.6.5.4 Reference message

Select Element.

10.7.6.6 Connect inverted contact in parallel

10.7.6.6.1 Introduction

Connect an inverted contact in parallel to a contact, block element or multiple elements.

10.7.6.6.2 Requirement

The element is selected.

10.7.6.6.3 Steps

To connect an inverted contact in parallel, follow the steps below:

(1) Select the [Insert] - [Parallel Negate Contact] command from the menu bar, or click the “Parallel Inverted Contact” icon  on the toolbar.

(2) At this time, an inverted contact is connected in parallel to the currently selected element, and is highlighted in blue, indicating it is selected.

10.7.6.6.4 Reference message

Select Element.

10.7.7 Insert Coil

10.7.7.1 Insert coil

10.7.7.1.1 Introduction

Insert a coil at the end of the current section busbar. Or connect a coil in parallel below an existing output element.

10.7.7.1.2 Requirement

The current section is selected.

10.7.7.1.3 Steps

To insert a coil, follow the steps below:

(1) Right-click the section.

(2) In the right-click menu, select the Coil command. Or click the Toolbar icon.

(3) At this time, a coil is inserted at the end of the section and is highlighted in blue, indicating it is selected.

(4) To continue connecting coil in parallel, repeat the above steps.

10.7.7.1.4 Reference message

Select Element

LD Elements

10.7.7.2 Insert set coil

10.7.7.2.1 Introduction

Insert a set coil at the end of the current section busbar. Or connect a new set coil in parallel below the existing one.

10.7.7.2.2 Requirement

The current section is selected.

10.7.7.2.3 Steps

To insert a set coil, follow the steps below:

(1) Right-click the section.

(2) In the right-click menu, select the Set Coil command. Or click the Toolbar icon .

(3) At this time, a set coil is inserted at the end of the section, and is highlighted in blue, indicating it is selected.

(4) To continue connecting the set coils in parallel, repeat the above steps.

10.7.7.2.4 Reference message

Select Element

10.7.7.3 Insert reset coil

10.7.7.3.1 Introduction

Insert a reset coil at the end of the current section busbar. Or connect a new reset coil in parallel below the existing one.

10.7.7.3.2 Requirement

The current section is selected.

10.7.7.3.3 Steps

To insert a reset coil, follow the steps below:

(1) Right-click the section.

(2) In the right-click menu, select the Reset Coil command. Or click the Toolbar icon .

(3) At this time, a reset coil is inserted at the end of the section, and is highlighted in blue, indicating it is selected.

(4) To continue connecting the reset coils in parallel, repeat the above steps.

10.7.7.3.4 Reference message

Select Element

10.7.7.4 Insert inverted coil

10.7.7.4.1 Introduction

Insert an inverted coil at the end of the current section busbar. Or connect a new inverted coil in parallel below the existing one.

10.7.7.4.2 Requirement

The current section is selected.

10.7.7.4.3 Steps

To insert an inverted coil, follow the steps below:

(1) Right-click the section.

(2) In the right-click menu, select the Invert Coil command. Or click the Toolbar icon  .

(3) At this time, an inverted coil is inserted at the end of the section, and is highlighted in blue, indicating it is selected.

(4) To continue connecting the inverted coils in parallel, repeat the above steps.

10.7.7.4.4 Reference message

Select Element

10.7.7.5 Set and reset coil

10.7.7.5.1 Introduction

By setting/resetting, set the coil variable to 1 to set it as a set coil or reset coil.

Set (S): When the logic operation value of the input coil is 1, the coil variable is set to TRUE. The variable remains TRUE until it is reset to FALSE.

Reset (R): When the logic operation value of the input coil is 1, the coil variable is set to FALSE. The variable remains FALSE until it is set to TRUE.

When there are both set and reset signals, the reset has a higher priority.

Select the Set/Reset command via:

- Programming area: Coil right-click menu - [Set/Reset];
- Shortcut key: Ctrl + T;

- Toolbar:  .

10.7.7.5.2 Requirement

The coil is selected.

10.7.7.5.3 Set coil

To set a coil, follow the steps below:

(1) Right-click the coil, or reset the coil.

(2) In the right-click menu, select the Set/Reset command.

(3) The coil is set as a set coil.

10.7.7.5.4 Reset coil

To reset a coil, follow the steps below:

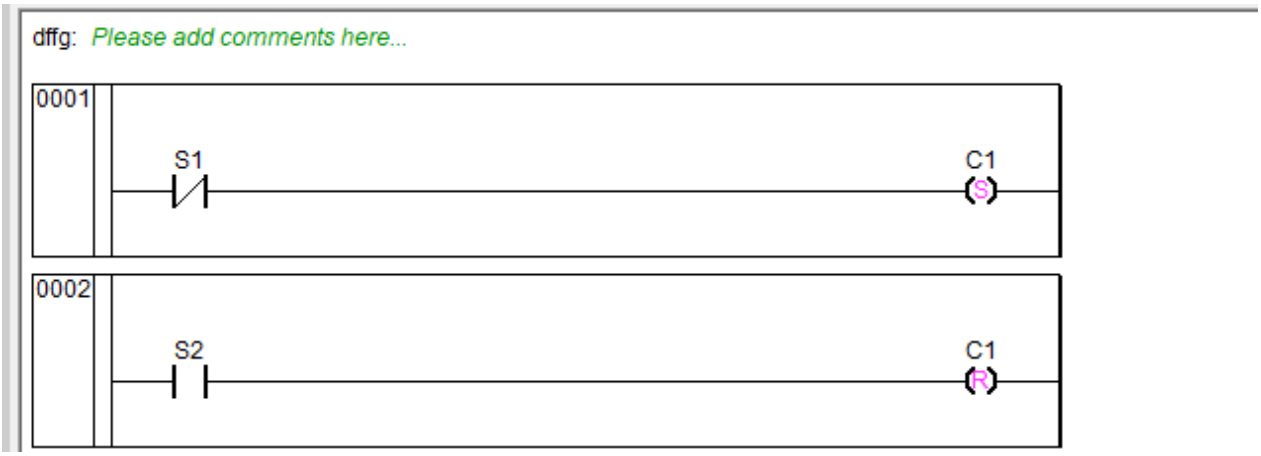
(1) Right-click the coil, or set the coil.

(2) In the right-click menu, select the Set/Reset command.

(3) The coil is set as a reset coil.

10.7.7.5.5 Example

When S1 is FALSE, the coil variable C1 is set to TRUE; Only when S2 is TRUE, the coil variable C1 is reset to FALSE. If coil C1 is in the reset state, when S1 is FALSE, the coil remains in the reset state until S2 is FALSE. At this time, the coil is set to TRUE again.



10.7.7.5.6 Reference message

Select Element

10.7.8 Insert Block Element

10.7.8.1 Connect block element in series

10.7.8.1.1 Introduction

Connect a block element in series to the element. The block element defaults to an AND functional block and has an enabled input pin EN. When EN is TRUE, the block element is executed.

The block element is inserted in a different position depending on the selected area.

- Select the section. Execute the Block Element command. The block element is inserted at the beginning of the section.
- Select the contact and execute the Block Element command. Then the block element is inserted behind the contact.
- Select the coil and execute the Block Element command. Then the block element is inserted before the coil.

You can execute the Block Element command in the following ways:

- Programming area: Right-click menu - Block Element;
- Toolbar: ;
- Shortcut keys: Alt + F9.

10.7.8.1.2 Requirement

The area where the element needs to be inserted has been selected.

10.7.8.1.3 Steps

Follow the steps below to insert a block element:

(1) Right-click the selected area.

(2) Select the Block Element command in the menu that appears.

(3) A block element is inserted and highlighted in blue, indicating it is selected.

10.7.8.1.4 Reference message

Select Element

10.7.8.2 Add parallel block element

10.7.8.2.1 Introduction

Add the parallel output block element to the coil.

10.7.8.2.2 Requirement

The coil is selected.

10.7.8.2.3 Steps

Follow the steps below to add a parallel block element:

(1) Right-click the coil.

(2) Select the Parallel Block Element command in the menu that appears.

(3) A block element is inserted below the coil and highlighted in blue, indicating it is selected.

10.7.8.2.4 Reference message

Select Element

10.7.9 Execute ST

Supports inserting ST language program segments. This feature enables mixed programming with ladder diagrams and ST language, making it more flexible and convenient to write and view programs for large-scale analog calculations and positioning algorithms. It allows for both logical sequence control and data computation within the same function block or subroutine.

- Menu Bar: [Insert] - [Execute ST];
- Program Area: Right-click Menu—“Execute ST”;
- Toolbar: ST;

10.7.9.1 Requirement

Select the area where you want to insert the Execute ST segment.

10.7.9.2 Steps

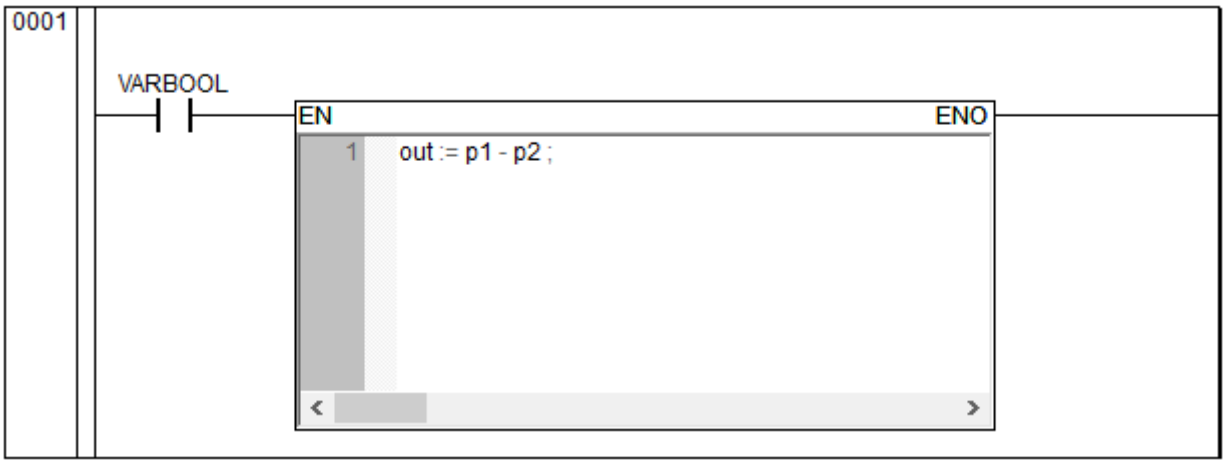
(1) Right-click the selected area.

(2) In the right-click menu, choose the "Execute ST" command.

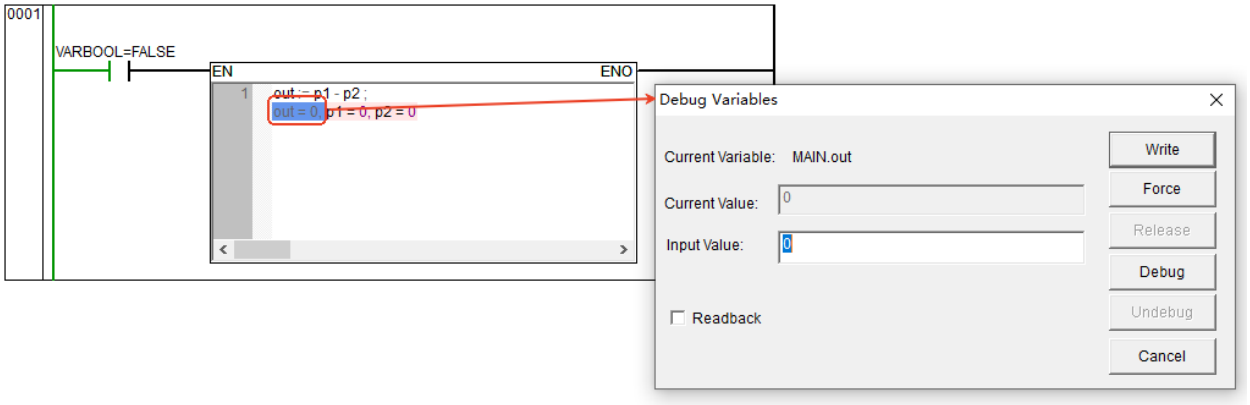
(3) An ST language program segment will be inserted and highlighted in blue.

10.7.9.3 Example

After inserting the ST language program segment into the program section, expressions can be configured in the ST program area, as shown in the figure below. For the method of creating expressions, refer to the relevant content on creating expressions in ST programs.



In online mode, you can double-click on a variable for debugging, as shown in the figure below.



- When there is an illegal condition in the ST expression, the system does not provide a prompt. However, the "Syntax Check" window will display an error during compilation.

10.7.9.4 Reference Information

Select Element

10.7.10 Configure Block Element Pin

10.7.10.1 Add input pin

10.7.10.1.1 Introduction

When the default input pins of a block element are not sufficient, additional input pins can be added. Only one input pin can be inserted at a time. The total number of input pins shall not exceed 64.

- When the input pin is added, the pin insertion position differs depending on the area selected.
- Select the block element. Execute the Multiple Inputs command. The new pin is inserted into the last one.
- Select the input pin. Execute the Multiple Inputs command. The new pin is inserted in front of the selected pin.

You can execute the More Input Pins command in the following ways:

- Programming area: Block element right-click menu - [Advanced] - [Multi-Inputs (D)];
- Programming area: Input pin right-click menu - Multi-Inputs (D);
- Shortcut key: Ctrl + D.

10.7.10.1.2 Requirement

The block element or input pin has been selected.

10.7.10.1.3 Steps

Follow the steps below to add an input pin:

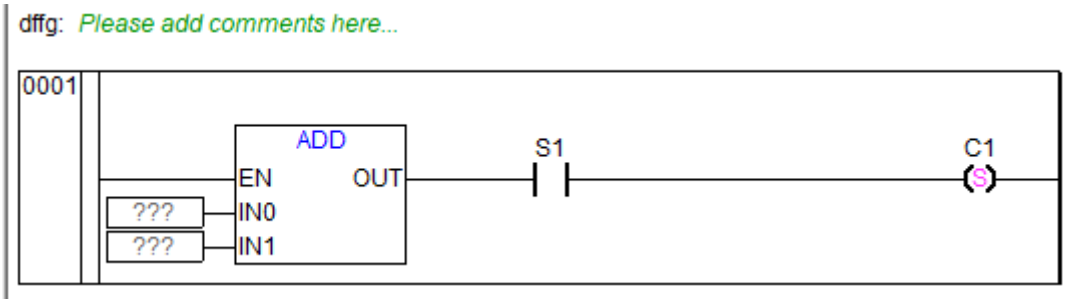
(1) Right-click the selected area.

(2) Select the More Input Pins command in the menu that appears.

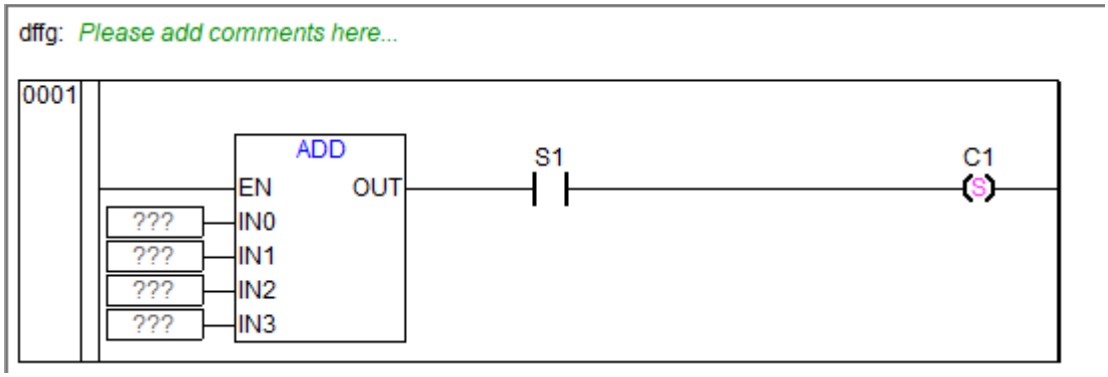
(3) An input pin is inserted and highlighted in blue, indicating it is selected.

10.7.10.1.4 Example

As shown in the figure, for the selected block element, execute 2 times Multiple Inputs command to add 2 input pins, IN2 and IN3.



(a)



(b)

10.7.10.1.5 Reference message

Select Element

10.7.10.2 Show or hide pins

10.7.10.2.1 Introduction

When configuring the block element, set the showing and hiding of enabled pins, input and output pins as needed.

All pins of the functional block can be set; For other block elements, only the EN and ENO pins can be set.

10.7.10.2.2 Requirement

The block element has been configured.

10.7.10.2.3 Hide pins

Follow the steps below to hide input pins:

(1) Right-click the block element.

(2) In the right-click menu, select the [Advanced] - [Pins (Show/Hide)] command.

(3) The Set Hide/Show for Pins dialog box is displayed.

(4) Click the checkbox to uncheck the pins.

(5) Click OK.

(6) The corresponding pins of the block element are hidden.

10.7.10.2.4 Show pins

Follow the steps below to hide input pins:

(1) Right-click the block element.

(2) In the right-click menu, select the [Advanced] - [Pins (Show/Hide)] command.

(3) The Set Hide/Show for Pins dialog box is displayed.

(4) Check the pins to be displayed.

(5) Click OK.

(6) The input pins of the block element are displayed.

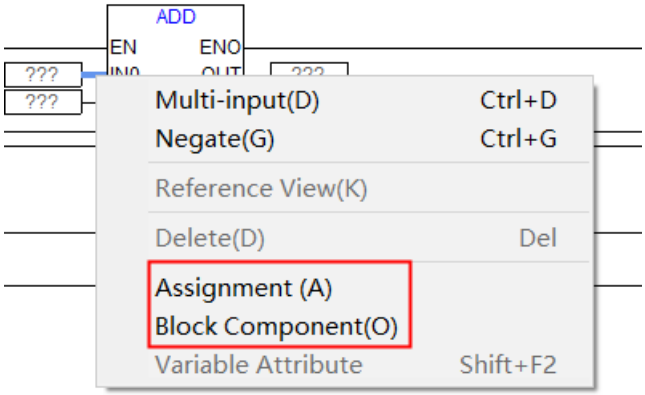
10.7.10.2.5 Reference message

Select Element

10.7.11 Cascade Block Element Pin

Cascade configuration means that the input pins of a block element can be connected to other block elements and assigned values. This reduces intermediate variables and makes the logic connection tight, structurally clear, and intuitively easy to understand.

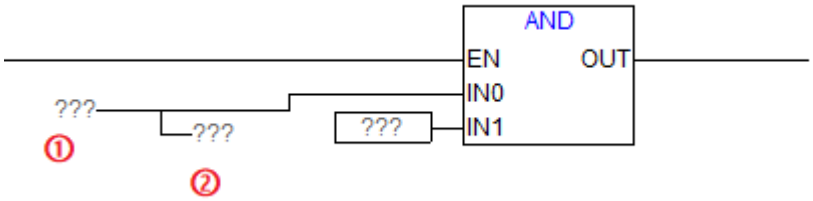
The right-click menu of block element input pin is shown in the figure below.



10.7.11.1 Cascade assignment element

10.7.11.1.1 Introduction

When the cascade object of the input pin of a block element is Assignment, two elements are added at the same time. The cascade result is shown in Positions ① and ② in the figure below:



Where Position ① is the input of the Assignment element, and Position ② is the output of the Assignment element; And Position ① assigns values to Position ② and the IN0 item of the AND block at the same time.

The Assignment element cannot be copied, pasted, or cut.

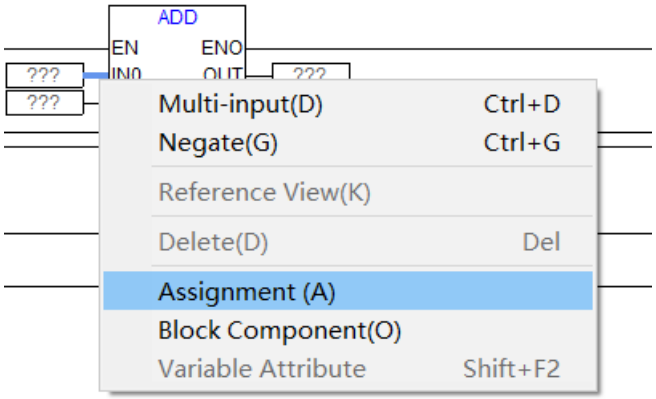
10.7.11.1.2 Requirement

There is a block element that needs to be cascaded.

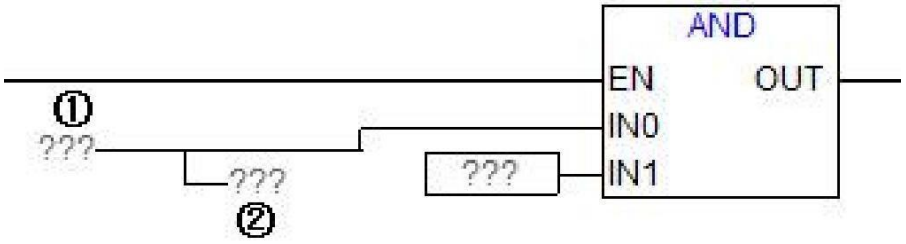
10.7.11.1.3 Steps

Follow the steps below to cascade an assignment element:

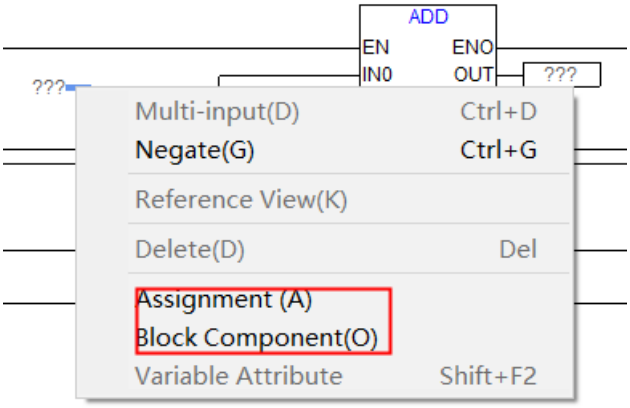
(1) Right-click one of the input pins of the block element. Select the Assign command, as shown in the figure below.



(2) The input pin is cascaded with two assignment elements

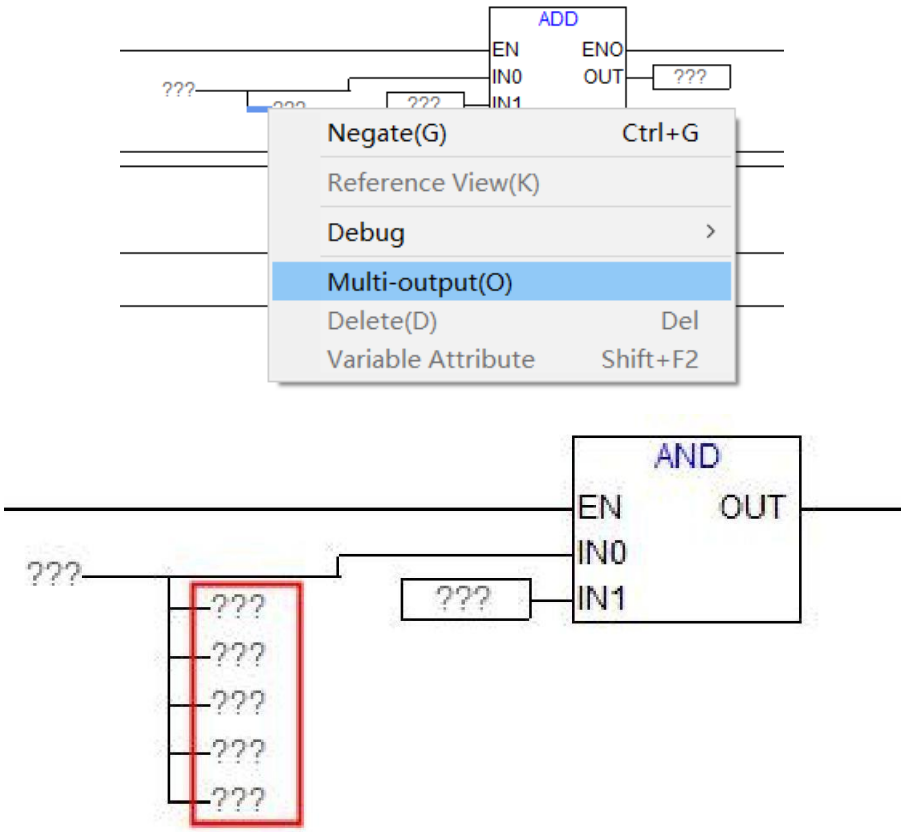


(3) The Assign or Block Element command in the right-click menu of Position ① allows users to expand Position ① to the next level. As shown in the figure below.



(4) The Multiple Outputs command in the right-click menu of Position ② allows users to add outputs

to expand Position ② in parallel. As shown in the figure below.



10.7.11.2 Delete assignment element

10.7.11.2.1 Introduction

The Delete command of a node allows you to remove a cascaded Assignment element or an output. The scope of the deletion depends on the position of the selected node.

10.7.11.2.2 Steps

To delete a cascaded assignment element, follow the steps below:

(1) Select the cascaded node position.

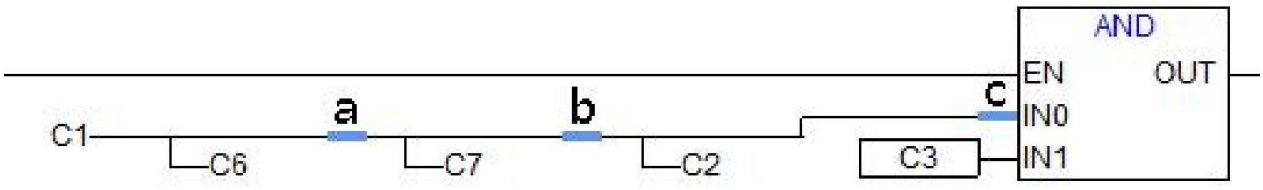
(2) Right-click the node. Click the Delete command. The assigned endpoints of the corresponding scope are deleted.

10.7.11.2.3 Example

Endpoints C1 and C6 can be deleted through Node (a) in the figure below;

Endpoints C1, C6, and C7 can be deleted through Node (b) in the figure below.

Endpoints C1, C6, C7, and C2 can be deleted through Node (c) in the figure below.



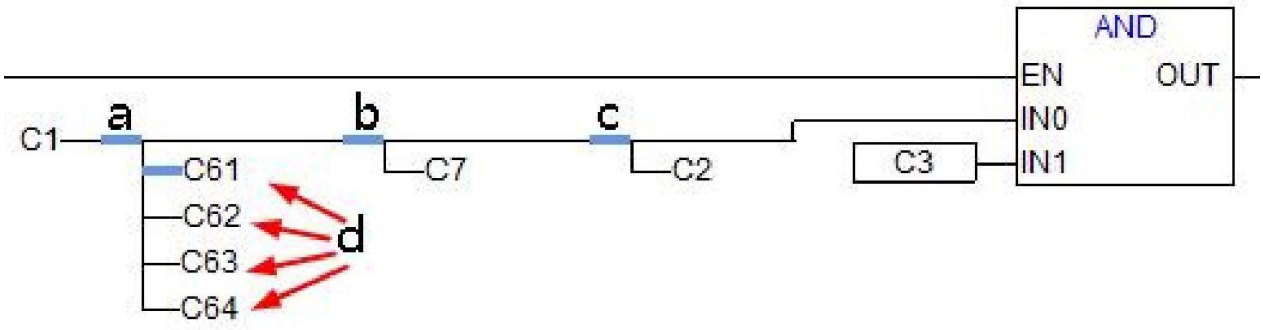
10.7.11.2.4 Delete multiple output nodes

Endpoints C61 - C64 can be deleted through Node (a) in the figure below.

Endpoint C7 can be deleted through Node (b) in the figure below.

Endpoint C2 can be deleted through Node (c) in the figure below.

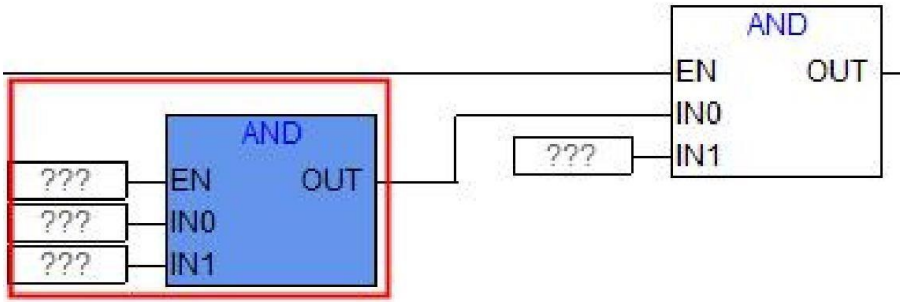
Endpoints C61, C62, C63, and C64 can be deleted through Node (d) in the figure below.



10.7.11.3 Cascade block element

10.7.11.3.1 Introduction

When the cascade object of the input pin of a block element is Block Element, an AND block is added by default. The cascade result is shown in the figure below.



When the Block Element is cascaded, the first output pin of the Block Element is connected by default. When users need to change the connected outputs, the output pins can be adjusted via the Pin (Show/Hide) command.

See the relevant contents of the Assignment element for the operation method and influence the scope of deleting block elements, which will not be repeated here.

10.7.11.3.2 Requirement

A block element has been added.

10.7.11.3.3 Steps

The operation method and scope of influence of Cascade block element refer to Cascade assignment element.

10.7.12 Rename Block Element

10.7.12.1 Introduction

The block element is added as an AND block by default. However, you can manually modify the block element name to the needed one. When the block element name is entered, the software automatically associates all the block element names that match the entered keyword for choice, making it easy to enter quickly.

10.7.12.2 Requirement

A block element has been created.

10.7.12.3 Steps

To rename the block element, follow the steps below:

(1) Double-click the block element name. The cursor is displayed at the block element name.

(2) Select the block element name. Enter the new name. All associated block elements are automatically displayed.

(3) Double-click the selected block element. The block element name is modified.

10.7.13 Add Jump Lable

10.7.13.1 Introduction

When the user program needs to jump to other program segments, the jump destination can be specified by the jump tag. The program jumps to the destination program segment for further execution.

Up to 32 characters can be entered for the jump tag. The character requirements are the same as the variable definition.

10.7.13.2 Requirement

The destination program segment has been configured.

10.7.13.3 Steps

Follow the steps below to add a jump codename:

(1) Right-click the destination program section.

(2) In the right-click menu, select the Lable command. The identification label and the placeholder ??? for the jump tag name are displayed at the top of the program segment.

(3) Click the placeholder ???. Enter the jump tag name.

(4) The jump tag name shall be entered at the jump destination.

10.7.13.4 Reference message

Jump

10.7.14 Jump

10.7.14.1 Introduction

The Jump command can be used to interrupt the sequential execution of a program, and to jump to the destination program segment for further execution. To identify the destination program segment, it is

necessary to add the Jump Codename. The jump codename name shall be specified at the jump destination, which is ??? by default.

The jumps are all at the end of the network section. If the logical operation result is TRUE, it jumps to the program segment with the specified codename. If the logical operation result is FALSE, the program is executed sequentially downwards.

You can execute the Jump command in the following ways:

- Menu Bar: [Insert] - [Jump];
- Programming area: Right-click the menu of the last input element or any output element in the current section > Jump;
- Toolbar:  .

10.7.14.2 Requirement

The coil has been selected

10.7.14.3 Steps

Follow the steps below to add a jump:

In the interrupted program section, right-click the coil.

In the right-click menu, select the Jump command. A jump arrow and a placeholder ??? for the jump tag name are displayed at the bottom of the coil.

Click the placeholder. Enter the jump tag name.

10.7.14.4 Reference message

Add Jump Codename

Select Element

Return

10.7.15 Return

10.7.15.1 Introduction

When calling a POU, users can use the Return command to return to the called POU when the conditions are met, instead of continuing to execute the called POU.

The return placeholder defaults to Return.

You can execute the Return command in the following ways:

- Menu Bar: [Insert] - [Return];
- Programming area: Right-click the last input element or any output element in the current section, and click Return in the menu that appears.
- Toolbar: RET.

10.7.15.2 Requirement

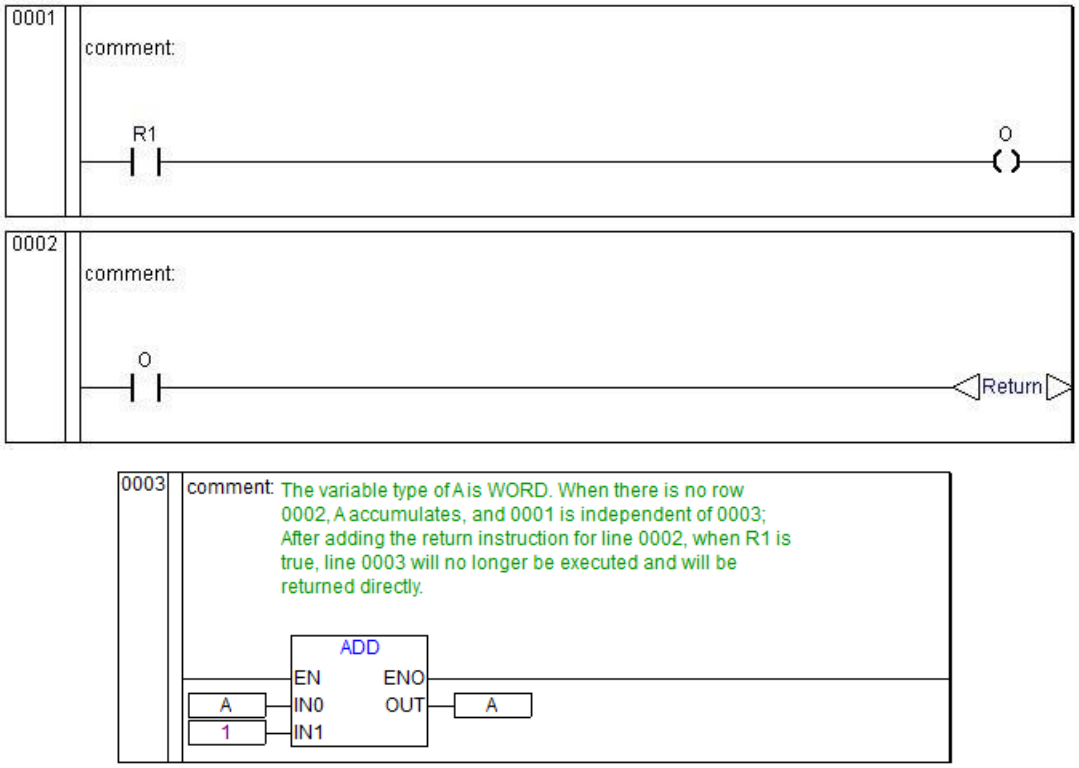
The section has been selected.

10.7.15.3 Steps

Follow the steps below to insert a Return element:

(1) Right-click the section.

(2) In the right-click menu, select the Return command. At the end of the section, insert the return character Return. As shown in Section 0002 in the figure below: When contact O of section 0002 is closed and turned on, jump out of the program, and return to the POU called it, without executing the program in section 0003 and below.



10.7.16 Negate

10.7.16.1 Introduction

The operation allows you to negate the pins of a contact, coil or block element.

Select the Invert command via:

- Menu Bar: [Insert] - [Negate];
- Programming area: Click negate in the right-click menu.
- Toolbar:  ;
- Shortcut key: Ctrl + G.

10.7.16.2 Requirement

The element is selected.

10.7.16.3 Steps

To negate an element, follow the steps below:

(1) Right-click the element.

(2) In the right-click menu, select the negate command. The element is negated.

10.7.16.4 Reference message

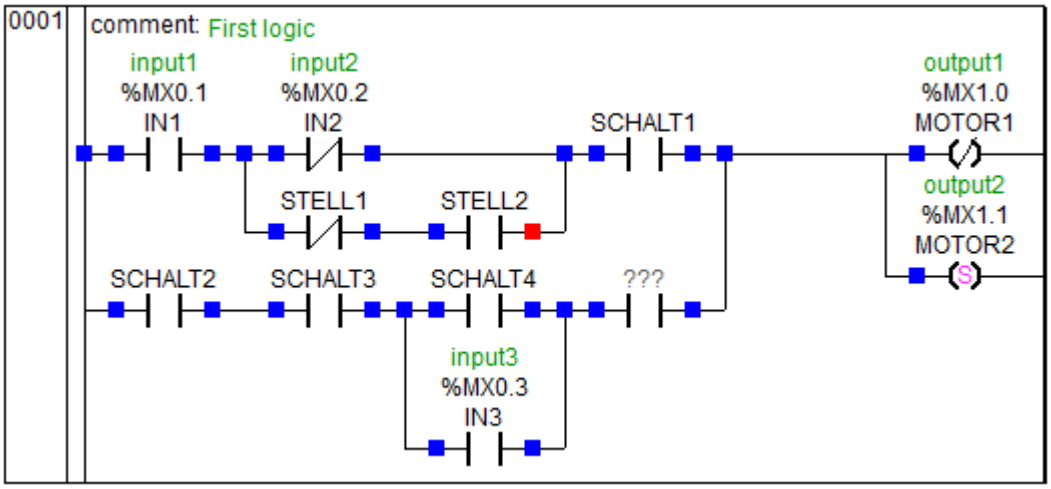
Select Element

10.7.17 Move

To move an element to another position, follow the steps below:

(1) Select the element. Hold down the left mouse button and drag it. At this time, a bright blue mark  is displayed in front of and behind all the elements that can be moved (the coil only has a moving rectangle box on the left side), as shown in the figure.

(2) When the element is moved to the target position, the box closest to the mouse is highlighted in red  . Release the mouse to move the element to this position.



10.7.18 Add Program Comment

10.7.18.1 Introduction

Use program comments to describe the program content of each network section. Program contents include the process description of the program section, the logical function description, or characteristics requiring the user's attention.

10.7.18.2 Requirement

The program section has been configured.

10.7.18.3 Steps

To add a program section comment, follow the steps below:

(1) Right-click the destination program section.

(2) In the right-click menu, select the Change Comment command. The mark comment is displayed at the top of the program section.

(3) Click Please add a comment here.... Enter the comment content.

10.7.19 Switch Comment Status

10.7.19.1 Overview

The annotation status section maintains the same display effect whether offline or online. When in annotation mode, the program section does not participate in syntax error checking, is not included in the compilation logic for itself and its sub-elements, and cannot be run online or perform online-related operations.

You can execute the Switch Comment Status command in the following ways:

- Program area: Right-click program section - 【Switch Comment Status】;
- Shortcut key: CTRL+SHIFT+O.

10.7.19.2 Requirement

Select the section you want to modify.

10.7.19.3 Steps

To change the annotation status, follow these steps:

(1) Right-click the target program section.

(2) In the right-click menu, select the "Toggle Annotation Status" command. The program section will then be in annotation status.

(3) To cancel the annotation status, select the "Switch Comment Status" command again.

10.7.20 Copy Element

10.7.20.1 Introduction

The section, contact, coil, block element, multiple elements or program segment in a program can be copied and pasted.

10.7.20.2 Requirement

There are available elements.

10.7.20.3 Steps

Follow the steps below to execute the Copy command:

Select the element to be copied. Execute the Copy via:

- Element selected area: Right-click menu - Copy;
- Menu Bar: [Edit] - [Copy];
- Toolbar:  ;
- Shortcut key: Ctrl + C.

-  Note: If you copy a program segment or multiple elements via the right-click menu, press and hold Shift after selecting them until the Copy operation is executed.

10.7.20.4 Result

The program is copied to the clipboard, and can be pasted to the target position.

10.7.20.5 Reference message

Select Element

10.7.21 Cut Element

10.7.21.1 Introduction

You can cut sections, contacts, coils, block elements, multiple elements, and program segments in a program.

10.7.21.2 Requirement

There are available elements.

10.7.21.3 Steps

Follow the steps below to execute the Cut command:

(1) Select the element to be cut.

(2) Execute the Cut command via:

- Element selection area: Click Cut in the right-click menu.
- Menu bar: Choose Edit > Cut.
- Toolbar: ;
- Shortcut key: Ctrl + X.

- Note: When you cut a program segment or multiple elements, press and hold the Shift key after you select the segment or elements until the Cut operation is performed if you are using the right-click menu.

10.7.21.4 Result

The object is cut and saved in the clipboard. You need to paste it to the target position.

10.7.21.5 Reference message

Select Element

10.7.22 Paste Element

10.7.22.1 Introduction

Users can paste the copied and cut element to a certain position in the program.

- Paste section: Paste the copied section in front of the current section.
- Paste contact and block element: Paste the copied element in front of the current element.
- Paste coil: Paste the coil to the end of the section. If the current section already has a coil, connect that copied coil in parallel.
- Paste to the below or right: Contact and block elements can be pasted to the right of the currently selected element or in parallel to the current element.

You can execute the Paste command in the following ways:

- Programming area: Click Paste in the right-click menu.
- Menu bar: Choose [Edit] - [Paste].
- Toolbar: ;
- Shortcut key: Ctrl + V.

10.7.22.2 Requirement

The element to be pasted has been copied or cut.

10.7.22.3 Paste

Follow the steps below to paste an element:

(1) Right-click the element.

(2) Select the Paste command in the menu that appears.

10.7.22.4 Paste to the right

Follow the steps below to paste an element to the right:

(1) Right-click the element.

(2) Select the Paste to Right command in the menu that appears.

10.7.22.5 Paste to the below

Follow the steps below to paste an element to the below:

(1) Right-click the element.

(2) Select the Paste to Bottom command in the menu that appears.

10.7.23 Delete Element

10.7.23.1 Requirement

There are available elements.

10.7.23.2 Steps

Follow the steps below to delete an element:

(1) Select the element to be deleted.

(2) You can execute the Delete command in the following ways:

- Programming area: Click Delete in the right-click menu.
- Menu bar: Choose [Edit] - [Delete].

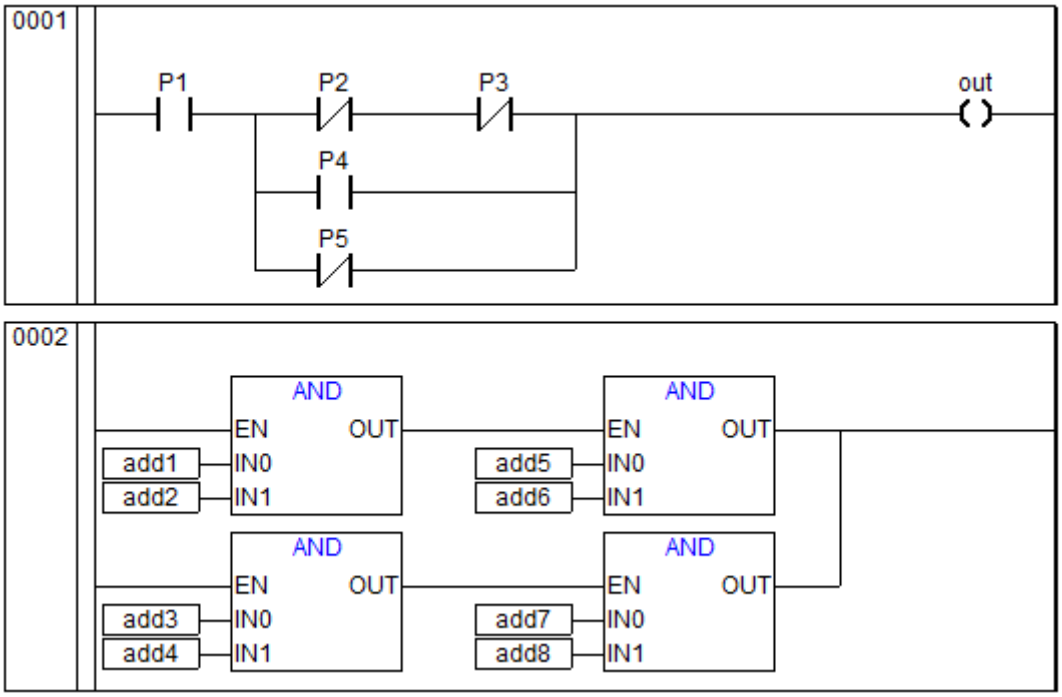
- Shortcut key: Delete.

- Note: When you delete a program segment or multiple elements, press and hold the Shift key after you select the segment or elements until the Delete operation is performed if you are using the right-click menu.

10.7.23.3 Reference message

Select Element

10.7.24 LD Programming Example



10.8 Create ST Program

10.8.1 ST Programming Overview

ST is short for structured text, a text-based programming language. ST consists of a series of statements, similar to a high-level language.

The POU operation process is determined by the statement.

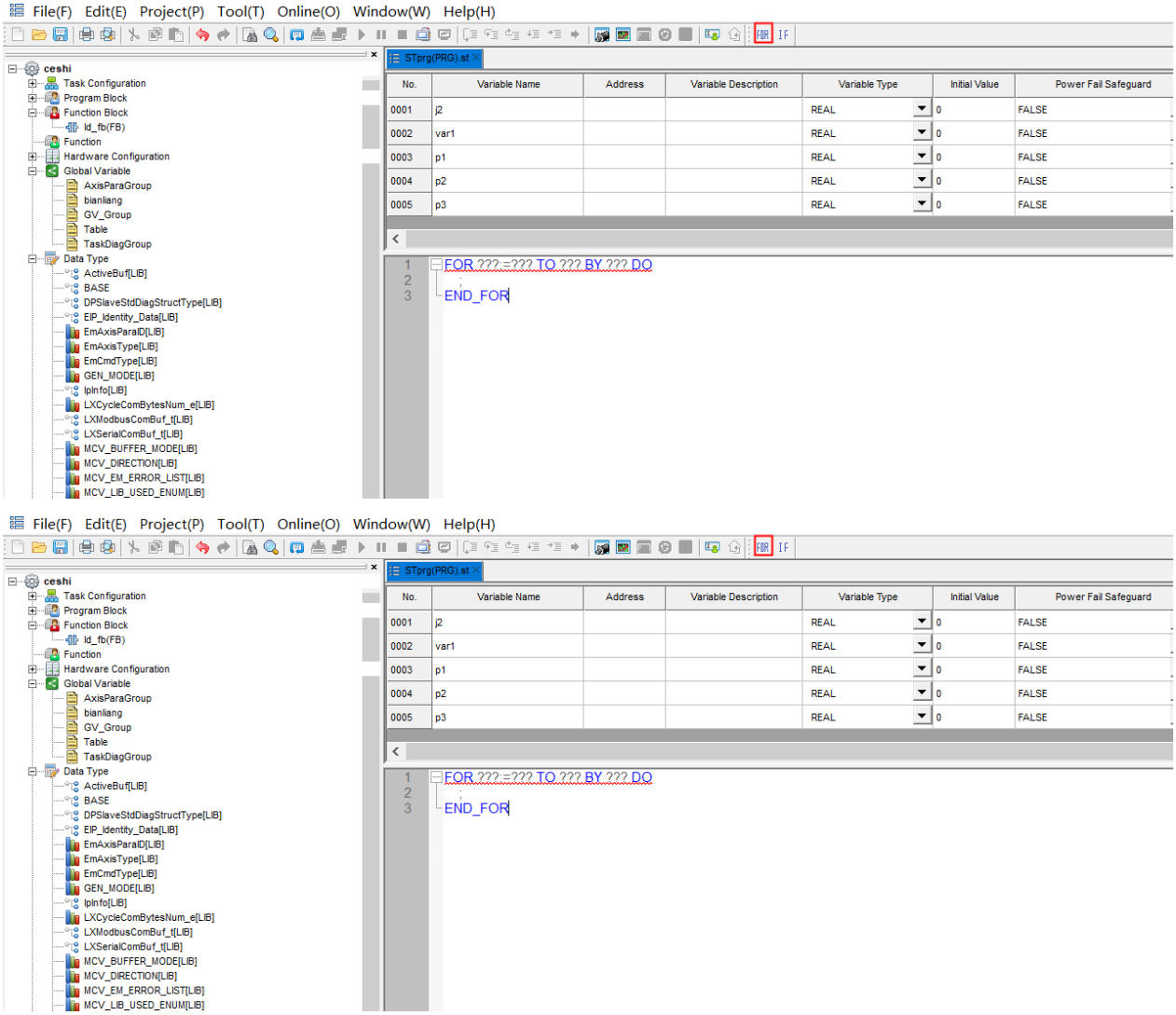
All functions and functional blocks in the Library Manager can be called in the ST language. Part of the display when the function is called is obscured by the operator. The functional block can be used after it

has been declared in the ST POU. The function and functional block are parameterized in the same way as in LD or CFC language.

The execution sequence can be obtained from the statement sequence orchestrated in the ST Editor, from left to right and from top to bottom. The sequence can only be changed by inserting a loop statement.

The ST Editor is shown in the figure below.

In addition to the basic buttons, the Toolbar of ST language also provides FOR and IF buttons. By clicking these two buttons at the cursor in the editing area, users can call the template statements to facilitate programming. An example is shown in the figure below.



10.8.2 ST Element

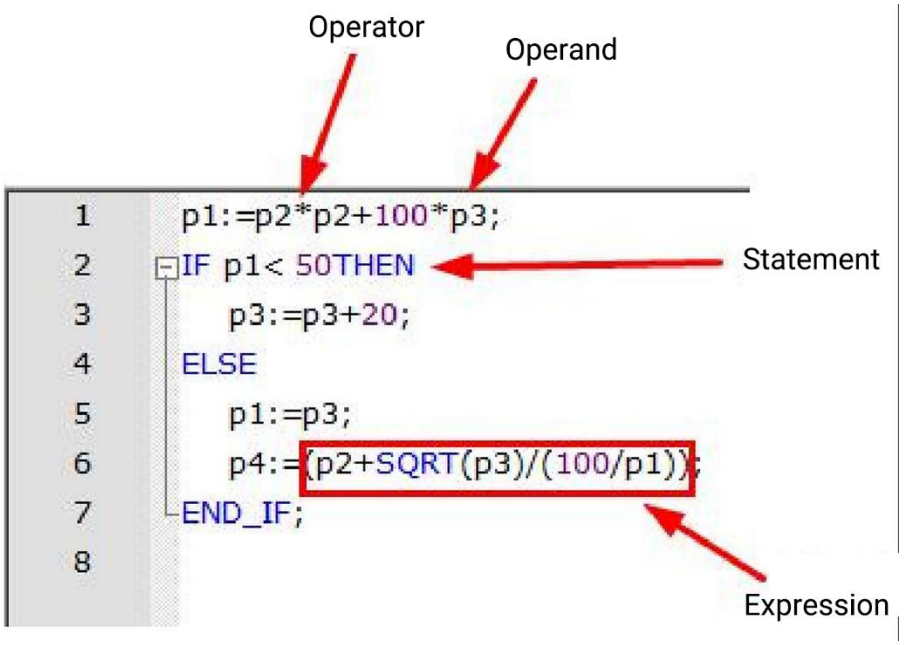
Users can use the structured text (ST) programming language to execute a variety of operations, such as calling functional blocks, assigning values, and executing instructions and repetitive tasks conditionally.

ST programming language consists of various elements as follows:

- Expression: A structure consisting of operands and operators. An expression returns a value when it is executed.
- Operand: An operand represents a variable, a value, an address, a functional block, etc.
- Operator: A symbol used in the execution of an operation.
- Statement: Used to assign the value returned by the expression to the actual parameters and to

construct and control the expression.

The legend description of each element is shown in the figure.



- The editing content of a single POU shall not exceed 9999 lines when edited in the ST language.

10.8.3 Expression

An expression is a structure that returns the evaluated value of a variable. This return value is required in the statement. An expression consists of operators and operands. Operands can be constants, variables, return values of function calls, or other expressions. Example:

```
12          (*Constant*)
Var1        (*Variable*)
Fun(a, b, c)        (*Function call*)
a + b       (*Operator call*)
I + (x * y * z) (*Operator call*)
```

When an expression is computed, the operators are applied to the operand table in the sequence defined by the operator priority. The operator with the highest priority in the expression is executed first, followed by the operator with the second highest priority; It continues in this manner until the entire calculation is completed. The operators with the same priority are executed from left to right according to the sequence in which they are written in the expression. Parentheses can be used to change this sequence.

For example, if the values of A, B, C, and D are 1, 2, 3, and 4, respectively, A - B - C * D results in -13. While (A - B - C) * D results in -16.

If the operator is related to two operands, the left operand is executed first. For example, in the expression SIN(x) * COS(y), the expression SIN(x) is computed first, followed by COS(y), and then the product of the two is computed.

10.8.4 ST Operator

The common ST operators and the operations they execute are shown in the table below.

Operator	Executed Operation	Priority
(Expression)	Parentheses	High
Function Name (Parameter List)	Call function	
EXPT	Exponential operation	
-, NOT	Invert	
*	Multiply	
/	Divide	
MOD	Take the remainder	
+	Add	
-	Subtract	
<, >, ≤, ≥	Comparison operation	
=	Equal to	
<>	Not equal to	
AND	AND	
XOR	XOR	Low
OR	Or	

10.8.5 Operand

Operands can be addresses, values, variables, array variables, structure variables, elements of an array/structure variable, function calls, functional block outputs, etc.

The data types shall be the same in statements that process operands. If users need to process operands of different types, the type conversion shall be performed beforehand.

In the following example, the integer variable Var1 is converted to a real variable before it is added to the real variable R1.

R2: =R1+SIN (INT_TO_REAL (Var1));

10.8.6 ST Statement

10.8.6.1 Assignment

An assignment statement assigns the evaluation result of an expression to a variable.

The data types on both sides of the assignment operator shall be the same. The data type of the assignment statement is determined by the data type of the variable on the left.

10.8.6.1.1 Syntax

A : = B;

Where A is a variable, and B is an expression. Expressions can be values, variables, operands, functions, or functional blocks.

10.8.6.1.2 Example

- Assign the value to the variable

  If A and B are basic data types, a single value of B is passed to A. If A and B are derived data types, all elements of A are replaced by the current values of the corresponding elements of variable B.

  - Statement C := 25;

    Used to assign the value 25 to variable C.

- Assign the operation value to a variable

  - Statement X := (A + B - C) * D;

    Used to assign the operation result of (A + B - C) * D to the variable X.

- Assign the functional block output to the variable

  - Statement A := MY_TON.Q;

    Used to assign the Q output value of the MY_TON functional block (an instance of the TON functional block) to variable A.

- Assign the function return value to the variable

  - Statement B := C MODA;

    Used to call the MOD (modulus) function and assign the operation result to variable B.

10.8.6.2 IF statement

The IF statement selects the branch of the program to be executed based on a condition. The condition is a Boolean expression. If the expression value is TRUE, and the condition is met, the THEN statement is executed; And if the expression value is FALSE, and the condition is not met, the relevant statement is executed according to the branch.

10.8.6.2.1 Syntax

IF <BOOL Expression 1> THEN (Determine whether the BOOL expression is TRUE)

<IF Statement> (Execute the statement if TRUE)

ELSIF<BOOL expression 2 >THEN (Optional. Check whether the BOOL expression is true)

<ELSIF statement 1>　　(Execute this statement if it is true)

...

...

ELSIF <BOOL Expression n> THEN

<ELSIF Statement n - 1>

ELSE　　(Optional. Execute this ELSE statement if none of the preceding IF statements meet the condition)

<ELSE Statement>

END_IF　　(The IF statement ends)

Description:

- If Conditional Expression 1 is TRUE, only the IF instruction is executed and no other statements are executed.
- Otherwise, the ELSIF expression values are determined sequentially until one of the expressions is TRUE, and the statement following the expression is executed.
- If none of the preceding conditional expression values are TRUE, the statement following ELSE is executed.
- Conditional expressions can be BOOL variables, comparison operations, and logical operations.
- The structure of the IF statement, ELSE statement, and ELSIF statement all follow the syntax structure above. Please choose according to the user program when applying it in practice.

10.8.6.2.2　Simple IF statement

A simple IF statement is an IF structure with only one determination condition. If the IF determination condition is TRUE, the statement is executed. Otherwise, it is not executed.

10.8.6.2.2.1　Syntax

IF<BOOL expression>THEN　　(Check whether the BOOL expression is true)

<IF Statement>　　(Execute the statement if TRUE)

END_IF　　(The IF statement ends)

10.8.6.2.2.2　Example

When the temperature value is greater than the set value or the stop button is pressed, the pump stops.

No.	Variable Name	Address	Variable Description	Variable Type	Initial Value	Power Fail Safeguard	NetWork Public
0001	Temp		Temperature point	DINT	0	FALSE	Not Public
0002	HEX_ENGIN1			HEX_ENGIN		FALSE	Not Public
0003	Temp_M		Engineering quantity temperature value	REAL	0	FALSE	Not Public
0004	PUMP		Pump stop signal	BOOL	FALSE	FALSE	Not Public
0005	Temp_SV		Temperature setting value	REAL	0	FALSE	Not Public
0006	Stop_B		Stop button	BOOL	FALSE	FALSE	Not Public

```
1  HEX_ENGIN1(
2  WH := Temp,
3  MU := 200,
4  MD := 50,
5  WU := 65535,
6  WD := 0,
7  AV=>Temp_M); (* Convert temperature code values to corresponding engineering values*)
8
9  IF Temp_M >=Temp_SV OR Stop_B = 1 THEN   (*When the temperature value exceeds the set value or stops pressing the button quickly *)
10
11 PUMP:=0;
12
13  END_IF;
```

10.8.6.2.3 IF ELSE statement

The IF ELSE statement is an extension of the simple IF statement; There shall be only one ELSE in an IF statement.

10.8.6.2.3.1 Syntax

IF<BOOL expression>THEN (Check whether the BOOL expression is true)

<IF Statement> (Execute the statement if TRUE)

ELSE (Execute this statement if none of the preceding IF statements meet the conditions)

<ELSE Statement>

END_IF (The IF statement ends)

10.8.6.2.3.2 Example

When the tank's low liquid level switch is closed, the inlet valve opens. Otherwise, it closes.

No.	Variable Name	Address	Variable Description	Variable Type	Initial Value	Power Fail Safeguard
0001	LSL101	%IX56.1	Liquid level switch	BOOL	FALSE	FALSE
0002	LV101	%QX0.1	Tank inlet valve	BOOL	FALSE	FALSE

```
1  IF  LSL101 =1 THEN  (* Low level switch closed*)
2      LV101 := 1;          (*Open LV101 valve *)
3  ELSE
4      LV101 := 0;          (*Close LV101 valve *)
5  END_IF
```

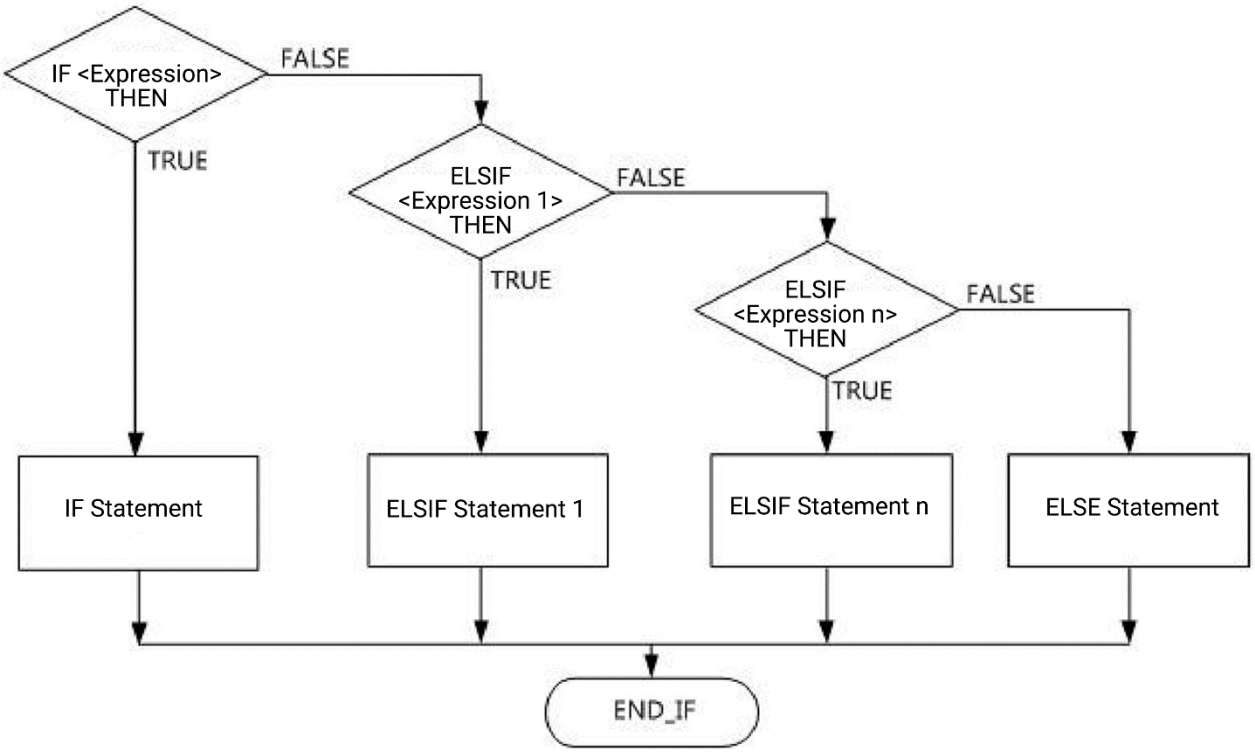
10.8.6.2.4 IF ELSIF statement

The insertion of one or more ELSIF statements into an IF ELSE statement enables the determination of multiple conditions.

10.8.6.2.4.1 Syntax

See IF Statement.

- Only if the Boolean expression value of the IF statement is FALSE, and the Boolean expression value of the ELSIF statement is TRUE, the statement or group of statements is executed.
- If the condition of the IF statement is TRUE, or the condition of the ELSIF statement is FALSE, the statement or group of statements is not executed.



10.8.6.2.4.2 Example

When the heating temperature of molecular sieve is lower than 190℃, Low Heating Temperature is displayed; When the heating temperature of molecular sieve is between 190 - 200℃, Heating OK is displayed; And when the heating temperature of molecular sieve is more than 200℃, High Heating Temperature is displayed.

No.	Variable Name	Address	Variable Description	Variable Type	Initial Value	Power Fail Safeguard	NetWork Public	Enable constant
0001	HEX_ENGIN1		Range conversion function block	HEX_ENGIN		FALSE	Not Public	
0002	TICA201	%IW8	Temperature point	DINT	0	FALSE	Not Public	
0003	TICA201_M		Engineering value of temperature	REAL	0	FALSE	Not Public	
0004	Temp_shown		Temperature display	STRING(80)	''	FALSE	Not Public	

```
1   HEX_ENGIN1(
2      WH := TICA201,
3      MU := 600,
4      MD := 0,
5      WU := 65535,
6      WD := 0,
7      AV=>TICA201_M);(*Convert temperature code values to corresponding engineering values*)
8   IF TICA201_M<190 THEN
9      Temp_shown:='Low heating temperature';
10  ELSIF TICA201_M>190 AND TICA201_M<230 THEN
11     Temp_shown:='Heating temperature is ok';
12  ELSE
13     Temp_shown:='High heating temperature';
14  END_IF
```

10.8.6.2.5 IF nesting

An IF nested statement is one or more simple IF statements inserted inside a simple IF statement. The nested IF statement has a lower priority than the upper level statement. To use the nested statement, IF is used in conjunction with END_IF.

10.8.6.2.5.1 Syntax

IF<BOOL expression 1 >THEN (Check whether the BOOL expression is true)

IF <BOOL Expression 2 > THEN (Execute the nested IF statement if TRUE)

END_IF (The nested IF statement ends)

END_IF (The upper-level IF statement ends)

10.8.6.2.5.2 Example

When there is an operating signal for the main oil pump P101A, and the lubricant pressure PIA103 is lower than 0.18 MPa, the auxiliary oil pump P101B starts.

No.	Variable Name	Address	Variable Description	Variable Type	Initial Value	Power Fail Safeguard
0010	XL101			BOOL	FALSE	FALSE
0011	PIA103			DINT	0	FALSE
0012	P101B			BOOL	FALSE	FALSE

```
1   HEX_ENGIN1(
2   WH := PIA103,
3   MU := 3,
4   MD := 0,
5   WU := 65535,
6   WD := 0,
7   AV=>PIA103_M); (* Convert temperature code values to corresponding engineering values*)
8
9   IF XL101 = 1 THEN
10
11    IF  PIA103_M <018  THEN
12     P101B_S:=1;
13
14    END_IF;
15  END_IF;
```

10.8.6.3 CASE statement

10.8.6.3.1 Syntax

CASE <Conditional Variable> OF (Determine the conditional variable value)

<Value 1> : <Statement 1> (Execute the statement if Value 1)

<Value 2>:<Statement 2> (Execute this statement if the variable value is Value 2)

<Value 3, Value 4, Value 5>: <Statement 3> (Execute the statement if the conditional variable value is Value 3, Value 4, or Value 5)

<Value 6 Value 10,>:<Statement 4> (Execute this statement if the variable value is in the range of Value 6 to Value 10)

…

…

<Value n> : <Statement n>

ELSE <ELSE Statement> (Execute the ELSE statement if the conditional variable value does not match any of the values)

END_CASE (CASE ends)

Note the following points when writing a CASE statement:

- The conditional variable shall be an integer or an enumeration;
- In the CASE clause, the value shall only be an integer constant or an enumeration constant, and no variable name or expression shall be used;
- The CASE clause value shall not be used overlappingly, including overlappingly in a single clause or in multiple clauses at the same time.

10.8.6.3.2 Rules

To execute a CASE statement, follow the rules below:

- If the conditional variable value matches <Value n>, <Statement n> is executed;
- If the conditional variable does not have any matching value, <ELSE Statement> is executed;
- If several values of the conditional variable are required to execute the same instruction, the values can be written one after the other and separated by commas. In this way, the same statement is executed;
- If the conditional variable executes the same statement within a certain range of values, users can write the initial and end values of the range values to separate them by operator ".." . Thus, the same statement is executed.

10.8.6.3.3 Example

The liquid level of the ammonia tank is measured by the liquid level meter LIA1001. When the liquid level is > 90%, the screen displays HIGH; When the liquid level is between 25% and 90%, the screen displays OK; When the liquid level is < 25%, the screen displays LOW; And when the level is lower than 1%, the alarm is issued.

No.	Variable Name	Address	Variable Description	Variable Type	Initial Value	Power Fail Safeguard
0020	Level_S			STRING(80)	''	FALSE
0021	L_ALARM			BOOL	FALSE	FALSE

```
1   HEX_ENGIN1(
2   WH := LIA1001,
3   MU := 100,
4   MD := 0,
5   WU := 65535,
6   WD := 0,
7   AV=>LIA1001_M);   (* Convert temperature code values to corresponding engineering values*)
8
9   LIA1001_INT := REAL_TO_INT(LIA1001_M);
10
11  CASE LIA1001_INT OF
12      91..100:
13      Level_S:='HIGH';
14      25..90:
15      Level_S:='OK';
16      2..24:
17      Level_S:='LOW';
18      0..1:
19      L_ALARM:=TRUE ;
20  END_CASE;
```

10.8.6.4 FOR loop

The FOR statement is used for programming where the number of loops can be predetermined. Otherwise, WHILE or REPEAT can be used.

10.8.6.4.1 Syntax

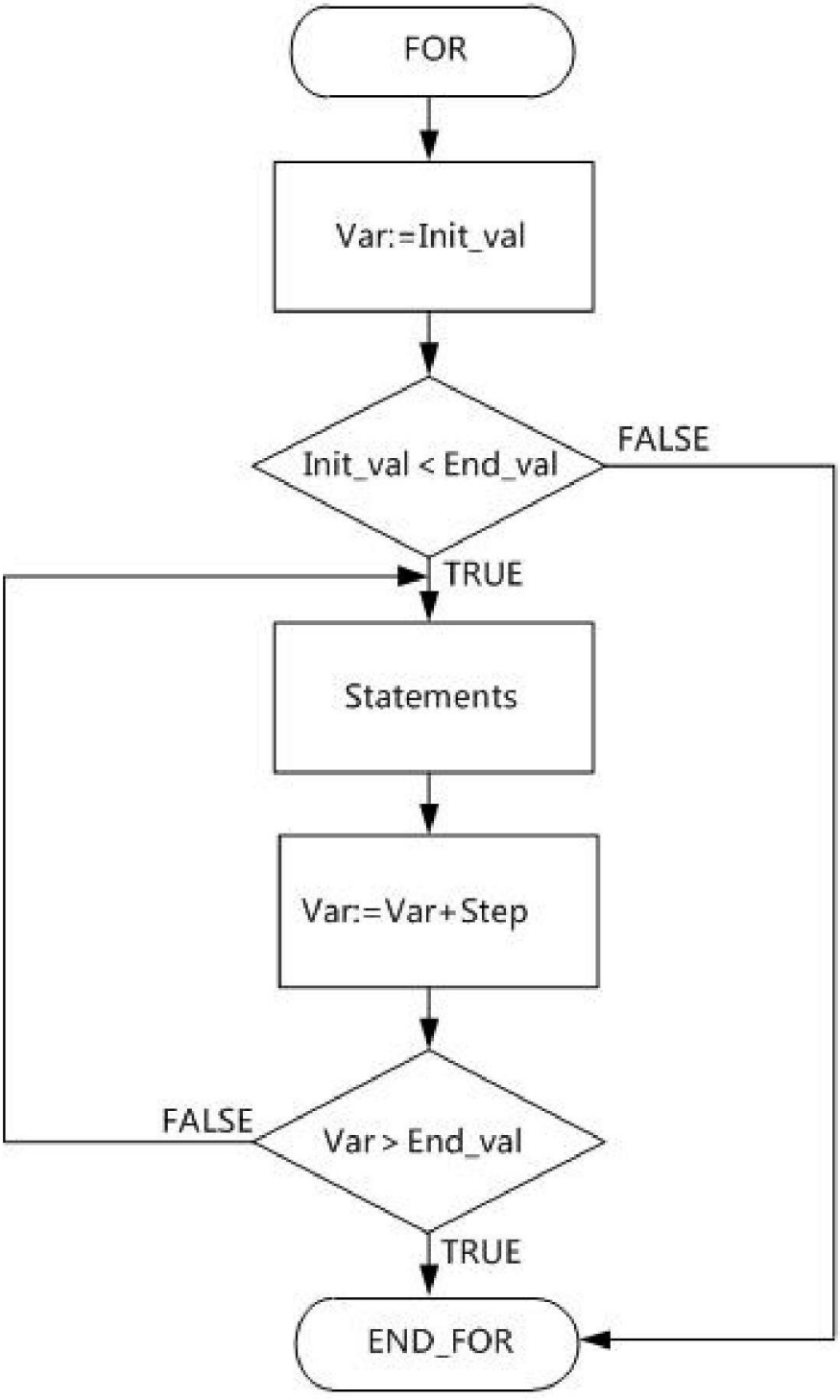
FOR<Var>:=<Init_val>TO<End_val>{BY<Step>} DO

<Statements>

END_FOR;

Specifically:

- <Var>: Control variable
- <Init_val>: The initial value of the control variable
- <End_val>: The end value of the control variable
- {BY<Step>}: Optional. Used to set the incremental value of the control variable
- <Statements>: Loop statement.
- END_FOR: The end flag of the FOR loop.



10.8.6.4.2 Rules

When the control variable value does not exceed the end value, the loop statement is executed. Before each loop, the control variable value is checked. If it exceeds the end value, the loop ends.

Each time after the execution of the loop statement, the control variable value is self-added according to

the specified incremental value. The incremental value can be any integer value. The setting of this value is optional. If not set, the value defaults to 1.

The control variable, initial value, and end value shall have the same data type (DINT or INT), and shall not be changed by repeated statements.

The initial value shall be smaller than the end value. Otherwise, no loop statement is executed.

10.8.6.4.3 Example

Write an example FOR loop to call the POU program under a single task.

No.	Variable Name	Address	Variable Description	Variable Type	Initial Value	Pov
0001	i			INT	0	FALSE
0002	Y			REAL	0	FALSE
0003	m			REAL	0	FALSE
0004	n			REAL	0	FALSE
0005	VAR1			REAL	0	FALSE

```
1  FOR i:=1 TO 5 DO (*When i is less than or equal to 5, execute the following statement, and the loop increment of i is 1*)
2    Y:= Y*3;
3    m:=n/2;
4  END_FOR
5  VAR1:=Y; (*After loop interpretation, assign the value of Y to the VAR1 variable*)
```

If the initial value of Y is 1, and the initial value of n is 32, then the value of Y is 243, the value of m is 1, and the value of i is 6 at the end of the loop.

When the POU is called under the periodic task, note that the POU is called for each IEC cycle, therefore, the values of Y and m are calculated cyclically.

10.8.6.5 WHILE loop

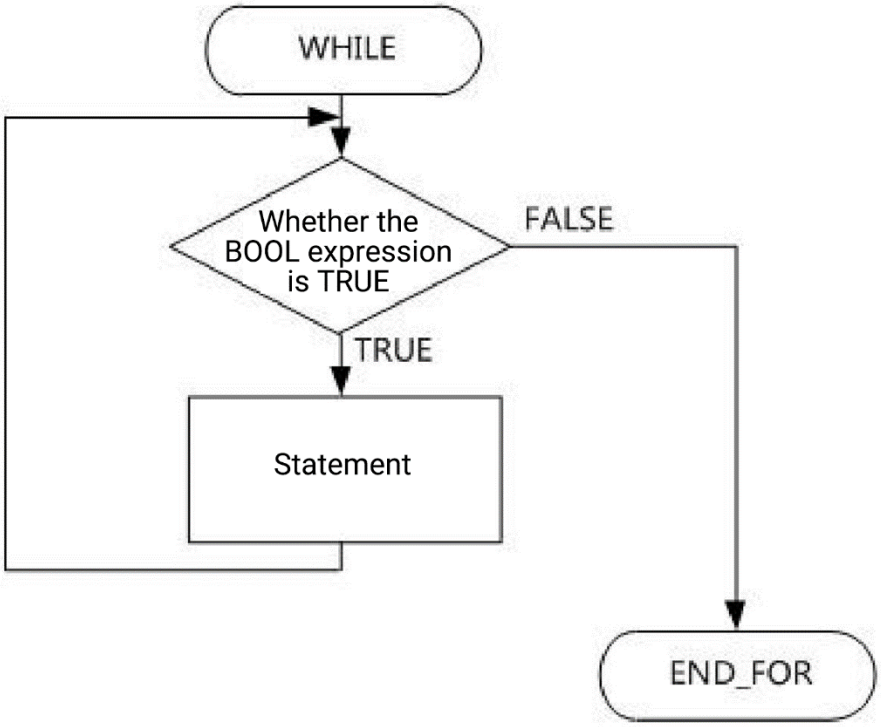
The WHILE loop is used for programming where the number of loops is unknown. It is used in a similar way to the FOR loop. The difference is that the determination condition of the WHILE loop can be any expression.

10.8.6.5.1 Syntax

WHIILE <BOOL expression> DO (Determination condition)

<Statement> (Loop statement)

END_WHILE; (WHILE loop ends)



10.8.6.5.2 Rules

- When the WHILE statement is executed, the value of the BOOL expression is first detected. If the value of the BOOL expression is TRUE, the statement is executed. After the statement is executed, the value of the expression is detected again. If it is TRUE, the statement is executed again. Until the value of the expression is FALSE, the loop ends.
- If the initial value of the BOOL expression is FALSE, the statement is not executed.
- The WHILE loop shall not be used in the following scenarios, as it may lead to an endless loop, which may cause the program to crash:
  - The value of the BOOL expression is always TRUE;
  - The WHILE loop shall not be used for synchronization between processes;

    For example, it shall not be used as a Wait Loop with an externally defined end condition.
  - The WHILE loop shall not be used in algorithms;

This is because there is no way to ensure completion of the loop end condition or execution of the EXIT statement.

10.8.6.5.3 Example

No.	Variable Name	Address	Variable Description	Variable Type	Initial Value	Power Fail Safeguard
0001	i			REAL	0	FALSE
0002	Value			REAL	0	FALSE

```
1  WHILE i< 4 DO
2      Value:=i*2;
3      i:=i+1;
4  END_WHILE;
5
```

If the initial value of i is 1, the value of Value is 6 and the value of i is 4 at the end of the loop.

10.8.6.6 REPEAT loop

The REPEAT loop differs from the WHILE loop in that it detects the end condition only after the execution of the loop statement. That means that the loop statement is executed at least once, regardless of whether the end condition is met or not.

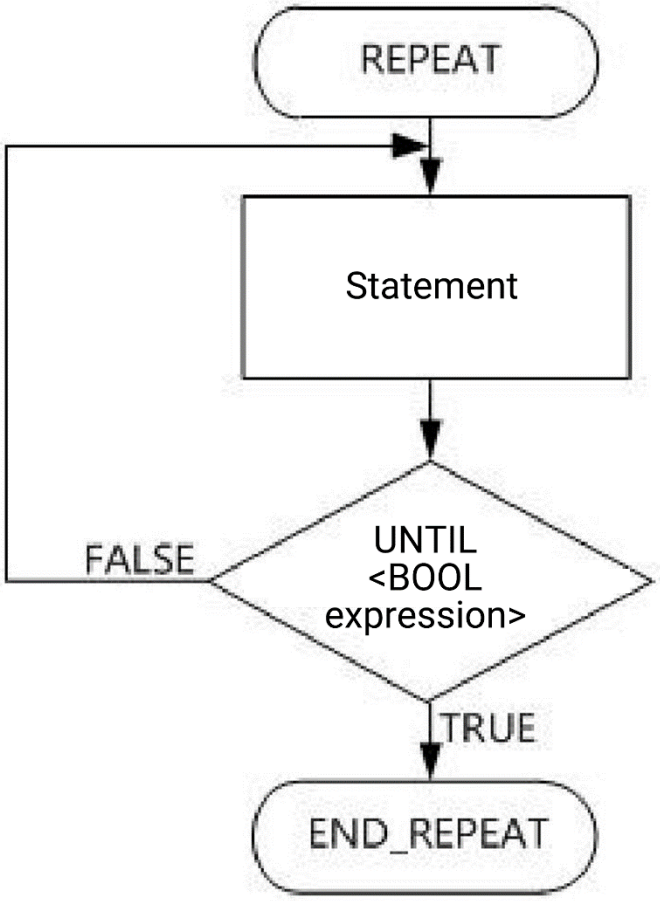
10.8.6.6.1 Syntax

REPEAT (*Loop starts*)

<Statement> (Loop statement)

UNTIL <BOOL Expression> (*Exit condition*)

END_REPEAT; (*Loop ends*)



10.8.6.6.2 Rules

- When the value of the BOOL expression is FALSE, the statement is looped. Until the value of the BOOL expression is TRUE, the REPEAT loop exits.
- If the value of the BOOL expression is already TRUE on the first execution, the statement is executed only once.
- REPEAT loop should not be used in the following scenarios because it may lead to an endless loop and thus cause a program crash:
  - The value of the BOOL expression is always FALSE;
  - REPEAT loop cannot be used for synchronization between procedures.

    For example, it shall not be used as a Wait Loop with an externally defined end condition.
  - The REPEAT loop shall not be used in algorithms;

This is because there is no way to ensure completion of the loop end condition or execution of the EXIT statement.

10.8.6.6.3 Example

No.	Variable Name	Address	Variable Description	Variable Type	Initial Value	Power Fail Safeguard	NetWork Public	Enable constant
0001	Value			REAL	0	FALSE	Not Public	
0002	i			REAL	0	FALSE	Not Public	

```
1  REPEAT
2      Value:=i*2;
3      i:=i+1;
4      UNTIL i>4
5  END_REPEAT
```

The initial value of i is 1. The POU is called by an event or status task. After the task runs once, the value of Value is 8 and the value of i is 5.

Note that if the POU is called by a periodic task, the run results are different from the run results of a single task call. The POU is executed once per IEC task cycle. Each time it is executed, the values of Value and i are executed once, leading to run results in an infinite loop to calculate the values of Value and i.

10.8.6.7 EXIT statement

10.8.6.7.1 Syntax

The EXIT statement is used in FOR, WHILE, and REPEAT loop statements. The EXIT statement exits the loop statement as soon as the end condition of the EXIT statement is met, regardless of whether the end condition is met or not. If the EXIT statement is inside a nested loop statement, it exits the loop in which EXIT is located and jumps to the first statement after the inner loop end statement (END_FOR, END_WHILE, or END_REPEAT) for further execution.

10.8.6.7.2 Example

If the value of FLAG is 0, the value of SUM is 15 after the statement is executed.

If the value of FLAG is 1, the value of SUM will be 6 after the statement is executed.

No.	Variable Name	Address	Variable Description	Variable Type	Initial Value	Power Fail Safeguard	NetWork Public	Enable constant
0001	SUM			DINT	0	FALSE	Not Public	
0002	I			DINT	0	FALSE	Not Public	
0003	J			DINT	0	FALSE	Not Public	
0004	FLAG			BOOL	FALSE	FALSE	Not Public	

```
1  SUM:=0;
2  FOR I :=1 TO 3 DO
3     FOR J:= 1TO 2 DO
4        IF FLAG=1 THEN
5        EXIT;
6        END_IF;
7        SUM:=SUM+J;
8     END_FOR;
9  SUM:=SUM +I;
10 END_FOR;
```

10.8.6.8 RETURN statement

10.8.6.8.1 Semantics

It provides early exit from a function, functional block, or program.

10.8.6.8.2 Example

When P1 is TRUE, P2 performs self-addition; And when P1 is FALSE, P2 stops self-addition and exits the IF statement.

No.	Variable Name	Address	Variable Description	/ariable Type	Initial Value	Power Fail Safeguard	NetWork Public	Enable constant
0001	p1			BOOL	FALSE	FALSE	Not Public	
0002	p2			DINT	10	FALSE	Not Public	

```
1  IF P1=TRUE THEN
2     p2:=p2+1;
3  ELSIF p1=FALSE THEN
4     RETURN;
5  END_IF;
```

10.8.6.9 Control statements for function and functional block

The control statements for function and functional block consist of a mechanism for calling the functional block, and a mechanism for returning control to the calling entity before physically ending the function or functional block.

The function (FUN) evaluation shall be called as part of the expression evaluation.

10.8.6.9.1 Call functional block

The functional block (FB) shall be called via a statement that consists of the functional block instance

name followed by an argument list in parentheses.

In the following example, a timer is called by assigning values to the two parameters IN and PT, and then the value of the result variable Q is assigned to the variable OUT.

The result variable is expressed as: Functional block name.variable name.

```
1  CMD_TMR (IN:= DM01, PT:=T#300ms);
2  OUT:=CMD_TMR.Q;
```

10.8.6.9.2 Call function

A textual call to a function shall consist of the function name followed by an argument list. The arguments shall be separated by commas. The left and right sides of the list shall be delimited by parentheses.

Rules for function calls:

- The assignment of a function output variable shall either be null or assign a value to the variable.
- The assignment to the VAR_IN_OUT argument shall be a variable.
- The assignment to the VAR_INPUT argument can be a null value, a constant, a variable, or a function call. In the latter case, the function result is used as the actual argument.

10.8.7 Create Expression

10.8.7.1 Overview

When configuring an expression, users need to add operands and operators from left to right. Users can enter them manually or select them automatically. When users enter manually, the ST Editor automatically associates relevant operands and operators based on the characters entered. Or users can select operands and operators via the Input Assistant.

Characters that can currently be associated include:

- POU names, including program, function, and functional block names.
- Global and PRG variable names.
- Variable members of structures or functional blocks.
- The function is associated only with the function name; And the variable names of the function shall be entered manually or selected via the Input Assistant.

10.8.7.2 Rules

To enter an operand, follow the rules below:

- The format of variable members of functional blocks or structures: Functional block or structure variable name.Member variable name. Enter . after the variable name to automatically associate the member variable name.
- The characters in the expression shall be English characters.
- Press Enter for a line feed.

10.8.7.3 Requirement

The ST program has been created

10.8.7.4 Steps

To enter an operand manually, follow the steps below:

(1) At the current position of the cursor, enter the operand.

(2) The associated operands are displayed below the input line, with the first operand selected by default.

(3) Use ↑ and ↓ keys to select the operand.

(4) Press Enter.

Or select the operand via Help Manager:

(1) At the current position of the cursor, Press F2. The Help Manager window is displayed.

(2) Select the operand or operator.

(3) Click OK.

10.8.8 Copy Statement

To reuse an ST statement line, copy the desired statement or program segment and paste it into the target position, which can be the current POU or another POU.

10.8.8.1 Requirement

There are available ST program segments.

10.8.8.2 Steps

Follow the steps below to copy the ST program segment:

(1) Hold down the left mouse button and drag the cursor.

(2) After selecting the program segment, release the mouse. The selected program segment area is highlighted in blue.

(3) You can execute the Copy command in the following ways:

- Selected part: Click Copy in the right-click menu.
- Menu Bar: [Edit] - [Copy];
- Shortcut key: Ctrl + C.

10.8.8.3 Result

The program is copied to the clipboard, and can be pasted to the target position.

10.8.9 Cut Statement

To cut the current statement and paste it into another position or another POU, execute a cut operation.

10.8.9.1 Requirement

There are available ST program segments.

10.8.9.2 Steps

To cut an ST program segment, follow the steps below:

(1) Hold down the left mouse button and drag the cursor.

(2) After selecting the program segment, release the mouse. The selected program segment area is highlighted in blue.

(3) Execute the Cut command via:

- Selected part: Click Cut in the right-click menu.
- Menu bar: Choose [Edit] - [Cut].
- Shortcut key: Ctrl + X.

10.8.9.3 Result

The object is cut and saved in the clipboard. You need to paste it to the target position.

10.8.10 Paste Statement

You can paste the copied or cut statements to a certain position in the program.

10.8.10.1 Requirement

Statements to be pasted have been copied or cut.

10.8.10.2 Steps

Follow the steps below to paste statement lines:

(1) Click the position where the statement line to be pasted.

(2) You can execute the Paste command in the following ways:

- Programming area: Click Paste in the right-click menu.
- Menu bar: Choose [Edit] - [Paste].
- Shortcut key: Ctrl + V.

10.8.11 Delete Statement

10.8.11.1 Requirement

There are available elements.

10.8.11.2 Steps

Follow the steps below to delete statement lines:

(1) Hold down the left mouse button and drag the cursor.

(2) After selecting the program segment, release the mouse. The selected program segment area is highlighted in blue.

(3) You can execute the Delete command in the following ways:

- Selected part: Click Delete in the right-click menu.
- Menu bar: Choose [Edit] - [Delete].
- Shortcut key: Delete.

10.8.11.3 Reference message

Undo

10.8.12 ST Programming Example

ST1(PRG).st | Manua(PRG).ld | Auto(PRG).st | Failure(PRG).st

No.	Variable Name	Address	Variable Description	Variable Type	Initial Value	Power Fail Safeguard
0007	Initialize_Sign			BOOL	FALSE	FALSE
0008	p2			BOOL	FALSE	FALSE

```
IF NOT Initialize_Enable  THEN

 Initialize_Sign:= FALSE;
 ELSE
 Initialize_Sign:= TRUE;

  END_IF



CASE i OF
 10:
 Manua();




 20:
 Auto();




 30:
 Failure ();
 END_CASE
```

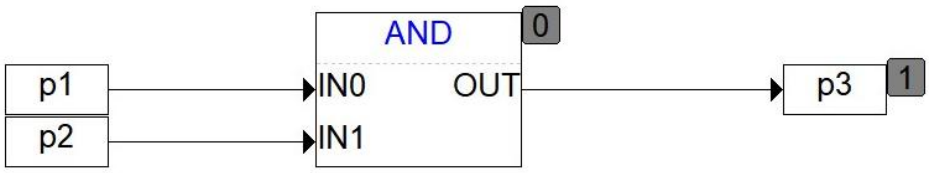
10.9 Create CFC Program

10.9.1 CFC Programming Overview

CFC is short for continuous function chart. It is a visual graphic editing language, which features easy operation and intuitive display. It is suitable for the configuration of continuous process control.

As shown in the figure below, the CFC Editor is a graphical editor that is edited via the Toolbar or right-click menu commands.

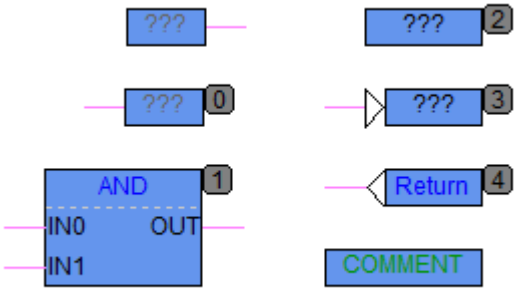
No.	Variable Name	Address	Variable Description	Variable Type	Initial Value	Power Fail Safeguard	NetWork Public	Enable constant
0001	p1			INT	0	FALSE	Not Public	
0002	p2			INT	0	FALSE	Not Public	
0003	p3			INT	0	FALSE	Not Public	



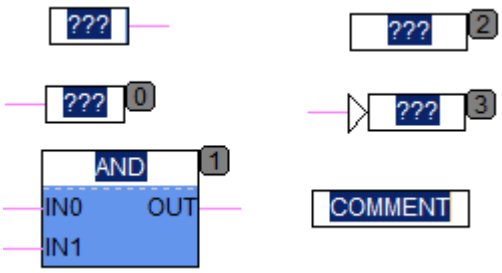
10.9.2 Cursor Position

Each text is a candidate position for the cursor. The selected text fades to blue and can be modified. Here is an example of the candidate positions for the cursor.

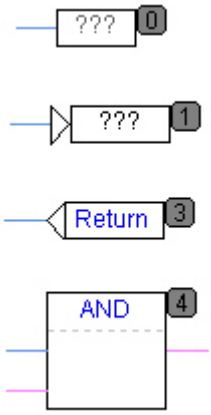
(1) Select the input, output, jump, tag, return, block, and comment elements. Click the text area. The cursor is placed, as shown in the figure.



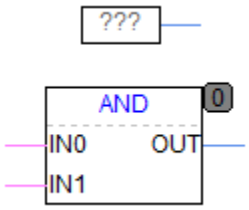
(2) You can double-click the text area of an input element, output element, jump, tag, return element, block element, or comment to select the element, as shown in the figure below.



(3) Select the input of the output, jump, tag, block, and return elements. Click the input. The cursor (in blue) is placed. Only one element can be selected at a time. As shown in the figure.



(4) Select the output of the input and block elements.



10.9.3 Insert Input Component

10.9.3.1 Introduction

This command is used to insert an input. The text ??? displayed can be selected and replaced by a variable or constant.

The input component has only one output pin, which can only be used as an input. For the input component, users can declare variables and assign them values while online.

Users can execute it via:

- Menu Bar: [Insert] - [Input Component];
- Programming area: Right-click the blank area. Click Input Input Component;
- Toolbar:  ;
- Shortcut key: Ctrl + I.

10.9.3.2 Requirement

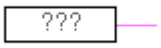
The CFC program has been opened.

10.9.3.3 Steps

Follow the steps below to insert an input component:

(1) Right-click the program editing area. In the right-click menu, click Input component. A floating input component is displayed at the cursor.

(2) The newly inserted input component moves with the mouse. Move it to the appropriate position and click the position. The insertion is successful. As shown in the figure.



10.9.4 Insert Output Component

This command is used to insert an output component. The text ??? can be selected and replaced by a variable.

The output component has only one input pin, which is used to receive data from the output pins of other components. The output component is associated with an execution number to indicate the execution sequence.

Users can execute it via:

- Programming area: Right-click in the blank area and click Output component.
- Toolbar:  ;
- Shortcut key: Ctrl + U.

10.9.4.1 Requirement

The CFC program has been opened.

10.9.4.2 Steps

Follow the steps below to insert an output component:

(1) Right-click in the program editing area and click Output Component in the menu that appears. A floating output component then appears at the cursor.

(2) Move the cursor to the appropriate position to insert the output component. As shown in the figure.

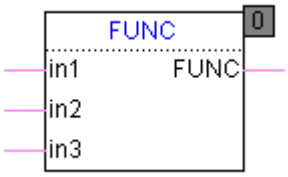
10.9.5 Insert Block Component

10.9.5.1 Introduction

The block component contains functions, functional blocks, and programs. The enabled terminals EN and ENO, which themselves contain the BOOL, are in the same state. If the block is in the hidden EN and ENO terminal state, the EN and ENO states are TRUE; Otherwise, the state depends on the input state.

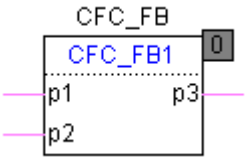
- Function

  Functions for performing certain operations can be used in the CFC configuration as blocks (for calling by CFC users). Function blocks can be provided by libraries or customized by users using IEC programming language. A function exists with a number of input variables and a unique return value.



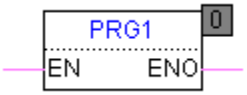
- Functional Block

  Functions for performing certain operations can be used in the CFC configuration as blocks (for calling by CFC users). Functional blocks can be provided by standard libraries or customized by users using IEC programming languages. A functional block exists with a number of inputs and a number of outputs.



- Procedure

  Program blocks are separate units that perform certain operations, which can be used in the CFC configuration as blocks (for calling by CFC users), and can be called directly by the program. A program block exists with an input and an enabled output (Enable is not displayed by default. You can select "Enable" in the CFC area of "Project > Option > Configuration Language" to display it).



10.9.5.2 Requirement

The CFC program has been opened.

10.9.5.3 Steps

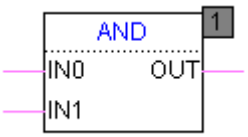
Follow the steps below to insert an block component:

(1) Right-click the program editing area. Select the following methods to insert block component.

- Programming area: Right-click in the blank area and click Block Component.
- Toolbar: ;
- Shortcut key: Ctrl + B.

(2) A floating block component is displayed at the cursor.

(3) Move the cursor to the appropriate position to insert the block component. As shown in the figure.

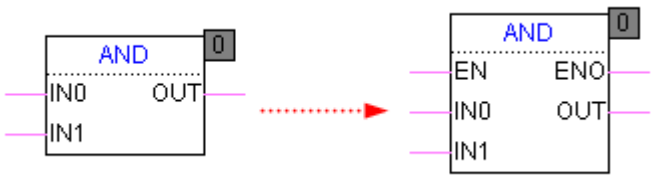


10.9.6 Enable Block Component

10.9.6.1 Introduction

This command is used to show and hide the enabled input and enabled output of the selected block component. Repeated activation of this command allows the block component to switch between the showed and hidden states on the enabled terminal.

In the CFC program, the enabled terminal of the block component is hidden by default.



10.9.6.2 Requirement

A block component has been added.

10.9.6.3 Steps

Follow the steps below to enable a block component:

(1) Select the block component. Execute the Enable command via:

- Programming area: Right-click the block component. Click Advanced > Enable;
- Toolbar:  ;
- Shortcut key: Ctrl + E.

10.9.7 Configure Block Component Pin

10.9.7.1 Add input pin

10.9.7.1.1 Introduction

This command is used to add an input to the block component.

10.9.7.1.2 Requirement

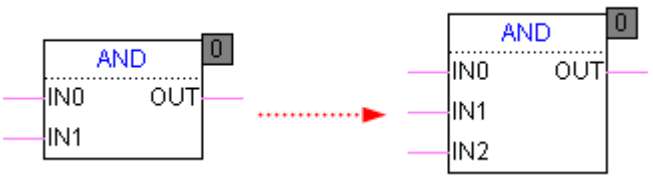
A block component has been added.

10.9.7.1.3 Steps

To add a block component input pin, follow the steps below:

Select the block component. Execute the following operations:

- Programming area: Right-click the block component and choose [Advanced] - [Multiple Input Components].
- Toolbar:  ;
- Shortcut key: Ctrl + D.



Select the added input. Execute the Delete command to delete the pin.

10.9.7.2 Show or hide pin

10.9.7.2.1 Introduction

Users can set the functional blocks to show and hide the block component input and output pins.

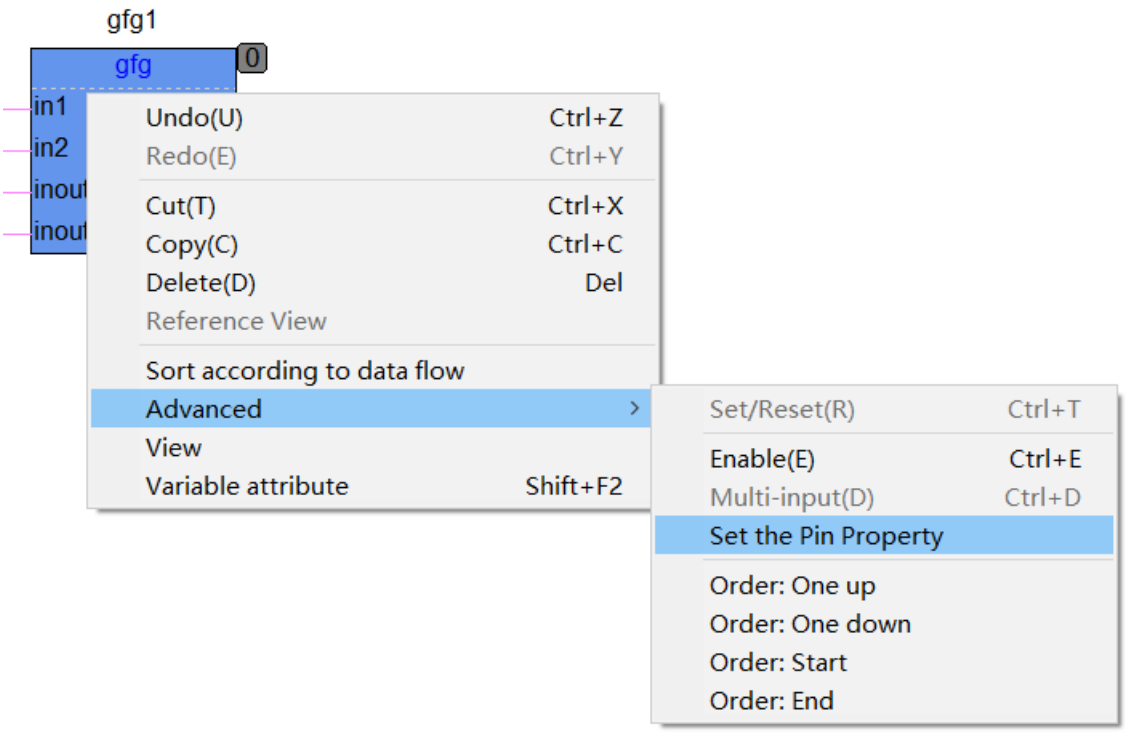
10.9.7.2.2 Requirement

The functional block has been added.

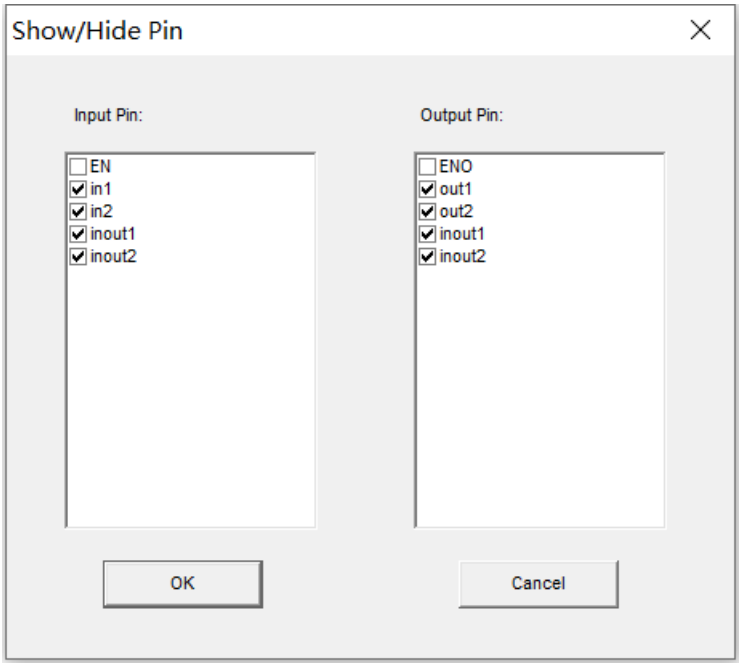
10.9.7.2.3 Steps

To set the showing and hiding of the functional block pins, follow the steps below:

(1) Right-click the functional block. Click [Advanced] - [Set the Pin Property].



(2) Check the input and output pins that need to be displayed. The unchecked pins are not displayed on the functional block.



10.9.8 Rename Block Component

10.9.8.1 Introduction

The block component is added as an AND block by default. However, you can manually modify the block component name to the needed one. When the block component name is entered, the software automatically associates all the block component names that match the entered keyword for choice, making it easy to enter quickly.

10.9.8.2 Requirement

A block component has been created.

10.9.8.3 Steps

To rename the block component, follow the steps below:

(1) Double-click the block component name. The cursor is displayed at the block component name.

(2) Select the block component name. Enter the new name. All associated block components are automatically displayed.

(3) Double-click the selected block component. The block component name is modified.

10.9.9 Insert Lable

10.9.9.1 Introduction

The lable component is used to indicate information about the execution position in the program. The component is the jump destination of the jump instruction. There is no input or output pin for the lable instruction.

10.9.9.2 Requirement

The CFC program has been opened.

10.9.9.3 Steps

Follow the steps below to insert a tag component:

(1) Users can select the lable component insertion command via:

- Programming area: Right-click in the blank area and click Tag Eomponent.
- Toolbar:  ;
- Shortcut key: Ctrl + L.

(2) A floating lable component then appears at the cursor.

(3) Move the cursor to the appropriate position to insert the tag component. As shown in the figure.

- In the text area ???, enter the tag name of the codename. The name shall be consistent with the jump name.

10.9.10 Insert Jump

10.9.10.1 Introduction

A jump enables the transition of program execution. When the instruction is executed, the program jumps to the tag referenced by the instruction. There is a unique BOOL value input pin for the jump instruction.

There shall be a BIT or BOOL variable (including one that can be converted to a BOOL variable) in order for the jump condition to be met.

10.9.10.2 Requirement

The CFC program has been opened.

10.9.10.3 Steps

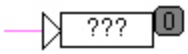
Follow the steps below to insert a jump element:

(1) You can execute the jump insertion command in the following ways:

- Programming area: Right-click in the blank area and click Jump Element.
- Toolbar:  ;
- Shortcut key: Ctrl + J.

(2) A floating jump element then appears at the cursor.

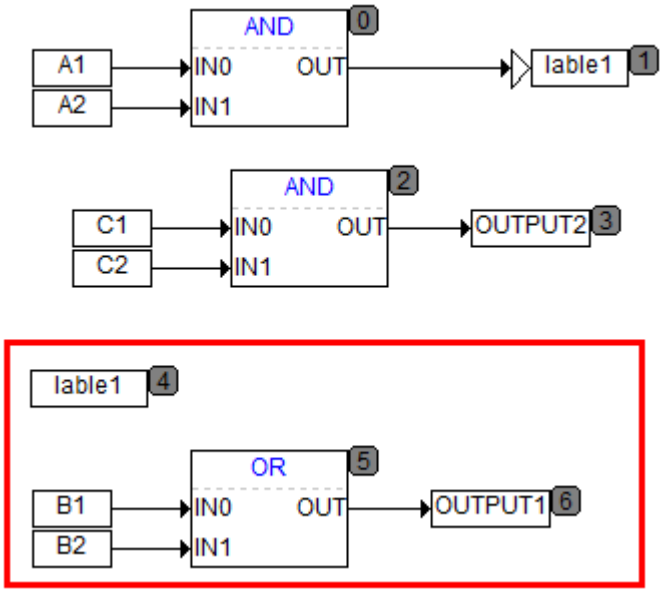
(3) Move the cursor to the appropriate position to insert the jump element. As shown in the figure.



(4) In the text area ???, In the text area ???, enter the codename to be jumped to. The name shall be consistent with the name defined for the tag.

10.9.10.4 Example

As shown in the figure below, when the jump condition is met, the CFC program jumps directly to the red marked area where the tag is located, and executes the program in the direction of the signal flow. The program lines in which signal flows and are located in front of the jump tag is not executed.



10.9.11 Insert Return

10.9.11.1 Introduction

After this instruction is executed, the current POU returns directly to the program that called the POU. There is a unique input pin for the return instruction, with a BOOL input value.

10.9.11.2 Requirement

The CFC program has been opened.

10.9.11.3 Steps

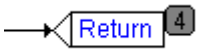
Follow the steps below to insert a return element:

(1) You can execute the return insertion command in the following ways:

- Programming area: Right-click in the blank area and click Return Element.
- Toolbar:  ;
- Shortcut key: Ctrl + R.

(2) A floating return element then appears at the cursor.

(3) Move the cursor to the appropriate position to insert the return element. As shown in the figure.



10.9.12 Insert Comment

10.9.12.1 Introduction

The comment element does not participate in the execution of the program. It is used to insert explanatory text information in the CFC program interface. Similar to comment information in other languages, there is no input or output information for this instruction.

10.9.12.2 Requirement

The CFC program has been opened.

10.9.12.3 Steps

Follow the steps below to insert a comment element:

(1) You can execute the comment insertion command in the following ways:

- Programming area: Right-click in the blank area and click Comment Element.
- Toolbar:  ;
- Shortcut key: Ctrl + K.

(2) A floating comment element then appears at the cursor.

(3) Move the cursor to the appropriate position to insert the comment element. As shown in the figure.

10.9.13 Delete Element

10.9.13.1 Introduction

This command is used to delete one or more CFC elements.

10.9.13.2 Requirement

The element has been added to the CFC program

10.9.13.3 Steps

Follow the steps below to delete an element:

(1) You can execute the comment insertion command in the following ways:

- □ Programming area: Right-click the block element and click Delete.
- □ Shortcut key: Delete.

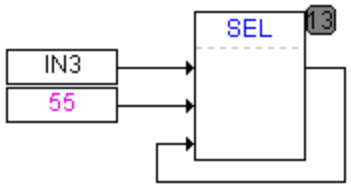
(2) When an element is deleted, it is deleted along with the connecting line.

10.9.14 Add Connecting Line

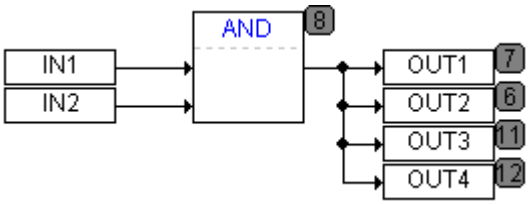
10.9.14.1 Introduction

Connect the pins of the block components via connecting lines to configure the user logic.

The input pin of an component can only be connected to one variable (the output pin of this component or the output pin of another component). While the output pin of an component can be connected to multiple variables (input variables of this component or input variables of other components), as shown in the figure.



(a)



(b)

The Connection Ways of Pin Signal

10.9.14.2 Requirement

The components that need to be connected have been created.

10.9.14.3 Steps

Follow the steps below to connect an component pin:

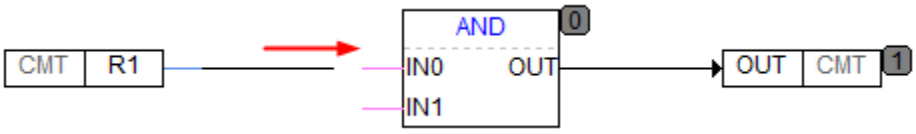
(1) Select a pin of the component. Drag the cursor to the pin of another component. The connecting line is automatically adjusted when the component is moved.

(2) Drag the cursor to the target pin position. Release the left mouse button. The connecting line is created automatically.

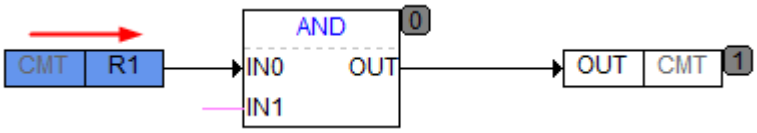
10.9.14.4 Example

There are two ways to connect the R1, AND and OUT components, as shown in the figure.

- Click the output pin of R1. Drag the cursor to the input pin of AND. Release the left mouse button. As shown in the figure.
- Drag R1 to coincide its output pin line with the input pin line of AND. Release the left mouse button. The connection is complete, as shown in the figure.



(a)



(b)

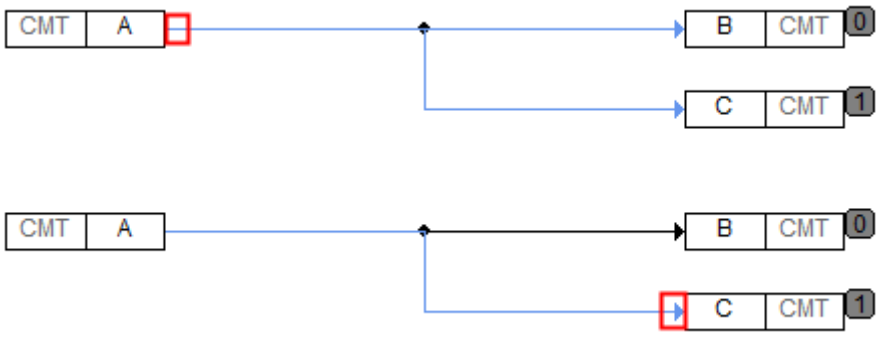
Connection Lead

10.9.15 Delete Connecting Line

10.9.15.1 Introduction

This operation is used to delete the connecting line between elements.

There are two ways to delete the connecting line, as shown in the figure. Select the output pin of A. Execute the Delete command. All lines connected to the output pin A are deleted. Select the input/output pin of C. Execute the Delete command. Only the connecting line between the input pin of C and the output pin of A is deleted. While the connecting line between A and B is not affected.



Delete the Selected Position of Connection Lead

10.9.15.2 Steps

To delete a connecting line, follow the steps below: Select the element pin. Then right-click Delete command, or press Delete on the keyboard. The connecting line is deleted.

10.9.16 Modify Component Text

10.9.16.1 Introduction

When a CFC component is added, the default component text is ???. This text can be modified as needed.

10.9.16.2 Steps

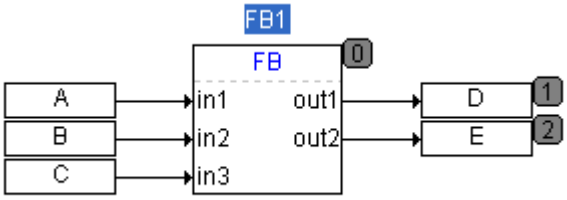
Follow the steps below to modify the component text:

(1) Double-click the text area. The component text editing box is displayed.

(2) In the editing box, enter the new text.

10.9.16.3 Example

Take the CFC components shown in the figure below as an example. To modify the functional block text FB1: Users can double-click the functional block text FB1. The editing box is displayed at the original text, as shown in the figure below. In the editing box, users can enter the new text.



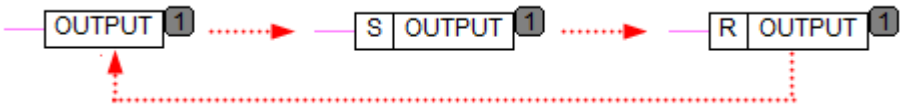
Double Click the Text to Edit Text

10.9.17 Set/Reset Output Component

10.9.17.1 Introduction

The CFC output element can be set/reset. A set output element can be reset, and a reset output element can be restored.

By continuously executing the set/reset operation, the output element is transitioned between the set, reset, and restored states. As shown in the figure.



Execute Set and rReset Operations Continuously

10.9.17.2 Requirement

The output element has been added.

10.9.17.3 Steps

To set/reset an output element, follow the steps below:

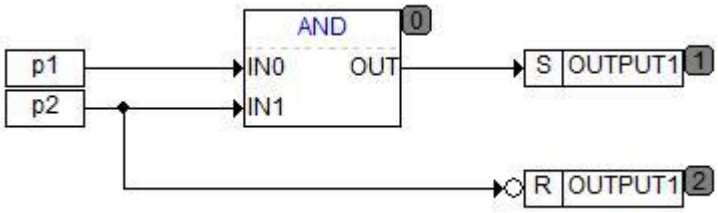
Select the output element. Execute the set/reset operation via:

- Programming area: Right-click the output element. Click Advanced > Set/Reset;
- Toolbar:  ;
- Shortcut key: Ctrl + G.

10.9.17.4 Example

As shown in the figure, for the set output, if OUT of the AND block element is TRUE, OUTPUT1 is set to TRUE and remains unchanged in the set state, even if OUT changes to FALSE. Only when the input

element p3 changes to FALSE, OUTPUT1 is reset to FALSE and remains unchanged in the reset state. Until the set condition is met, it is set again.



10.9.18 Negate and Restore Pin

10.9.18.1 Introduction

The inverse operation is to negate the BOOL value. After an inversion operation is executed, the selected pin is negated. Select the negated pin again. Execute the negate command. The pin is restored, as shown in the figure below.

Execute Negate Operation

10.9.18.2 Steps

Follow the steps below to invert a pin:

Select the pin and execute the negate command in the following ways.

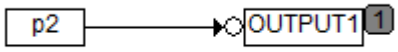
Programming area: Right-click the element pin. Click Invert;

Toolbar:  ;

Shortcut key: Ctrl + G.

10.9.18.3 Example

As shown in the figure, if p2 is TRUE, the value of OUTPUT1 is FALSE after the Invert operation; And if p2 is FALSE, the value of OUTPUT1 is TRUE.



The Example of Negating

10.9.19 Move Component

10.9.19.1 Introduction

In CFC, users can freely drag the component to the target position in the editing area. If the default collision detection is checked for the option, that is, if there is already an component at the new position to be moved to, the move fails and restores to the state before the move.

10.9.19.2 Requirement

The component has been added.

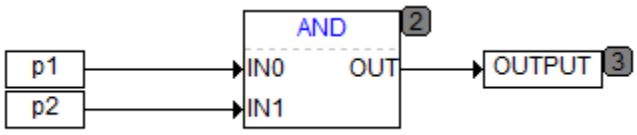
10.9.19.3 Steps

Follow the steps below to move an component:

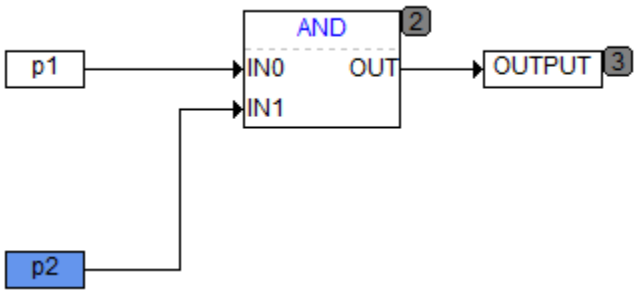
Select the component. Hold down the left mouse button. Drag the component to move it.

10.9.19.4 Example

Select the component p2. Drag it to a new position to realize the movement of the component, as shown in the figure below.



(a)



(b)

Moving Component

10.9.20 Cut, Paste and Copy

10.9.20.1 Introduction

Copy: Select the component to be copied. Execute the Copy command, or Press Ctrl + C. The selected component data is stored on the clipboard. If multiple components are selected and there are connecting lines between the components, the original connecting lines are maintained.

Cut: Select the component to be cut. Execute the Cut command. The selected component data is stored on the clipboard, and the selected component is deleted at the same time.

Paste: Add the component data that has been copied or cut on the clipboard as an component to the CFC editing interface.

10.9.20.2 Steps

To execute Copy/Cut command, follow the steps below:

(1) Right-click the component. Click Copy/Cut. After the above operations, execute the Paste operation.

(2) Right-click the blank area. Click Paste. The component is pasted to the current position.

10.9.21 Undo and Redo

10.9.21.1 Introduction

Undo is to cancel the execution of the last command, and to restore to the time before the last execution of the operation command.

The Redo command restores the operation to before the Undo command was executed.

All configuration operation commands for CFC programs are divided into two types: Those that cannot be

undone, and those that can be undone and restored. The operations on the POU itself such as renaming the POU, disabling calls and enabling calls are not included here.

- The commands that cannot be undone and restored include selecting element, selecting input, selecting output, copying element, and finding.

10.9.21.2 Steps

Follow the steps below to perform cancellation or restoration:

(1) Right-click the program editing area. In the right-click menu, select the Undo command. After the above operations are performed, the previous operation is undone. To restore, execute Step 2.

(2) Right-click in the program editing area and select the Restore command in the menu that appears. The canceled operation will be restored.

10.9.22 Adjust Execution Sequence

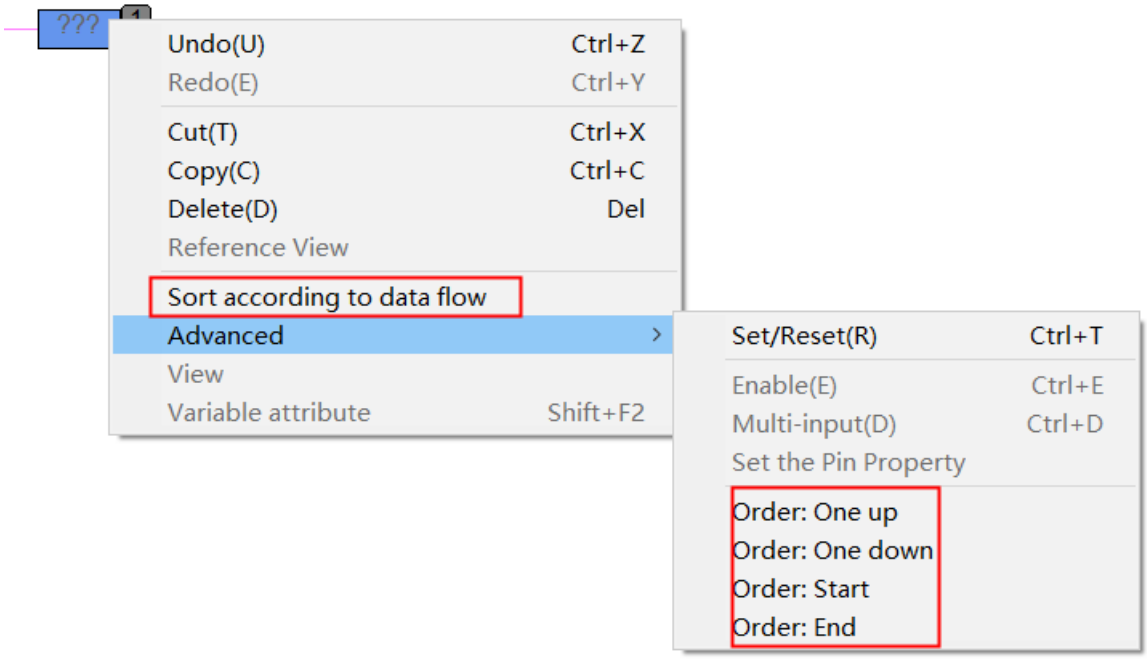
10.9.22.1 Introduction

In addition to the input element and comment, all CFC elements have a number that indicates the program execution sequence, which is displayed in the upper right corner of the element.

The CFC program completes the program flow (such as assignment, calculation, jump, and return) in sequence according to the execution sequence numbers (SNs) of each element (starting from No. 0). The execution SNs are sorted by default in the order of element input. The element execution SNs are always sorted consecutively in natural numbers from smallest to largest.

Users can choose to sort by data flow sequence or specify the element execution sequence.

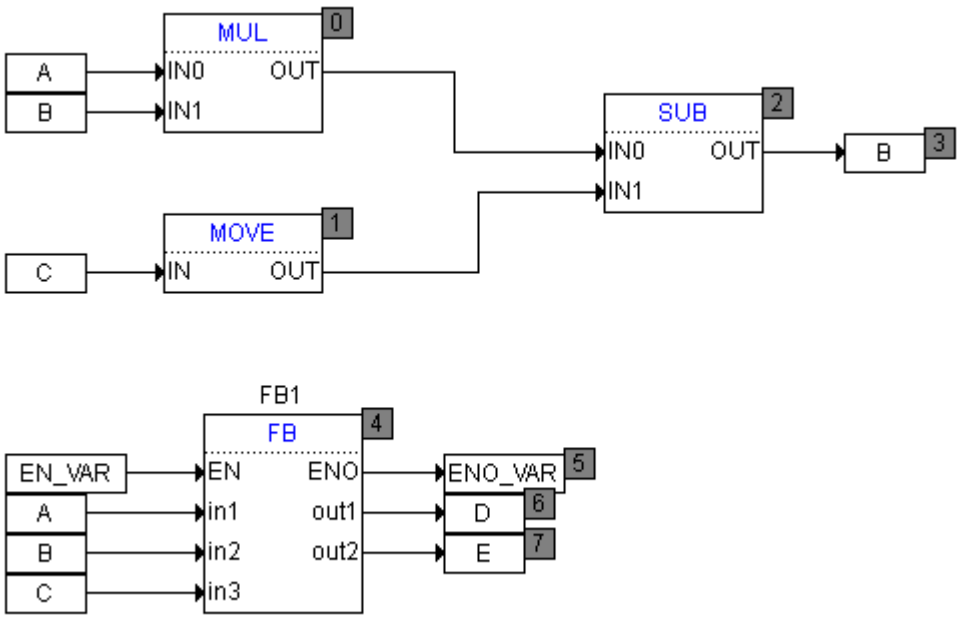
In the right-click menu of the element, select the sorting method, as shown in the figure below.



(1) Sort according to data flow

Principle of sorting by data flow: Based on inputs, execution SNs are sorted by reverse Polish notation.

Example: As shown in the figure below, a CFC program that sorts by data flow.

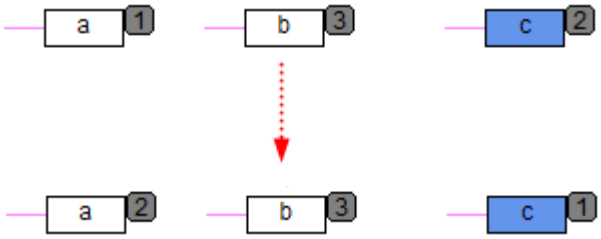


(2) One up

Select an element with an execution SN. Execute this command. The selected element exchanges its execution SN with the element whose execution SN is 1 less than it.

After selecting the element with the smallest SN and executing this command, the execution sequence remains unchanged.

As shown in the figure, the Decrease execution SN by 1 command is executed for output element c. Its execution SN is exchanged with its previous execution SN.



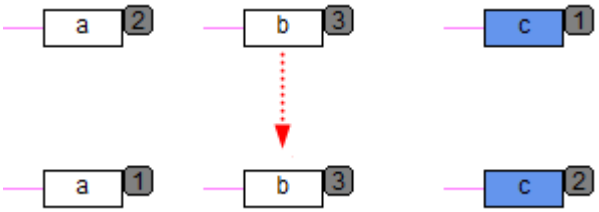
Execute the Order of One Up

(3) One down

Select an element with an execution SN and execute this command. The execution SN of the selected element will be exchanged with that of the element that will be executed right before this element.

If the execution SN of the selected element is the largest, the SN will not change after this command is executed.

As shown in the figure below, the execution SN of the output element c is exchanged with that of a after this command is executed on c.

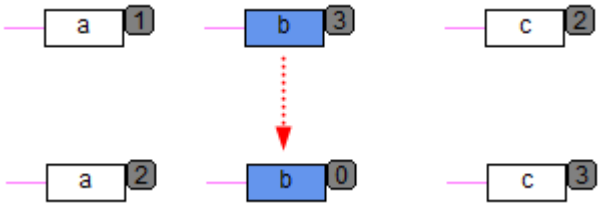


Execute the Order of Down

(4) Start

Set the SN of the selected element to 0. Increase the element execution SNs in front of the original execution SN of the selected element by 1 respectively. If the execution SN of the selected element is 0, there is no effect after executing this command.

As shown in the figure, the Decrease execution SN to Min command is executed on the output element b. Its execution SN is changed to the minimum SN, which is 0; And the SNs and in front of its original execution SN are increased by 1 respectively.

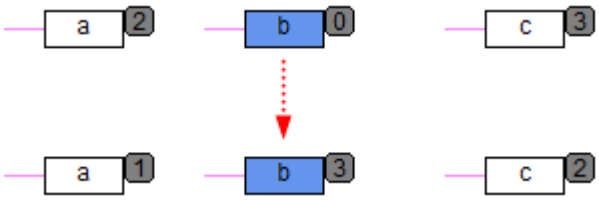


Execute the Order of Start

(5) End

Set the SN of the selected element to the maximum value in the existing SN range. Decrease the element execution SNs behind the original execution SN of the selected element by 1 respectively. If the execution SN of the selected element is the largest, this command will not take effect

As shown in the figure below, the execution SN of the output element b becomes the largest after this command is executed on b. The execution SNs (and) of other output elements are reduced by 1.



Execute the Order of End

10.9.22.2 Requirement

The program has been configured

10.9.22.3 Steps

To adjust the element execution sequence, follow the steps below:

(1) If users want to sort by data flow, right-click the blank area of the programming area. Click the Sort by Data Flow command. All the elements of the programming area are sorted automatically. If users want to adjust the sequence of an element, execute Step 2.

(2) Right-click the element. Click Advanced. According to the position that needs to be adjusted, select the corresponding command. The execution SN of this element is adjusted accordingly.

10.9.23 Element Alignment

10.9.23.1 Introduction

Users can align the block components left and right horizontally, and arrange them vertically at a minimum spacing distance.

Horizontal left/right alignment: Taking the leftmost/rightmost component of the selected component as the benchmark, the other selected components are horizontally aligned with the benchmark.

Vertical alignment: Taking the topmost component of the selected component as the benchmark, the other selected components are arranged vertically at a minimum spacing distance from each other, with no movement in the horizontal direction.

10.9.23.2 Requirement

Two or more components have been added.

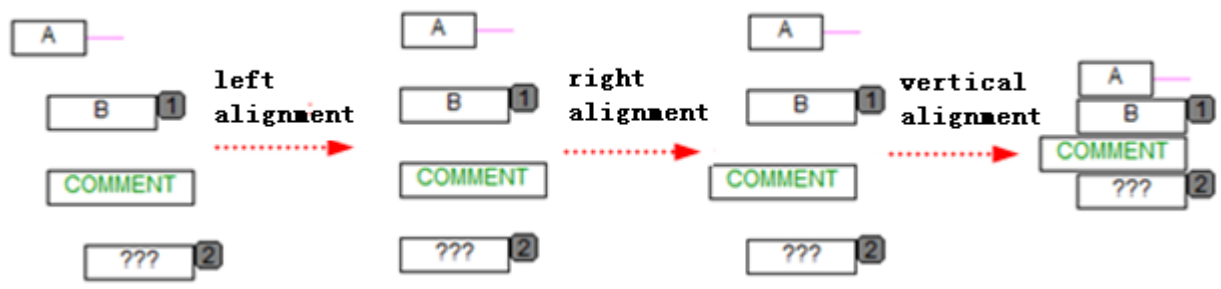
10.9.23.3 Steps

Follow the steps below to align components:

(1) Select two or more block components.

(2) Right-click the selected components. Click [Component Alignment] - [Horizontal left alignment]/[Horizontal right alignment]/[Vertical alignment]. The selected components are arranged according to the corresponding operations.

10.9.23.4 Example



Component Alignment

10.9.24 Show and Hide Variable Area

10.9.24.1 Introduction

This command is used to show or hide the variable area.

10.9.24.2 Requirement

The CFC Editor has been opened.

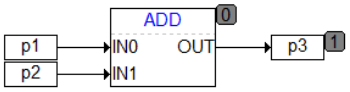
10.9.24.3 Steps

Follow the steps below to hide the variable area:

(1) Right-click the blank area of the program. Click Variable Area (Show/Hide). The variable area is hidden.

10.9.25 CFC Programming Example

No.	Variable Name	Address	Variable Description	Variable Type	Initial Value	Power Fail Safeguard	NetWork Public	Enable constant
0001	p1			INT	0	FALSE	Not Public	
0002	p2			INT	0	FALSE	Not Public	
0003	p3			INT	0	FALSE	Not Public	



10.10 Create SFC Program

10.10.1 SFC Programming Overview

SFC is short for sequential function chart, a programming language in the IEC 61131-3 standard for establishing and modifying sequential controls.

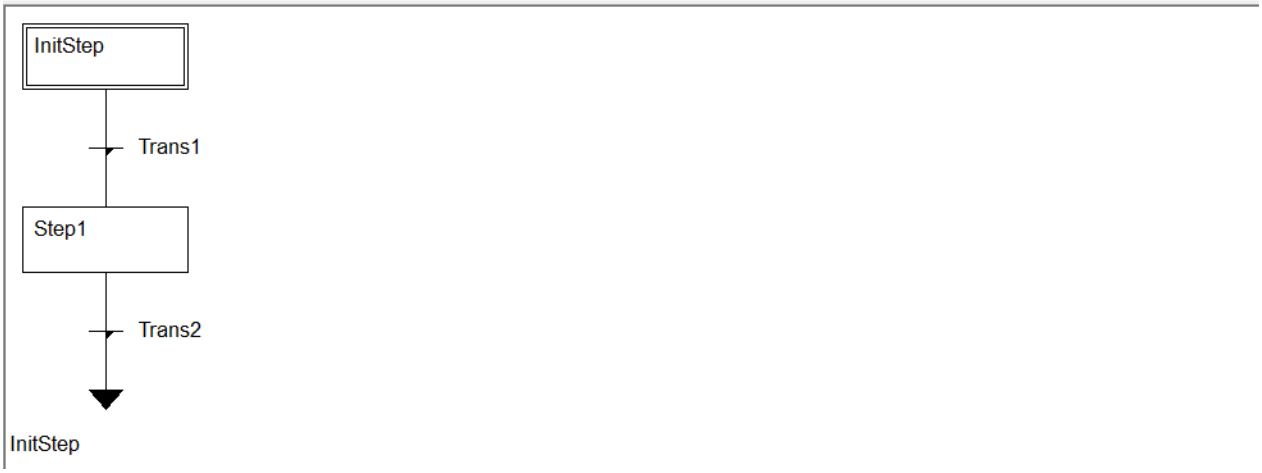
With SFC, single to complex tasks can be accomplished with a simple configuration method. This structure represents the user's subprograms with separate elements, similar to network node elements.

The subtask is associated with transitions and steps, which are embodied in the form of a program. These programs can be edited in CFC, LD, and ST languages. The transition represents an enabling condition to

activate a succeeding step. The step is processed periodically until the next transition condition is met.

The transitions are connected by lines, branches and separate elements that control the process. There is a difference between alternative branch and simultaneous branch. In the case of alternative branch, only one process can be running at a time; Conversely, when there are simultaneous branches in the process, several steps are processed in parallel.

No.	Variable Name	Address	Variable Description	Variable Type	Initial Value	Power Fail Safeguard	NetWork Public	Enable constant
0001	c1			BOOL	FALSE	FALSE	Not Public	
0002	p2			BOOL	FALSE	FALSE	Not Public	
0003	c3			BOOL	FALSE	FALSE	Not Public	
0004	p4			BOOL	FALSE	FALSE	Not Public	



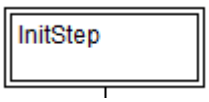
10.10.2 SFC Step Element

A step represents a state. Either a step is active or inactive. At any moment, the state of the program organization unit (POU) is defined by the settings of some active steps and the values of their internal variables.

1. Initial step

   Each SFC-edited POU starts with an initial step. This step is the entry point to POU.

   Only the initial step always exists in the SFC POU.

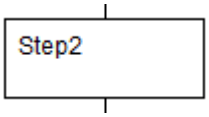


Initial Step

2. Step

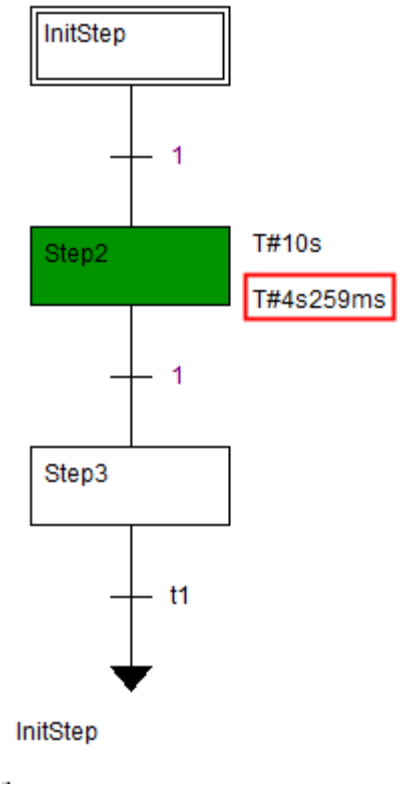
   The step describes what is controlled at this process step. The action of the step describes the configured program, which can be edited in ST, LD, or CFC language.

   To insert a step, the step name needs to be specified in the SFC POU. A step can be associated with up to 9 actions.



Step Element

- StepName.t(TIME): Step running time. When a step is deactivated, the value of the time elapsed by the step shall remain at the value it had when this step was deactivated; And when the step is activated, the value of the time elapsed by the step shall be reset to t#0s, and the timing shall be restarted. As shown in the figure, T#10s is the minimum time set for Step2, and T#4s259ms is the value of Step2.t. When the minimum step time is set, the step is activated. StepName.t is displayed as shown in the figure.



Online Value of StepName

- StepName.x(BOOL): Step mark, a BOOL element that marks the current active state of the step. Where 1 indicates step active; And 0 indicates step inactive. As shown in the figure above, when the transition condition in front of Step2 is changed to TRUE, Step2 is activated. At this time, StepName.x is 1 and this step is active.

- The step name, step running time, and step mark are only valid for the current POU, and are not supported for references to other POUs.

10.10.3 SFC Action Element

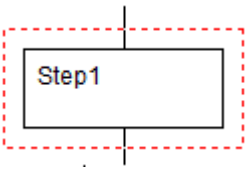
Each step shall be associated with a number of actions. A step with zero associated actions shall be considered to have a wait function. That is, it waits for a succeeding transition condition to change to TRUE.

An action can be a set of statements in ST language, a set of rungs in LD language, or a set of programs in CFC language.

10.10.4 Select Element

10.10.4.1 Introduction

The element is selected by clicking. The selected element is identified by an enclosed dashed box. During configuration, multiple elements often need to be selected to execute the relevant operations.



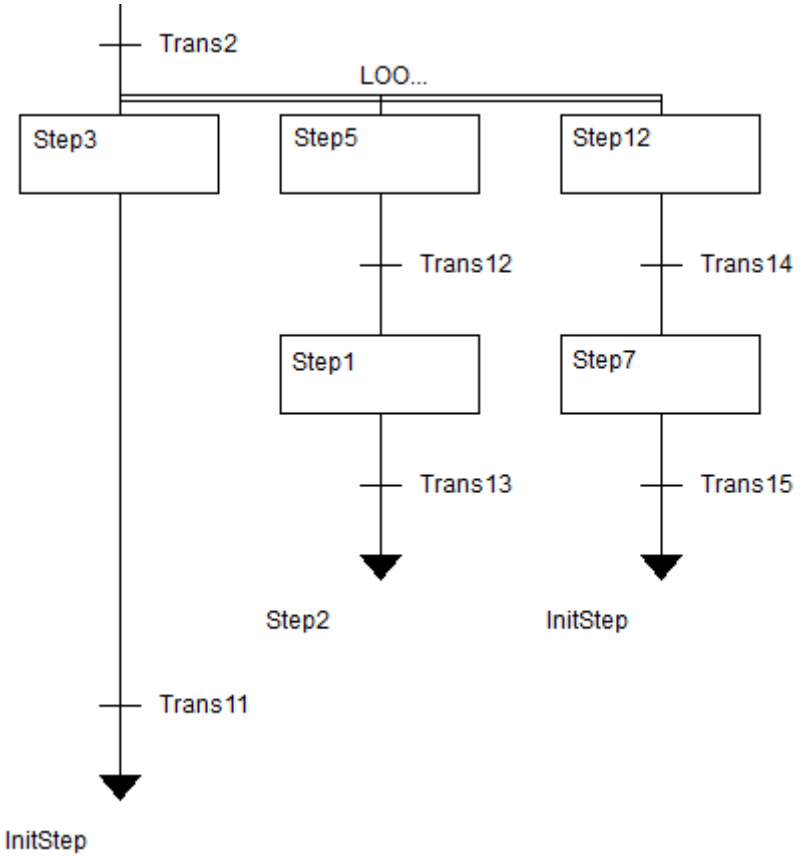
10.10.4.2 Steps

(1) Select an element: Move the cursor over an element. Press the left mouse button. The element (step, transition or jump) is selected.

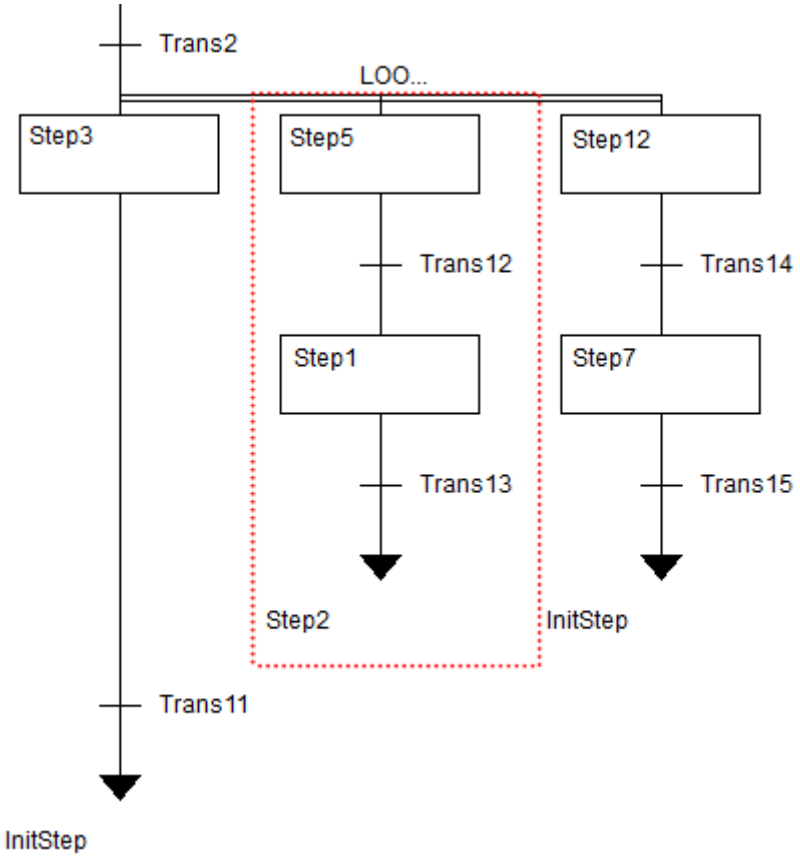
(2) To select multiple elements in a group, press Shift on the selected elements. Select the elements in the lower-left or lower-right corners of the group. The smallest group that includes these elements is thus selected.

10.10.4.3 Example

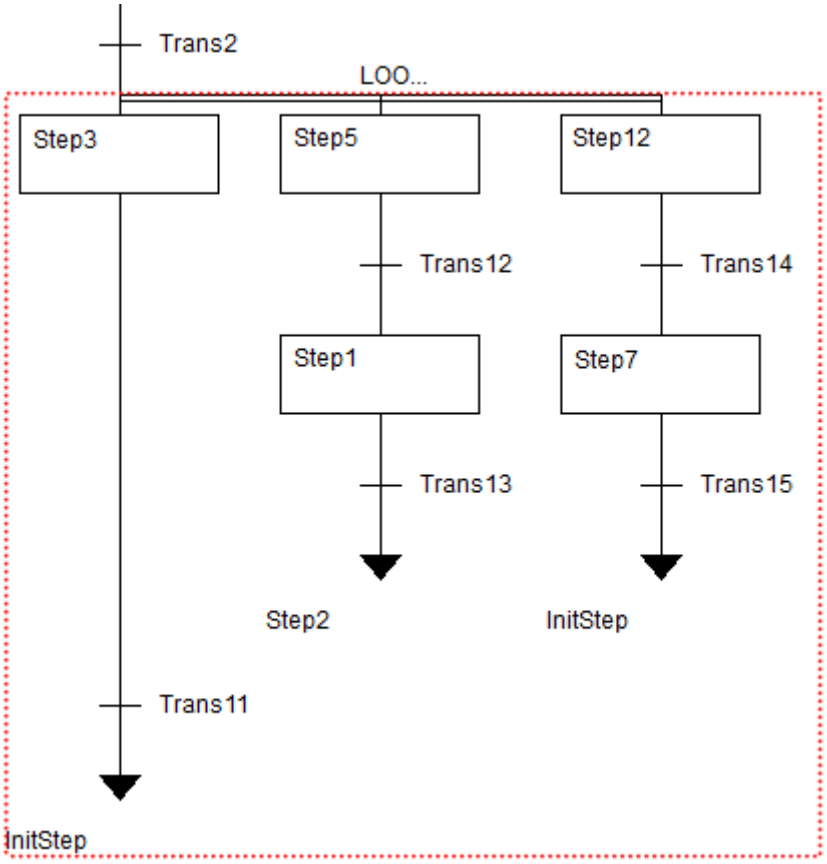
Selecting different combinations of elements may result in different results.



Select the element Step5 first. Press Shift, and simultaneously select the jump element Step2. All the elements in the branch where they are located are selected.



Select the element Step3 first. Press Shift. Meanwhile, select any other element outside of the branch Step3 belongs to (Trans11 and jump InitStep). In the end, the result is the same.



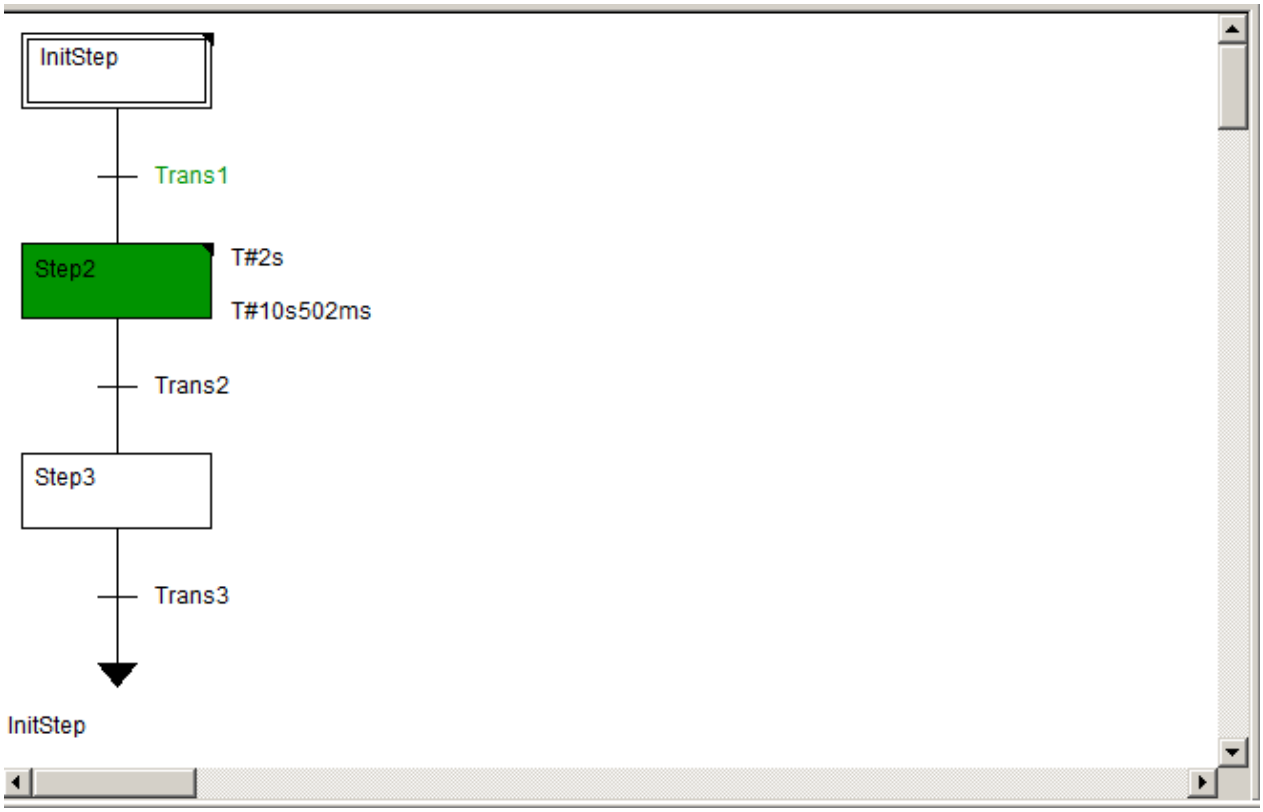
A step can only be deleted with the transition that precedes or follows it.

10.10.5 Identifier

An identifier is used to control the program flow in the SFC POU, which is implicitly created during project runtime. In order to be able to read these identifiers, appropriate local variables shall be defined.

- SFCInit: A BOOL variable. When the value of this variable is TRUE, the sequential function chart (SFC) returns to the initial step. Meanwhile, the other SFC identifiers are reset. The initial step remains activated, but is not executed. Only when SFCInit is reset to FALSE can the sequential function chart resume normal execution.
- SFCPause: BOOL variable. Only when this variable is TRUE can the sequential function chart stop executing, as shown in the figure below.

No.	Variable Name	Address	Variable Description	Variable Type	Online Value	Power Fail Safeguard	NetWork Public	Enable constant
0001	Trans1			BOOL	TRUE	FALSE	Not Public	
0002	Trans2			BOOL	FALSE	FALSE	Not Public	
0003	sfcinit			BOOL	FALSE	FALSE	Not Public	
0004	sfcreset			BOOL	FALSE	FALSE	Not Public	
0005	sfcpause			BOOL	TRUE	FALSE	Not Public	
0006	g1			INT	1	FALSE	Not Public	
0007	Trans3			BOOL	FALSE	FALSE	Not Public	

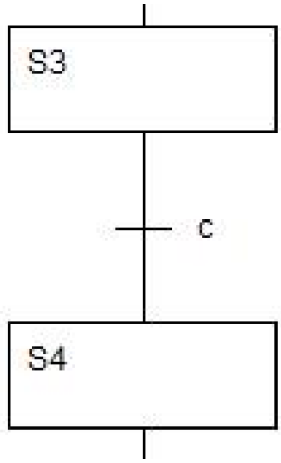


- SFCReset: BOOL variable. When this variable is TRUE, the SFC program is reset and returns to the initial step. Meanwhile, the other SFC identifiers are also reset, and the initial step remains activated. In contrast to SFCInit, when the value of the SFCReset variable is TRUE, the action of the initial step is not affected and continues to be executed. Meanwhile, the SFCReset identifier can be reset to FALSE in the initial step.

10.10.6 SFC Evolutionary Rules

- Single branch: Step-transition alternations are repeated continuously.

  Example:The transition from S3 to S4 works only when step S3 is active and the transition condition c is TRUE.

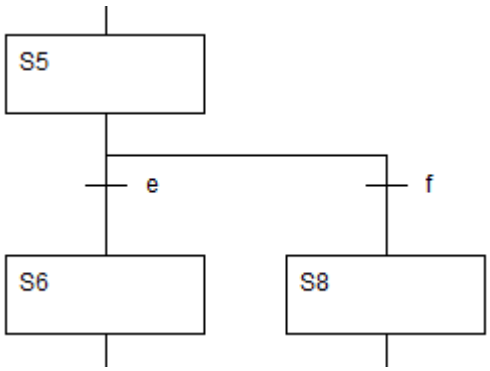


- Selection - Branch: Several alternative branches can be represented by a number of transition symbols below the horizontal line; With several transition symbols, there are several possible different transitions.

  Example:

  The transition from S5 to S6 works only when S5 is active and the transition condition e is TRUE;

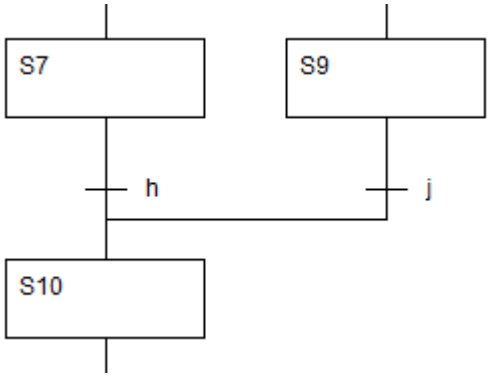
  The transition from S5 to S8 works only when S5 is active, f is true, and e is false.



- Convergence of selection branches: Convergence of alternative branches: The end of an alternative branch is represented by a number of transition symbols above the horizontal line; With several transition symbols, there are several alternative paths that end.

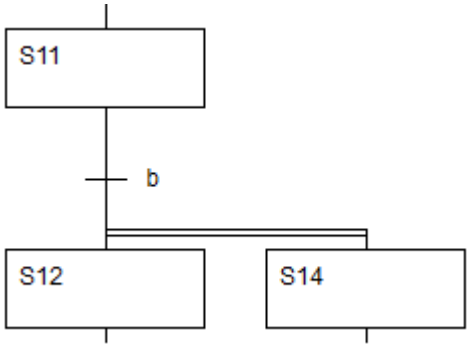
  Example:The transition from S7 to S10 works only when S7 is active and the transition condition h is true.

  Or the transition from S9 to S10 works only when S9 is active and j is TRUE.



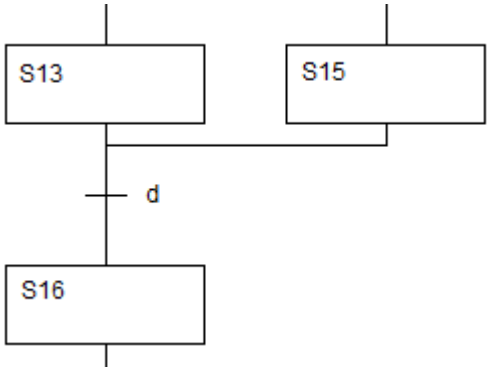
- Simultaneous branch: Only one shared transition symbol can exist above the double horizontal lines of the simultaneous branch.

  Example:The transition from S11 to S12, S14, ... works only when S11 is active and the transition condition b associated with the common transition is TRUE. After S12, S14, ... are activated simultaneously, each branch is executed independently.



- Simultaneous branch - convergence: Only one shared transition symbol can exist below the horizontal line of the simultaneous branch.

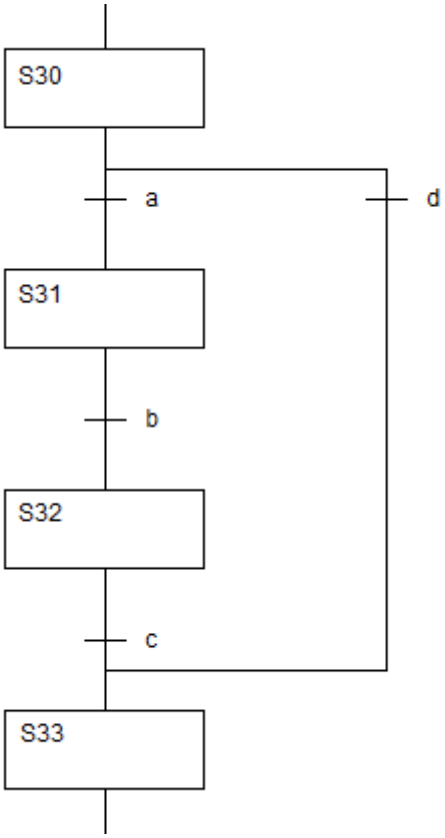
  Example:The transition from S13, S15, ... to S16 works only when all the steps connected to the horizontal line above are active and the transition condition d associated with the common transition is TRUE.



- Branch skip: A branch skip is a special case of alternative branches in which one or more branches do not contain steps.

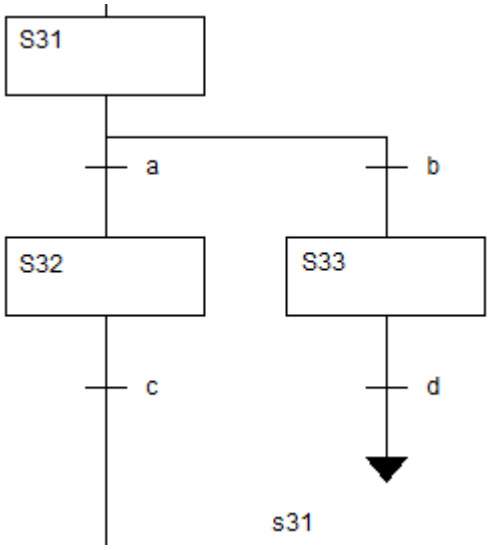
  Example:The evolution from S30 to S33 works when a is FALSE and d is TRUE. That is, the sequences

S31 and S32 are skipped.



- Branch jump: A special case of alternative branches or simultaneous branches in which one or more branches jump to other steps.

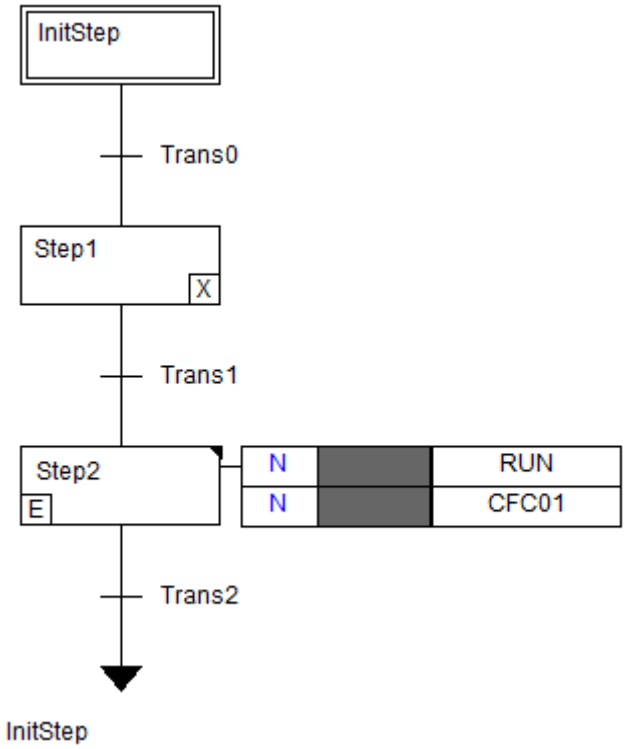
  Example: The transition from S33 to S31 works when S33 is active and d is TRUE.



10.10.7 SFC Step Action Execution Timing

The entry action, exit action, step action and IEC-associated action in each step are executed in a certain sequence.

As shown in the figure, Step1 adds the exit action, and Step2 adds the entry action, step action, and IEC-associated action.



In the current scan cycle, the step actions are executed as shown in the table.

Execution	Action	Action Processing

Timing		
1	Step1 Exit Action	If Step1.x = TRUE and Trans1 = TRUE, this action is executed. (Its entry and step actions have been executed in the previous cycle)
2	Step2 Entry Action	If Trans1 = TRUE and Step2 is activated (that is, Step2.x = TRUE), the entry action is executed
3	Step2 Step Action	If Step2 is activated and the entry action has been executed, the step action is executed
4	Step2 IEC-associated Action	If the step action has been executed, the IEC-associated actions are executed in alphabetical sequence
5	Transition to activate next step	If Step2 is activated, Trans2 is TRUE, and Step2.t is greater than the minimum time, the subsequent steps are activated

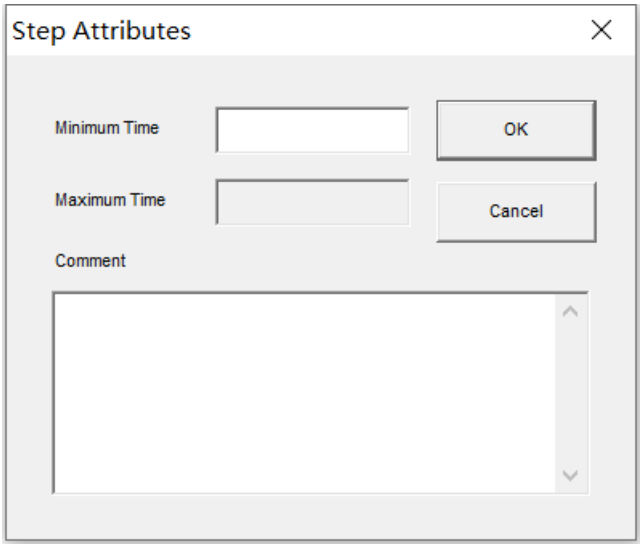
10.10.8 Set Step Attributes

10.10.8.1 Introduction

It is used to set the step-related properties, including the minimum step time and the step comment. The maximum step time cannot be set.

Minimum time: The lower limit of the step running time. When the step running time is smaller than the minimum time, the program does not execute the transition even if its exit transfer condition is met. The program does not execute the transition until the step execution time exceeds the minimum time. It is recommended to set the minimum time. Time can be represented by the TIME variable or by time data; Its constant is written in the following format: T + # + Time value.

- Comment: A step-related description in an unlimited number of characters. After determining the setting contents, the comment information of the setting is displayed on the right side of this step, as shown in the figure below. How much comment information is displayed is related to the set step height and whether or not the maximum and minimum times are displayed.



10.10.8.2 Requirement

The step has been added.

10.10.8.3 Steps

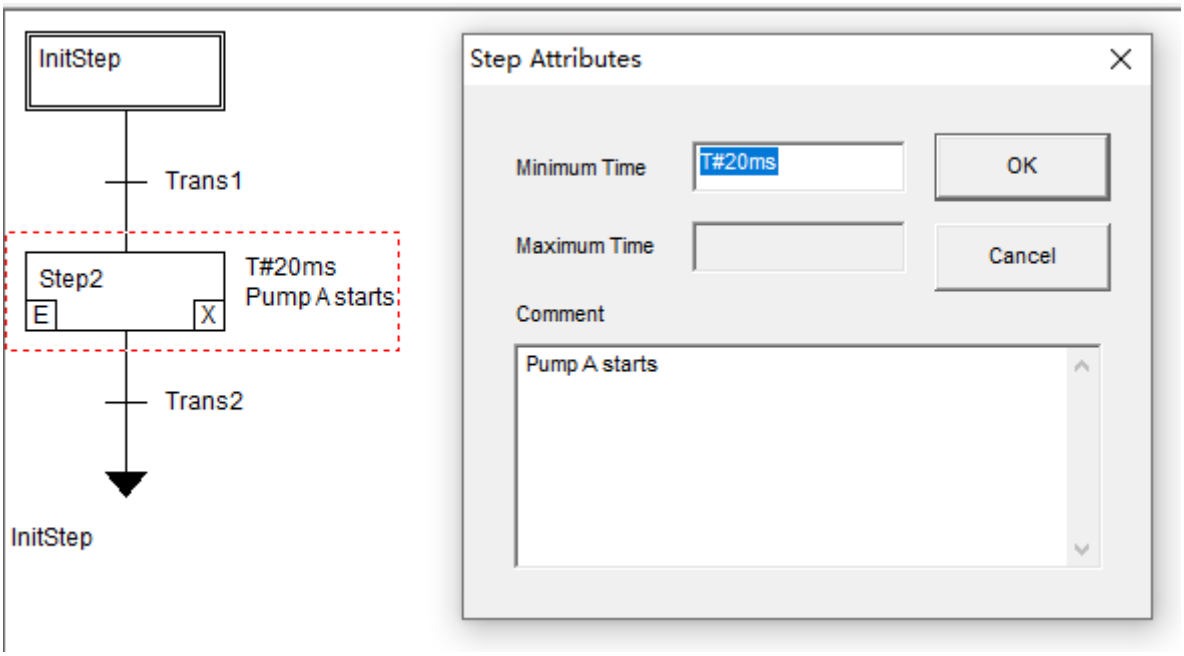
Follow the steps below to set step attributes:

(1) Select a step. Open the Step Properties window via:

- Menu bar: Click [Extras] - [Step Attributes];
- Programming area: Right-click the step. Click Step Attributes.

(2) In the Step Attributes window, set the Minimum Step Time and Comment.

10.10.8.4 Example



10.10.9 Insert Step Forward

10.10.9.1 Introduction

When you insert a step, the step-transition pair is inserted, so that the original SFC program maintains a basically correct logical flow.

You can insert a new transition and step in front of an existing transition or step. For the initial step (InitStep), no step can be inserted forward. Up to 2000 steps can be added in a SFC POU.

10.10.9.2 Requirement

The step and transition have been added.

10.10.9.3 Steps

Follow the steps below to insert a step:

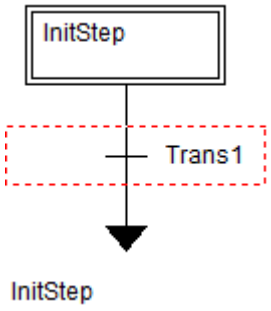
(1) Select a step or transition. Insert it via:

- Menu bar: [Click Insert] - [Step-Transition (Forward)];
- Programming area: Right-click the transition or step. Click Step-Transition (Forward);

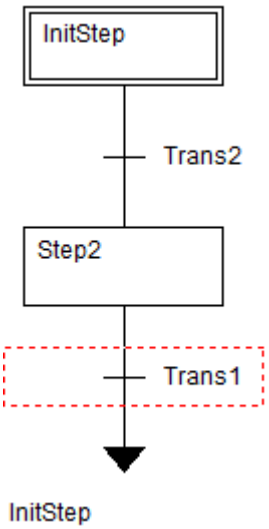
- Toolbar:  ;
- Shortcut key: Ctrl + T.

10.10.9.4 Example

Select a transition. Insert a new transition and a new step in front of the transition.



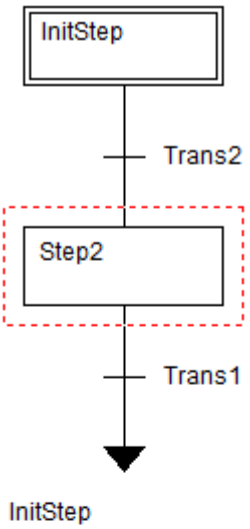
(a)



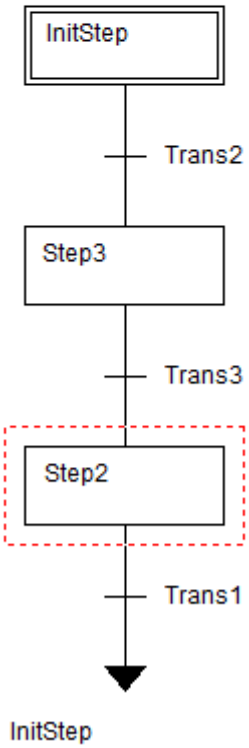
(b)

Insert Step Forward - Selected Transformation

When a step is selected, insert a new step and a new transformation before the step. The initial step (InitStep) cannot have a step inserted forward.



（a）



（b）

Insert Step Forward - Selected Step

10.10.10 Insert Step to the Back

10.10.10.1 Introduction

When you insert a step, the step-transition pair is inserted, so that the original SFC program maintains a basically correct logical flow.

You can insert a transition or step after an existing transition or step. Up to 2,000 steps can be added in a SFC POU.

10.10.10.2 Requirement

The step and transition have been added.

10.10.10.3 Steps

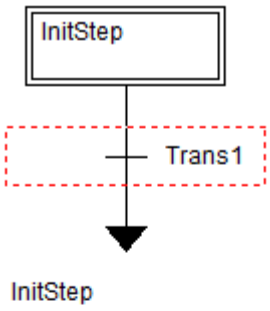
Follow the steps below to insert a step:

Select a step or transition. Insert it via:

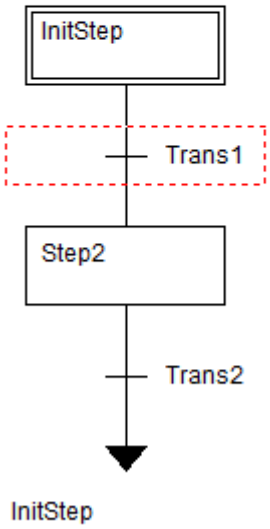
- Menu bar: Choose [Insert] - [Step-Transition (Back)].
- Programming area: Right-click a transition or step and click Step-Transition (Back).
- Toolbar:  ;
- Shortcut key: Ctrl + E.

10.10.10.4 Example

Select a transition and insert a new transition and a new step after the transition.



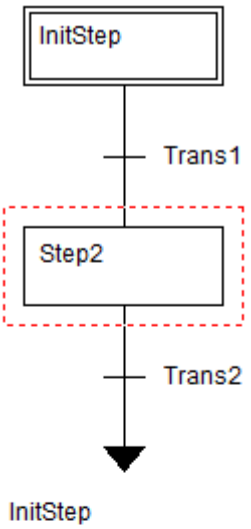
（a）



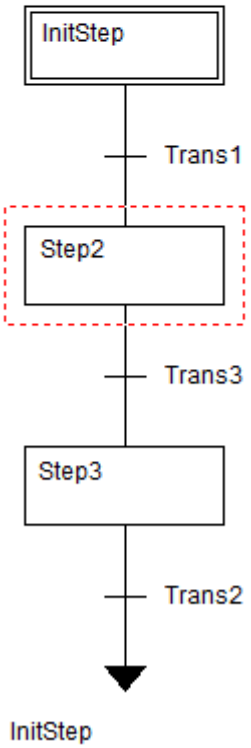
（b）

Insert Step Backward - Selected Transformation

Select a step. Insert a new step and a new transition behind the step.



（a）



（b）

Insert Step Backward - Selected Step

10.10.11 Delete Step

10.10.11.1 Introduction

When you delete a step, the step-transition pair is deleted so that the original logic of the SFC program remains correct. The initial step cannot be deleted.

10.10.11.2 Delete single step-transition pair

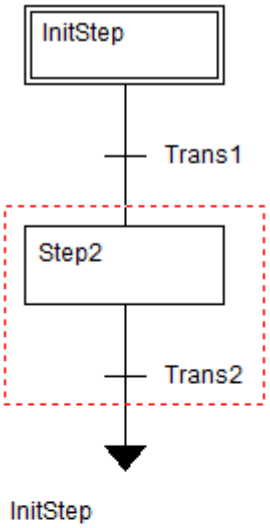
Follow the steps below:

(1) Click the left mouse button to select the step.

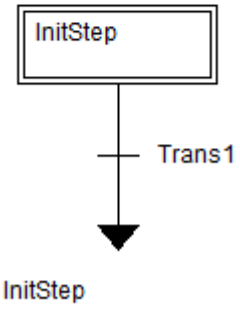
(2) Press Ctrl or Shift. Meanwhile, click the transition corresponding to that step.

(3) Right-click the selected step-transition pair. Click the Delete command.

Example: Select the step-transition pairs that need to be deleted: Click the left mouse button to select Step2. Press Ctrl or Shift. Meanwhile, click Trans2 to select both Step2 and Trans2. Execute the Delete command.



（a）



（b）

10.10.11.3 Delete multiple step-transition pairs

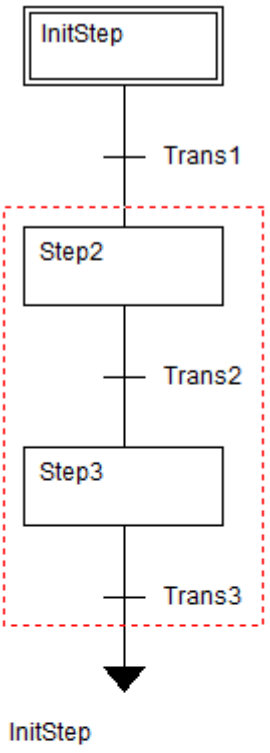
Follow the steps below:

(1) Click the left mouse button to select the step.

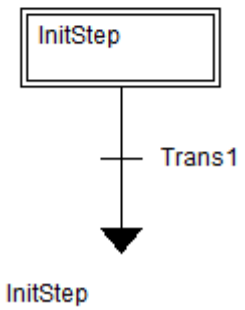
(2) Press Ctrl or Shift. Click the step-transition pairs to be deleted.

(3) Right-click the selected step-transition pair. Click the Delete command.

Example: Select the step-transition pairs that need to be deleted: Click the left mouse button to select Trans3. Press Ctrl, and click Step3, Trans2 and Step2 in sequence; Or press Shift, and select Trans3 and Step2. Execute the Delete command.



（a）



（b）

Delete multiple step-transition pairs

10.10.12 Modify Step Name

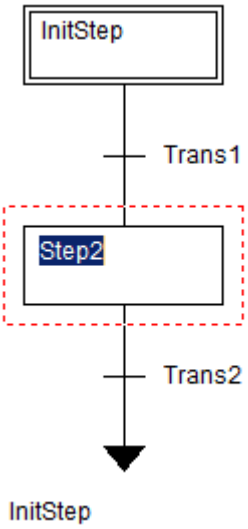
10.10.12.1 Introduction

When users create a new step, the system automatically assigns a name to it. To modify it, it is necessary to set the new name in accordance with the common variable naming rules and to ensure that the name is not the same as the one in the project.

10.10.12.2 Steps

Follow the steps below to modify the step name:

(1) Double-click the step name text area. The step name is editable. Users can enter a new step name.



Modify Step Name

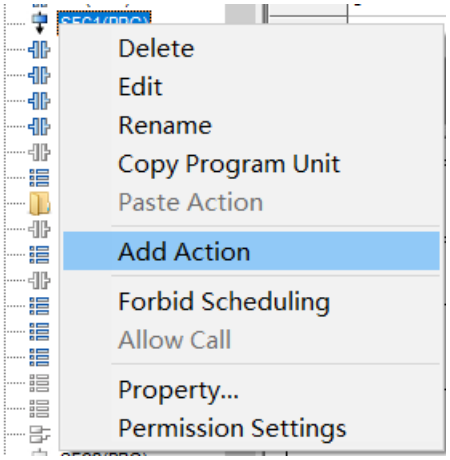
(2) After entering the new name, click any blank area or Press Enter to confirm. The editing is completed.

10.10.13 Add Action

10.10.13.1 Steps

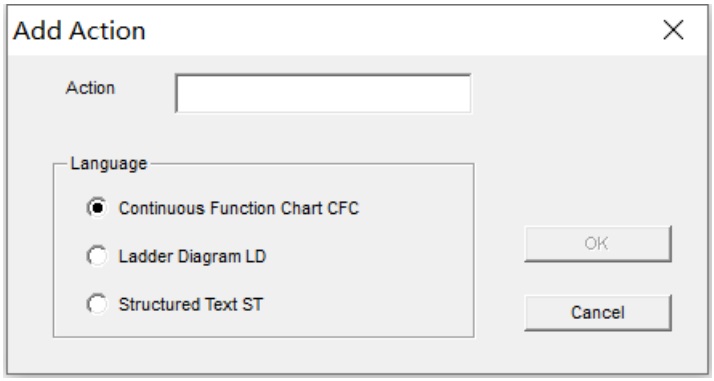
Follow the steps below to add an action:

(1) In the Project Management Tree, right-click the POU name of the SFC language. Select Add Action option to add an action to this POU.



Operation of Adding Action

The Add Action dialog box appears.



Add Action

(2) Enter the action name, with no more than 30 characters. Select the editing language. Click OK. The action is added. The action name is displayed in the POU subdirectory.

As shown in the figure below, an action with editing language LD and action name test1(ACTION) is added under fff(PRG).

Added Action Node

10.10.14 Add Entry Action

10.10.14.1 Introduction

Entry action refers to the action executed when this step first enters the activated state. For the initial step (InitStep), no entry action can be added.

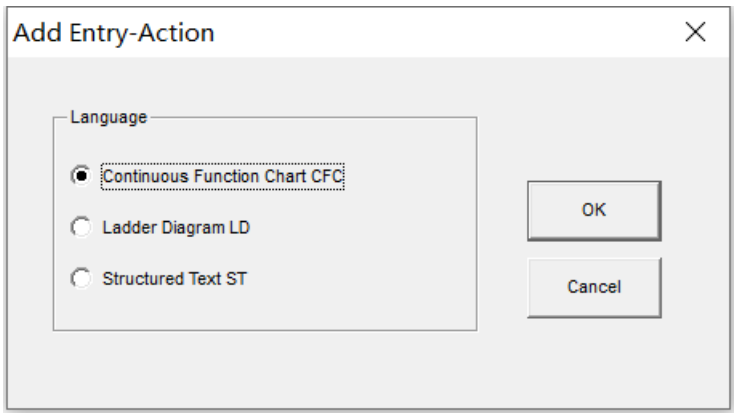
10.10.14.2 Steps

Follow the steps below to add an entry action:

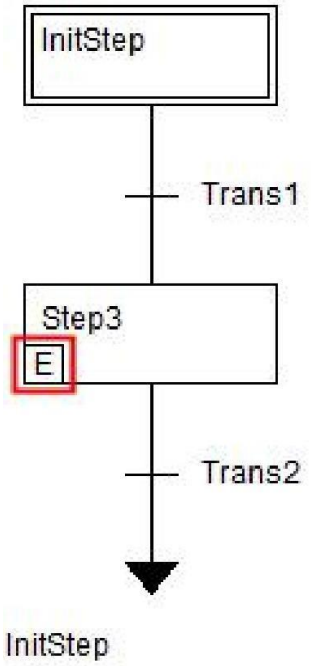
(1) Select the step. Add the entry action via:

- Menu bar: Click [Insert] - [Add Entry-Action];
- Programming area: Right-click the step. Click Add Entry-Action;
- Shortcut key: Ctrl + I.

(2) A dialog box for selecting the action language type is displayed.



(3) Select the action description language. Click OK to add the entry action. The mark E is displayed in the lower-left corner of the step.



(4) Double-click this mark to edit the entry action of the step.

10.10.15 Add Exit Action

10.10.15.1 Introduction

Exit action refers to the action executed when the step is about to exit the activated state.

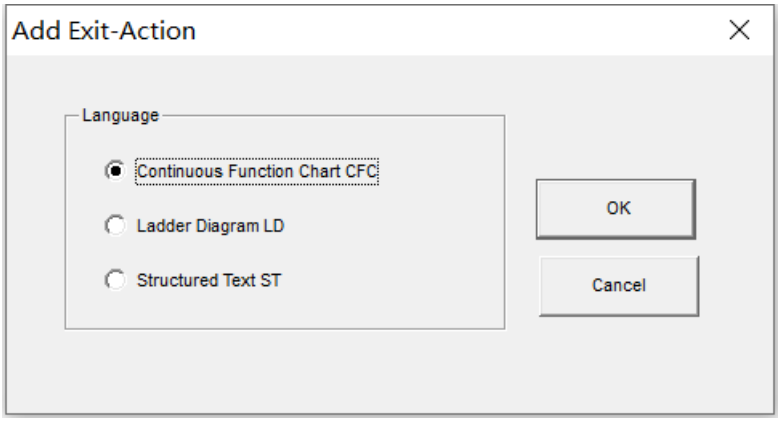
10.10.15.2 Steps

Follow the steps below to add an exit action:

(1) Select the step and add an exit action in the following ways:

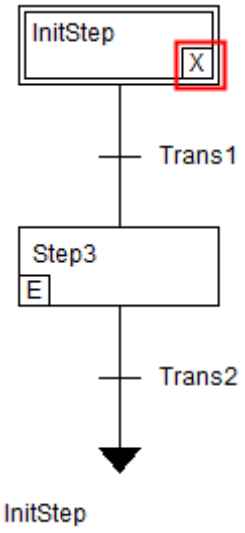
- Menu bar: Choose [Insert] - [Add Exit-Action].
- Programming area: Right-click the step and choose Add Exit-Action.
- Shortcut key: Ctrl + G.

(2) A dialog box for selecting the action language type is displayed.



Language Type of Exit Action

(3) Select the language and click OK to add the exit action. A X symbol then appears in the lower-right corner of the step.



Exit Action Logo

(4) You can double-click the symbol to edit the exit action.

10.10.16 Associate Action

10.10.16.1 Introduction

The Associate Action operation allows you to associate an action added to the SFC POU to a step.

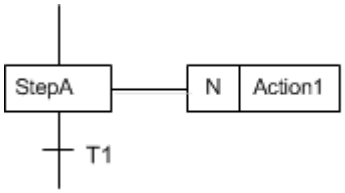
The association relationship between an action and a step is determined by the action qualifier.

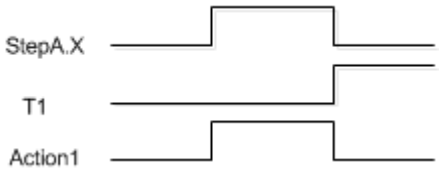
Action Qualifier

Area A is the action qualifier with its own drop-down box, from which you can select the desired qualifier; If you select a time-related qualifier in area A, you can edit the time in Area B, with the default value of T#1S; Area C is the action name, which is displayed in the drop-down box in Area C when the action is added to the POU (SFC) (see Add Action) and can be called; whether or not the action is activated is closely related to the qualifier in Area A. The following table lists the relationship between different qualifiers and the execution of the action; Whether the action is activated or not is closely related to the qualifier in Area A, as shown below, which lists the execution relationship between different qualifiers and actions.

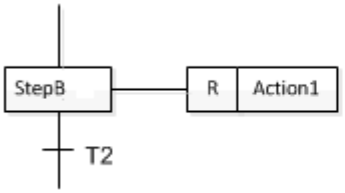
- Action Qualifiers

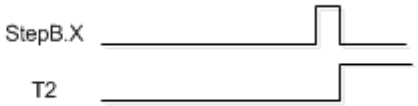
N: Non-storage



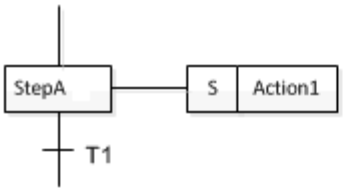


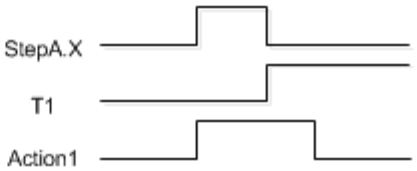
R: Storage reset



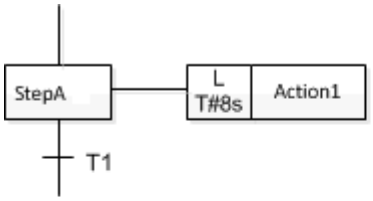


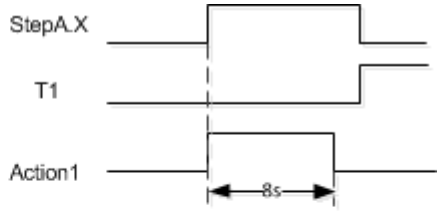
S: Storage settings

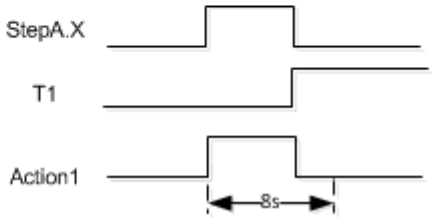




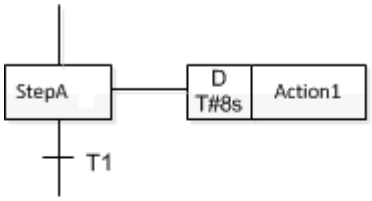
L: Time limit

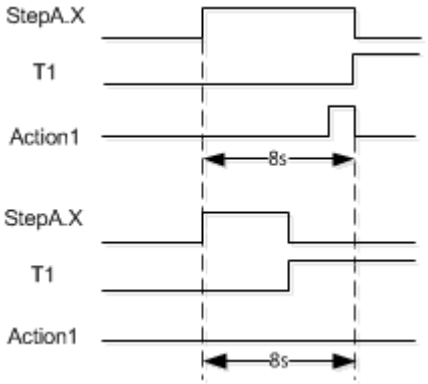




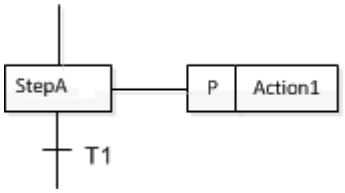


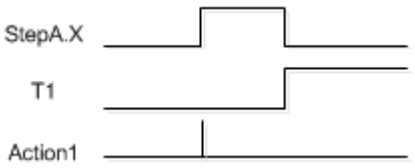
D: Delay



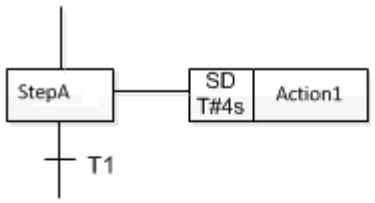


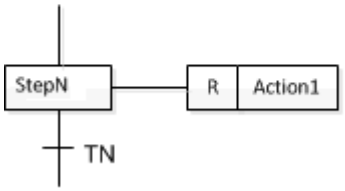
P: Pulse

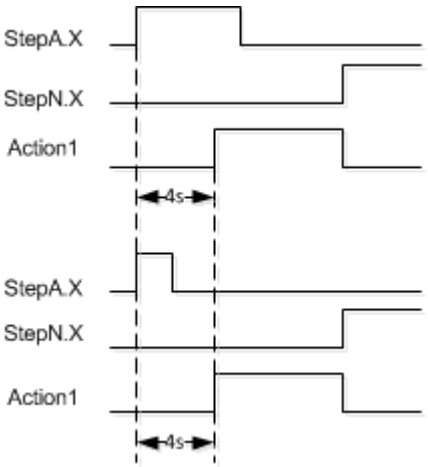




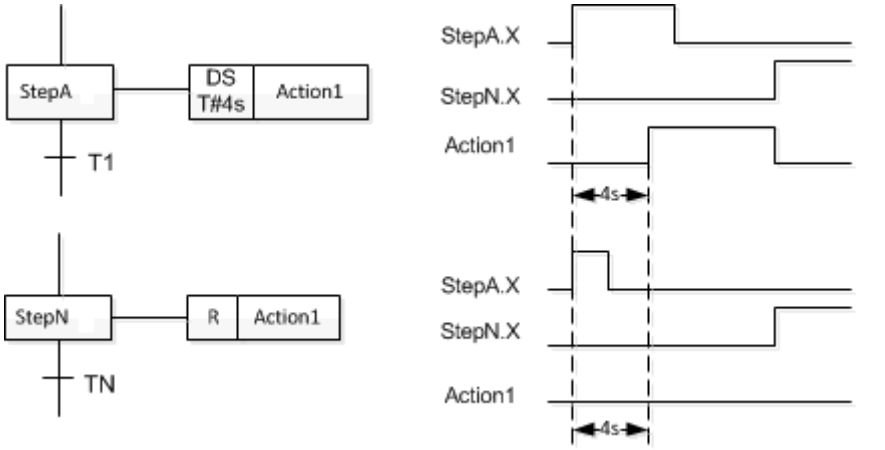
SD: Storage and delay



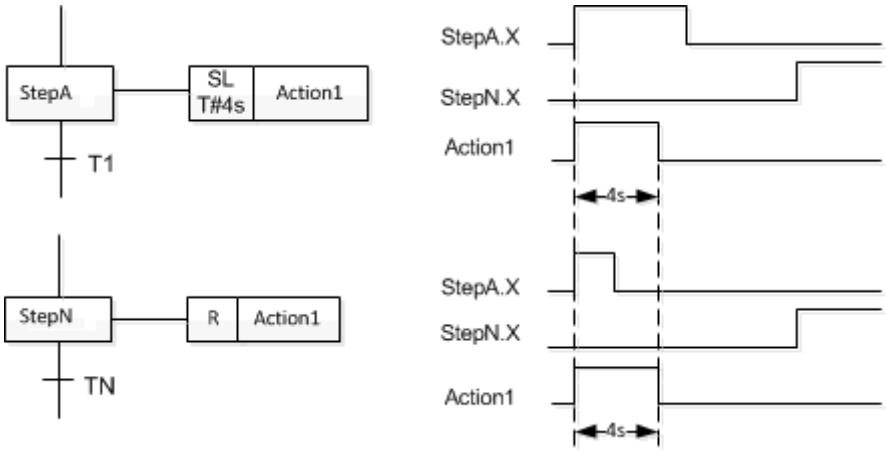




DS: Delay and storage

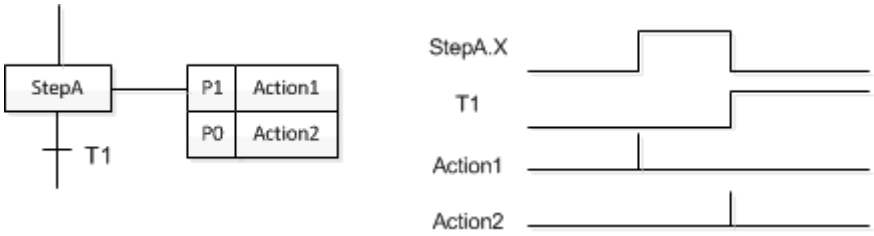


SL: Storage and time limit



P1: Rising edge pulse

P0: Falling edge pulse



10.10.16.2 Requirement

The action has been added under the current SFC POU node.

10.10.16.3 Steps

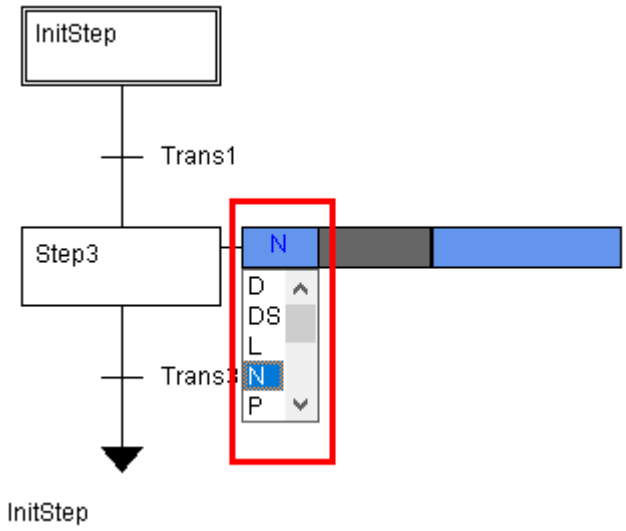
Follow the steps below to associate a step with an action:

(1) Select the step and add associate it with an action in the following ways:

- Menu bar: Click [Extras] - [Associate Action];
- Programming area: Right-click the transition. Click Associate Action;
- Shortcut key: Ctrl + K.

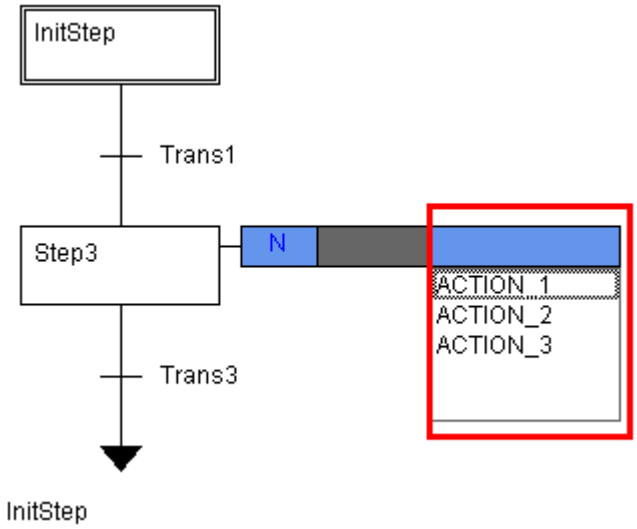
This associates a new action with the step.

(2) Double-click the action qualifier area. In the drop-down box, select the action qualifier.



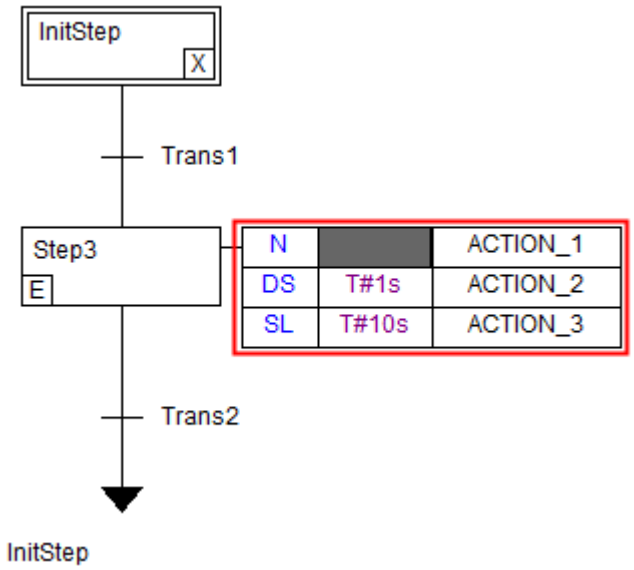
Select Action Qualifier

(3) Click the action name selection area to select an associated action.



Select Associated Action

(4) For example, Step3 is associated with 3 actions.



Action Association

10.10.17 Add Step Action

10.10.17.1 Introduction

It is used to add a step action to a step.

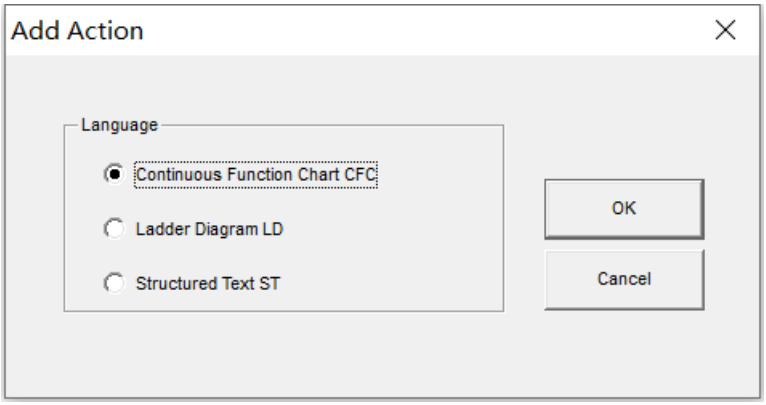
10.10.17.2 Steps

Follow the steps below to add an action to a step:

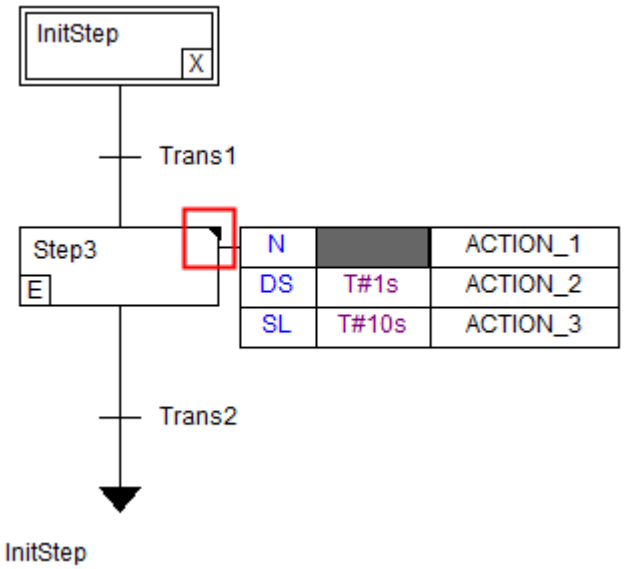
(1) Select the step and add an action in the following ways:

- Menu bar: Choose [Extras] - [Add Action/Transition].
- Programming area: Right-click the step and click Add Action/Transition.
- Shortcut key: Ctrl + W.

(2) The Add Action dialog box appears.



(3) Select the action description language. Click OK to complete the Add Step Action operation. The mark is displayed in the upper right corner of the step.



Step Action Mark

(4) Double-click the mark. Open the step editing page of the corresponding language environment, where users can edit the step action.

10.10.18 Delete Action

10.10.18.1 Introduction

For the step whose action has been added, uses can delete the corresponding action as needed. Deletable actions contain all added actions in the step, including entry action, exit action, step action, and associated action.

10.10.18.2 Requirement

The action has been added to the current step.

10.10.18.3 Steps

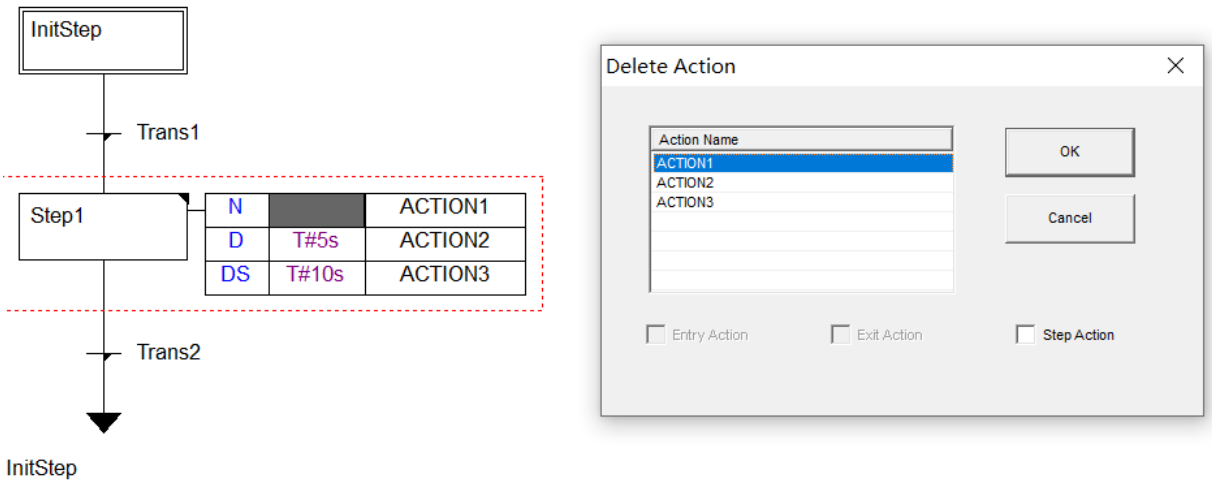
Follow the steps below to remove an action:

(1) Select the step and remove the action in the following ways:

- Menu bar: Choose [Extras] - Clear Action/Transition.
- Programming area: Right-click the step and click Remove Action/Transition.
- Shortcut key: Ctrl + U.

(2) The Delete Action dialog box is displayed.

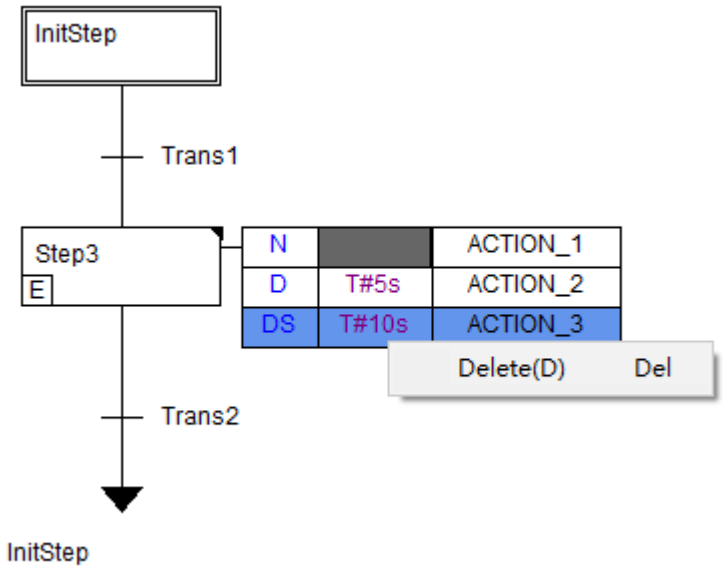
When the step has only one action, users can execute the Delete Action/Transition command to delete the action directly, without the Delete Action dialog box displayed.



Remove Action

(3) Select the action to be deleted. Click OK to complete the deletion.

- To delete an entry action, check the entry action.
- To delete an exit action, check the exit action.
- To delete a step action, check the step action.
- To delete an associated action, select the name of the action to delete it. To delete an associated action, users can also right-click the action that needs to be deleted and select the Delete command.



Delete Associated Action

10.10.19 Copy Action

10.10.19.1 Introduction

It is used to copy the action added under the SFC POU, and to paste it to multiple SFC POUs.

Each step can be associated with up to 9 actions.

The maximum number of actions associated with each SFC is 1024, including step actions (exit/entry action, and step action), transition actions, and IEC actions of SFC POUs.

To copy an action, the copied content shall not contain the definition of the local variables in the action.

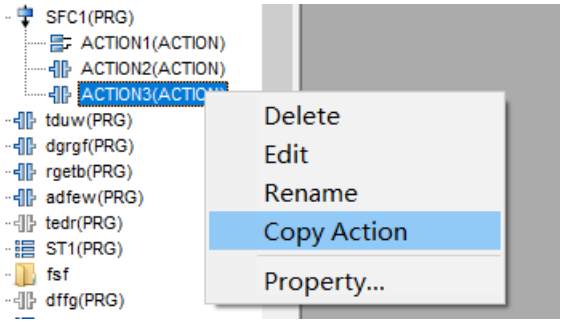
10.10.19.2 Requirement

An action has been added in the SFC POU.

10.10.19.3 Steps

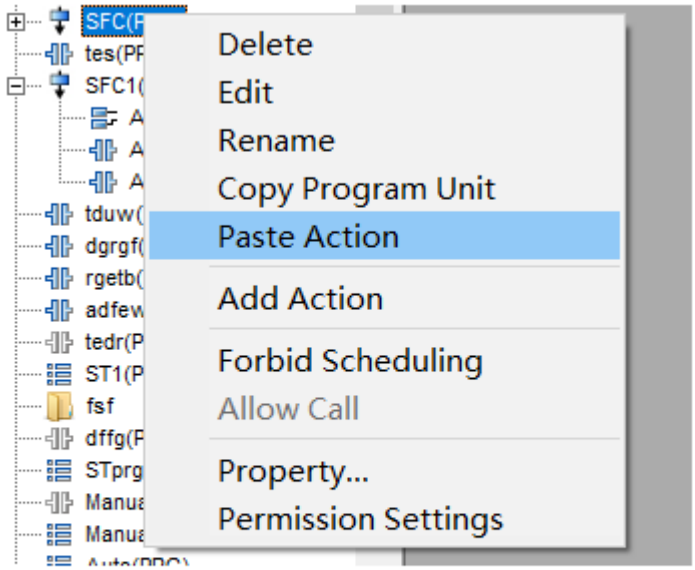
Follow the steps below to copy an action:

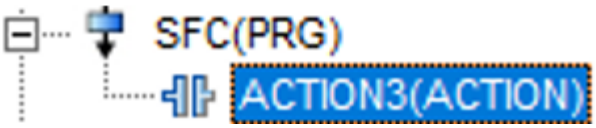
(1) Right-click the action POU name. Select Copy Action.



After executing the Copy Action command, users can paste the copied content to multiple SFC POUs.

(2) Right-click the SFC POU that needs to be pasted. Select Paste Action.

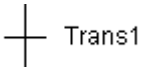




10.10.20 Transition Element

10.10.20.1 Introduction

A transition represents a condition under which control is transitioned from one or more preceding steps to one or more succeeding steps along a corresponding directed connection. The direction of transition is from the bottom of the preceding step to the top of the succeeding step. Up to 2,000 transitions can be added in a SFC POU.

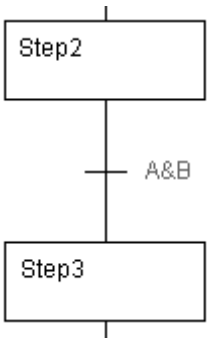


Transformation

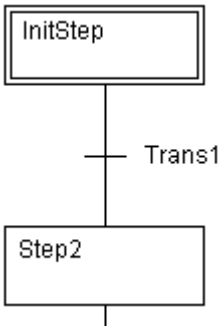
Each transition shall have a transition condition, which is the evaluation result of a single BOOL expression. In short, a true transition condition shall be indicated by the symbol 1 or the keyword TRUE. The transition evaluation can be derived from LD, CFC and ST.

During transition condition evaluation, it is an error case to assign a value to a variable instead of to transition a value. Transition-related descriptions are shown below.

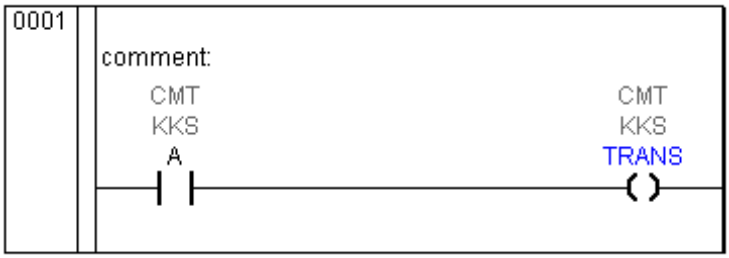
- The transition condition described in ST language, which is a single BOOL evaluation expression



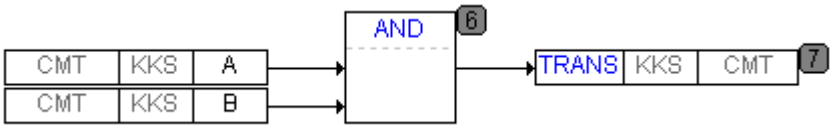
- Transition name



- Transition condition described in LD language



- Transition condition in the CFC language

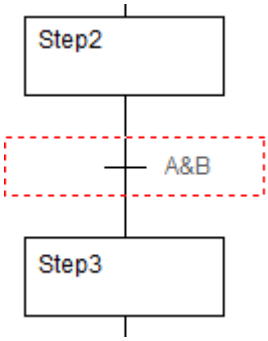


- Transition condition described in ST language

```
TRANS:=A&B;
```

10.10.20.2 Single BOOL expression as transition condition

Write a single BOOL expression directly at the transition name as a transition condition.



Transformation Condition—Single BOOL Expression

10.10.20.3 Edit transition

Whether it is a transition condition or an added transition, when users need to edit the transition content, just double-click the transition condition or the mark of the added transition.

10.10.21 Add Transition

10.10.21.1 Introduction

It is used to add a transition for a step to be transitioned.

When ST editing language is selected for the transition language, the transition content shall be written with an assignment statement, starting with Trans:=.

When a transition is added, there are two transition conditions. At this time, only the transition conditions edited in CFC/LD/ST are valid.

10.10.21.2 Steps

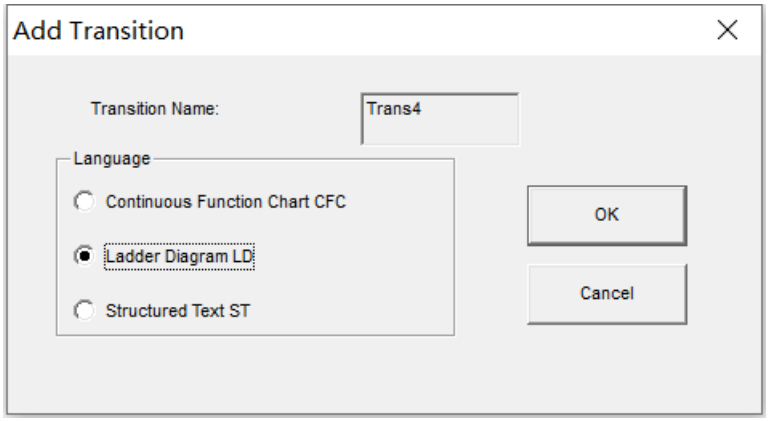
Follow the steps below to add a transition:

(1) Select the transition condition. Execute Add Transition via:

- Menu bar: Choose [Extras] - [Add Action/Transition].
- Programming area: Right-click the step and click Add Action/Transition.

- Shortcut key: Ctrl + W.

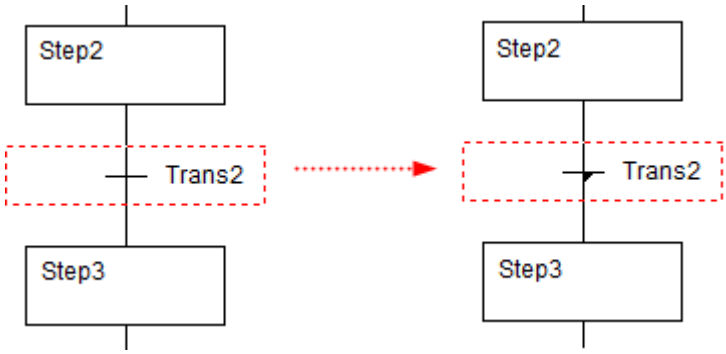
(2) The Add Transition dialog box appears.



Transition Language

(3) Select the transition condition description language. Click OK.

After the transition is added, a ⊣⊢ mark is displayed at the transition position.



The identifier for the added transformation.

10.10.22 Delete Transition

10.10.22.1 Introduction

It is used to delete the transition that has been added for a transition.

10.10.22.2 Requirement

The transition has been added to the current transition.

10.10.22.3 Steps

Follow the steps below to remove a transition:

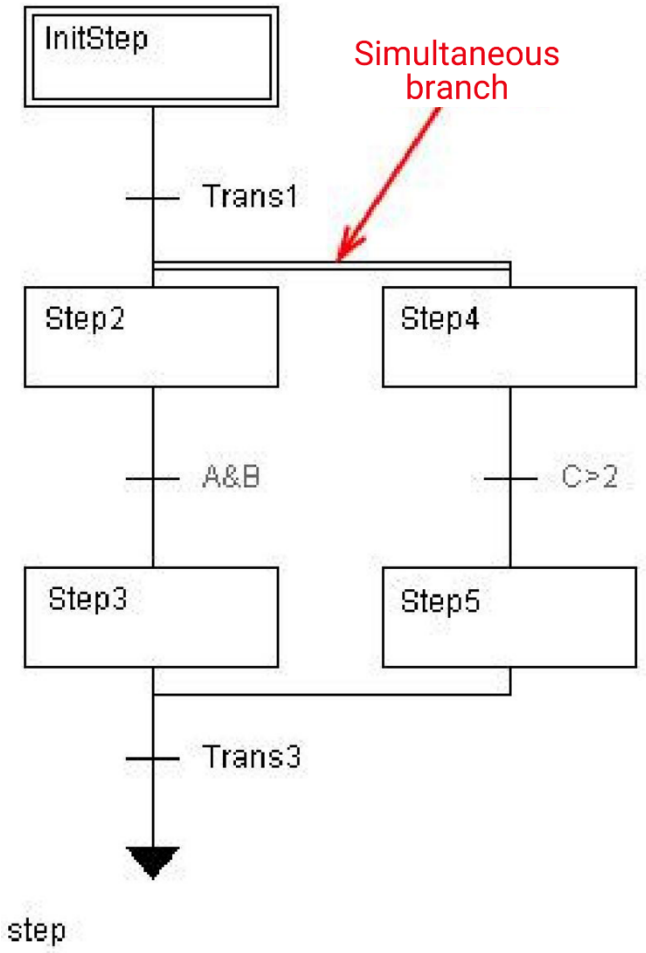
Select the transition and remove it in the following ways:

- Menu bar: Choose [Extras] - [Remove Action/Transition].
- Programming area: Right-click the transition and click Remove Action/Transition.
- Shortcut key: Ctrl + U.

10.10.23 Add Parallel Branch

10.10.23.1 Introduction

Two or more SFC units that start and end with a step are connected in parallel by double solid lines parallel to each other to form a parallel branch.



Parallel Branch

When the transition TRANS1 is TRUE and the initial step (InitStep) is active, Step2 and Step4 are activated at the same time.

This document describes only how to add parallel branches to the right; It is similar to how to add parallel branches to the left.

10.10.23.2 Requirement

The step-transition pair has been added.

10.10.23.3 Steps

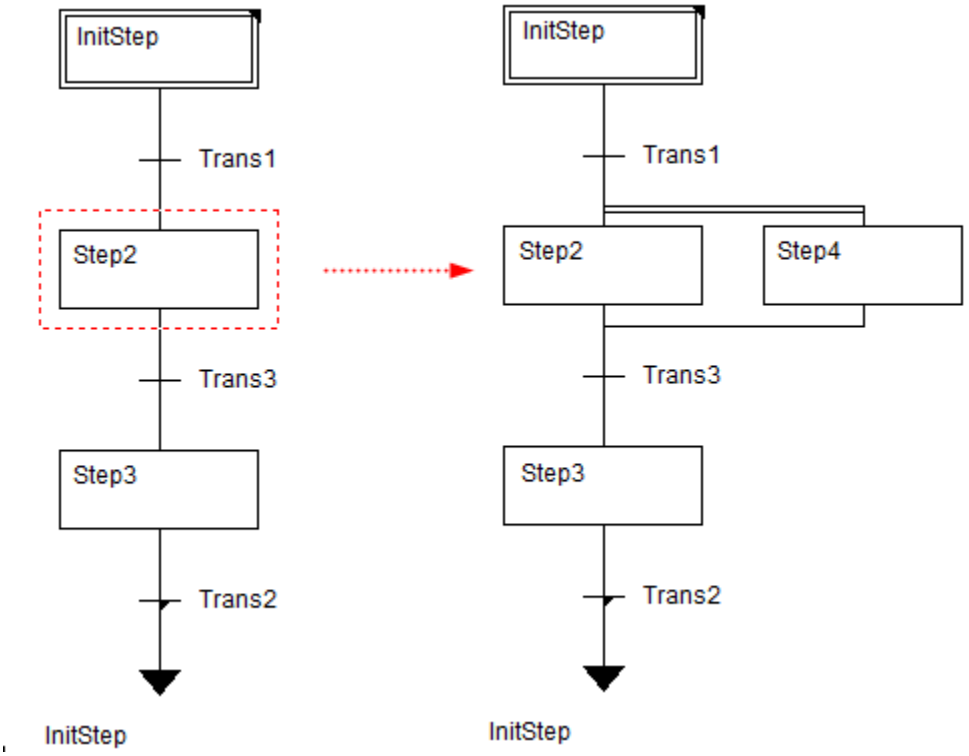
Follow the steps below to add a parallel branch:

Select a single step or a unit that starts and ends with a step. Add a parallel branch via:

- Menu bar: Click [Insert] > [Parallel Branch (Right)/(Left)];
- Toolbar:  .
- Programming area: Right-click the selected step or unit. Click Parallel Branch (Right)/(Left).

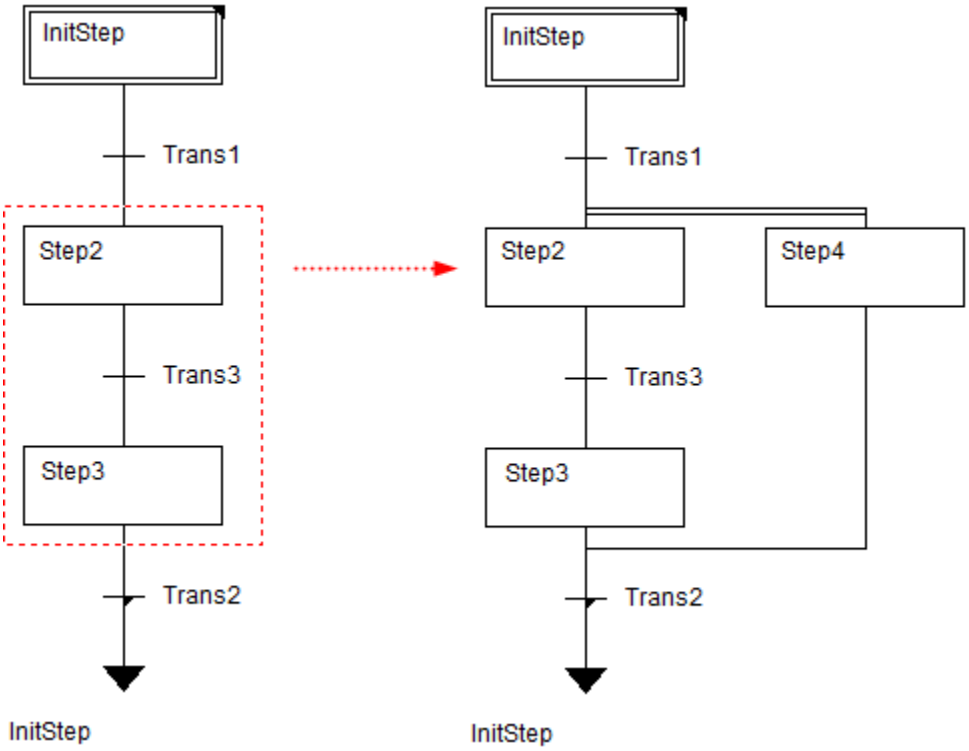
10.10.23.4 Example

Select a single step. Add a parallel branch to the right, as shown in the figure.



Add Parallel Branch to the Right--Single Step

To select a unit that starts and ends with a step, users need to press Ctrl or Shift to select the step units. Add a parallel branch to the right, as shown in the figure.



Add parallel Branch to the Right—Step Unit

10.10.24 Add Label to Parallel Branch

10.10.24.1 Introduction

You can add a label to the entire parallel branch as the identifier of the jump destination.

10.10.24.2 Requirement

A parallel branch has been added.

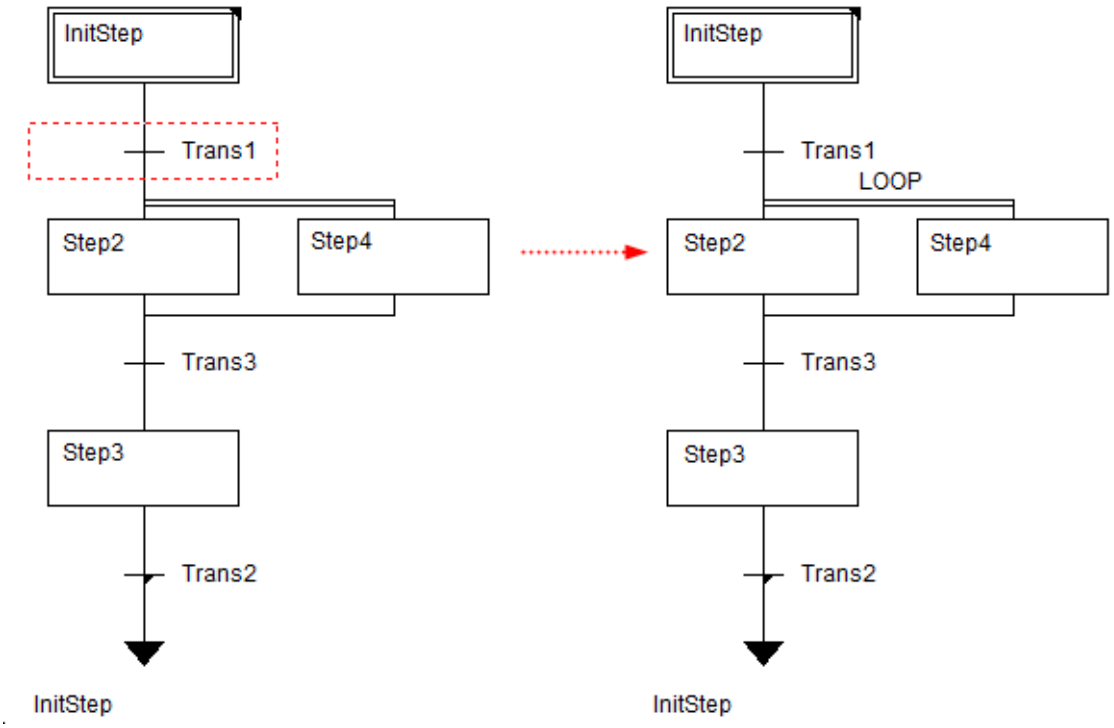
10.10.24.3 Steps

Follow the steps below to add the parallel branch label:

(1) Select the transition before the parallel branch and add the label in the following ways.

- Menu bar: Choose [Extras] - [Add Label to Parallel Branch];
- Programming area: Right-click the transition, and click Add Label to Parallel Branch;
- Shortcut key: Ctrl + Q.

The default label is LOOP, as shown in the figure below. It can be modified.



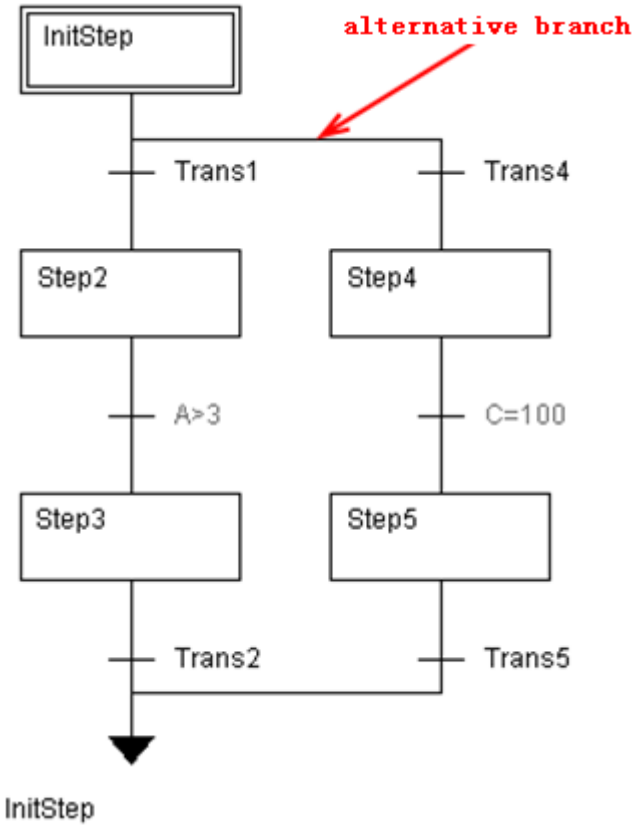
Parallel Branch Label

(2) Double-click to select the label LOOP, enter the actual label (transition name-jump destination), and left-click on any blank area or press the Enter key to complete renaming.

10.10.25 Add Alternative Branch

10.10.25.1 Introduction

An alternative branch contains two or more SFC units that begin and end with a transition and are connected using mutually parallel single solid lines.



Alternative Branch

After the initial step (InitStep) is activated, activate Step4 if Trans4 is true while Trans1 is false, or activate Step2 if Trans1 is true.

This document describes only adding alternative branches to the right here. The requirement and process of adding alternative branches to the left are very much the same.

10.10.25.2 Requirement

The step-transition pair has been added.

10.10.25.3 Steps

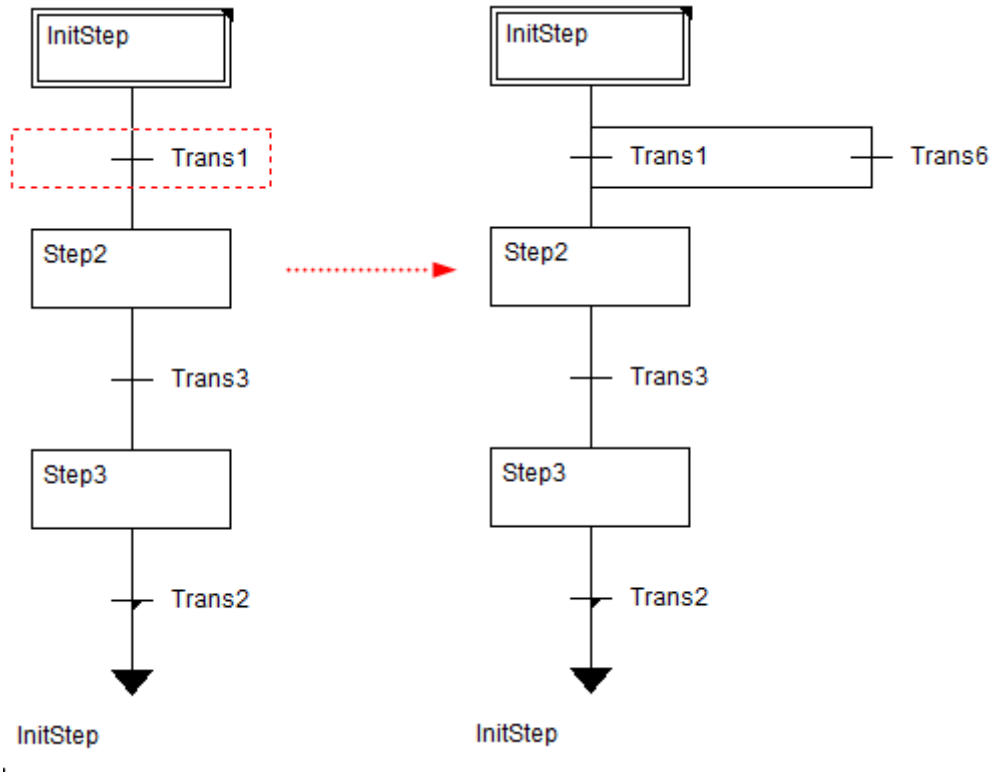
Follow the steps below to add an alternative branch:

Select a transition or a unit that begins and ends with a transition and add an alternative branch in the following ways:

- Menu bar: Choose [Insert] - [Alternative Branch (Right)/(Left)].
- Toolbar:  .
- Programming area: Right-click the selected transition or unit and click Alternative Branch (Right)/(Left).

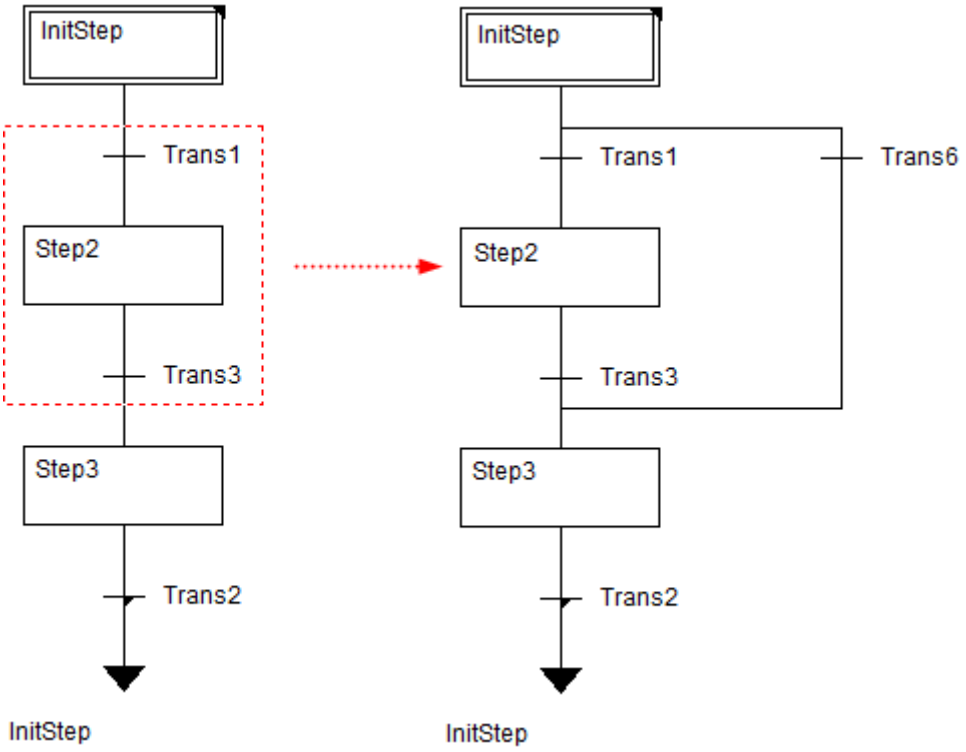
10.10.25.4 Example

The comparison before and after an alternative branch is added is shown in the figure below.



Selection Branch Adding of Single Transformation

Select a unit that starts and ends with a transition. To select multiple units, press the Ctrl key or the Shift key. The comparison before and after an alternative branch is added is shown in the figure below.



Selection Branch Adding for Unit

- If conditions of both branch transitions (Trans1 and Trans4 in the figure) are met, the program executes the left branch preferentially.

10.10.26 Delete Branch

10.10.26.1 Introduction

Added parallel or alternative branch can be deleted.

To delete the entire branch, delete all step-transition pairs.

To delete a part of the branch, delete all corresponding elements.

The deletion of the entire branch is taken as an example here.

10.10.26.2 Requirement

A branch has been added to the SFC program.

10.10.26.3 Delete parallel branch

Follow the steps below to delete a parallel branch:

(1) Selects all elements of the parallel branch and the transitions before and after the branch.

(2) Right-click in the selected area and click Delete.

10.10.26.4 Delete alternative branch

Follow the steps below to delete an alternative branch:

(1) Select all elements of the alternative branch and the transitions before and after the branch.

(2) Right-click in the selected area and click Delete.

10.10.27 Jump Element

10.10.27.1 Introduction

Jump is a kind of transition. SFC program can implement loops using jumps. POUs edited using the SFC language should contain at least one jump.

As a kind of transition, jump can achieve the transfer from a certain step to another step inside or outside the current branch directly. A jump is used to replace a step and is executed by the transition at its location. The jump is executed only when the transition conditions are met.

Jump Element

A jump must be after a transition. Therefore, jumps can be inserted directly into alternative branches. For parallel branches, use the "Transition-Jump" command to insert a jump. The destination of a jump element must be indicated by a step name or branch codename.

10.10.27.2 Reference message

Add jump to alternative branch

Add Jump to Parallel Branch

10.10.28 Add Jump to Alternative Branch

10.10.28.1 Introduction

Add jumps to alternative branches.

10.10.28.2 Requirement

An alternative branch has been added.

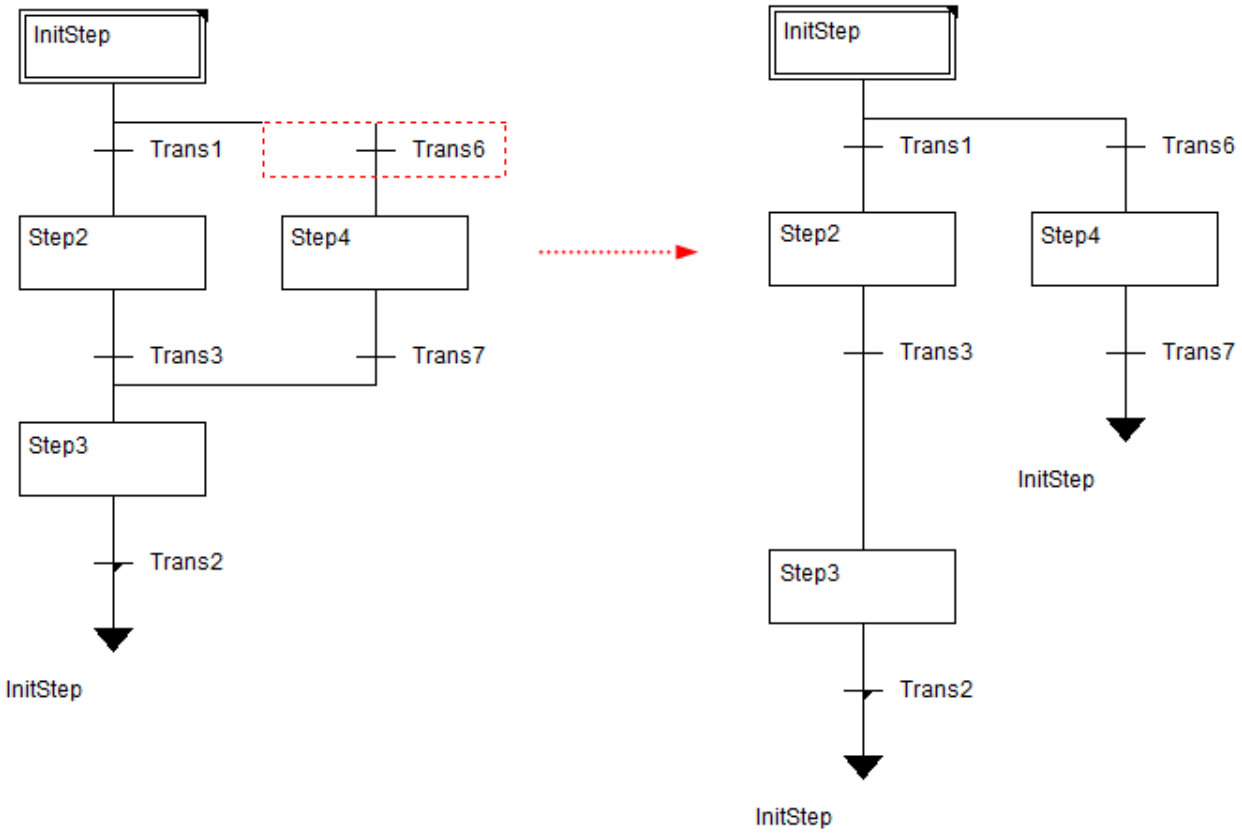
10.10.28.3 Add jump

Follow the steps below to add a jump to an alternative branch:

(1) Select an element on the alternative branch and add a jump in the following ways:

- Menu bar: Choose [Insert] - [Jump].
- Programming area: Right-click the step or transition of the branch, and click Jump.
- Toolbar:  ;
- Shortcut key: Ctrl + J.

The jump destination is displayed in the lower-left corner of the jump element, which is InitStep by default, indicating that the sequencer jumps to the initial step by default.



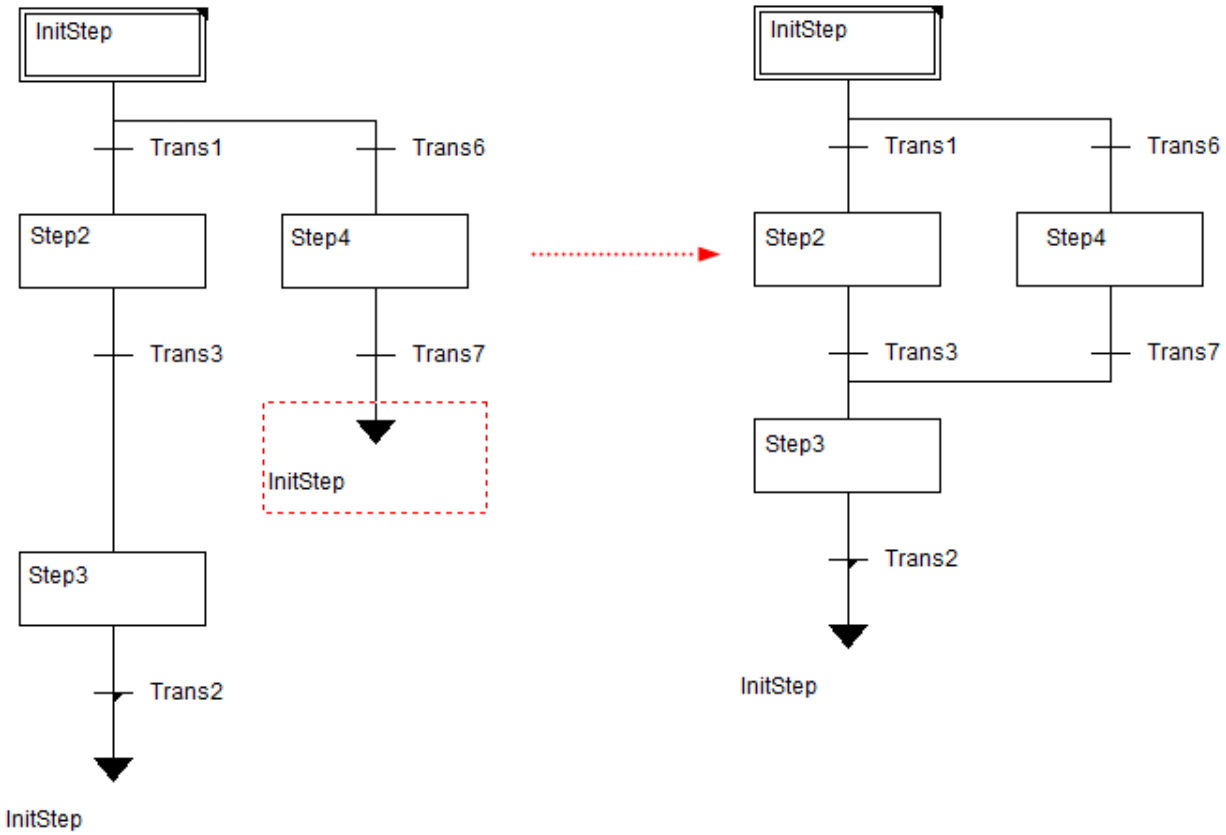
Insert Jump in Selection Branch

(2) Select the jump element, double-click the jump codename, and enter the name of the destination step or branch.

10.10.28.4 Delete jump

Follow the steps below to delete a jump from an alternative branch:

Select the jump element and execute the Delete command, as shown in the figure below.



Delete Jump on Selection Branch

10.10.29 Add Jump to Parallel Branch

10.10.29.1 Introduction

Add jumps to parallel branches. A jump is added to a parallel branch in the form of a transition-jump element. The destination of a jump element must be indicated by a step name or parallel branch codename. The transition-jump element is added after a step.

10.10.29.2 Requirement

A parallel branch has been added.

10.10.29.3 Add jump

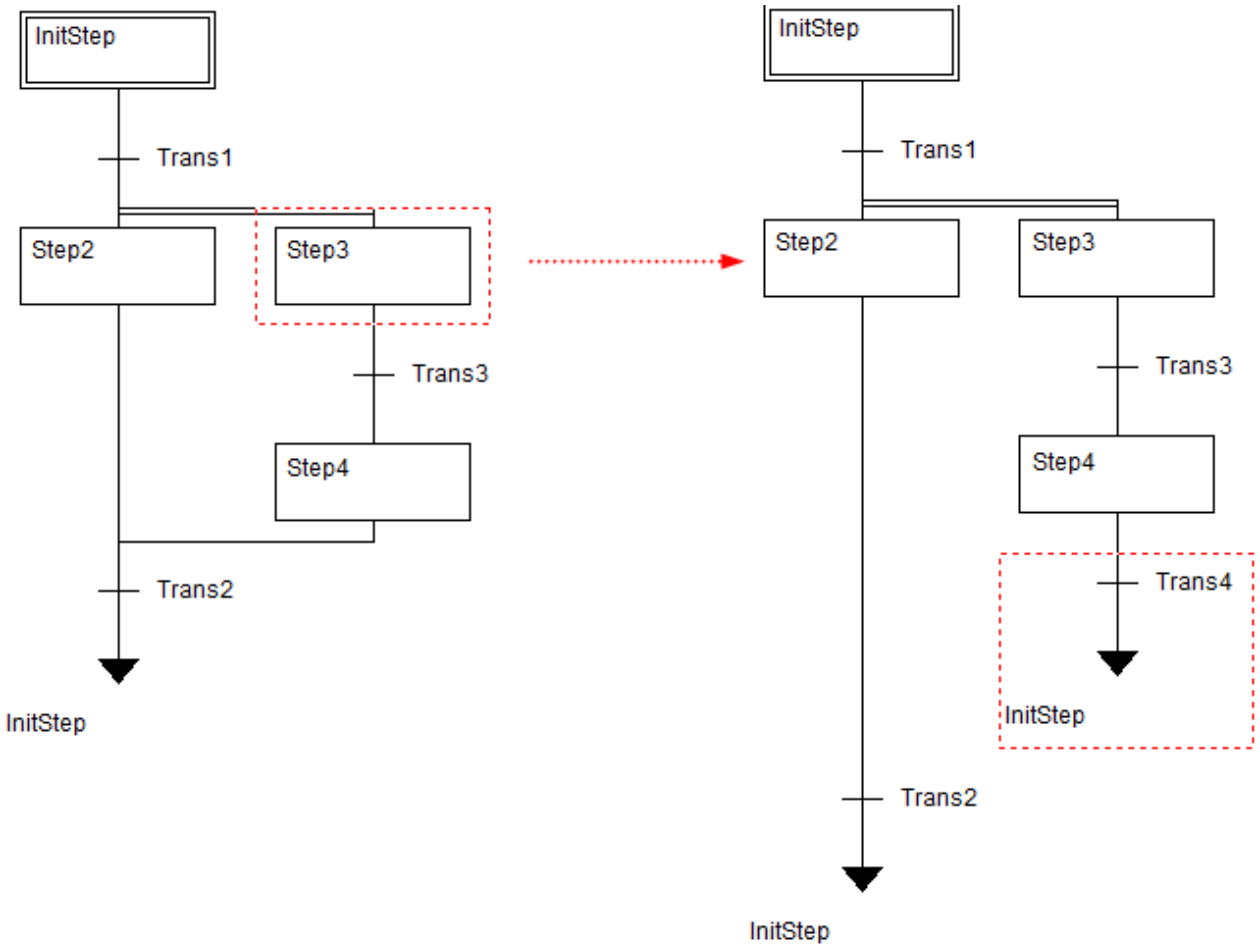
Follow the steps below to add a jump to a parallel branch:

(1) Select an element in the parallel branch and add a jump in the following ways:

- Menu bar: Choose [Insert] - [Transition-Jump (R)].

- Programming area: Right-click the step or transition of the parallel branch, and click Transition-Jump (R).
- Toolbar: ;
- Shortcut key: Ctrl + R.

The jump destination is displayed in the lower-left corner of the transition-jump element, which is InitStep by default, indicating that the sequencer jumps to the initial step by default.



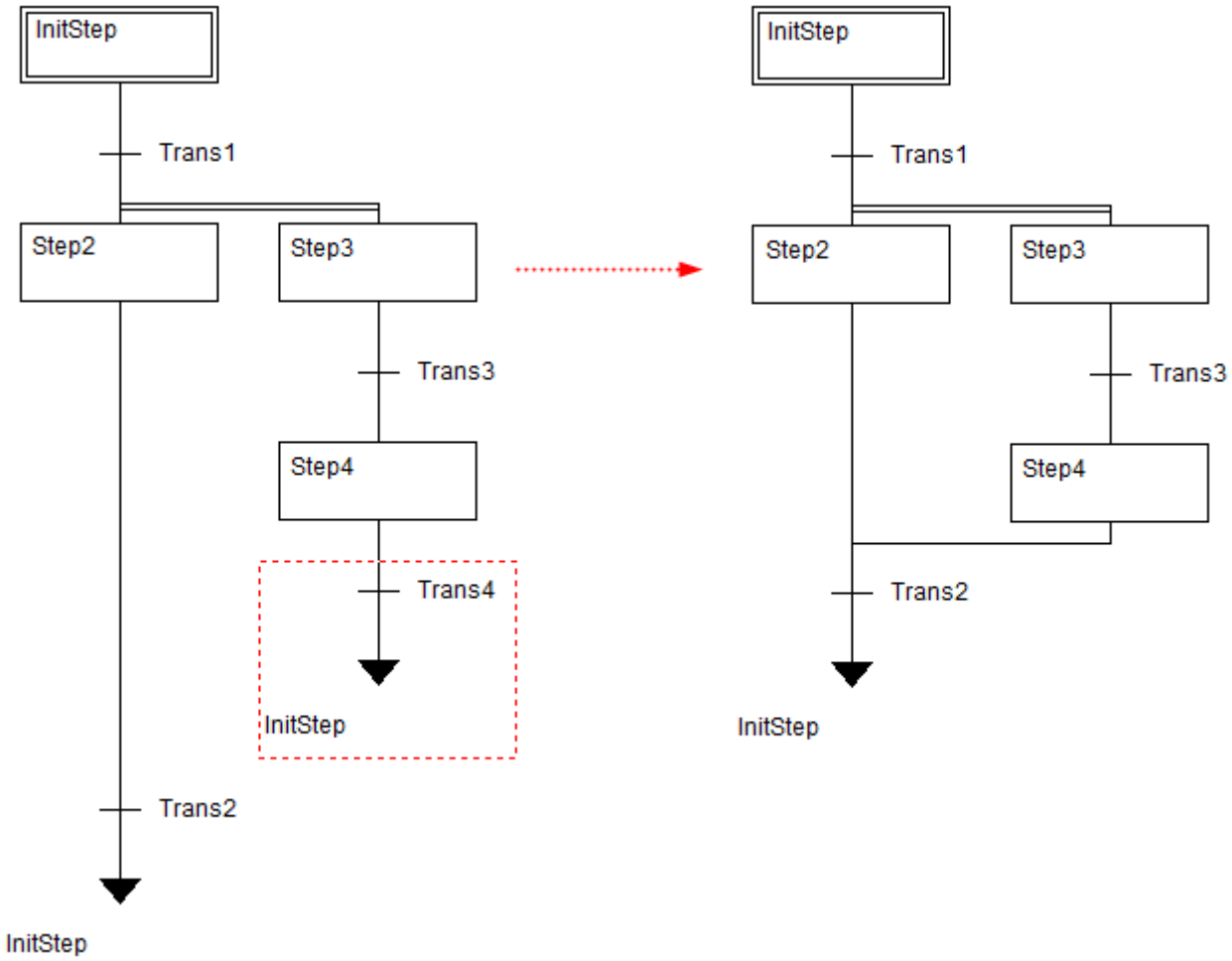
Parallel Branch Insert Transformation—Jump

(2) Select the jump element, double-click the jump codename, and enter the name of the destination step or branch.

10.10.29.4 Delete jump

Follow the steps below to delete a jump from a parallel branch:

Press the Shift key, select Transition-Jump, and execute the "Delete" command to delete the transition-jump element, as shown in the figure.



Delete Transformation—Jump

10.10.30 Cut, Copy, and Paste

Units that can be cut and copied include sequence pairs of steps and transitions, simultaneous branches, and alternative branches. Copied units cannot contain initial steps.

The paste command can be used when pasting any of the following units on the clipboard:

- Units that start and end with a step;
- Units that start and end with a transition;
- Units that start with a step and end with a transition;
- Units that start with a transition and end with a step.

10.10.31 Paste to the Right

10.10.31.1 Introduction

Paste the copied individual step or transition to the right of a step or transition.

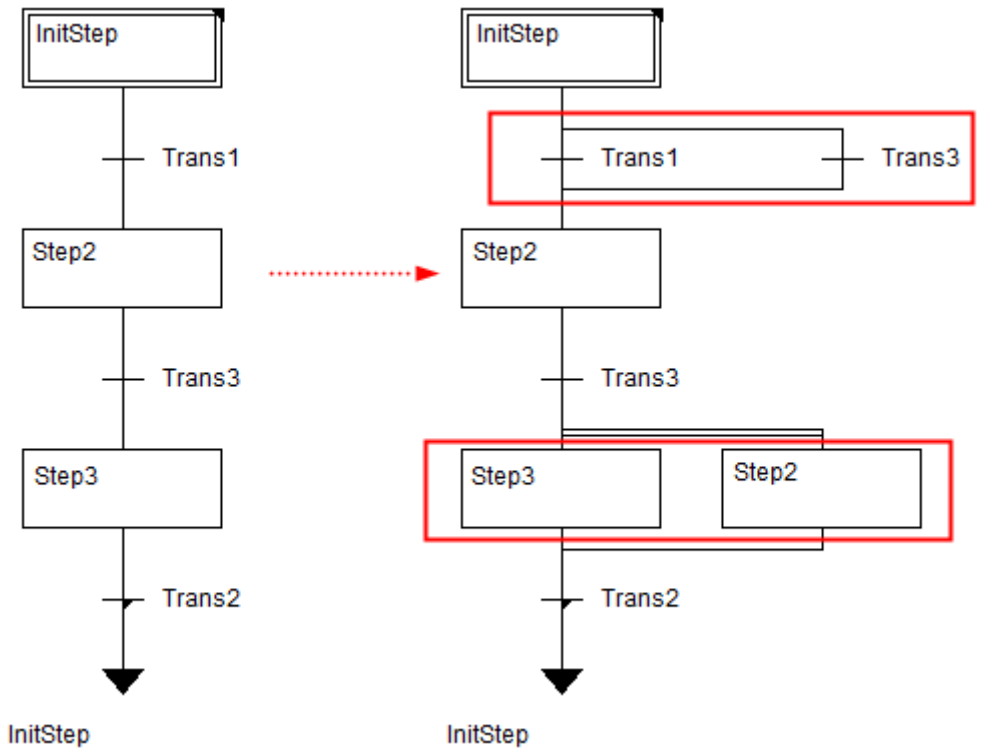
10.10.31.2 Steps

Follow the steps below to paste the content to the right:

(1) Copy a step or transition.

(2) Select the step or transition to be pasted, and paste it in the following ways.

- Menu bar: Choose [Extras] - [Paste Parallel Branch(Right)].
- Programming area: Right-click the step or transition, and click Paste Parallel Branch(Right).
- Shortcut key: Ctrl + V.



Paste to the Right

10.10.32 Paste After

10.10.32.1 Introduction

Paste the copied step-transition pair after a transition.

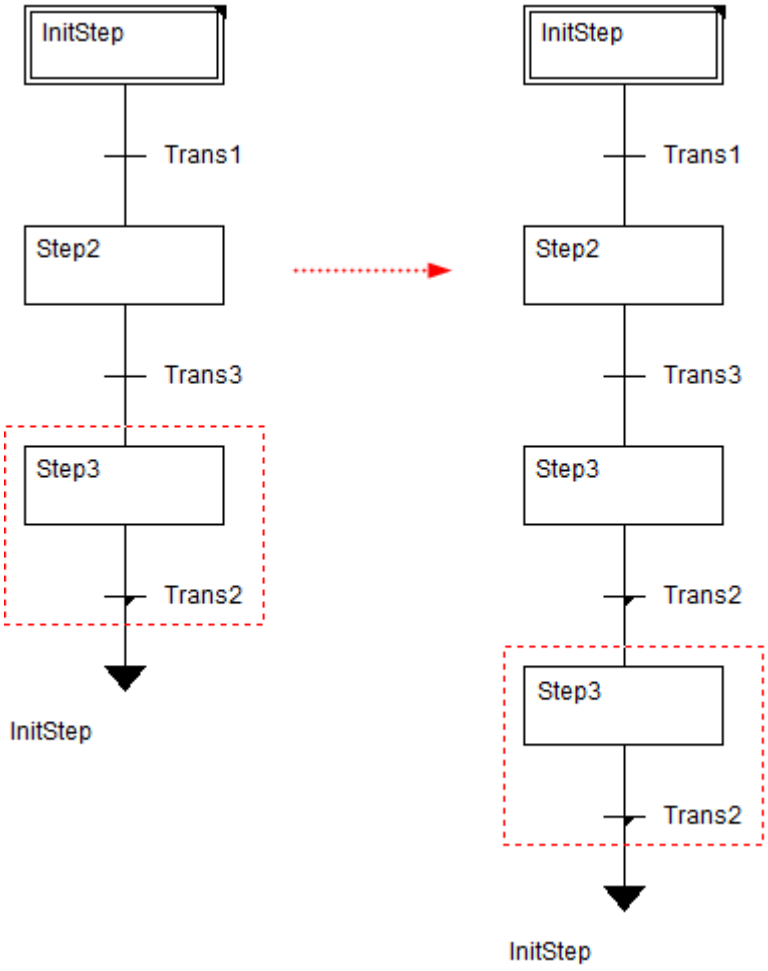
10.10.32.2 Steps

Follow the steps below to paste the step-transition pair to the back:

(1) Copy the step-transition pair.

(2) Select the target transition and paste the step-transition pair in the following ways.

- Menu bar: Choose [Extras] - [Paste After].
- Programming area: Right-click the transition and click Paste After.
- Shortcut key: Ctrl + V.

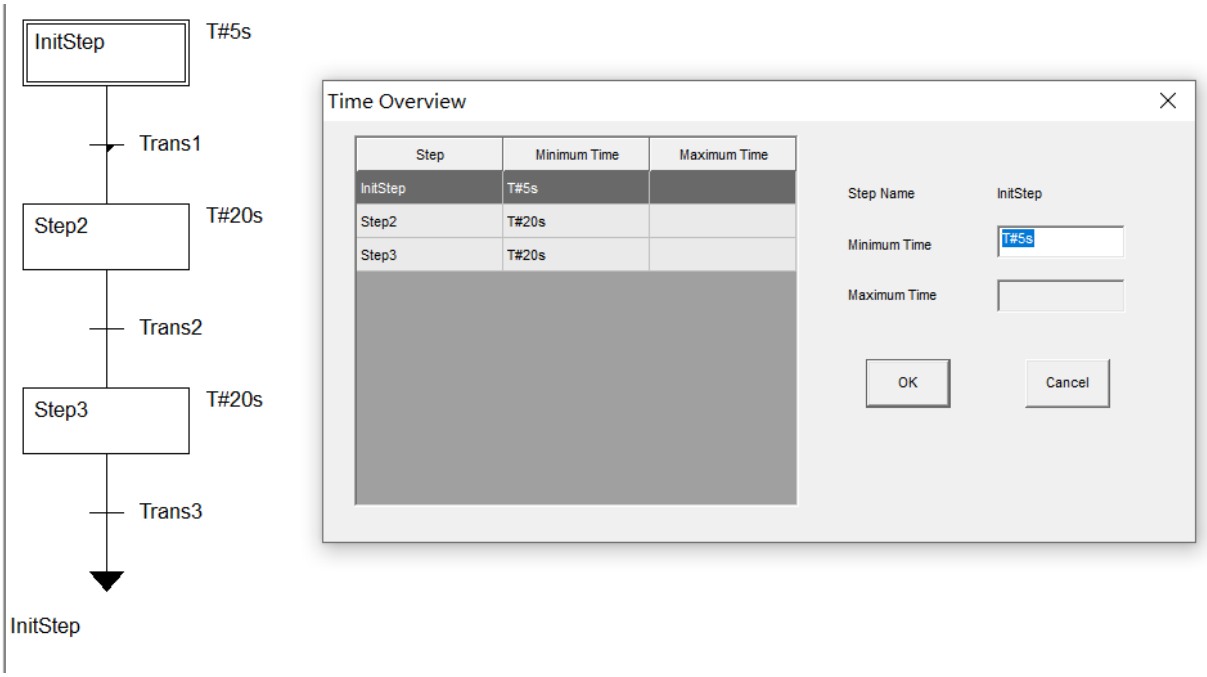


Paste to the Back

10.10.33 Time Overview

10.10.33.1 Introduction

In the Time Overview window, you can view and modify the settings of all step times on the SFC (PRG) program page.



10.10.33.2 Requirement

SFC POU is opened.

10.10.33.3 Steps

Follow the steps below to set the SFC step time:

(1) Open the Time Overview dialog box in the following ways:

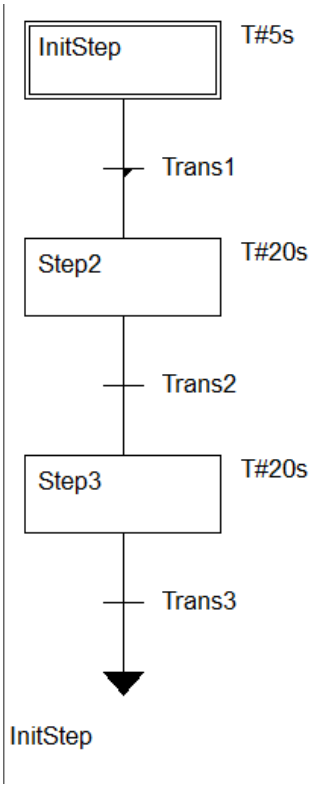
- Menu bar: Click [Extras] - [Time Overview (T)].
- Programming area: Right-click the blank area, and click Time Overview (T).

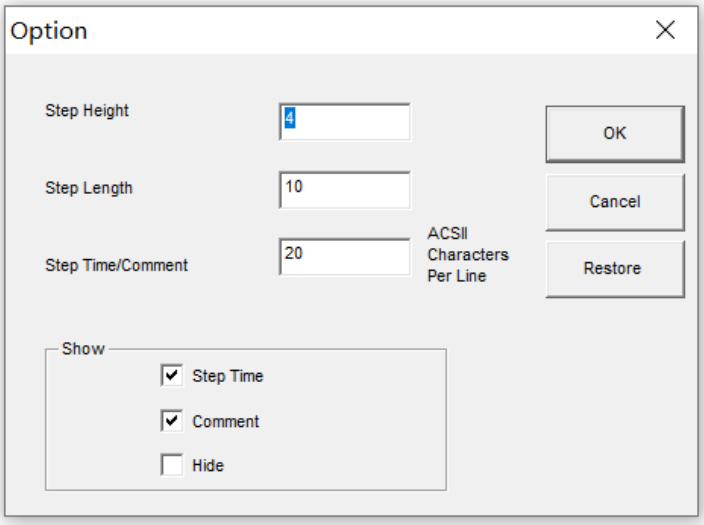
(2) Set the step time according to the program, and click OK.

10.10.34 Set Step Display Parameters

10.10.34.1 Introduction

In the Options window, you can set the height, length, and comment width of steps and other parameters for step display (identification box), as shown in the figure below.

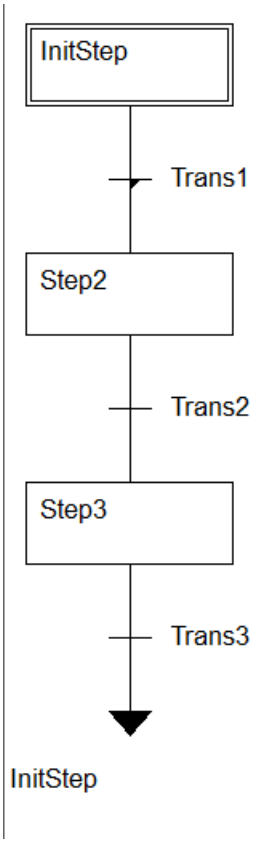


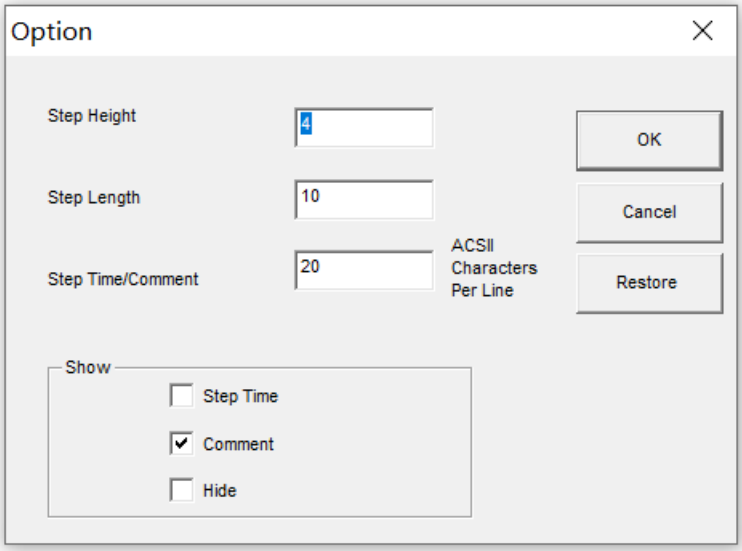


Valid value ranges

Parameter	Valid Range
Step Height	4-16 unit lengths (10 pixels).
Step Length	8-16 unit lengths (10 pixels).
Step Time/Comment	4-20 characters in each line.
Step Display	Specify whether to display the step time and comment set in Step Properties. After the minimum time and comment of a step are set in Step Properties, you can decide the step content to be displayed in the Options dialog box. By default, the Step Time option is checked. If the comment needs to be displayed, check the Step Comment option. You can also check the Hide option to display no content.
Restore Default	Click this button to restore the set values to default ones.

If the Step Comment option is checked, the displayed SFC program content is as shown in the figure below.





10.10.35 SFC Programming Example

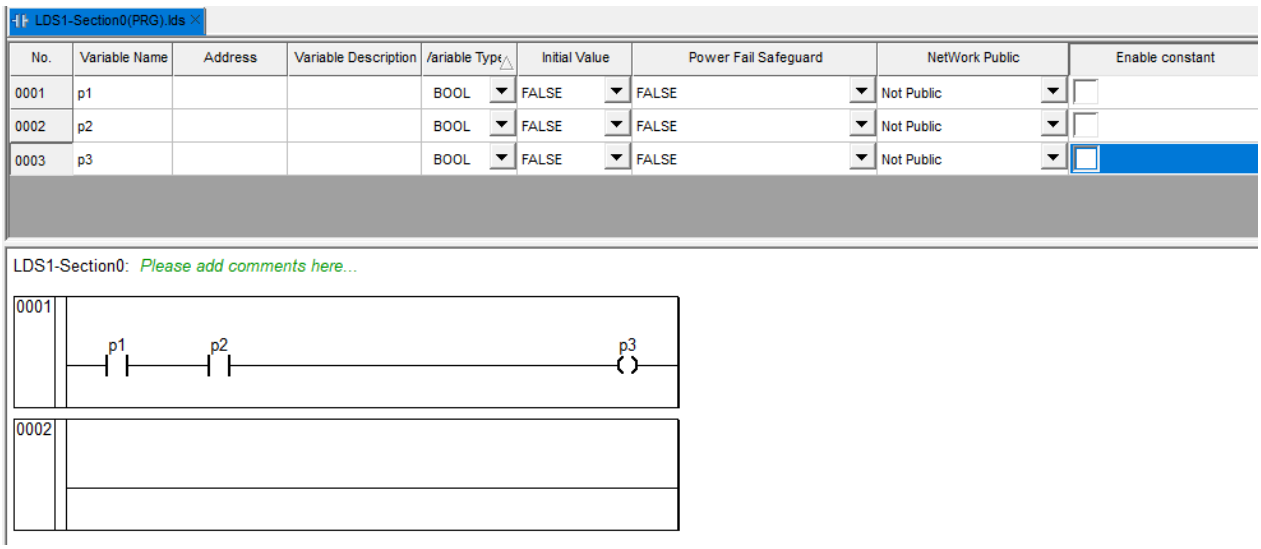
The example below can help users better understand SFC programming. In the example, row and column numbers indicate the location of specific elements.

10.11 Create LDS Program

10.11.1 LDS Programming Overview

LD is short for Ladder Diagram Segmented, a graphical programming language.

LDS consists mainly of programming elements such as contacts, coils, functional blocks, and connecting lines, which are connected by horizontal and vertical lines to form a planar mesh diagram, as shown in the figure below.



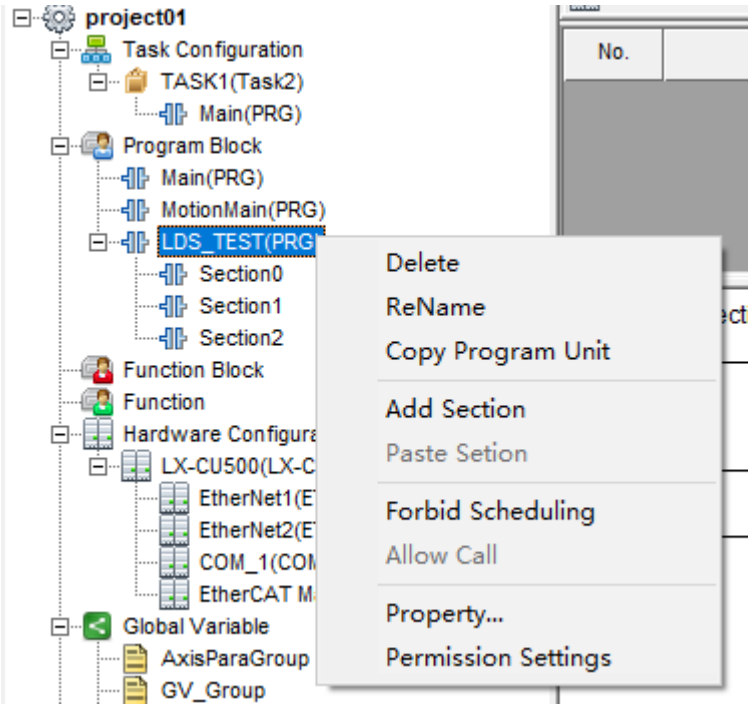
The leftmost vertical line of the section is generally called the Energy Line. and its state is always TRUE. Each programming element is connected to each other by certain rules, and finally connected to this energy line, forming a section, segment, or network, which accomplishes a specific logic operation. The line on each section is called a busbar, where contact, coil, and functional block can be added.

10.11.2 LDS Elements

10.11.2.1 Program Segment

A program segment is a part of a program. When the entire program content is too long, for ease of reading, the program can be segmented and placed in different program segments, with all program segments sharing a variable area.

Select a program segment and drag it to adjust its position. The program is executed in order from top to bottom during runtime. For example, as shown in the following figure, drag Section 0 to the position of Section 2, release the mouse, and Section 0 will move to the front of Section 2.



10.11.2.2 Section

A section is the basic unit of an LDS program. Each POU written in LDS consists of sections. The section number defaults from 0001. Up to 999 sections can be added to each POU.

- No more than 64 columns horizontally and 64 rows vertically shall be included in the network of each section of the LDS programming area.

10.11.2.3 Contact

Each contact represents a BOOL variable, which can be a channel input variable such as a switch or button, or an internal variable. The contact passes the value from left to right.

┤ ├: Normally open contact. When the variable value is TRUE, the contact is turned on and the energy flow is transmitted to the right. Otherwise, the energy flow is broken at the contact.

10.11.2.4 Coil

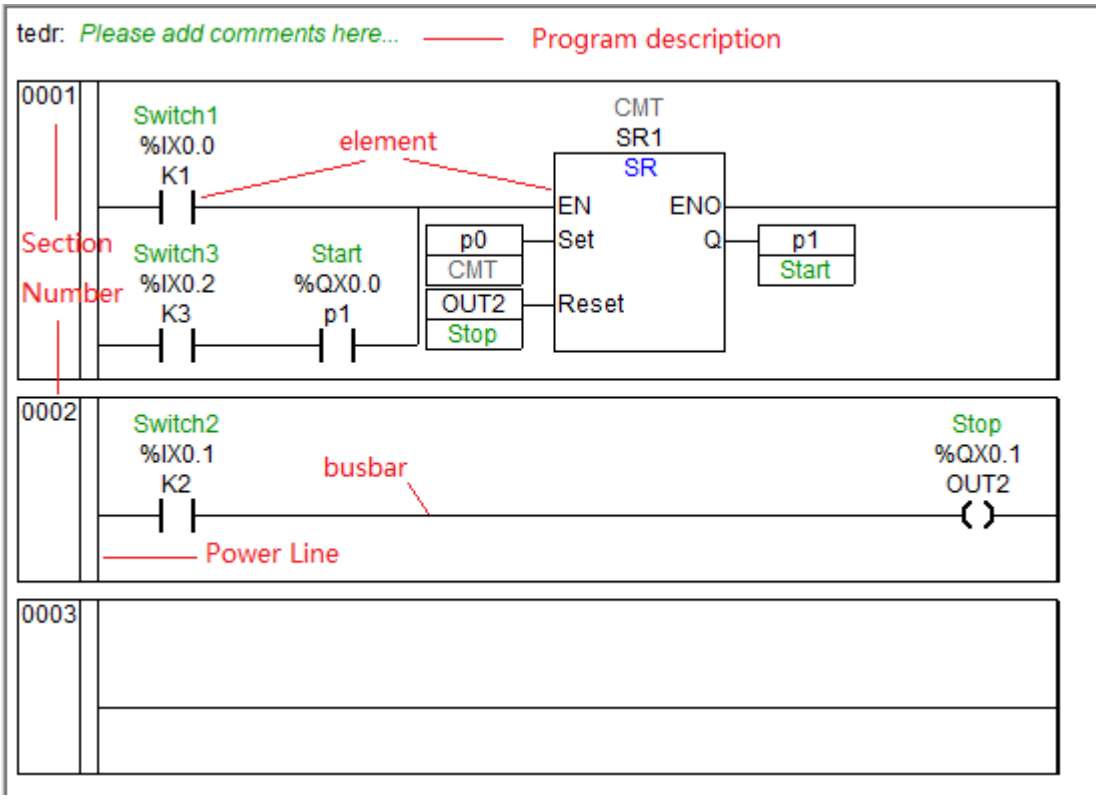
The coil represents the output of a logic operation, denoted by (). It transmits the energy flow from left to right, which can be connected to the channel output variable to control the start and stop of equipment, the start, stop and shutdown of the motor, etc.

10.11.2.5 Block elements

The block elements include three types: functional block, function, and program. See POU Types for a detailed description.

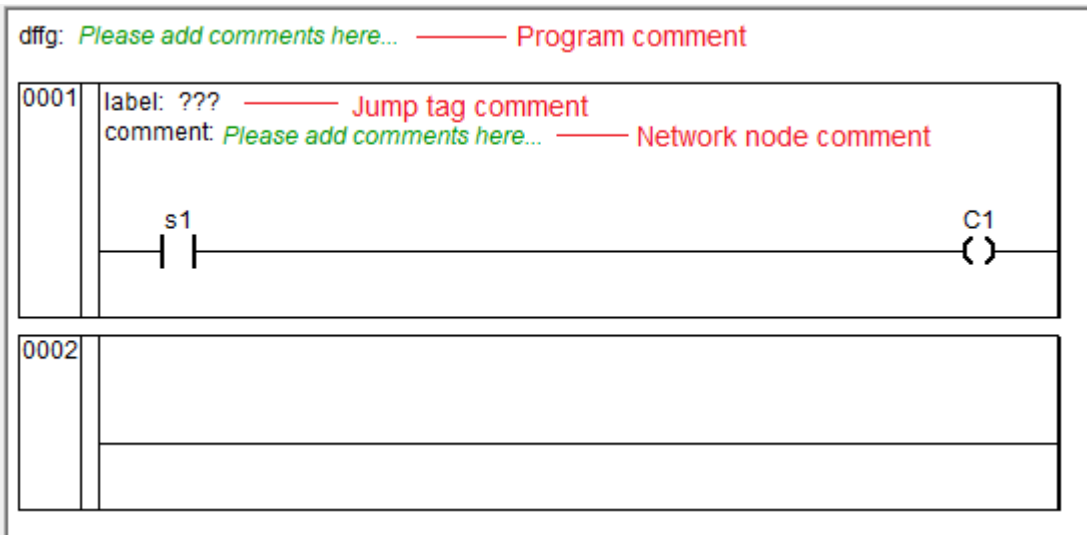
10.11.2.6 Legend description

An LDS program consists of a series of sections. The energy lines on the left and right and the elements on the middle busbar form the entire program network.



10.11.2.7 Comment description

- Program comment: Used to describe the POU program, such as Pump start/stop.
- Jump tag comment: Used to identify the destination program segment of a jump. It is not shown by default. Users need to add Jump Codename to show it. Up to 32 characters can be displayed.
- Network node comment: Used to describe the current section of the program. It is not shown by default. Users need to select the Modify Comment command in the right-click menu of the node to show the comment area.

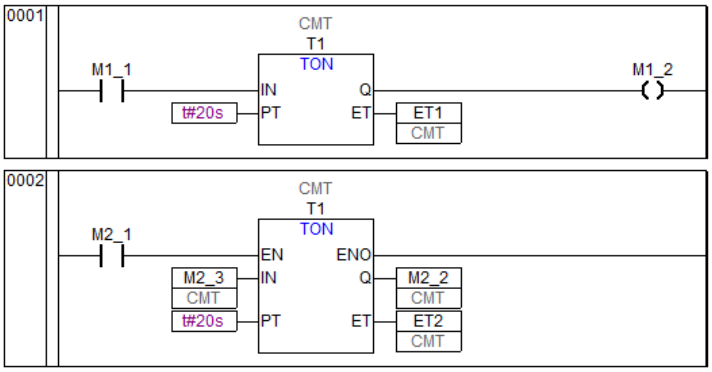


10.11.2.8 Functional block and enable operator

The fundamental difference between a functional block and an enable operator is the way in which they are executed. For a functional block, it is executed when the program is running, regardless of whether it is enabled or not. However, for an enable operator, it is executed only when the enable terminal EN is valid. A simple program written in LD language is described here.

No.	Variable Name	Address	Variable Description	Variable Type	Initial Value
0003	T1			TON	
0004	ET1			TIME	T#0MS
0005	M2_1			BOOL	FALSE
0006	M2_3			BOOL	FALSE
0007	M2_2			BOOL	FALSE
0008	TON1			TON	
0009	ET2			TIME	T#0MS

tes: Please add comments here...

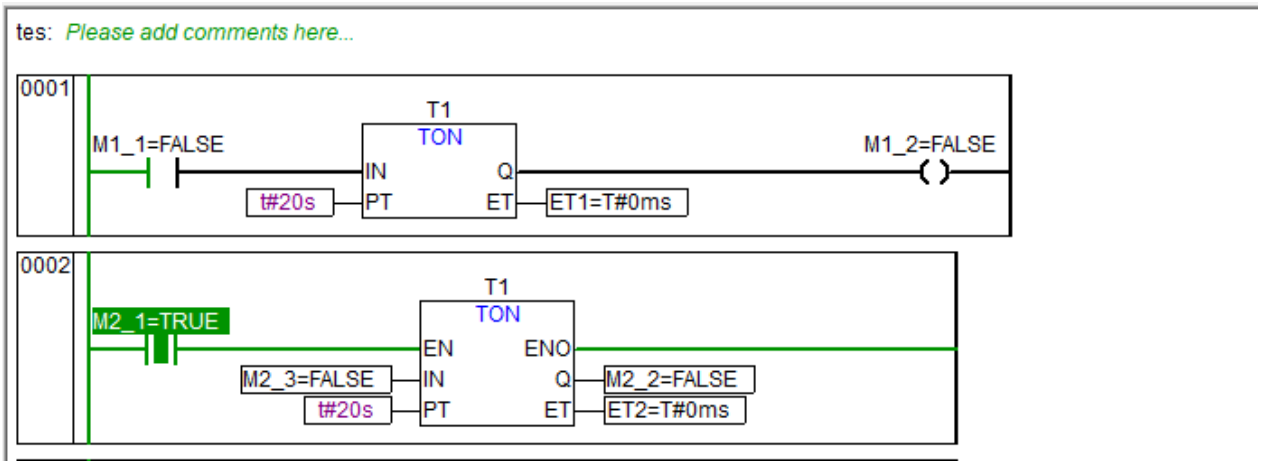


As shown in the figure, the power-on delay timers are called in sections 0001 and 0002 with a functional block and an enable operator, respectively.

When starting to run the program, first turn on M1_1, M2_1, and M2_3 at the same time. At this time, the

two timers are running in the same state. Disconnect M1_1 and M2_1, and keep M2_3 turned on. At this time, ET1 is cleared, while ET2 is not cleared.

Then disconnect M2_3. The ET2 time is still not cleared. Only when M2_1 is kept turned on and M2_3 is detected as disconnected, ET2 is cleared. This is because only when the enable terminal is valid, the code inside the TON is executed.



To use the enable operator to call the subprogram, when the enable terminal is invalid, the subprogram is not executed. To use the functional block to call the subprogram, when the functional block is enabled, the enabled part is run; And when the functional block is not enabled, the non-enabled branch is run.

10.11.3 Select Element

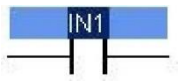
Elements must be selected for all operations related to LDS configuration. Users can select a single element or multiple elements to execute related operations.

The commands that can be executed vary depending on the selected position.

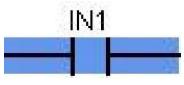
The element is highlighted in blue when it is selected by the cursor.

See the following for details of the selected positions and operations:

- Select Text Area: Click the variable name or block name of the element



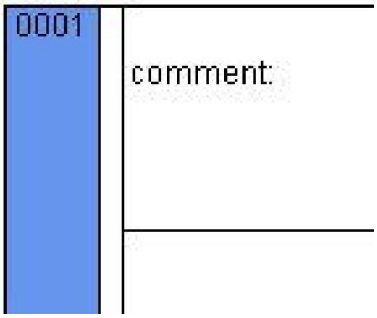
- Select Contact: Click the contact



- Select Block Elements: Click the block element

- Select Coil: Click the coil

- Select Section: Click the section



- Select Comment for Network Node: Click the Please add a comment here...

- Select Multiple Elements:

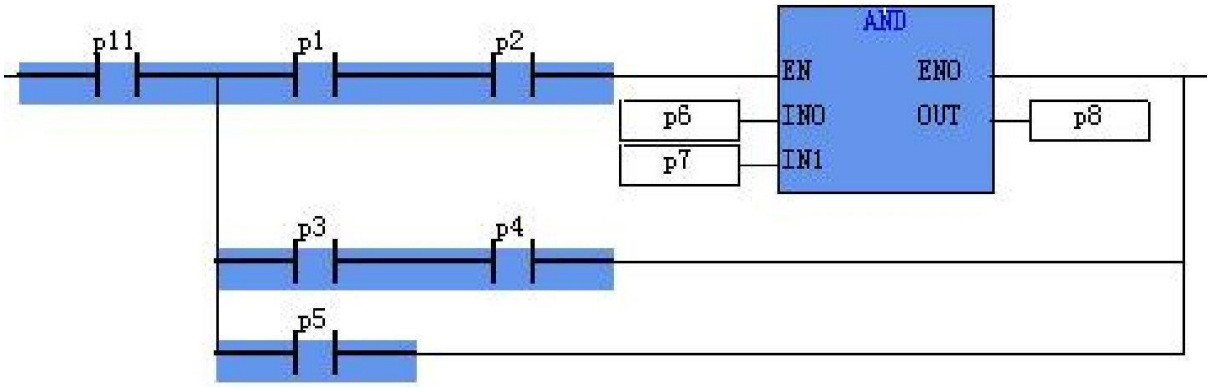
(1) Select an element on the section with the cursor as the starting element.

(2) Press and hold Shift, and click the ending element. At this time, all elements between the two elements are selected.

Note:

The starting and ending elements here refer to the positions where the elements are located, counted sequentially from top to bottom and from left to right.

When multiple elements are selected to perform operations, keep pressing the Shift key until the operation is performed.



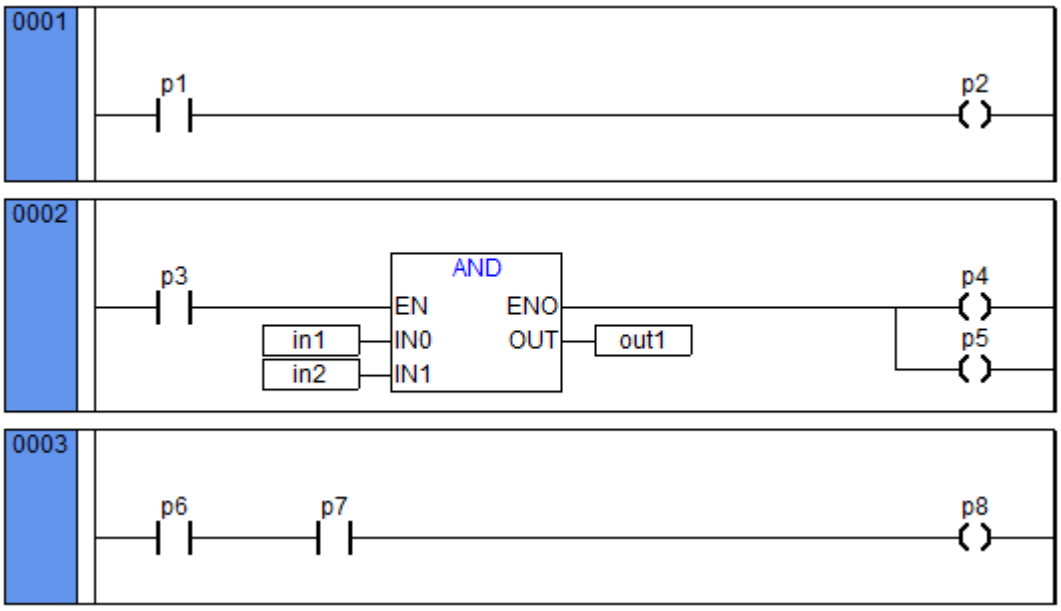
■ Select Program Segment:

(1) Select the program section.

(2) Press and hold Shift.

(3) Click the ending section of the program segment.This program section is selected.

When performing a right-click operation, keep pressing Shift until the operation command is executed.



• In the LDS programming area, when you add a block element, an AND functional block with enabled input pin EN and output pin ENO. The output value of ENO is equal to the input value of EN.

10.11.4 Insert in Front of the Section

10.11.4.1 Introduction

Insert a new section in front of the section where the cursor is located.

10.11.4.2 Steps

To insert a new section in front of the selected section, follow the steps below:

(1) Select the section. Add the entry action via:

- Menu bar: Click [Insert] - [Network(Before)];
- Programming area: Right-click the section. Click Network(Before).

(2) Right-click the section. Select the Previous Section command. At this point, a new section is inserted in front of the current section and is highlighted in blue, indicating it is selected.

10.11.5 Insert After the Section

10.11.5.1 Introduction

Insert a new section after the section where the cursor is located.

10.11.5.2 Steps

To insert a new section after the selected section, follow the steps below:

(1) Select the current section with the cursor.

- Menu bar: Click [Insert] - [Network(After)];
- Programming area: Right-click the section. Click Network(After).

(2) Right-click the section. Select the Next Section command. At this point, a new section is inserted after the current section, and is highlighted in blue, indicating it is selected.

10.11.6 Insert Contact

10.11.6.1 Insert contact forward

10.11.6.1.1 Introduction

Connect a contact in series in front of a contact, block element, coil element, or parallel logic.

10.11.6.1.2 Requirement

The element is selected.

10.11.6.1.3 Steps

To connect a contact in series forward, follow the steps below:

(1) Right-click the element.

- Note: If the currently selected element is parallel logic, users need to press and hold Shift key, and right-click the blue highlighted area. If users right-click other blank areas, the selected status is invalid.

(2) In the right-click menu, select Insert Contact command. When the right-click menu pops up, release the Shift key. At this time, a contact is inserted and is highlighted in blue, indicating it is selected.

10.11.6.1.4 Reference message

Select Element.

10.11.6.2 Insert contact backward

10.11.6.2.1 Introduction

Connect a contact in series behind a contact, block element, or parallel logic.

10.11.6.2.2 Requirement

The element is selected.

10.11.6.2.3 Steps

To connect a contact in series backward, follow the steps below:

(1) Right-click the element.

- Note: If the currently selected element is parallel logic, users need to press and hold Shift key, and right-click the blue highlighted area. If users right-click other blank areas, the selected status is invalid.

(2) Select the Append Contact command in the right-click menu or click the Connect Contact in Series icon in the toolbar. When the right-click menu is displayed, release Shift key.

(3) At this time, a contact is inserted and is highlighted in blue, indicating it is selected.

10.11.6.2.4 Reference message

Select Element.

10.11.6.3 Insert contact at the beginning of the section

10.11.6.3.1 Introduction

Insert a contact at the beginning of the current section.

10.11.6.3.2 Requirement

The current section is selected.

10.11.6.3.3 Steps

To insert a contact at the beginning of the section, follow the steps below:

(1) Right-click the section. In the right-click menu, select the Insert Contact command.

(2) At this time, a contact is inserted and is highlighted in blue, indicating it is selected.

10.11.6.3.4 Reference message

Select Element.

10.11.6.4 Connect contact in parallel

10.11.6.4.1 Introduction

Connect a contact in parallel to a contact, block element or multiple elements.

10.11.6.4.2 Requirement

The element is selected.

10.11.6.4.3 Steps

To connect a contact in parallel, follow the steps below:

(1) Click the menu bar [Insert] - [Parallel Contact], or click the “Parallel Contact” icon on the toolbar.

(2) At this time, a contact is connected in parallel to the currently selected element and is highlighted in blue, indicating it is selected.

10.11.6.4.4 Reference message

Select Element.

10.11.6.5 Connect Negated contact in series

10.11.6.5.1 Introduction

Insert a negated contact behind a contact, block element, or in front of a coil. Or connect an negated contact in series for parallel logic.

10.11.6.5.2 Requirement

The element is selected.

10.11.6.5.3 Steps

To insert an negate contact, follow the steps below:

(1) Right-click the element.

- Note: If the currently selected element is parallel logic, users need to press and hold Shift key,

and right-click the blue highlighted area. If users right-click other blank areas, the selected status is invalid.

(2) Select the [Insert] - [Negate Contact] command from the menu bar, or choose the “Negate Contact” command from the right-click menu, or click the toolbar icon  .

(3) When the right-click menu is displayed, release Shift key.

(4) At this time, a negate contact is inserted and is highlighted in blue, indicating it is selected.

10.11.6.5.4 Reference message

Select Element.

10.11.6.6 Connect Negated contact in parallel

10.11.6.6.1 Introduction

Connect a negate contact in parallel to a contact, block element or multiple elements.

10.11.6.6.2 Requirement

The element is selected.

10.11.6.6.3 Steps

To connect a negate contact in parallel, follow the steps below:

(1) Select the [Insert] - [Parallel Negate Contact] command from the menu bar, or click the “Parallel Inverted Contact” icon  on the toolbar.

(2) At this time, an negate contact is connected in parallel to the currently selected element, and is highlighted in blue, indicating it is selected.

10.11.6.6.4 Reference message

Select Element.

10.11.7 Insert Coil

10.11.7.1 Insert coil

10.11.7.1.1 Introduction

Insert a coil at the end of the current section busbar. Or connect a coil in parallel below an existing output element.

10.11.7.1.2 Requirement

The current section is selected.

10.11.7.1.3 Steps

To insert a coil, follow the steps below:

(1) Right-click the section.

(2) In the right-click menu, select the Coil command. Or click the Toolbar icon  .

(3) At this time, a coil is inserted at the end of the section and is highlighted in blue, indicating it is selected.

(4) To continue connecting coil in parallel, repeat the above steps.

10.11.7.2 Insert set coil

10.11.7.2.1 Introduction

Insert a set coil at the end of the current section busbar. Or connect a new set coil in parallel below the existing one.

10.11.7.2.2 Requirement

The current section is selected.

10.11.7.2.3 Steps

To insert a set coil, follow the steps below:

(1) Right-click the section.

(2) In the right-click menu, select the Set Coil command. Or click the Toolbar icon .

(3) At this time, a set coil is inserted at the end of the section, and is highlighted in blue, indicating it is selected.

(4) To continue connecting the set coils in parallel, repeat the above steps.

10.11.7.2.4 Reference message

Select Element

LDS Elements

10.11.7.3 Insert reset coil

10.11.7.3.1 Introduction

Insert a reset coil at the end of the current section busbar. Or connect a new reset coil in parallel below the existing one.

10.11.7.3.2 Requirement

The current section is selected.

10.11.7.3.3 Steps

To insert a reset coil, follow the steps below:

(1) Right-click the section.

(2) In the right-click menu, select the Reset Coil command. Or click the Toolbar icon .

(3) At this time, a reset coil is inserted at the end of the section, and is highlighted in blue, indicating it is selected.

(4) To continue connecting the reset coils in parallel, repeat the above steps.

10.11.7.3.4 Reference message

Select Element

10.11.7.4 Insert inverted coil

10.11.7.4.1 Introduction

Insert an inverted coil at the end of the current section busbar. Or connect a new inverted coil in parallel below the existing one.

10.11.7.4.2 Requirement

The current section is selected.

10.11.7.4.3 Steps

To insert an inverted coil, follow the steps below:

(1) Right-click the section.

(2) In the right-click menu, select the Invert Coil command. Or click the Toolbar icon .

(3) At this time, an inverted coil is inserted at the end of the section, and is highlighted in blue, indicating it is selected.

(4) To continue connecting the inverted coils in parallel, repeat the above steps.

10.11.7.4.4 Reference message

Select Element

10.11.7.5 Set and reset coil

10.11.7.5.1 Introduction

By setting/resetting, set the coil variable to 1 to set it as a set coil or reset coil.

Set (S): When the logic operation value of the input coil is 1, the coil variable is set to TRUE. The variable remains TRUE until it is reset to FALSE.

Reset (R): When the logic operation value of the input coil is 1, the coil variable is set to FALSE. The variable remains FALSE until it is set to TRUE.

When there are both set and reset signals, the reset has a higher priority.

Select the Set/Reset command via:

- Programming area: Coil right-click menu - [Set/Reset];
- Shortcut key: Ctrl + T;

- Toolbar:  .

10.11.7.5.2 Requirement

The coil is selected.

10.11.7.5.3 Set coil

To set a coil, follow the steps below:

(1) Right-click the coil, or reset the coil.

(2) In the right-click menu, select the Set/Reset command.

(3) The coil is set as a set coil.

10.11.7.5.4 Reset coil

To reset a coil, follow the steps below:

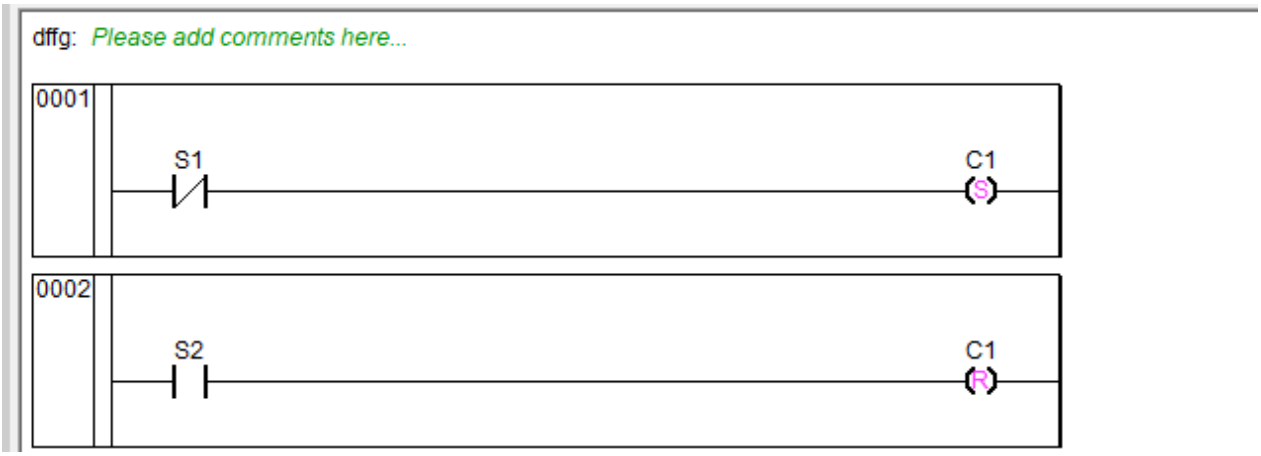
(1) Right-click the coil, or set the coil.

(2) In the right-click menu, select the Set/Reset command.

(3) The coil is set as a reset coil.

10.11.7.5.5 Example

When S1 is FALSE, the coil variable C1 is set to TRUE; Only when S2 is TRUE, the coil variable C1 is reset to FALSE. If coil C1 is in the reset state, when S1 is FALSE, the coil remains in the reset state until S2 is FALSE. At this time, the coil is set to TRUE again.



10.11.7.5.6 Reference message

Select Element

10.11.8 Insert Block Element

10.11.8.1 Connect block element in series

10.11.8.1.1 Introduction

Connect a block element in series to the element. The block element defaults to an AND functional block and has an enabled input pin EN. When EN is TRUE, the block element is executed.

The block element is inserted in a different position depending on the selected area.

- Select the section. Execute the Block Element command. The block element is inserted at the beginning of the section.
- Select the contact and execute the Block Element command. Then the block element is inserted behind the contact.
- Select the coil and execute the Block Element command. Then the block element is inserted before the coil.

You can execute the Block Element command in the following ways:

- Programming area: Right-click menu - Block Element;
- Toolbar: ;
- Shortcut keys: Alt + F9.

10.11.8.1.2 Requirement

The area where the element needs to be inserted has been selected.

10.11.8.1.3 Steps

Follow the steps below to insert a block element:

(1) Right-click the selected area.

(2) Select the Block Element command in the menu that appears.

(3) A block element is inserted and highlighted in blue, indicating it is selected.

10.11.8.1.4 Reference message

Select Element

10.11.8.2 Add parallel block element

10.11.8.2.1 Introduction

Add the parallel output block element to the coil.

10.11.8.2.2 Requirement

The coil is selected.

10.11.8.2.3 Steps

Follow the steps below to add a parallel block element:

(1) Right-click the coil.

(2) Select the Parallel Block Element command in the menu that appears.

(3) A block element is inserted below the coil and highlighted in blue, indicating it is selected.

10.11.8.2.4 Reference message

Select Element

10.11.9 Configure Block Element Pin

10.11.9.1 Add input pin

10.11.9.1.1 Introduction

When the default input pins of a block element are not sufficient, additional input pins can be added. Only one input pin can be inserted at a time. The total number of input pins shall not exceed 64.

- When the input pin is added, the pin insertion position differs depending on the area selected.
- Select the block element. Execute the Multiple Inputs command. The new pin is inserted into the last one.
- Select the input pin. Execute the Multiple Inputs command. The new pin is inserted in front of the selected pin.

You can execute the More Input Pins command in the following ways:

- Programming area: Block element right-click menu - [Advanced] - [Multi-Inputs (D)];
- Programming area: Input pin right-click menu - Multi-Inputs (D);
- Shortcut key: Ctrl + D.

10.11.9.1.2 Requirement

The block element or input pin has been selected.

10.11.9.1.3 Steps

Follow the steps below to add an input pin:

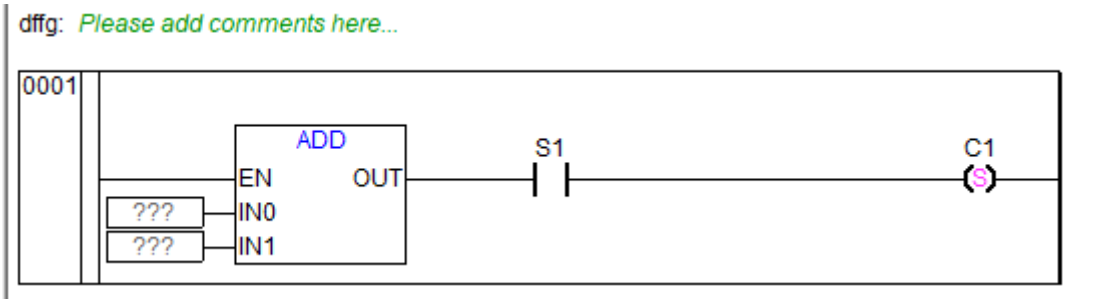
(1) Right-click the selected area.

(2) Select the More Input Pins command in the menu that appears.

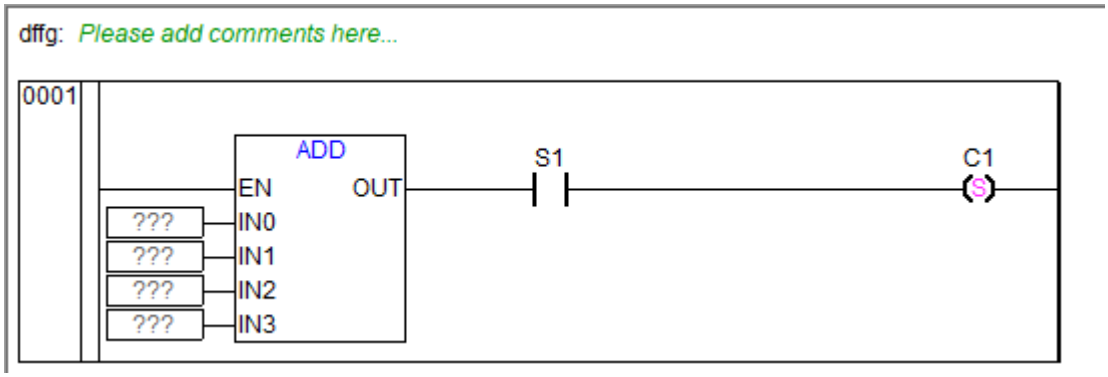
(3) An input pin is inserted and highlighted in blue, indicating it is selected.

10.11.9.1.4 Example

As shown in the figure, for the selected block element, execute 2 times Multiple Inputs command to add 2 input pins, IN2 and IN3.



(a)



(b)

10.11.9.1.5 Reference message

Select Element

10.11.9.2 Show or hide pins

10.11.9.2.1 Introduction

When configuring the block element, set the showing and hiding of enabled pins, input and output pins as needed.

All pins of the functional block can be set; For other block elements, only the EN and ENO pins can be set.

10.11.9.2.2 Requirement

The block element has been configured.

10.11.9.2.3 Hide pins

Follow the steps below to hide input pins:

(1) Right-click the block element.

(2) In the right-click menu, select the [Advanced] - [Pins (Show/Hide)] command.

(3) The Set Hide/Show for Pins dialog box is displayed.

(4) Click the checkbox to uncheck the pins.

(5) Click OK.

(6) The corresponding pins of the block element are hidden.

10.11.9.2.4 Show pins

Follow the steps below to hide input pins:

(1) Right-click the block element.

(2) In the right-click menu, select the [Advanced] - [Pins (Show/Hide)] command.

(3) The Set Hide/Show for Pins dialog box is displayed.

(4) Check the pins to be displayed.

(5) Click OK.

(6) The input pins of the block element are displayed.

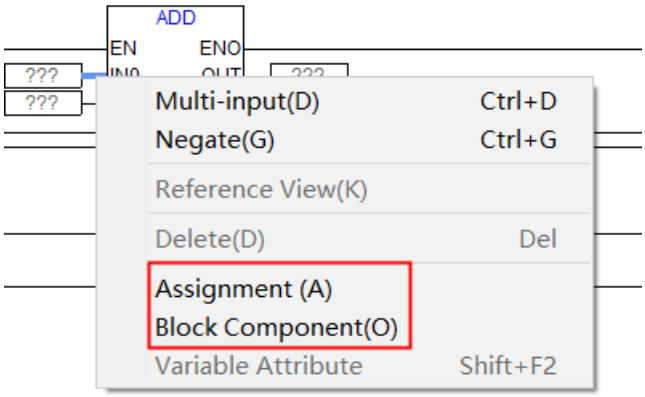
10.11.9.2.5 Reference message

Select Element

10.11.10 Cascade Block Element Pin

Cascade configuration means that the input pins of a block element can be connected to other block elements and assigned values. This reduces intermediate variables and makes the logic connection tight, structurally clear, and intuitively easy to understand.

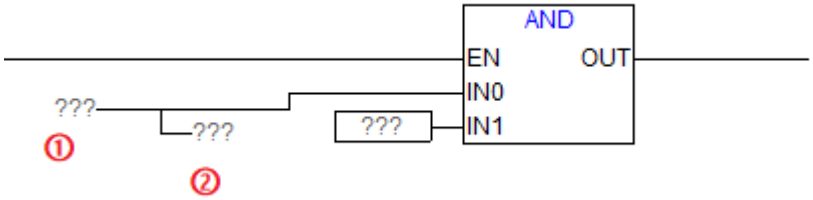
The right-click menu of block element input pin is shown in the figure below.



10.11.10.1 Cascade assignment element

10.11.10.1.1 Introduction

When the cascade object of the input pin of a block element is Assignment, two elements are added at the same time. The cascade result is shown in Positions ① and ② in the figure below:



Where Position ① is the input of the Assignment element, and Position ② is the output of the Assignment element; And Position ① assigns values to Position ② and the IN0 item of the AND block at the same time.

The Assignment element cannot be copied, pasted, or cut.

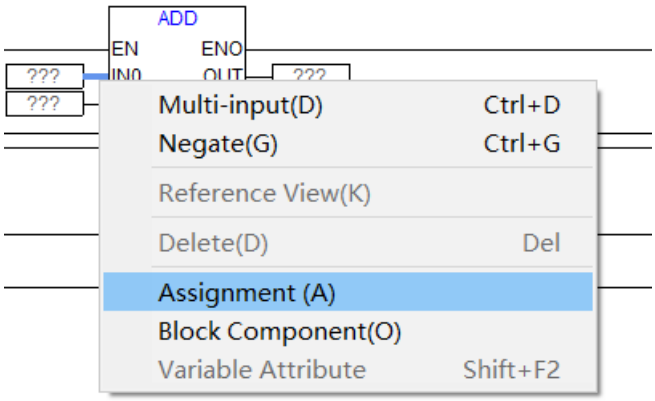
10.11.10.1.2 Requirement

There is a block element that needs to be cascaded.

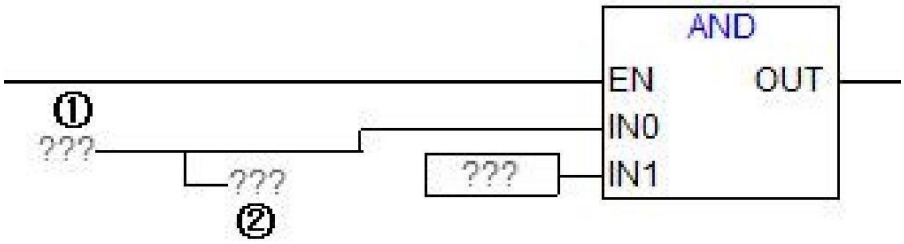
10.11.10.1.3 Steps

Follow the steps below to cascade an assignment element:

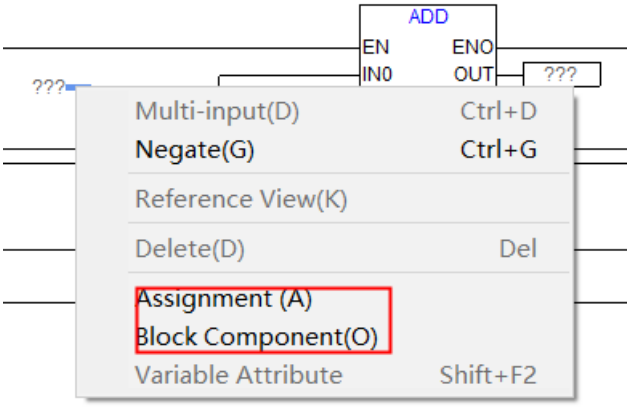
(1) Right-click one of the input pins of the block element. Select the Assign command, as shown in the figure below.



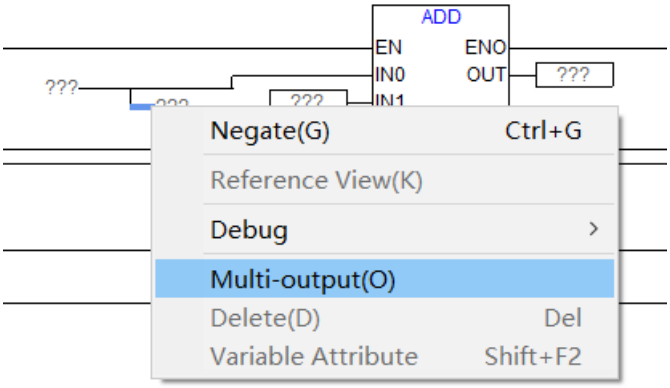
(2) The input pin is cascaded with two assignment elements

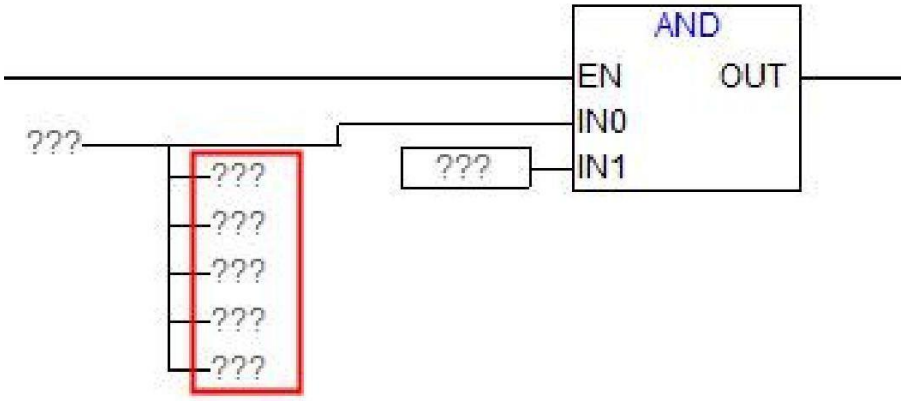


(3) The Assign or Block Element command in the right-click menu of Position ① allows users to expand Position ① to the next level. As shown in the figure below.



(4) The Multiple Outputs command in the right-click menu of Position ② allows users to add outputs to expand Position ② in parallel. As shown in the figure below.





10.11.10.2 Delete assignment element

10.11.10.2.1 Introduction

The Delete command of a node allows you to remove a cascaded Assignment element or an output. The scope of the deletion depends on the position of the selected node.

10.11.10.2.2 Steps

To delete a cascaded assignment element, follow the steps below:

(1) Select the cascaded node position.

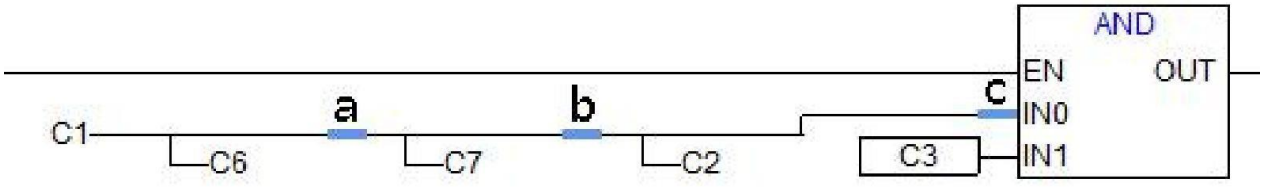
(2) Right-click the node. Click the Delete command. The assigned endpoints of the corresponding scope are deleted.

10.11.10.2.3 Example

Endpoints C1 and C6 can be deleted through Node (a) in the figure below;

Endpoints C1, C6, and C7 can be deleted through Node (b) in the figure below.

Endpoints C1, C6, C7, and C2 can be deleted through Node (c) in the figure below.



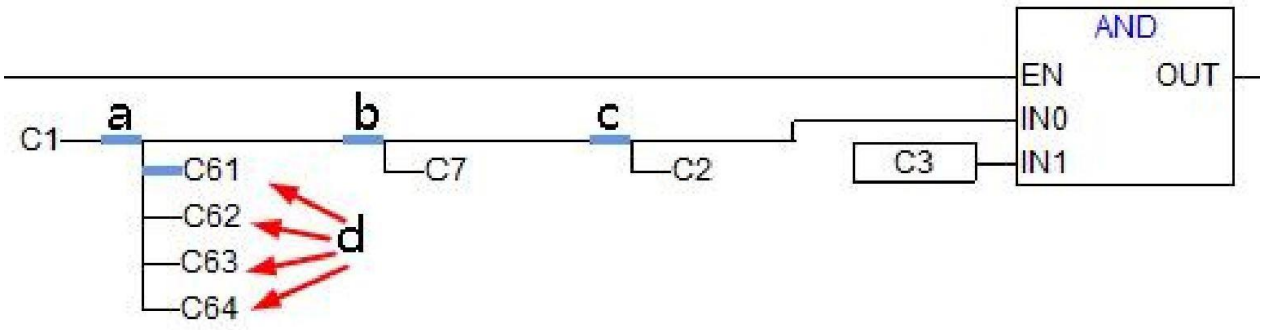
10.11.10.2.4 Delete multiple output nodes

Endpoints C61 - C64 can be deleted through Node (a) in the figure below.

Endpoint C7 can be deleted through Node (b) in the figure below.

Endpoint C2 can be deleted through Node (c) in the figure below.

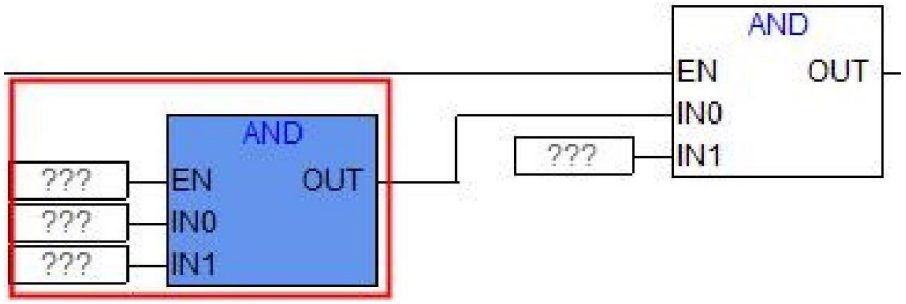
Endpoints C61, C62, C63, and C64 can be deleted through Node (d) in the figure below.



10.11.10.3 Cascade block element

10.11.10.3.1 Introduction

When the cascade object of the input pin of a block element is Block Element, an AND block is added by default. The cascade result is shown in the figure below.



When the Block Element is cascaded, the first output pin of the Block Element is connected by default. When users need to change the connected outputs, the output pins can be adjusted via the Pin (Show/Hide) command.

See the relevant contents of the Assignment element for the operation method and influence the scope of deleting block elements, which will not be repeated here.

10.11.10.3.2 Requirement

A block element has been added.

10.11.10.3.3 Steps

The operation method and scope of influence of Cascade block element refer to Cascade assignment element.

10.11.11 Rename Block Element

10.11.11.1 Introduction

The block element is added as an AND block by default. However, you can manually modify the block element name to the needed one. When the block element name is entered, the software automatically associates all the block element names that match the entered keyword for choice, making it easy to enter quickly.

10.11.11.2 Requirement

A block element has been created.

10.11.11.3 Steps

To rename the block element, follow the steps below:

(1) Double-click the block element name. The cursor is displayed at the block element name.

(2) Select the block element name. Enter the new name. All associated block elements are automatically displayed.

(3) Double-click the selected block element. The block element name is modified.

10.11.12 Add Jump Lable

10.11.12.1 Introduction

When the user program needs to jump to other program segments, the jump destination can be specified by the jump tag. The program jumps to the destination program segment for further execution.

Up to 32 characters can be entered for the jump tag. The character requirements are the same as the variable definition.

10.11.12.2 Requirement

The destination program segment has been configured.

10.11.12.3 Steps

Follow the steps below to add a jump codename:

(1) Right-click the destination program section.

(2) In the right-click menu, select the Lable command. The identification label and the placeholder ??? for the jump tag name are displayed at the top of the program segment.

(3) Click the placeholder ???. Enter the jump tag name.

(4) The jump tag name shall be entered at the jump destination.

10.11.12.4 Reference message

Jump

10.11.13 Jump

10.11.13.1 Introduction

The Jump command can be used to interrupt the sequential execution of a program, and to jump to the destination program segment for further execution. To identify the destination program segment, it is

necessary to add the Jump Codename. The jump codename name shall be specified at the jump destination, which is ??? by default.

The jumps are all at the end of the network section. If the logical operation result is TRUE, it jumps to the program segment with the specified codename. If the logical operation result is FALSE, the program is executed sequentially downwards.

You can execute the Jump command in the following ways:

- Menu Bar: [Insert] - [Jump];
- Programming area: Right-click the menu of the last input element or any output element in the current section > Jump;
- Toolbar:  .

10.11.13.2 Requirement

The coil has been selected

10.11.13.3 Steps

Follow the steps below to add a jump:

In the interrupted program section, right-click the coil.

In the right-click menu, select the Jump command. A jump arrow and a placeholder ??? for the jump tag name are displayed at the bottom of the coil.

Click the placeholder. Enter the jump tag name.

10.11.13.4 Reference message

Add Jump Lable

Select Element

Return

10.11.14 Return

10.11.14.1 Introduction

When calling a POU, users can use the Return command to return to the called POU when the conditions are met, instead of continuing to execute the called POU.

The return placeholder defaults to Return.

You can execute the Return command in the following ways:

- Menu Bar: [Insert] - [Return];
- Programming area: Right-click the last input element or any output element in the current section, and click Return in the menu that appears.
- Toolbar: 

.

10.11.14.2 Requirement

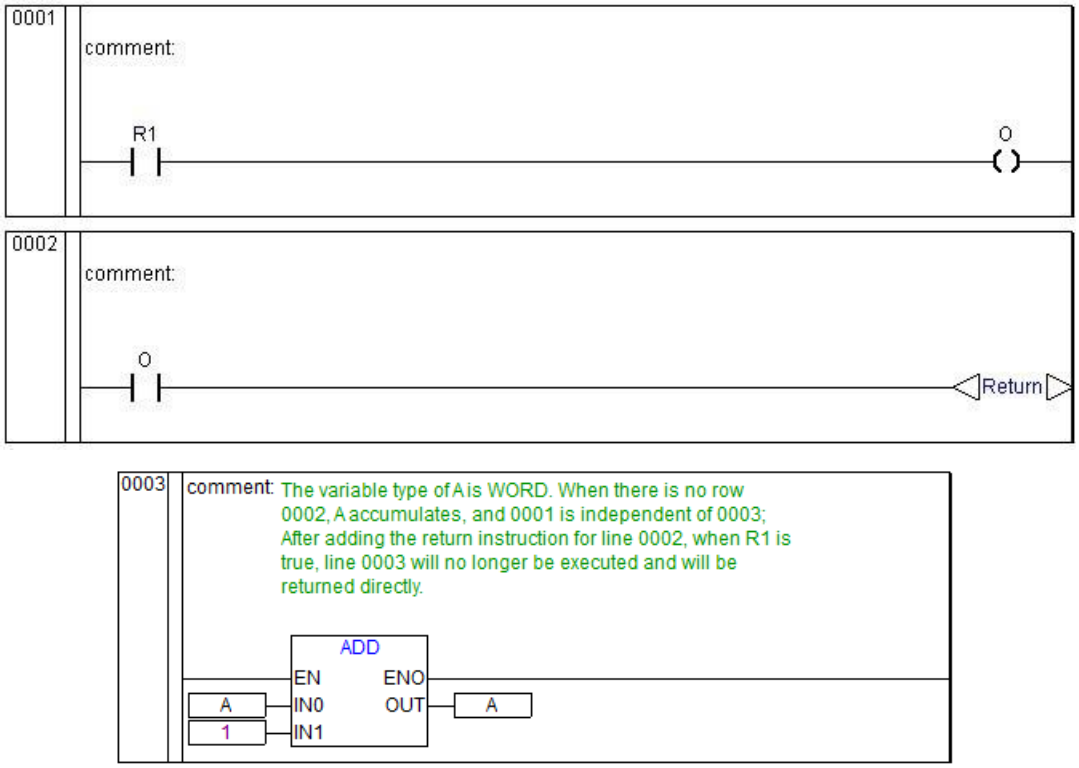
The section has been selected.

10.11.14.3 Steps

Follow the steps below to insert a Return element:

(1) Right-click the section.

(2) In the right-click menu, select the Return command. At the end of the section, insert the return character Return. As shown in Section 0002 in the figure below: When contact O of section 0002 is closed and turned on, jump out of the program, and return to the POU called it, without executing the program in section 0003 and below.



10.11.15 Negate

10.11.15.1 Introduction

The operation allows you to negate the pins of a contact, coil or block element.

Select the Invert command via:

- Menu Bar: [Insert] - [Negate];
- Programming area: Click negate in the right-click menu.
- Toolbar: ;
- Shortcut key: Ctrl + G.

10.11.15.2 Requirement

The element is selected.

10.11.15.3 Steps

To negate an element, follow the steps below:

(1) Right-click the element.

(2) In the right-click menu, select the negate command. The element is negated.

10.11.15.4 Reference message

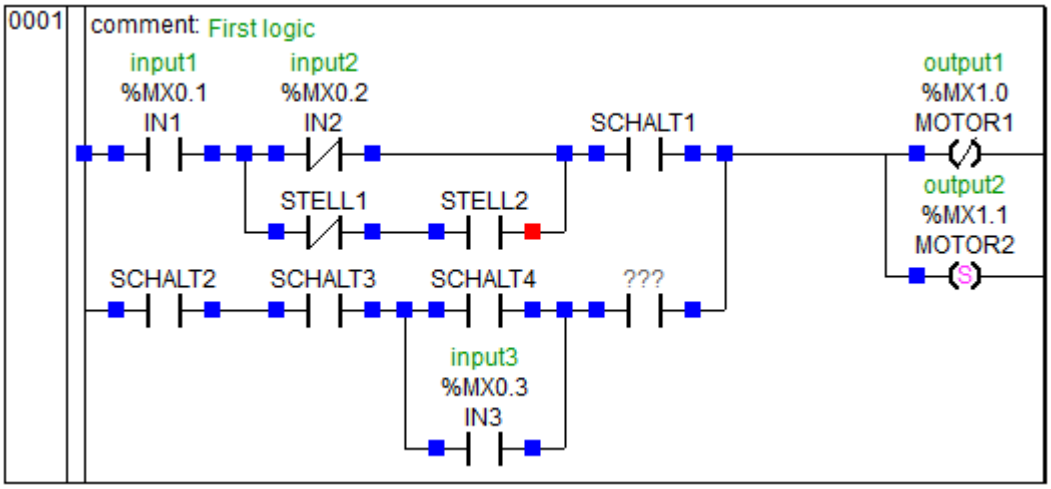
Select Element

10.11.16 Move

To move an element to another position, follow the steps below:

(1) Select the element. Hold down the left mouse button and drag it. At this time, a bright blue mark is displayed in front of and behind all the elements that can be moved (the coil only has a moving rectangle box on the left side), as shown in the figure.

(2) When the element is moved to the target position, the box closest to the mouse is highlighted in red. Release the mouse to move the element to this position.



10.11.17 Add Program Comment

10.11.17.1 Introduction

Use program comments to describe the program content of each network section. Program contents include the process description of the program section, the logical function description, or characteristics requiring the user's attention.

10.11.17.2 Requirement

The program section has been configured.

10.11.17.3 Steps

To add a program section comment, follow the steps below:

(1) Right-click the destination program section.

(2) In the right-click menu, select the Change Comment command. The mark comment is displayed at the top of the program section.

(3) Click Please add a comment here.... Enter the comment content.

10.11.18 Switch Comment Status

10.11.18.1 Overview

The annotation status section maintains the same display effect whether offline or online. When in annotation mode, the program section does not participate in syntax error checking, is not included in the compilation logic for itself and its sub-elements, and cannot be run online or perform online-related operations.

You can execute the Switch Comment Status command in the following ways:

- Program area: Right-click menu - 【Switch Comment Status】;
- Shortcut key: CTRL+SHIFT+O.

10.11.18.2 Requirement

Select the section you want to modify.

10.11.18.3 Steps

To change the annotation status, follow these steps:

(1) Right-click the target program section.

(2) In the right-click menu, select the "Switch Comment Status" command. The program section will then be in annotation status.

(3) To cancel the annotation status, select the "Switch Comment Status" command again.

10.11.19 Copy Element

10.11.19.1 Introduction

The section, contact, coil, block element, multiple elements or program segment in a program can be copied and pasted.

10.11.19.2 Requirement

There are available elements.

10.11.19.3 Steps

Follow the steps below to execute the Copy command:

Select the element to be copied. Execute the Copy via:

- Element selected area: Right-click menu - Copy;
- Menu Bar: [Edit] - [Copy];
- Toolbar:  ;
- Shortcut key: Ctrl + C.

- Note: If you copy a program segment or multiple elements via the right-click menu, press and hold Shift after selecting them until the Copy operation is executed.

10.11.19.4 Result

The program is copied to the clipboard, and can be pasted to the target position.

10.11.19.5 Reference message

Select Element

10.11.20 Paste Element

10.11.20.1 Introduction

Users can paste the copied and cut element to a certain position in the program.

- Paste section: Paste the copied section in front of the current section.
- Paste contact and block element: Paste the copied element in front of the current element.
- Paste coil: Paste the coil to the end of the section. If the current section already has a coil, connect that copied coil in parallel.
- Paste to the below or right: Contact and block elements can be pasted to the right of the currently selected element or in parallel to the current element.

You can execute the Paste command in the following ways:

- Programming area: Click Paste in the right-click menu.
- Menu bar: Choose [Edit] - [Paste].
- Toolbar:  ;
- Shortcut key: Ctrl + V.

10.11.20.2 Requirement

The element to be pasted has been copied or cut.

10.11.20.3 Paste

Follow the steps below to paste an element:

(1) Right-click the element.

(2) Select the Paste command in the menu that appears.

10.11.20.4 Paste to the right

Follow the steps below to paste an element to the right:

(1) Right-click the element.

(2) Select the Paste to Right command in the menu that appears.

10.11.20.5 Paste to the below

Follow the steps below to paste an element to the below:

(1) Right-click the element.

(2) Select the Paste to Bottom command in the menu that appears.

10.11.21 Delete Element

10.11.21.1 Requirement

There are available elements.

10.11.21.2 Steps

Follow the steps below to delete an element:

(1) Select the element to be deleted.

(2) You can execute the Delete command in the following ways:

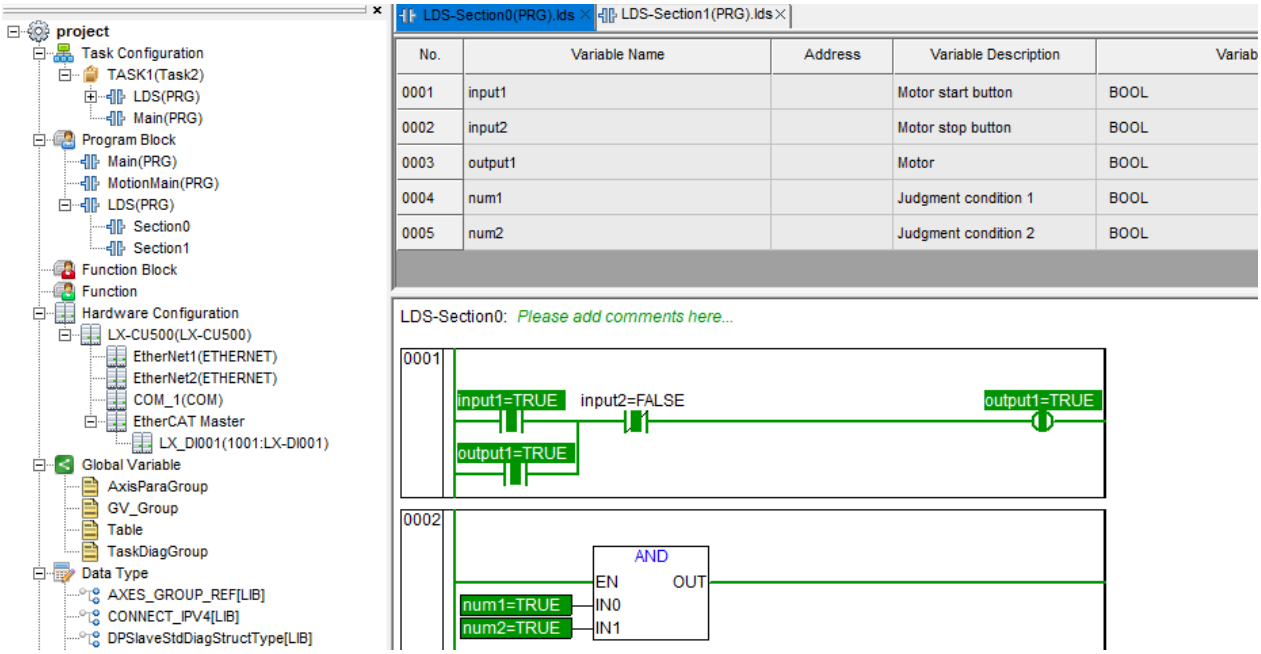
- Programming area: Click Delete in the right-click menu.
- Menu bar: Choose [Edit] - [Delete].
- Shortcut key: Delete.

- Note: When you delete a program segment or multiple elements, press and hold the Shift key after you select the segment or elements until the Delete operation is performed if you are using the right-click menu.

10.11.21.3 Reference message

Select Element

10.11.22 LDS Programming Example

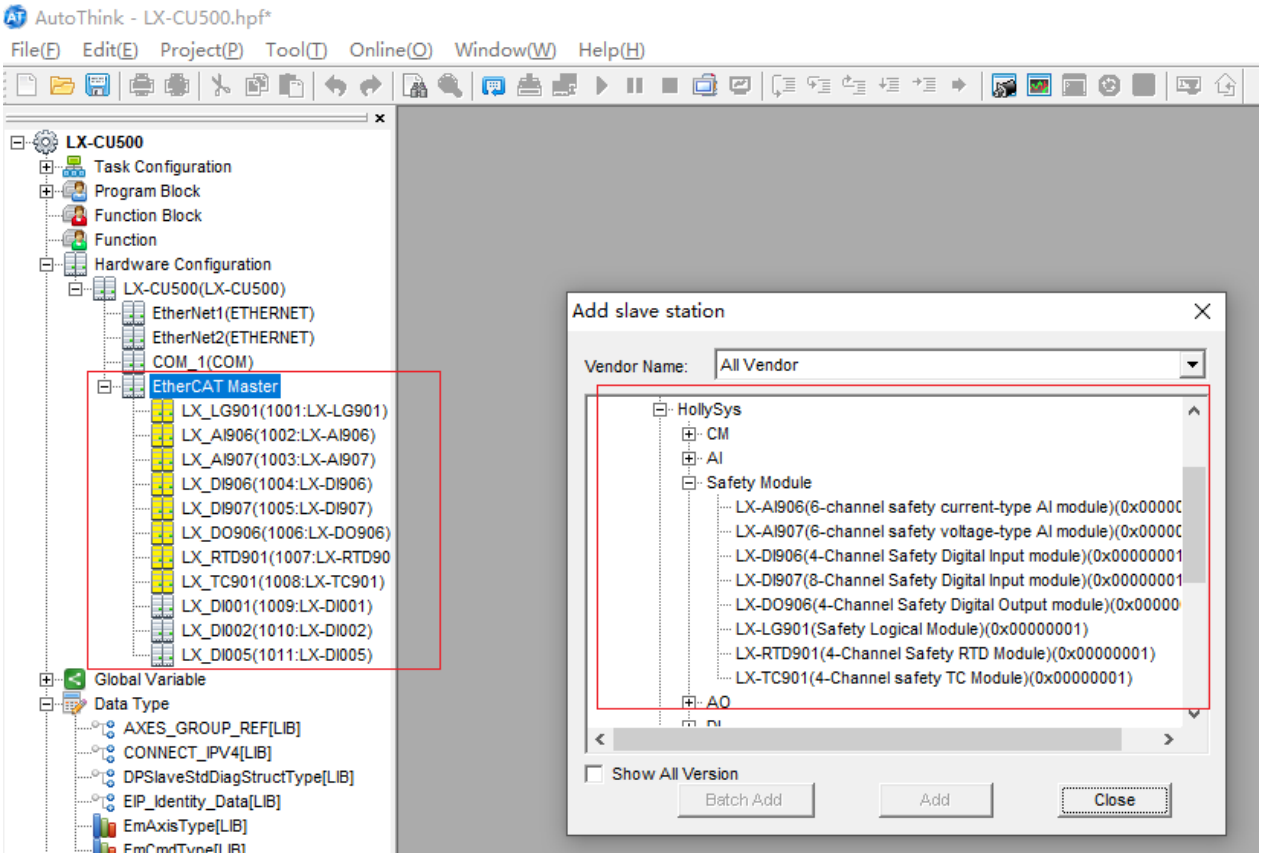


Chapter 11 LXS System Configuration

11.1 Non-secure Side Configuration

11.1.1 Add Safety Module

The safety module supports offline Add Slave Station and Scan Slave Station. When a slave station is added offline, the safety module is added under the Safety Module. Currently, the following modules are supported, as shown in the figure. The yellow ones are safety modules, and the non-yellow ones are ordinary modules.

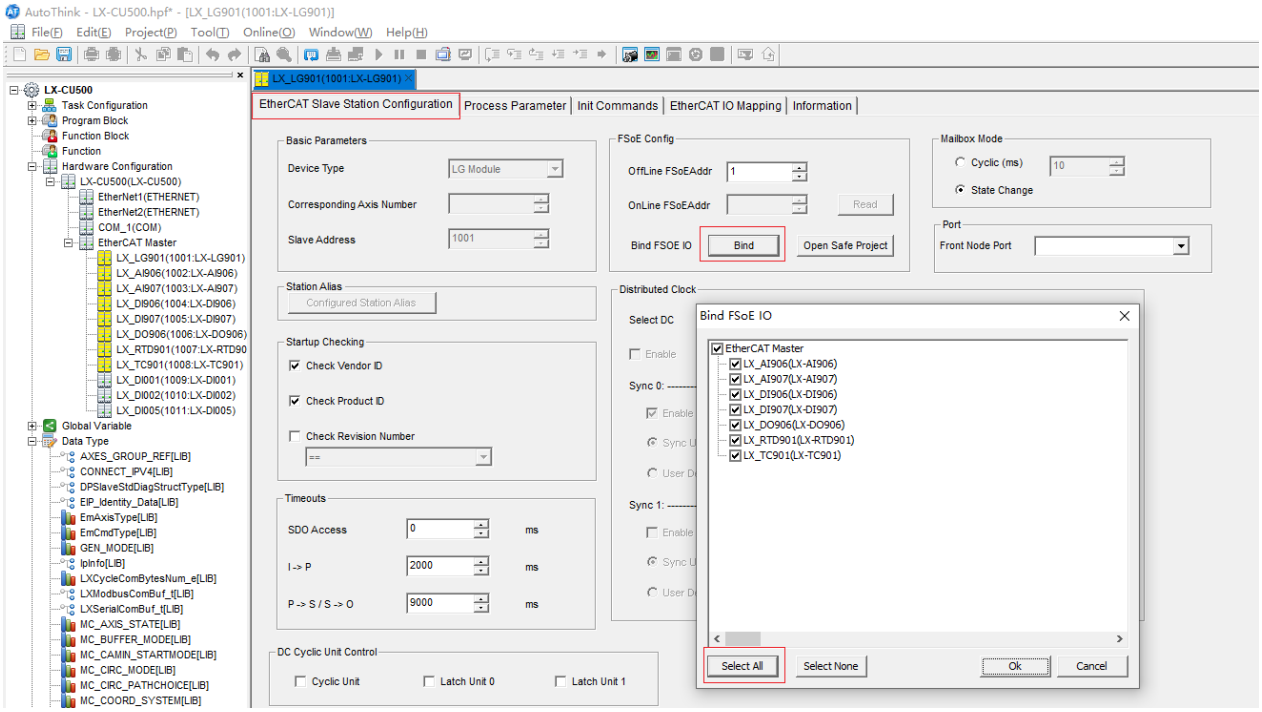


Add Safety Module Slave Station

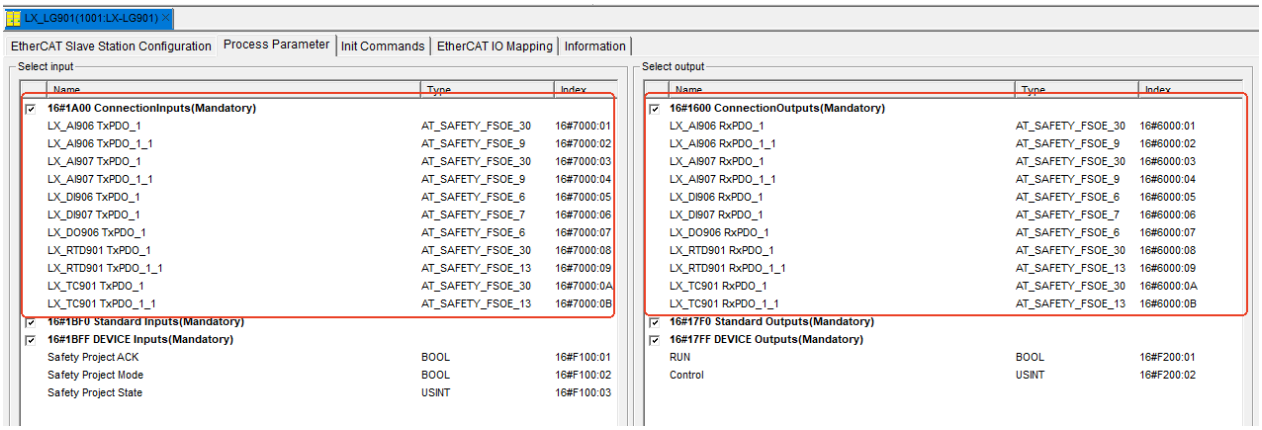
11.1.2 Bind FSOE IO Device

Double-click the LX-LG901 module to open the EtherCAT Slave Configuration window. In the FSOE Configuration interface, bind the safety module, as shown in Figure 2, EtherCAT Slave Configuration

Interface. At this time, the PDO of the safety module is automatically added in the Process Parameters window, as shown in the figure.



(a)

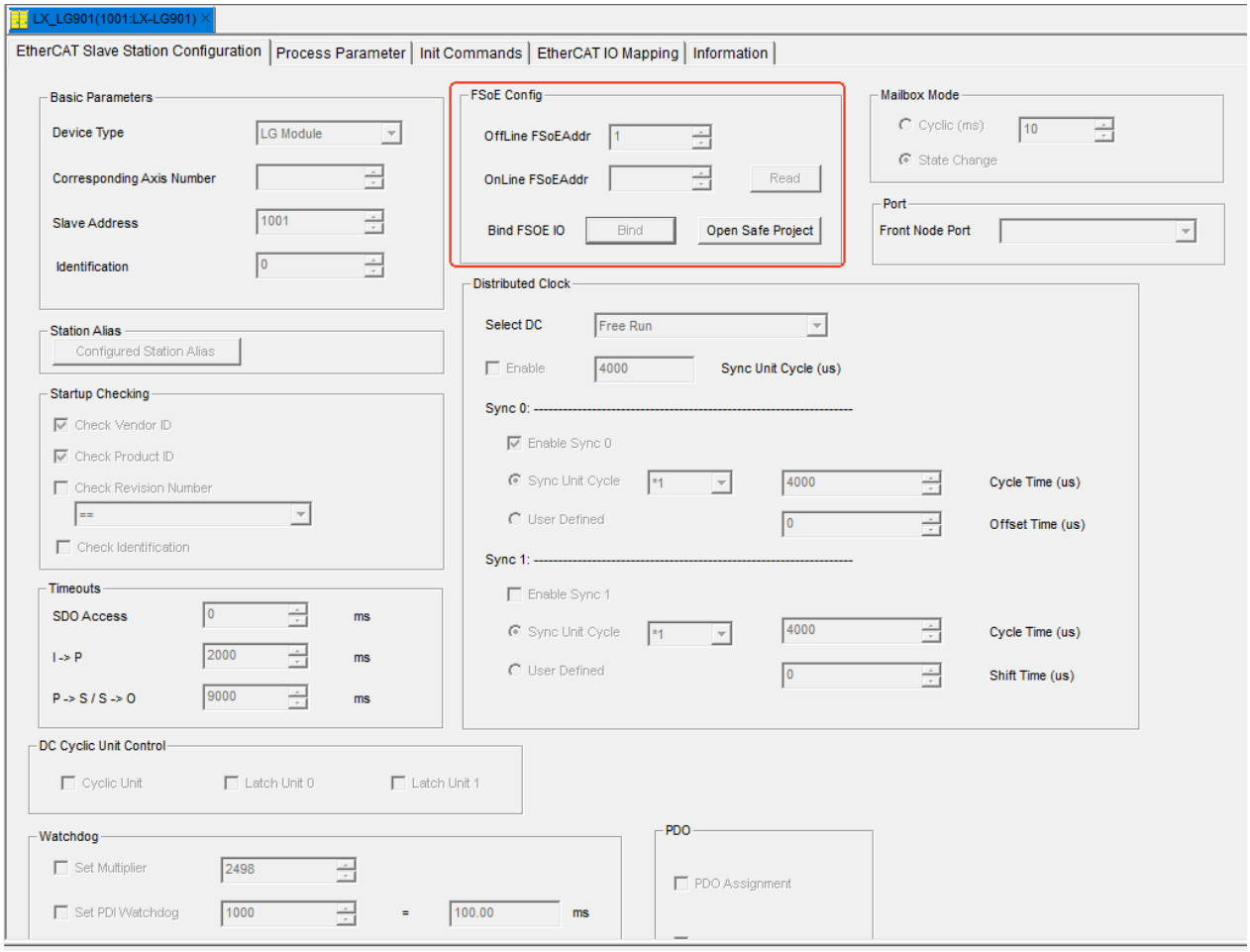


(b)

Process Parameters Interface

11.1.3 FSOE Address Configuration

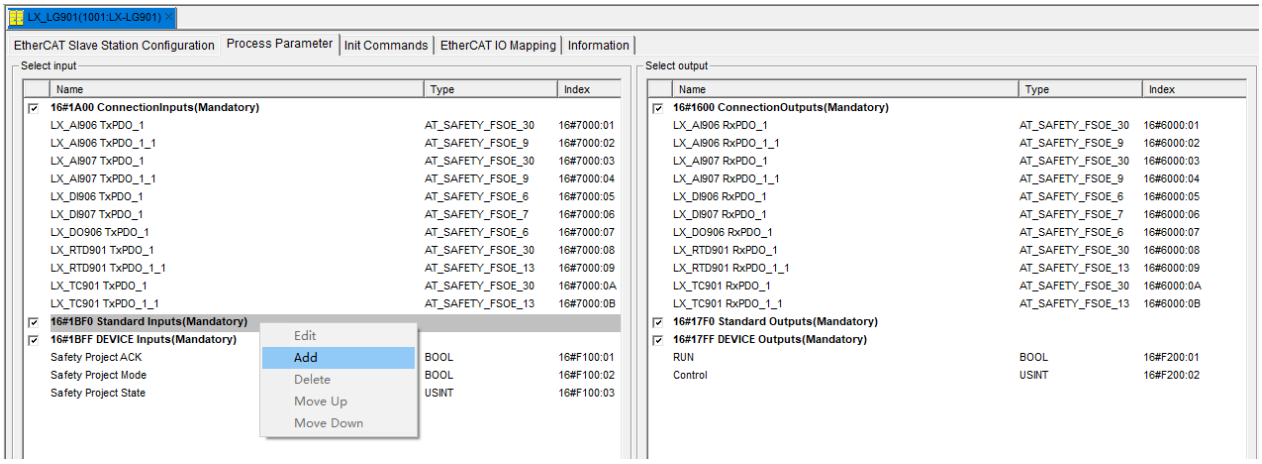
In the EtherCAT Slave Configuration interface, the FSOE soft dialing code is supported. Users can configure the FSOE address offline and read it online, as shown in the figure.



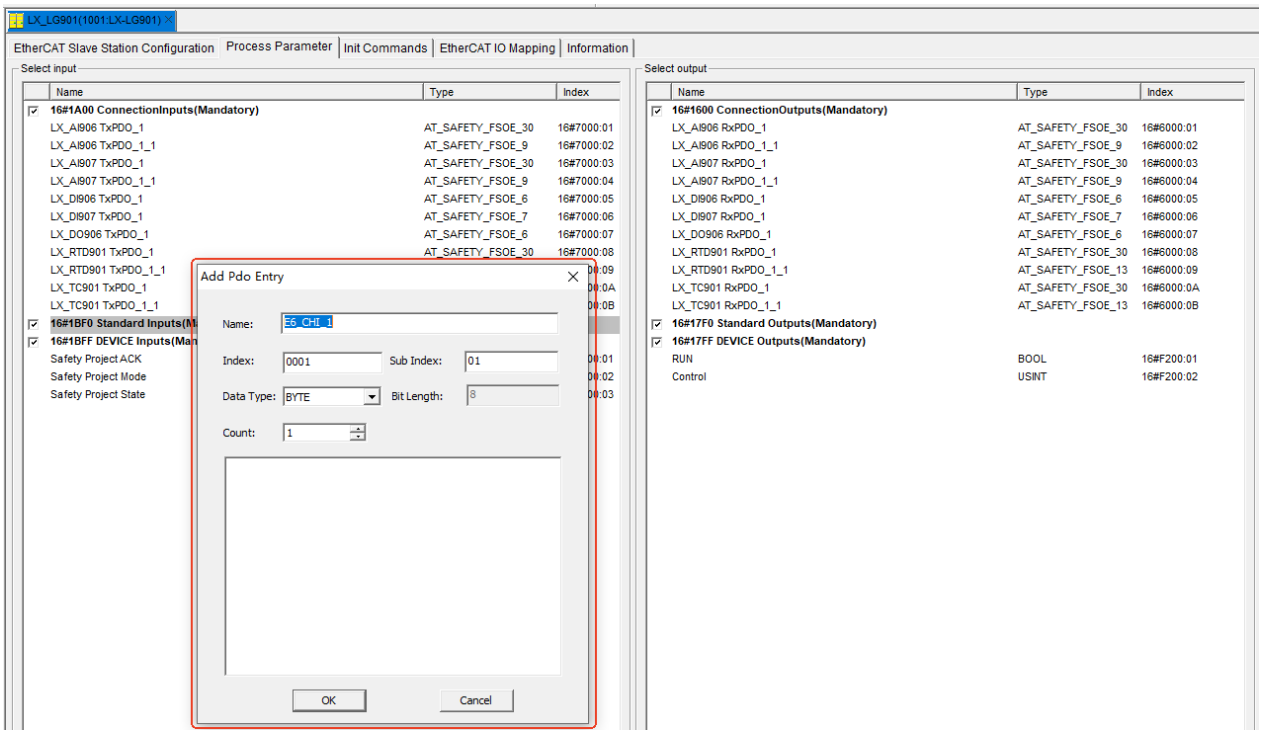
FSOE Address Configuration

11.1.4 Secure and Non-secure Data Interaction

When the non-secure side interacts with the secure side, it is necessary to add a parameter to the Standard Inputs PDO in the Process Parameters of the non-secure side. The maximum data interaction between the secure and non-secure sides is 512 bytes.



(a)



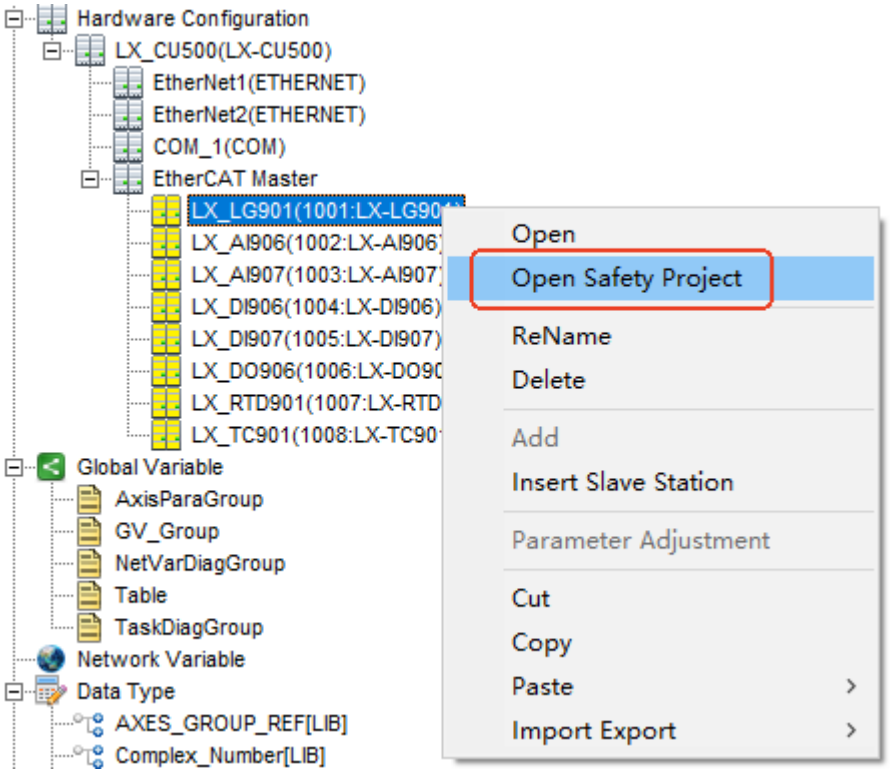
(b)

PDO Parameter Configuration

11.2 Secure-side Configuration

11.2.1 Open Secure-side Project

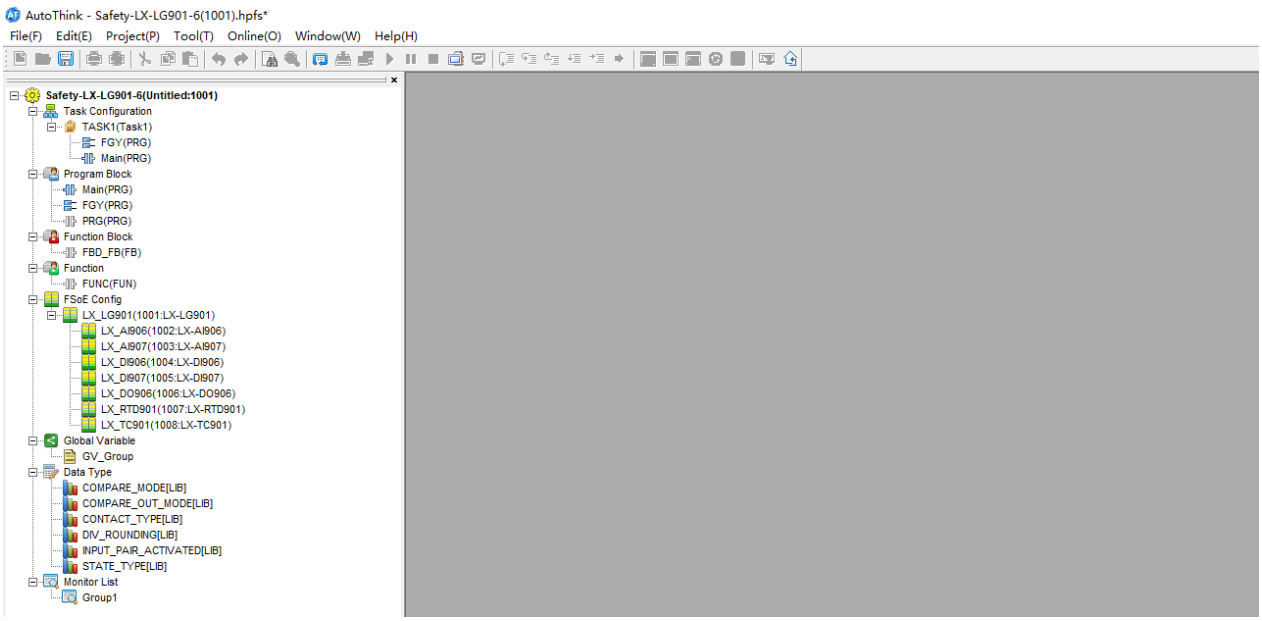
Select the safety logic module. Right-click it and click Open Safety Project to open the security-side project.



Open Safety Project

11.2.2 Secure-side Configuration

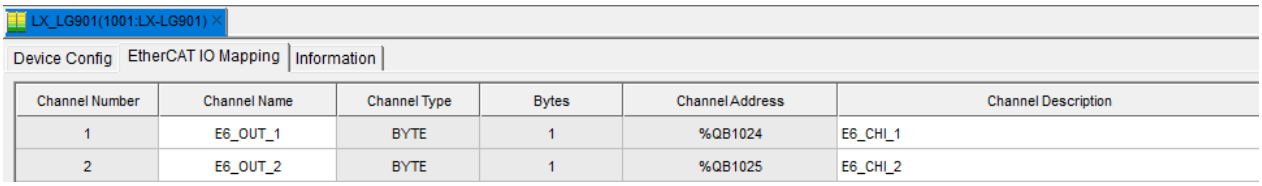
If a safety IO device is bound to the non-secure side, the safety IO module is automatically added to the configuration tree on the secure side, as shown in the figure.



Secure-side Configuration

11.2.3 Secure and Non-secure Data Interaction

When PDO parameters are added to the non-secure side, channel parameters are automatically added to the secure side.



Channel Number	Channel Name	Channel Type	Bytes	Channel Address	Channel Description
1	E6_OUT_1	BYTE	1	%QB1024	E6_CHI_1
2	E6_OUT_2	BYTE	1	%QB1025	E6_CHI_2

Secure-side Channel Parameters

11.3 Multiple LX-LG901 Configurations

AutoThink supports the configuration of multiple LX-LG901 functional safety logic modules, which includes:

- Mounting multiple LX-LG901 functional safety logic modules under one controller (such as: LX-CU500). The LX-LG901 modules can communicate with each other through custom connections, acting either as masters or slaves during communication.
- Mounting LX-LG901 functional safety logic modules under multiple controllers (such as: LX-CU500) respectively. The LX-LG901 modules can communicate with each other via network variables. A single controller can establish multiple network variables, which can both receive and send data.

11.3.1 Single Controller with LX-LG901

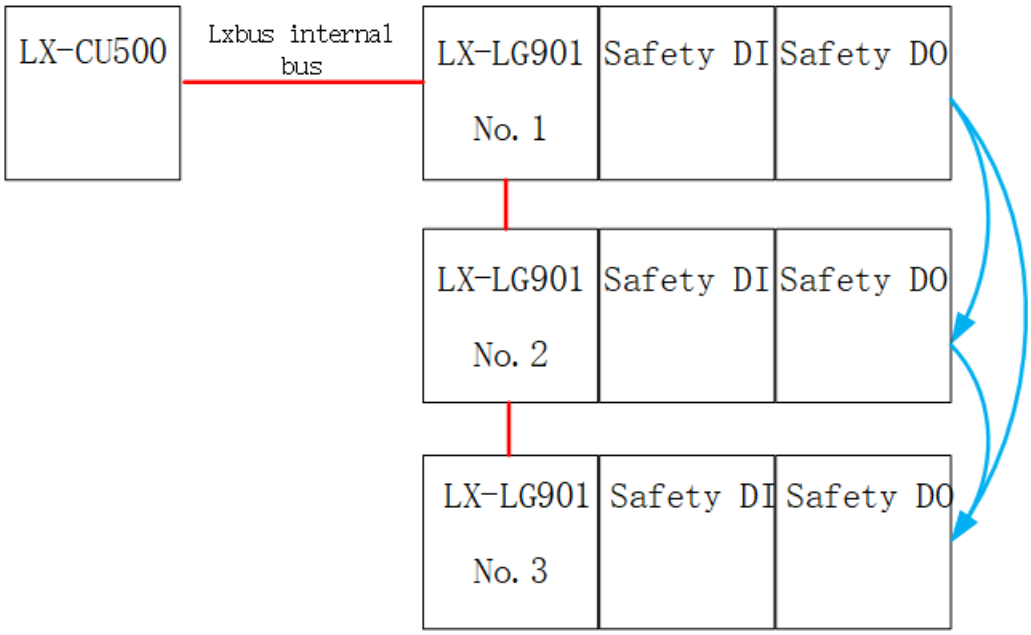
11.3.1.1 Overview

When multiple safety logic modules are connected to a single controller and communicate in an FSoE (Functional Safety over EtherCAT) master/slave relationship:

- As an FSoE Slave: The safety logic module acting as an FSoE slave needs to add a custom connection in the safety-side programming software and configure the safety data under this custom connection.
- As an FSoE Master: The safety logic module acting as an FSoE master needs to bind to this custom connection to enable normal master/slave communication.

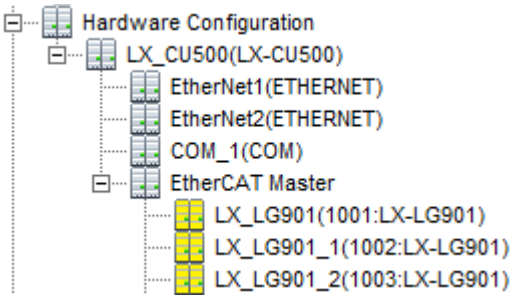
- Note: The master/slave roles and communication data of the LX-LG901 during communication are entirely determined by the configured custom connections. Therefore, the same LX-LG901 can act as both a master and a slave.

The following example provides a detailed explanation of the configuration process. As shown in the figure below, No. 2 LX-LG901 acts as a slave and communicates with No. 1. Simultaneously, No. 2 LX-LG901 acts as a master and communicates with No. 3. No. 3 LX-LG901 acts as a slave and communicates with both No. 1 and No. 2.



11.3.1.2 Adding Multiple LX-LG901 Modules

In the navigation tree on the left side of AutoThink, right-click on "Hardware Configuration" — "LX-CU500" — "EtherCAT Master," and select "Add Slave Station." In the "Add Slave" dialog box that appears, choose LX-LG901 and add three LX-LG901 modules.



- Note: A single controller can have a maximum of 8 LX-LG901 modules added.

11.3.1.3 Establishing Communication Between No. 1 and No. 2 LX-LG901

11.3.1.3.1 Configuring the Slave Custom Connection for No. 2 LX-LG901

(1) Right-click on the 2nd LX-LG901 and select "Open Safety Project."

(2) In the safety-side AutoThink interface that pops up, right-click on "FSoE Configuration" — "LX_LG901" in the left navigation tree, and select "Add." In the "Add" dialog box that appears, add one custom connection.

- Note: A single LX-LG901 can have a maximum of 8 custom connections.

(3) Double-click the custom connection added in the previous step to open the configuration window.

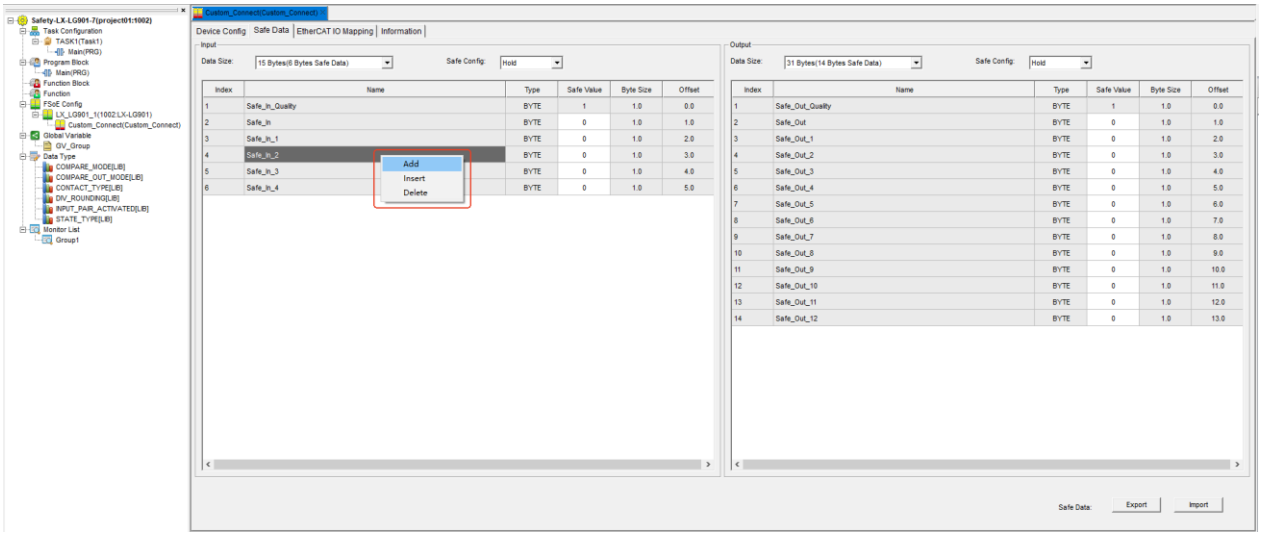
□ Click on the "Device Configuration" tab and configure the following parameters.

Base Info	
FSoEAddr	Represents the FSoE address of the custom connection. The value range is 1 to 65535. During configuration, it must not conflict with any existing FSoE addresses in the entire project.
Connection Setting	
Conn-ID	Configurable when the master/slave mode is set to "FSoE Master." It represents the ID of the custom connection. The value range is 1 to 65535. The system automatically assigns a unique and valid connection ID, so no modification is required. This parameter is not configurable when the Master/Slave mode is set to "FSoE Slave."
Mode	Since this LX-LG901 communicates with other LX-LG901 modules as a slave through the current custom connection, select "FSoE Slave." At this point, the Conn-ID and watchdog are not editable.
Watchdog(ms)	Represents the watchdog time for the connection. The default value is 300, and the value range is 1 to 65535. This parameter is not configurable when the Master/Slave mode is set to "FSoE Slave." Indicates that the FSoE message reception interval at the slave exceeds the watchdog timeout time.
Safe Param	
Length(Byte)	When the master/slave mode is set to "FSoE Master," configure the safety parameters in hexadecimal format. The maximum supported configuration size is 128 bytes. When the master/slave mode is set to "FSoE Slave," configure the length of the safety parameters. The value range is 1 to 128.

□ Configure the following parameters in the input/output areas respectively.

Parameter	Instruction
DataSize	Indicates the Length of Safety Data, choose based on the actual situation. Example: If the input data is 6 bytes, select “15 Bytes (6 Bytes Safe Data)” from the data length dropdown in the input area.
Safe Config	Actions Taken on Input/Output When the Module is Abnormal: Hold: The input/output retains the value before the abnormality. Clear: The input/output values are reset to zero. Safe Value: The input/output uses a safe value.

After selecting the data length, AutoThink automatically fills all bytes, with the default type being Byte. You can delete, insert, or add data according to the actual situation (the first byte is the safety quality parameter and cannot be modified, deleted, or added).



□ Click on the "EtherCAT IO Mapping" Tab, the input/output data configured in the "Safety Data"

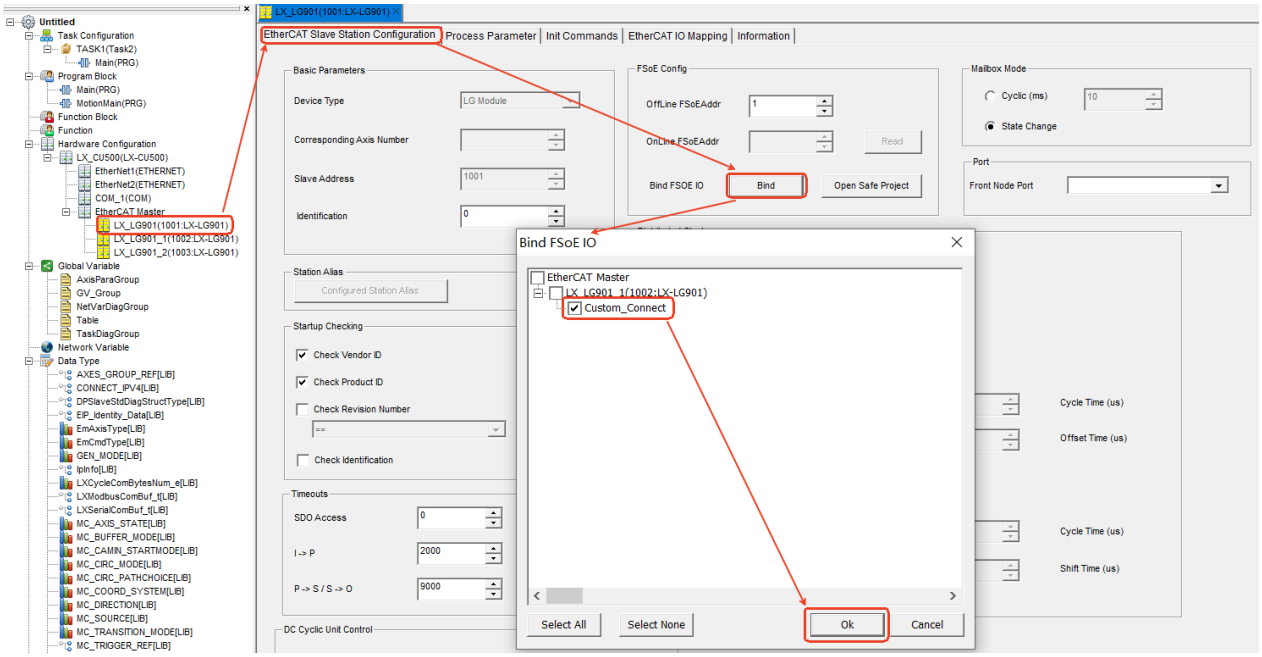
tab is automatically added to this tab. Parameters with a white background can be modified according to actual needs for easier identification and management.

11.3.1.3.2 Establish Communication Between LX-LG901 No. 1 and No. 2

(1) In the AutoThink software interface on the non-safe side, double-click on LX-LG901 No. 1 to open the configuration interface.

(2) Click on the "EtherCAT Slave Configuration" tab.

(3) In the "FSoE Config" — "Bind FSOE IO" section, click "Bind". In the "Bind FSoE IO" dialog box that appears, check the slave custom connection created for configuring LX-LG901 No. 2 (Slave Custom Connection 2), and then click "OK."



- Note: Since an LX-LG901 supports a maximum of 8 custom connections, if there are already 8 custom connections under this LX-LG901, it will not be possible to bind another LX-LG901 slave. However, binding a safety I/O module is not subject to this limitation.

(4) Right-click on LX-LG901 No. 1 and select "Open Safety Project." In the safety-side AutoThink interface that appears, you will see that AutoThink has automatically created a custom connection.

(5) Double-click the connection to open the configuration interface.

□ Click on the "Device Configuration" tab and configure the following parameters.

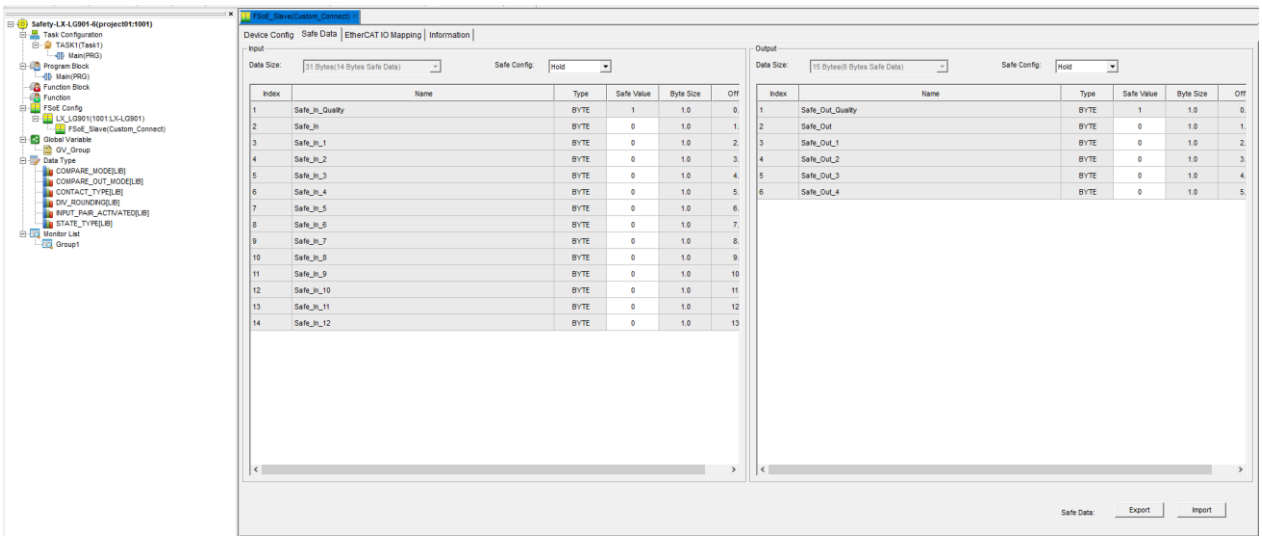
Base Info	
FSoEAddr	epresents the FSoE address of the custom connection. The value range is 1 to 65535. During configuration, it must not conflict with any existing FSoE addresses in the entire project.
Connection Setting	

Conn-ID	The system automatically assigns a unique and valid connection ID, so no modification is required. This parameter is not configurable when the Master/Slave mode is set to "FSoE Slave."
Mode	Default: "FSoE Master" (not modifiable).
Watchdog(ms)	Represents the watchdog time for the connection. The default value is 300, and the value range is 1 to 65535. This parameter is not configurable when the Master/Slave mode is set to "FSoE Slave." Watchdog timeout indicates that the FSoE message reception interval at the slave station is greater than the watchdog's timeout time.
Safe Param	
Param(Hex)	Configure the safety parameters in hexadecimal format. The maximum supported configuration size is 128 bytes.

- Click on the "Safe Data" tab. AutoThink automatically adds master station data based on the data situation of the slave station (i.e., the input/output data configured in the slave custom connection (2) for LX-LG901 No. 2). No modifications are required.

- Note:

- The output of the slave station is the input of the master station; the input of the slave station is the output of the master station. The length and type of data correspond and match.
- After the input/output data of the slave station is updated datasize, the master station will automatically synchronize and update.



- The "EtherCAT IO Mapping" tab displays the mapping relationship between the safe data and the I/O areas automatically generated by AutoThink. No modifications are required.

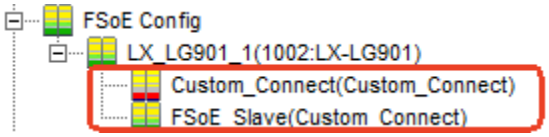
11.3.1.4 Establish Communication Between LX-LG901 No. 2 and LX-LG901 No. 3

11.3.1.4.1 Configure the Slave Custom Connection for LX-LG901 No. 3

Refer to the configuration of the slave custom connection for LX-LG901 No. 2 to add one slave custom connection for LX-LG901 No. 3.

11.3.1.4.2 Establish Communication Between LX-LG901 No. 2 and LX-LG901 No. 3

Refer to the establish communication between LX-LG901 No. 1 and LX-LG901 No. 2 to bind the slave custom connection for LX-LG901 No. 3 to LX-LG901 No. 2. After binding, LX-LG901 No. 2 will have two custom connections: one manually created slave custom connection and one master custom connection automatically created by AutoThink.

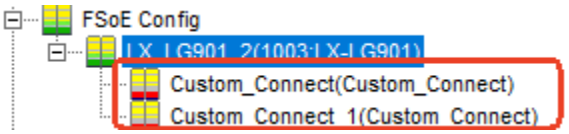


- FSoE Slave (Slave Type) connection icon is red. FSoE Master (Master Type) connection icon is green.

11.3.1.5 Establish Communication Between LX-LG901 No. 1 and LX-LG901 No. 3

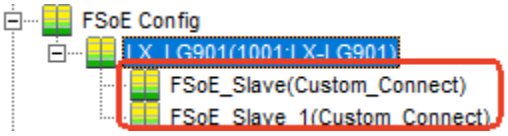
11.3.1.5.1 Configure the Slave Custom Connection for LX-LG901 No. 3

Refer to the configuration of the slave custom connection for LX-LG901 No. 2 to add one slave custom connection for LX-LG901 No. 3. After adding, LX-LG901 No. 3 will have two custom connections: one for communication with LX-LG901 No. 1 and one for communication with LX-LG901 No. 2.



11.3.1.5.2 Establish Communication Between LX-LG901 No. 1 and LX-LG901 No. 3

Refer to the establish communication between LX-LG901 No. 1 and LX-LG901 No. 2 to bind the slave custom connection for LX-LG901 No. 3 to LX-LG901 No. 1. After binding, LX-LG901 No. 1 will have two master custom connections, both automatically created by AutoThink: one for communication with LX-LG901 No. 2 and one for communication with LX-LG901 No. 3.



11.3.2 Multiple Controllers with LX-LG901

11.3.2.1 Overview

When safety logic modules between multiple controllers communicate in an FSoE master/slave relationship:

- As an FSoE Slave: The safety logic module that acts as an FSoE slave needs to add a slave custom connection in the safety-side programming software and configure the safety data under this custom connection. Then, create network variables for the controller to which the safety logic module belongs, and use these network variables as the sender to transmit data, while it is also necessary to establish a receiver network variable to accept the data originating from the master station.
- As an FSoE Master: The safety logic module that acts as an FSoE master needs to add a master custom connection in the safety-side programming software and import the safety data configuration information from the slave under this custom connection. Then, create network variables for the controller to which the safety logic module belongs, and use these network variables as the receiver to receive data, it is essential to create a sender network variable to transmit data to the master station.

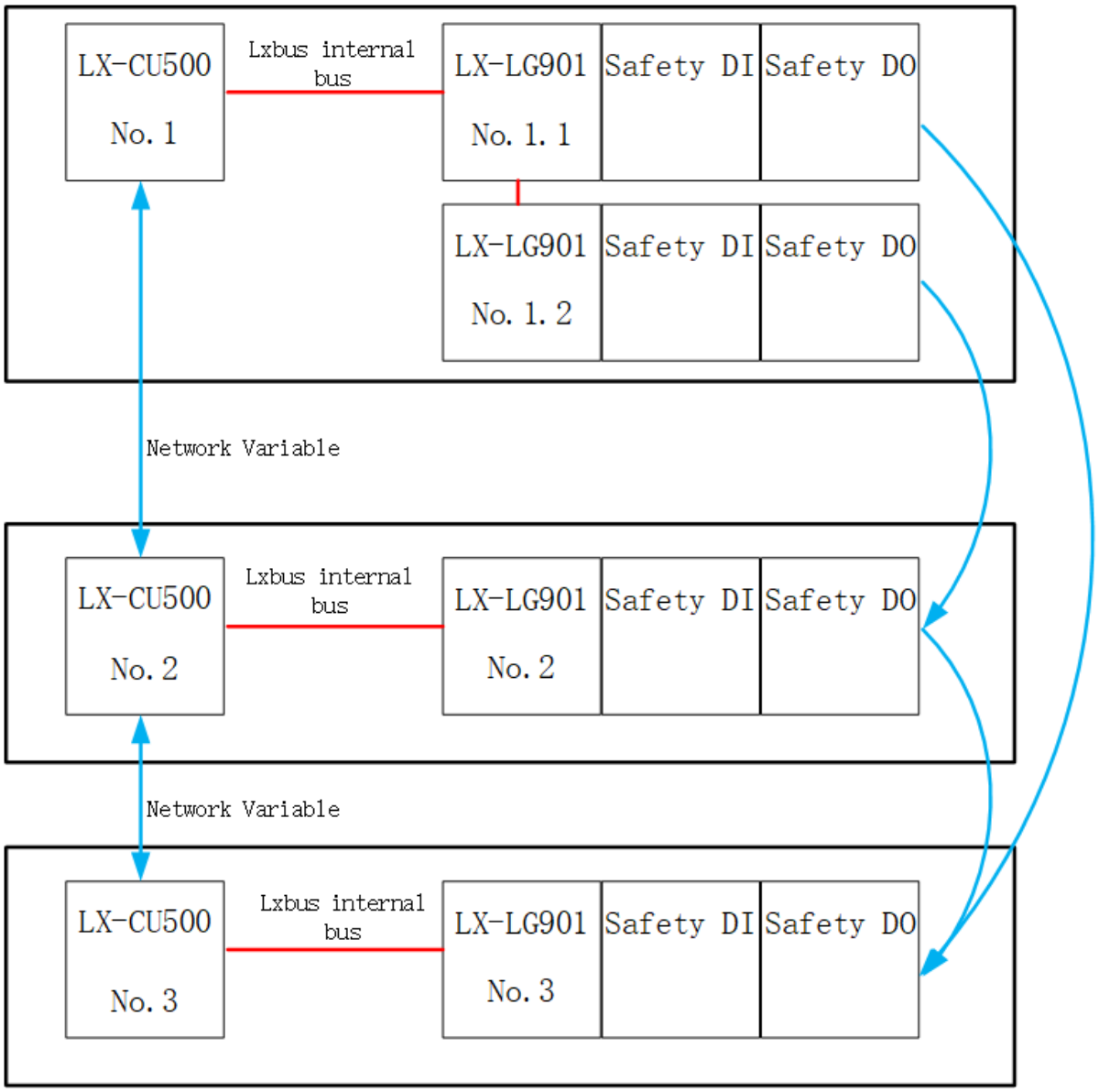
- The master/slave roles and communication data of the LX-LG901 during communication are entirely determined by the configured custom connections. Therefore, the same LX-LG901 can act as both a master and a slave.
- When the master/slave stations of FSoE communicate, it is necessary to configure both the sending network variable group and the receiving network variable group.

Detailed Configuration Process:The following diagram illustrates the configuration process, where Controller 1 has two LX-LG901 modules, while Controllers 2 and 3 each have one LX-LG901 module:

Controller 1: 1.1 LX-LG901 acts as the master station and communicates with 3 LX-LG901. 1.2 LX-LG901 acts as the master station and communicates with 2 LX-LG901.

Controller 2: 2 LX-LG901 acts as a slave and communicates with 1.2 LX-LG901, and also acts as a master to communicate with 3.

Controller 3: 3 LX-LG901 acts as a slave and communicates with both 1.1 and 2.



- Note: One AutoThink interface only supports creating one project, and one project only supports configuring and programming one controller. Since this scenario requires operations on three controllers, you need to open three AutoThink interfaces. In the following operations, they are distinguished by numbers 1/2/3. When performing actual operations, please be cautious to avoid confusion.

11.3.2.2 Establish Communication Between LX-LG901 No. 1.1 and LX-LG901 No. 3

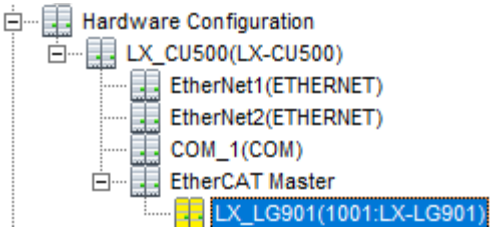
When communicating between LX-LG901 No. 1.1 and LX-LG901 No. 3, data exchange must be achieved through network variables of the controller. Therefore, the example for establishing communication follows the logical sequence below: Controller No. 1 sends network variables → Controller No. 3 receives network variables → Controller No. 3 returns network variables → Controller No. 1 receives network variables.

11.3.2.2.1 Creating Network Variables Sent by Controller No. 1

11.3.2.2.1.1 Add LX-LG901 No. 1.1

(1) In the navigation tree on the left side of Controller No. 1's AutoThink, right-click “Hardware Configuration” — “LX-CU500” — “EtherCAT Master,” and select “Add Slave Station” .

(2) In the “Add Slave Station” dialog box that appears, select LX-LG901 and add one LX-LG901 module, which will be designated as LX-LG901 No. 1.1.

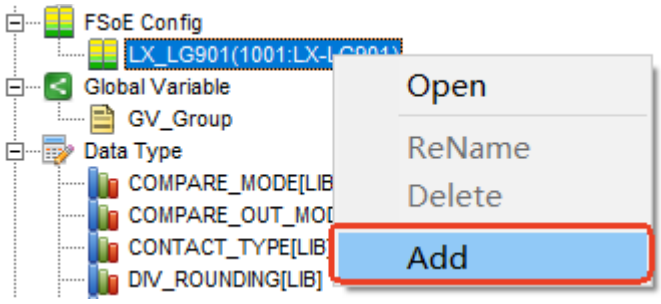


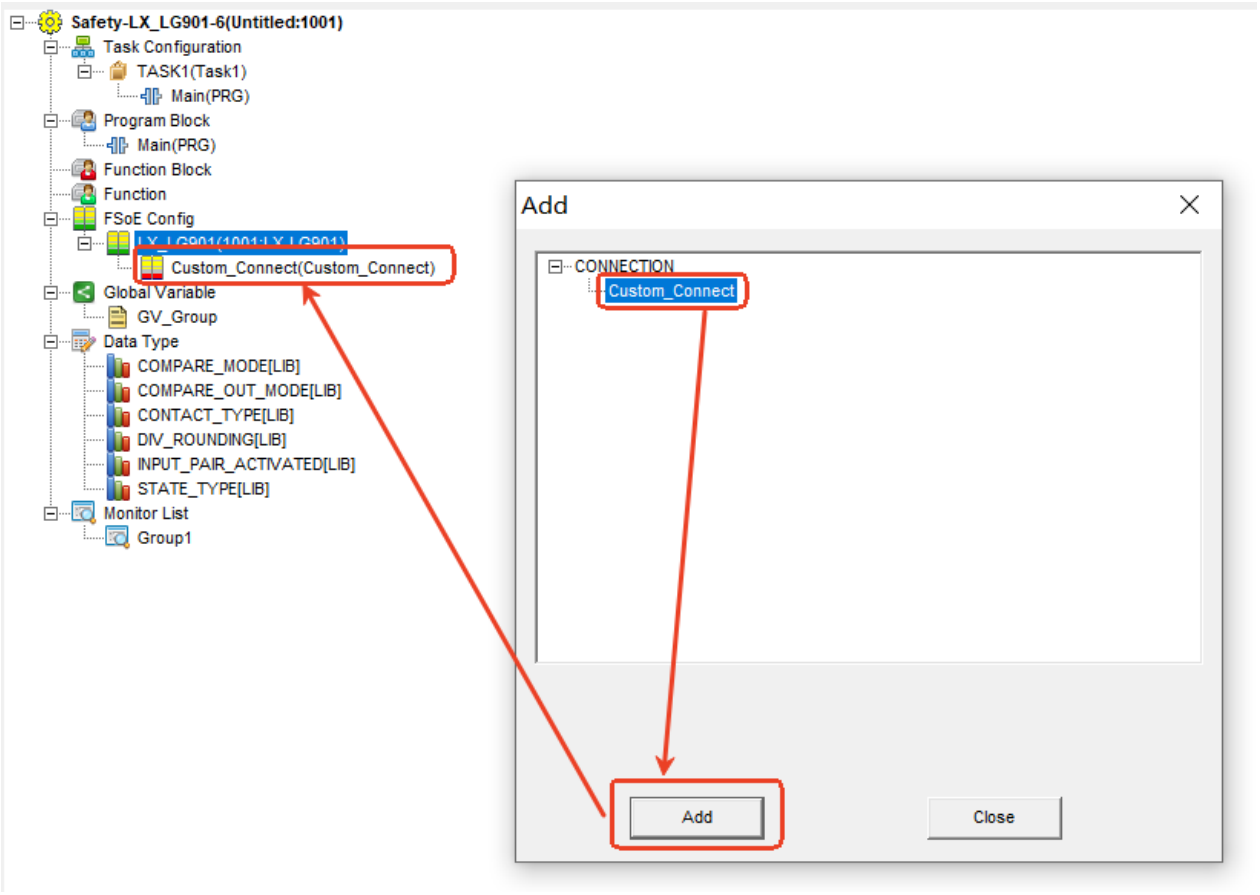
- Note: A single controller can have a maximum of 8 LX-LG901 modules added.

11.3.2.2.1.2 Configuring the Master Custom Connection for LX-LG901 No. 1.1

(1) Right-click on LX-LG901 No. 1.1 and select "Open Safety Project".

(2) In the safety-side AutoThink interface that pops up, right-click on "FSoE Configuration" — "LX_LG901" in the left navigation tree, and select "Add." In the "Add" dialog box that appears, add one custom connection.





- Note: An LX-LG901 can have a maximum of 8 custom connections.

(3) Double-click the custom connection added in the previous step to open the configuration window.

□ Click the "Device Configuration" tab and configure the following parameters.

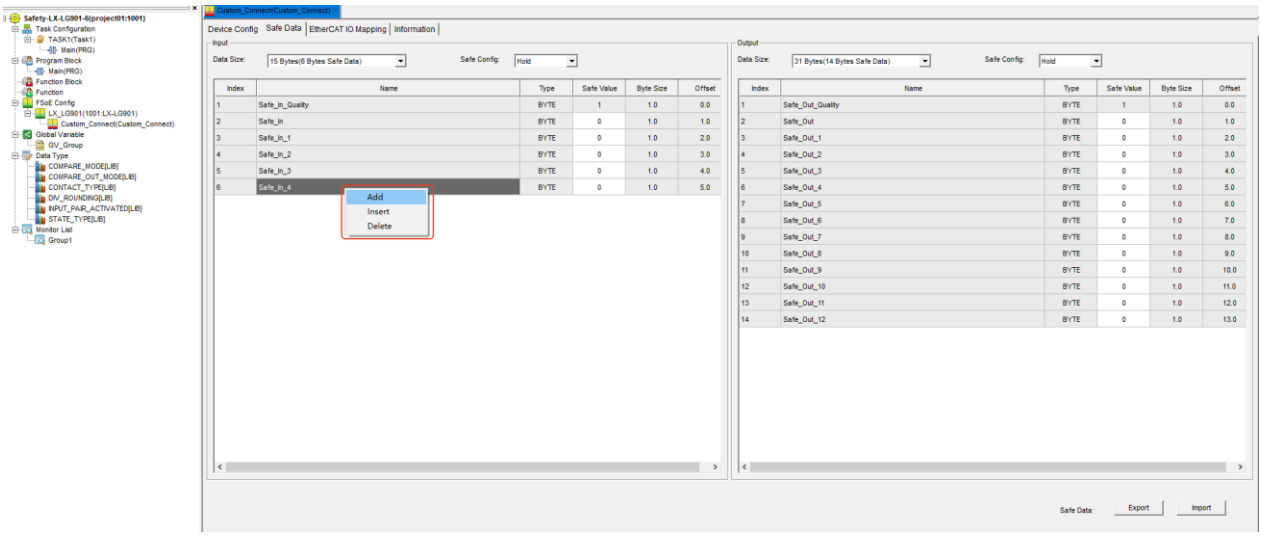
Base Info	
FSoEAddr	Represents the FSoE address of the custom connection. The value range is 1 to 65535. During configuration, it must not conflict with any existing FSoE addresses in the entire project.
Connection Setting	
Conn-ID	Configurable when the master/slave mode is set to "FSoE Master." It represents the ID of the custom connection. The value range is 1 to 65535. The system automatically assigns a unique and valid connection ID, so no modification is required. This parameter is not configurable when the Master/Slave mode is set to "FSoE Slave."
Mode	Since this LX-LG901 uses the current custom connection to communicate with other slave stations as the master station, select "FSoE Master."
Watchdog(ms)	Represents the watchdog time for the connection. The value range is 1 to 65535. This parameter is not configurable when the Master/Slave mode is set to "FSoE Slave."

	Watchdog timeout indicates that the FSoE message reception interval at the slave exceeds the watchdog timeout time.
Safe Param	
Param(Hex)	Configure the safety parameters in hexadecimal format. The maximum supported configuration size is 128 bytes.

□ Configure the following parameters in the input/output areas respectively.

Parameter	Instruction
DataSize	Indicates the Length of Safety Data, choose based on the actual situation. Example: If the input data is 6 bytes, select “15 Bytes (6 Bytes Safe Data)” from the data length dropdown in the input area.
Safe Config	Actions Taken on Input/Output When the Module is Abnormal: Hold: The input/output retains the value before the abnormality. Clear: The input/output values are reset to zero. Safe Value: The input/output uses a safe value.

After selecting the data length, the software automatically fills all bytes, with the default type being Byte. Data can be deleted, inserted, or added according to actual needs (the first byte is the safety quality parameter and cannot be modified, deleted, or added).



□ Click "Export" to save the configured input/output data locally (.sfd file). This file can be directly imported when configuring the slave station LX-LG901 for communication in the future.

□ Click on the "EtherCAT IO Mapping" tab. The input/output data configured in the "Safety Data" tab is automatically added to this tab. Parameters with a white background can be modified according to actual needs to facilitate identification and management.

11.3.2.2.1.3 Configure Network Variables for Controller No. 1

(1) Return to the non-safety side interface of AutoThink No. 1, double-click on LX-LG901 No. 1.1 to open the configuration interface.

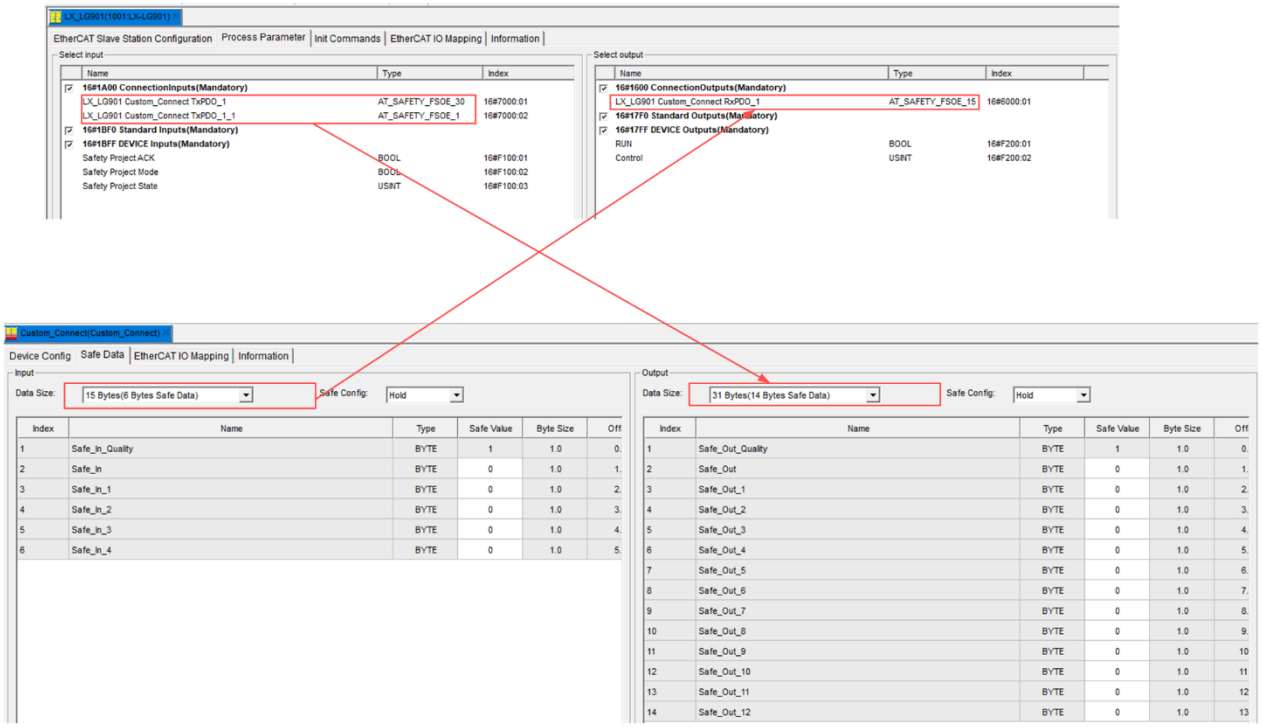
□ Click on the "Process Parameters" tab and check if the input/output parameters match the safety data configured in the safety side interface.

The output from the safety side corresponds to the input on the non-safety side, and the input from the safety side corresponds to the output on the non-safety side. Both the length and type of data should match.

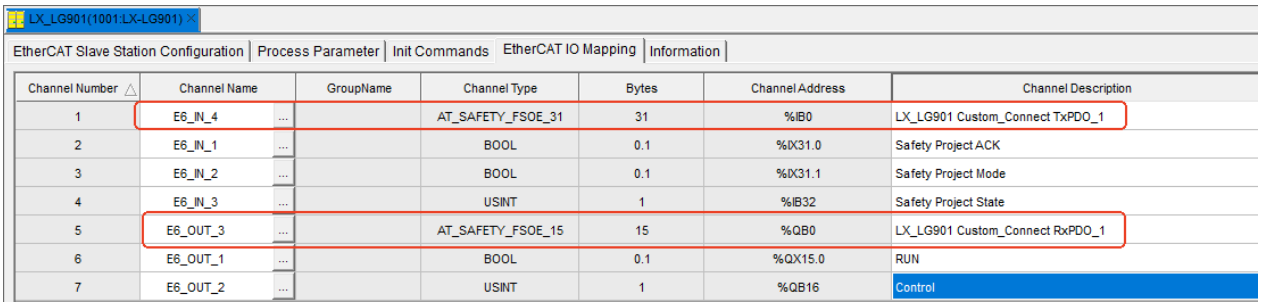
Due to the 30-byte length limitation on the non-safety side process parameters, if the input/output

data from the safety side exceeds 30 bytes, it will be automatically split. When checking, use custom connection names to distinguish them.

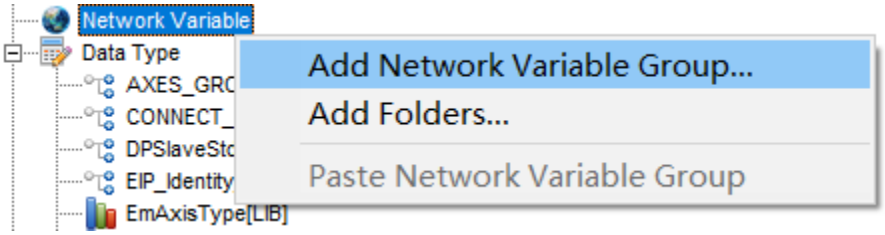
When the input/output data is updated datasize, the non-safety side will automatically synchronize these updates.



□ Click on the "EtherCAT IO Mapping" tab. AutoThink automatically generates the mapping relationship between process parameters and I/O areas. Check if the process parameters corresponding to the safety data have been correctly mapped.



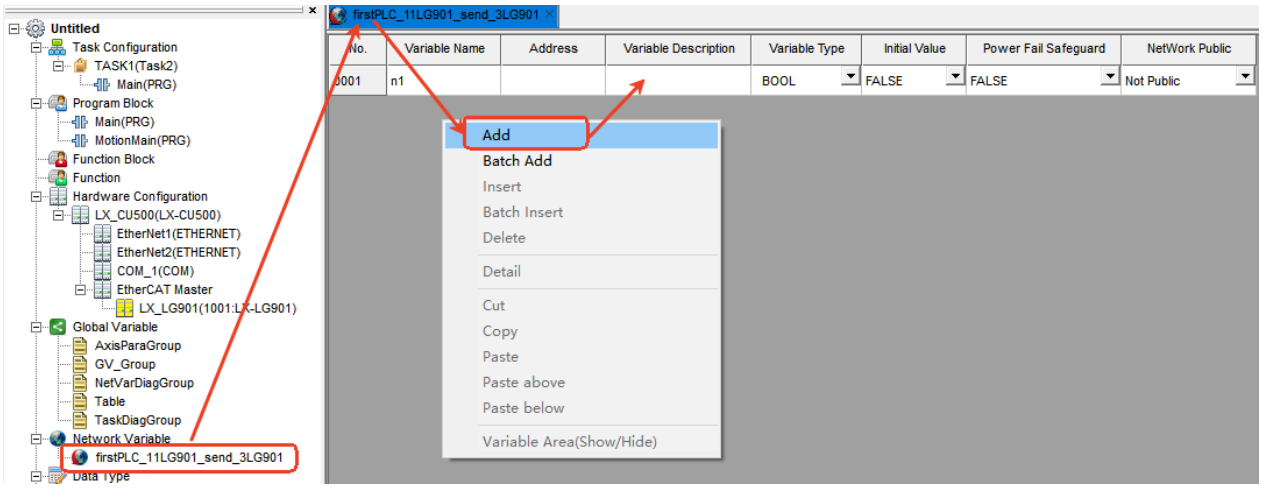
(2) In the navigation tree on the left, right-click "Network Variable" and select "Add Network Variable Group".



(3) In the "Add Network Variable Group" dialog box that appears, configure the following parameters and then click "OK".

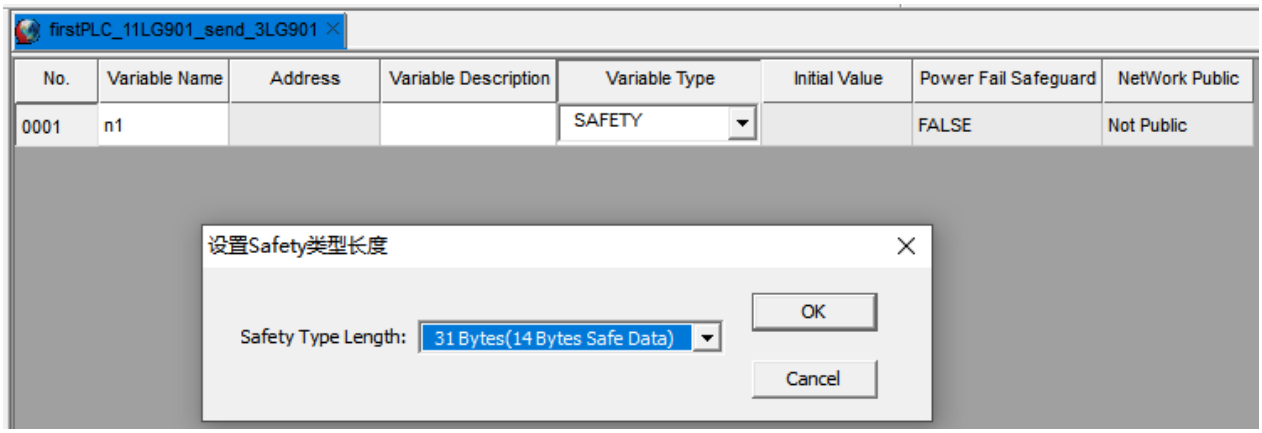
GroupSet	
Enable	Check
Sender/Receiver	select "Receiver"
GroupName	Enter the network variable group name, for example: firstPLC_11LG901_send_3LG901.
AssociateFile	Select the local .nvg file associated with the network variable group. If there is no associated file, you can manually enter the name, and the system will automatically create it locally , for example: firstPLC_send_11LG901_send_3LG901.
Export/Import	Select "Export" to export the configuration and variable list of the currently created network variable group to the associated file. Note: When selecting "Import," it indicates that the sender network variable group will be created based on the local associated file. After creation, editing is not supported.
NetworkSet	
NetworkType	Currently, only UDP transmission is supported.
BroadcastAddr	The destination IP address and port number for the UDP packet sent by the network variable sender default to 255.255.255.255:48040. When the target subnet is known, it is recommended to change the broadcast address to the subnet broadcast address of the specified segment (for example, when communicating with all controllers in the 192.168.0.x subnet through the P1 network interface, input 192.168.0.255 as the broadcast address). Note: If the broadcast address is in a different subnet from the controller's IP address, you need to configure routing information for the broadcast address in the "Network Configuration" tab under "Tool - Assistant Tool - Controller Operation" Otherwise, the network variable group may fail to send.
Port	
TransferSet	
AssociateTask	Network variable groups do not support association with Motiontask tasks.
Identifier	The identifier for the network variable group must be unique and cannot duplicate the identifiers of any existing network variable groups within the controller.
PackageVar	When checked, the network variable sender will send all variables within the group as a single package to minimize the number of packets sent and improve performance. When unchecked, the controller will generate and send a separate packet for each variable. It is recommended to check this option.
TransferCheck	It is recommended to check this option, which indicates that the sender provides a checksum for each packet, and the receiver only accepts packets with matching checksums.
Confirm	It is recommended to check this option, which indicates that the receiver sends an acknowledgment message to the sender for each received data packet.
CycleTransfer	When checked, the sender cyclically sends network variable data.
ChangeTransfer	When checked, the sender only sends variable data when the value of the variable changes.
EventTransfer	When checked, the sender only sends variable data when the specified variable's value is TRUE. This option only supports BOOL variables or BOOL member variables in complex data types.

(4) Double-click the network variable group created in the previous step to open the configuration interface. In the editing area, right-click the mouse and select "Add" to add a network variable.

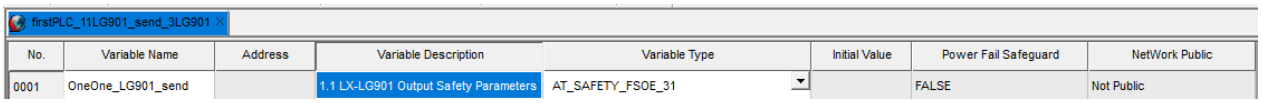


(5) Select the variable type as "AT_SAFETY_FSOE". In the "Set Safety Type Length" dialog box that

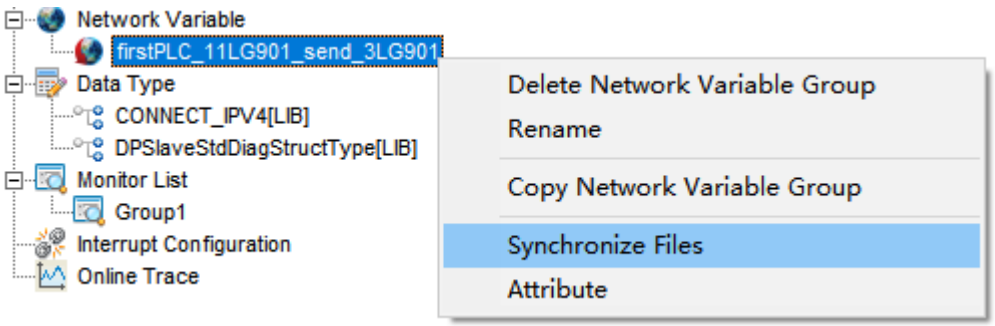
appears, choose the safety data output from No. 1.1 LX-LG901, which corresponds to the length of the input process parameters of No. 1 LX-CU500.



(6) Modify the variable name and variable description to help identify the purpose of the network variable.



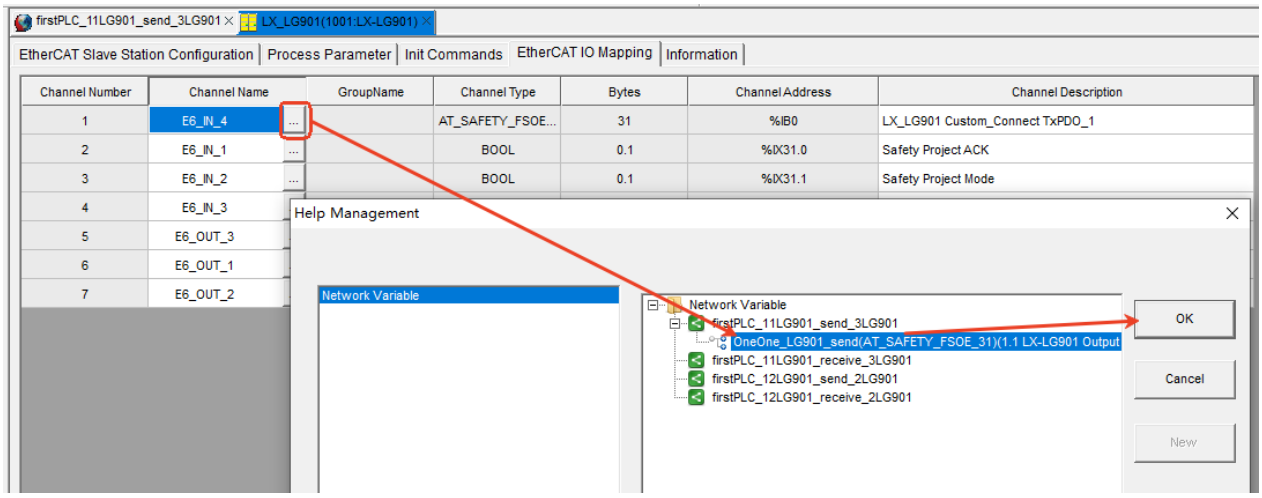
(7) Right-click the network variable group and select "Synchronize File" to ensure that the parameter information in the software matches the parameter information in the locally saved file.

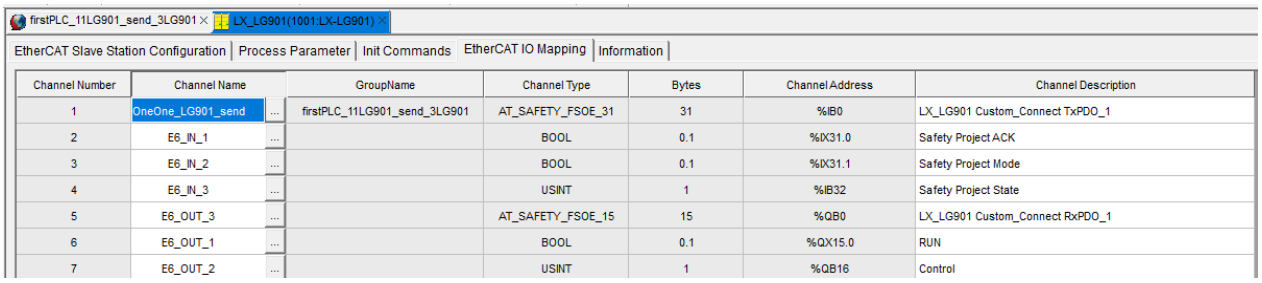


(8) Double-click No. 1.1 LX-LG901 to open the configuration interface.

(a) Click the "EtherCAT IO Mapping" tab, and then click the corresponding channel for the related input process parameter . . . . Select "Associate IEC Variables," choose the network variable created above, and click "OK."

(b) After associating the IEC variable, the one-to-one correspondence between the network variable (sender) of Controller 1, the IEC variable (input process parameter), and the safety data output of No. 1.1 LX-LG901 is established.





11.3.2.2.2 Create Network Variables for Controller No. 3

11.3.2.2.2.1 Add No. 3 LX-LG901

(1) In the navigation tree on the left side of 3 AutoThink, right-click “Hardware Configuration - LX-CU500 - EtherCAT Master” and select “Add Slave.”

(2) In the “Add Slave” dialog box that appears, select LX-LG901 and add one LX-LG901 module, which will be 3 LX-LG901.

11.3.2.2.2.2 Configure the Slave Custom Connection for No. 3 LX-LG901

(1) Right-click No. 3 LX-LG901 and select “Open Safety Engineering.”

(2) In the safety side AutoThink interface that appears, right-click “FSoE Configuration” - “LX_LG901” in the left navigation tree, select “Add,” and in the “Add” dialog box that appears, add one custom connection.

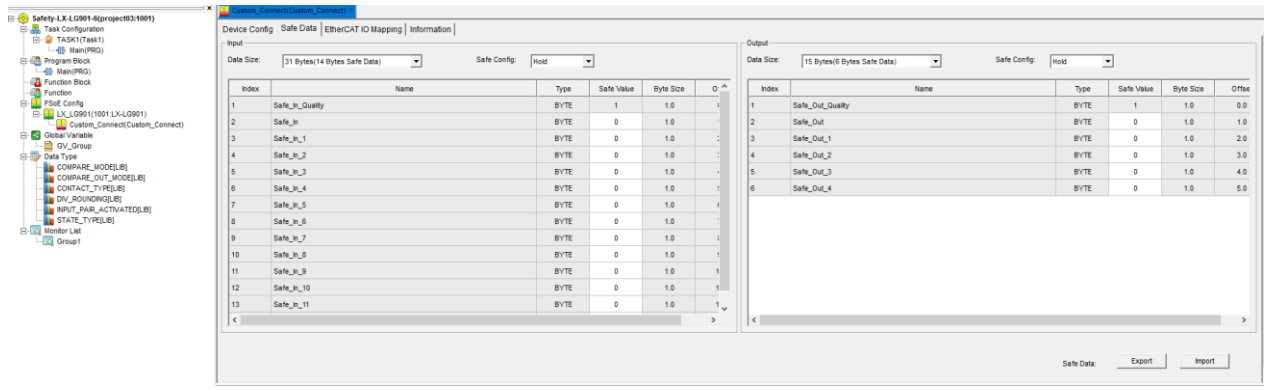
(3) Double-click the custom connection added in the previous step to open the configuration window.

□ Click the “Device Configuration” tab and configure the following parameters.

Base Info	
FSoEAddr	epresents the FSoE address of the custom connection. The value range is 1 to 65535. During configuration, it must not conflict with any existing FSoE addresses in the entire project.
Connection Setting	
Conn-ID	Configurable when the master/slave mode is set to "FSoE Master." It represents the ID of the custom connection. The value range is 1 to 65535. The system automatically assigns a unique and valid connection ID, so no modification is required. This parameter is not configurable when the Master/Slave mode is set to "FSoE Slave."
Mode	This indicates the master-slave mode of the connection. Since this LX-LG901 communicates with 1.1

	LX-LG901 as a slave through the current custom connection, "FSoE Slave" is selected.
Watchdog(ms)	Represents the watchdog time for the connection. The default value is 300, and the value range is 1 to 65535. This parameter is not configurable when the Master/Slave mode is set to "FSoE Slave." Watchdog timeout indicates that the FSoE message reception interval at the slave station is greater than the watchdog's timeout time.
Safe Param	
Length(Byte)	Configure the safety parameter length, with a value range of 1 to 128.

□ Click the “Safety Data” tab, select “Import,” and follow the prompts to import the file (with a .sfd extension) saved locally from the creation of network variables for Controller No. 1.



• Note: The input/output data length can also be manually selected here. When configuring manually, note that the output of the master station is the input of the slave station, and the input of the master station is the output of the slave station. The length and type of data must match; otherwise, normal communication will not be possible.

□ Click the “EtherCAT IO Mapping” tab. The input/output data configured in the “Safety Data” tab will be automatically added to this tab. Parameters with a white background can be modified according to actual needs for easier identification and management.

11.3.2.2.2.3 Configure the received network variables for Controller No. 3

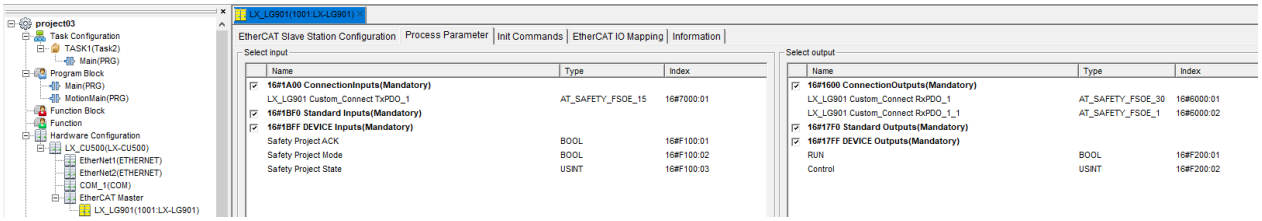
(1) Return to the non-safe side interface of AutoThink No. 3, double-click on No. 3 LX-LG901 to open the configuration interface.

□ Click the “Process Parameters” tab and check if the input/output parameters match the safety data configured in the safety side interface.

The output of the safety side is the input of the non-safe side; the input of the safety side is the output of the non-safe side. The length and type of data must match.

Since the process parameters on the non-safe side have a length limit of 30 bytes, when the input/output data from the safety side exceeds 30 bytes, it will be automatically split, and the distinction can be made through custom connection names during inspection.

After the input/output data is updated datasize, the non-safe side will automatically synchronize the updates.



□ Click the “EtherCAT IO Mapping” tab. AutoThink automatically generates the mapping relationship between process parameters and I/O areas. Check if the process parameters corresponding to the safety data have been mapped correctly.

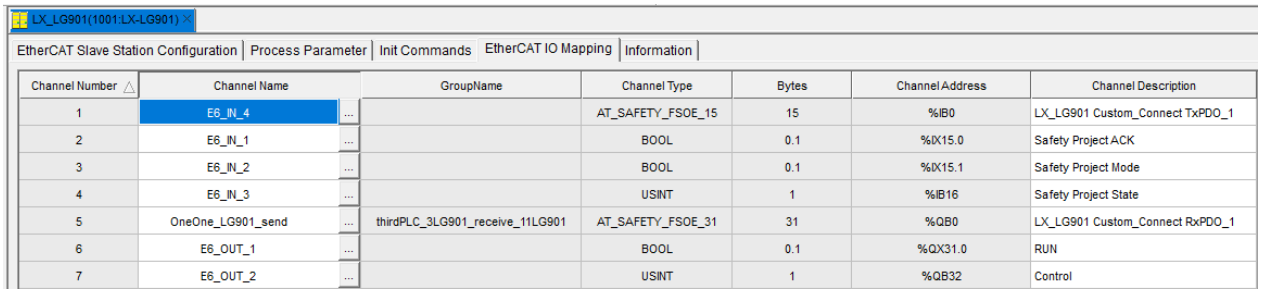
(2) In the navigation tree on the left, right-click “Network Variables,” select “Add Network Variable Group,” and in the “Add Network Variable Group” dialog box that appears, configure the following parameters, then click “OK.”

GroupSet	
Enable	Check
Sender/Receiver	select "Receiver"
GroupName	Enter the network variable group name, for example: thirdPLC_receive_11LG901.
AssociateFile	Select the associated file from Configure Network Variables for Controller No.1 (firstPLC_11LG901_send.nvg)
AssociateTask	The task associated with the network variable group does not support selecting the Motiontask task.

The system automatically imports the variable information from the associated file, and the imported information does not support additions, deletions, or modifications.

(3) Double-click on No. 3 LX-LG901 to open the configuration interface. Click the “EtherCAT IO Mapping” tab, then click on the channel corresponding to the relevant output process parameter ..., and select “Associate IEC Variables.” Choose the network variable created above, and click “OK”.

(4) After associating the IEC variable, the one-to-one correspondence between the network variable of Controller No. 3 (receiver) - IEC variable (output process parameter) - safety data of No. 3 LX-LG901 (input) is established.



11.3.2.2.3 Create Network Variables for Controller No. 3

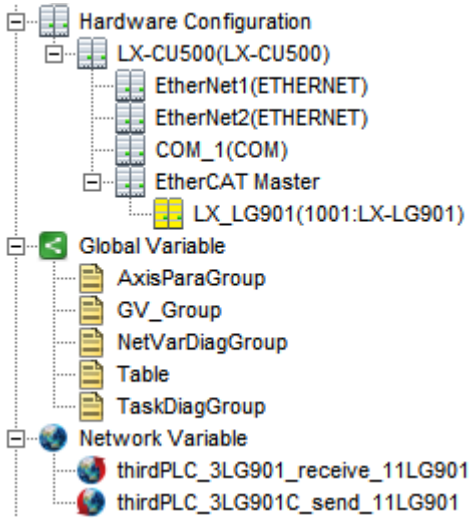
(1) In the navigation tree on the left side of AutoThink No. 3: Right-click on "Network Variables." Select "Add Network Variable Group."

(2) In the "Add Network Variable Group" dialog box, configure the following parameters and then click "OK." After adding, Controller No. 3 will have two network variable groups: one for sending, with the icon, and one for receiving, with the icon . Please note the difference between these two

network variable groups.

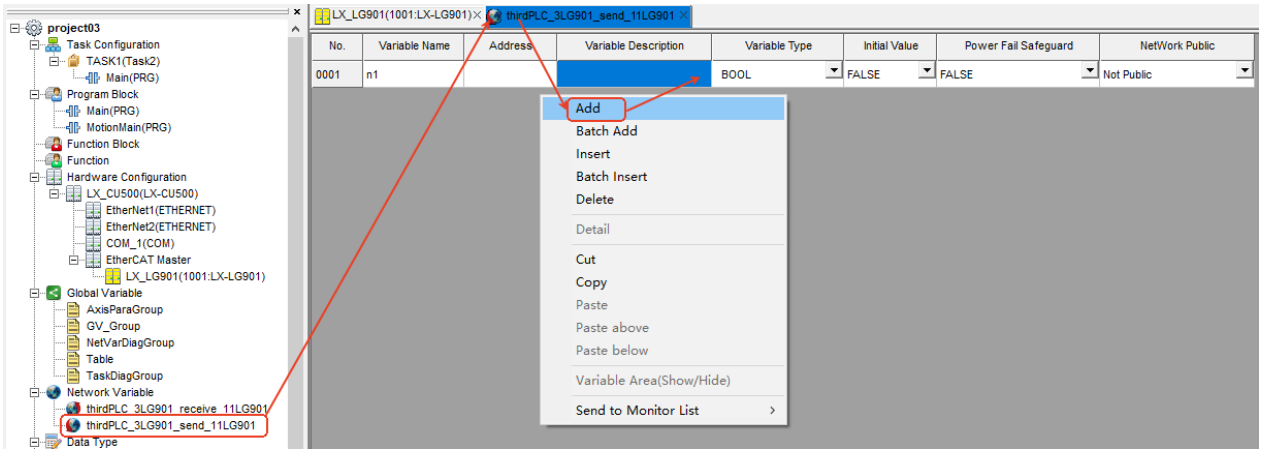
GroupSet	
Enable	Check
Sender/Receiver	select "Sender"
GroupName	Enter the network variable group name, for example: thirdPLC_3LG901C_send_11LG901.
AssociateFile	Select the local .nvg file associated with the network variable group. If there is no associated file, you can manually enter the name, and the system will automatically create it locally , for example: thirdPLC_3LG901C_send_11LG901.
Export/Import	Select "Export" to export the configuration and variable list of the currently created network variable group to the associated file.
NetworkSet	
NetworkType	Currently, only UDP transmission is supported.
BroadcastAddr	The destination IP address and port number for the UDP packet sent by the network variable sender default to 255.255.255.255:48040. When the target subnet is known, it is recommended to change the broadcast address to the subnet broadcast address of the specified segment (for example, when communicating with all controllers in the 192.168.0.x subnet through the P1 network interface, input 192.168.0.255 as the broadcast address). Note: If the broadcast address is in a different subnet from the controller's IP address, you need to configure routing information for the broadcast address in the "Network Configuration" tab under "Tool - Assistant Tool - Controller Operation" Otherwise, the network variable group may fail to send.
Port	
TransferSet	
AssociateTask	Network variable groups do not support association with Motiontask tasks.
Identifier	The identifier for the network variable group must be unique and cannot duplicate the identifiers of any existing network variable groups within the controller.
PackageVar	When checked, the network variable sender will send all variables within the group as a single package to minimize the number of packets sent and improve performance. When unchecked, the controller will generate and send a separate packet for each variable. It is recommended to check this option.
TransferCheck	It is recommended to check this option, which indicates that the sender provides a checksum for each packet, and the receiver only accepts packets with matching checksums.
Confirm	It is recommended to check this option, which indicates that the receiver sends an acknowledgment message to the sender for each received data packet.
CycleTransfer	When checked, the sender cyclically sends network variable data.
ChangeTransfer	When checked, the sender only sends variable data when the value of the variable changes.
EventTransfer	When checked, the sender only sends variable data when the specified variable's value is TRUE. This option only supports BOOL variables or BOOL member variables in complex data types.

The added sender and receiver network variable groups are shown in the figure below.

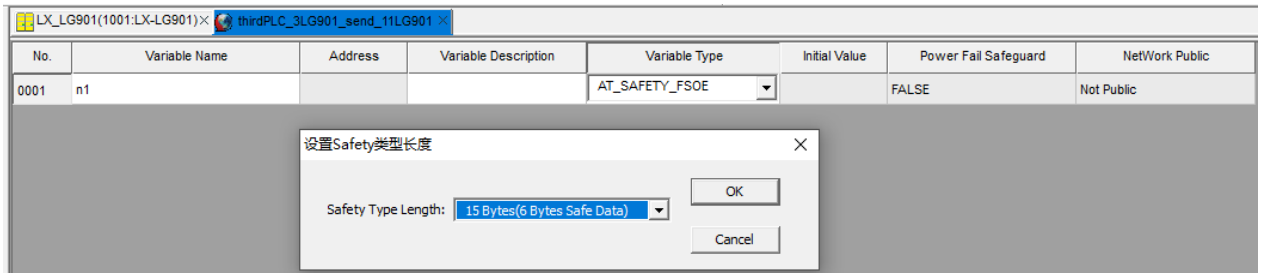


(3) Double-click the sender network variable group created in the previous step to open the configuration interface. In the editing area, right-click the mouse and select "Add" to add a network

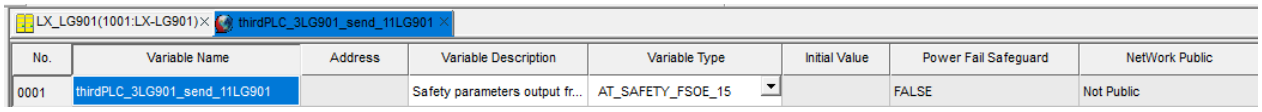
variable.



(4) Select the variable type as "AT_SAFETY_FSOE". In the "Set Safety Type Length" dialog box that appears, choose the output safety data length for No. 3 LX-LG901, which corresponds to the input process parameter length of No. 3 LX-CU500.



(5) Modify the variable name and variable description to help identify the purpose of the network variable.



(6) Right-click the network variable group and select "Synchronize Files" to ensure that the parameter information in the software matches the parameter information in the locally saved file.

(7) Double-click No. 3 LX-LG901 to open the configuration interface, and click the "EtherCAT IO Mapping" tab. Click the channel corresponding to the relevant input process parameter ..., and select "Associate IEC Variables." Choose the network variable created above and click "OK."

(8) After associating the IEC variable, the one-to-one correspondence between the network variable (sender) of Controller No. 3, the IEC variable (input process parameter), and the safety data output of No. 3 LX-LG901 is established.

11.3.2.2.4 Create a network variable group for receiving on Controller No. 1

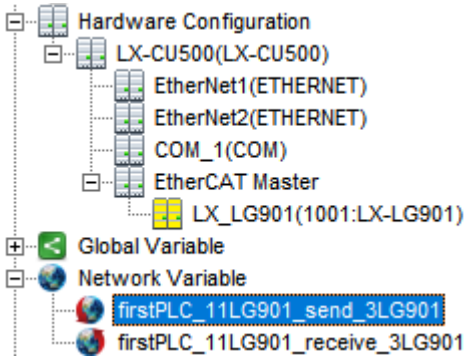
(1) In the navigation tree on the left side of No. 1 AutoThink, right-click "Network Variables" and select "Add Network Variable Group."

(2) In the "Add Network Variable Group" dialog box that appears, configure the following parameters

and then click "OK." After adding, Controller No. 1 will have two network variable groups: one for sending and one for receiving. Please note the difference between these two network variable groups.

GroupSet	
Enable	Check
Sender/Receiver	select "Receiver"
GroupName	Enter the network variable group name, for example: firstPLC_11LG901_receive_3LG901.
AssociateFile	Select the associated file from Create Network Variables for Controller No. 3 (thirdPLC_3LG901_send_11LG901.nvg)
AssociateTask	The task associated with the network variable group does not support selecting the Motiontask task.

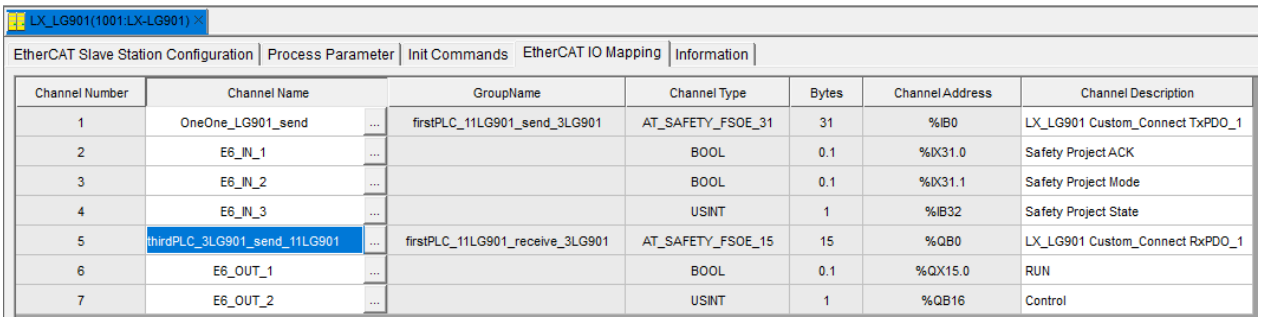
The added sender and receiver network variable groups are shown in the figure below.



The system automatically imports the variable information from the associated file, and the imported information cannot be modified, added, or deleted.

(3) Double-click No. 1.1 LX-LG901 to open the configuration interface, and click the "EtherCAT IO Mapping" tab. Click the channel corresponding to the relevant output process parameter , and select "Associate IEC Variable." Choose the receiver network variable created above and click "OK".

(4) After associating the IEC variable, the one-to-one correspondence between the network variable (receiver) of Controller No. 1, the IEC variable (output process parameter), and the safety data input of No. 1.1 LX-LG901 is established.



Channel Number	Channel Name	GroupName	Channel Type	Bytes	Channel Address	Channel Description
1	OneOne_LG901_send	firstPLC_11LG901_send_3LG901	AT_SAFETY_FSOE_31	31	%IB0	LX_LG901 Custom_Connect TxPDO_1
2	E6_IN_1		BOOL	0.1	%IX31.0	Safety Project ACK
3	E6_IN_2		BOOL	0.1	%IX31.1	Safety Project Mode
4	E6_IN_3		USINT	1	%IB32	Safety Project State
5	thirdPLC_3LG901_send_11LG901	firstPLC_11LG901_receive_3LG901	AT_SAFETY_FSOE_15	15	%QB0	LX_LG901 Custom_Connect RxPDO_1
6	E6_OUT_1		BOOL	0.1	%QX15.0	RUN
7	E6_OUT_2		USINT	1	%QB16	Control

11.3.2.3 Establish Communication Between LX-LG901 No. 1.2 and LX-LG901 No. 2

When 1.2 LX-LG901 and 2nd LX-LG901 communicate, they need to exchange data through the network

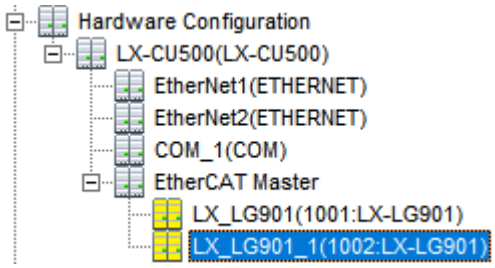
variables of the controllers. Therefore, the example of establishing communication follows the logical sequence below: Controller 1 sends network variable → Controller 2 receives network variable → Controller 2 returns network variable → Controller 1 receives network variable.

11.3.2.3.1 Create a network variable for sending on Controller No. 1

11.3.2.3.1.1 Add 1.2 LX-LG901

(1) In the navigation tree on the left side of 1st AutoThink, right-click "Hardware Configuration"—"LX-CU500"—"EtherCAT Master," and select "Add Slave Station".

(2) In the "Add Slave" dialog box that appears, select LX-LG901 and add one LX-LG901 module, which will be No. 1.2 LX-LG901. After adding, Controller 1 will have two LX-LG901 modules: the one with ID 1001 is No. 1.1 LX-LG901, and the one with ID 1002 is No. 1.2 LX-LG901.



11.3.2.3.1.2 Configure the master custom connection for No. 1.2 LX-LG901

(1) Right-click No. 1.2 LX-LG901 and select "Open Safety Engineering".

(2) In the safety-side AutoThink interface that pops up, right-click "FSoE Configuration"—"LX_LG901" in the left navigation tree, and select "Add". In the "Add" dialog box that appears, add one custom connection. Double-click the custom connection added in the previous step to open the configuration window.

□ Click the "Device Configuration" tab and configure the following parameters.

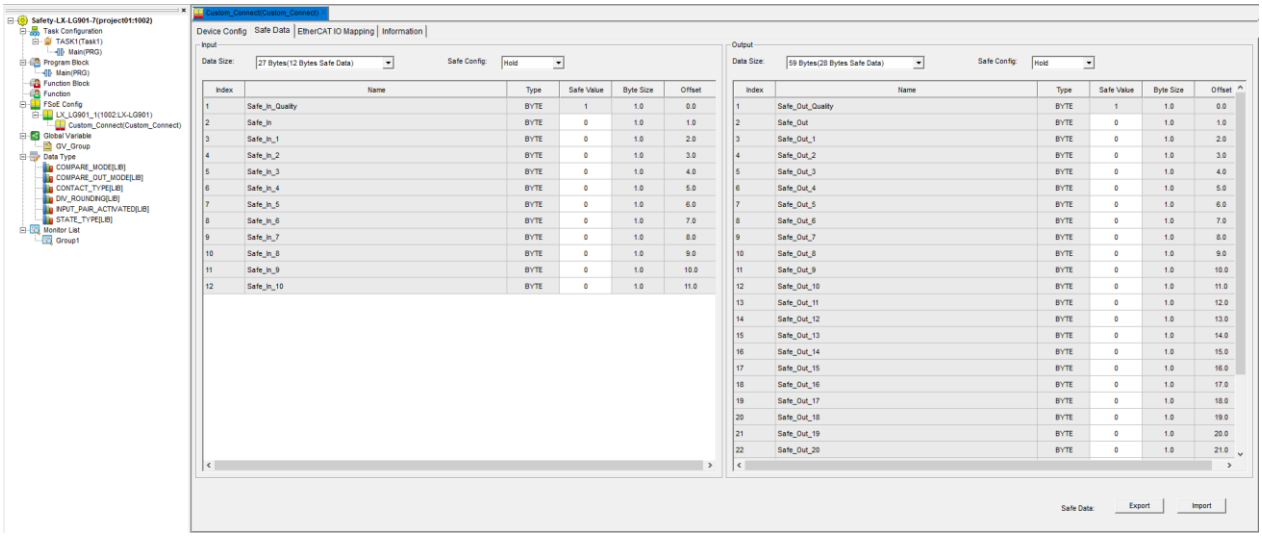
Base Info	
FSoEAddr	Represents the FSoE address of the custom connection. The value range is 1 to 65535. During configuration, it must not conflict with any existing FSoE addresses in the entire project.
Connection Setting	
Conn-ID	Configurable when the master/slave mode is set to "FSoE Master." It represents the ID of the custom connection. The value range is 1 to 65535. The system automatically assigns a unique and valid connection ID, so no modification is required. This parameter is not configurable when the Master/Slave mode is set to "FSoE Slave."
Mode	Since this LX-LG901 uses the current custom connection to communicate with other slave stations as the master station, select "FSoE Master".
Watchdog(ms)	Represents the watchdog time for the connection. The default value is 300, and the value range is 1 to 65535. This parameter is not configurable when the Master/Slave mode is set to "FSoE Slave." Watchdog timeout indicates that the FSoE message reception interval at the slave station is greater than the watchdog's timeout time.
Safe Param	
Param(Hex)	Configure the safety parameters in hexadecimal format. The maximum supported configuration size is 128 bytes.

□ Configure the following parameters in the input/output areas respectively.

Parameter	**Instruction**
DataSize	Indicates the Length of Safety Data, choose based on the actual situation.

	Example: If the input data is 6 bytes, select “15 Bytes (6 Bytes Safe Data)” from the data length dropdown in the input area.
Safe Config	Actions Taken on Input/Output When the Module is Abnormal: Hold: The input/output retains the value before the abnormality. Clear: The input/output values are reset to zero. Safe Value: The input/output uses a safe value.

After selecting the data length, the software automatically fills all bytes, with the default type being Byte. Data can be deleted, inserted, or added according to actual needs (the first byte is the safety quality parameter and cannot be modified, deleted, or added).



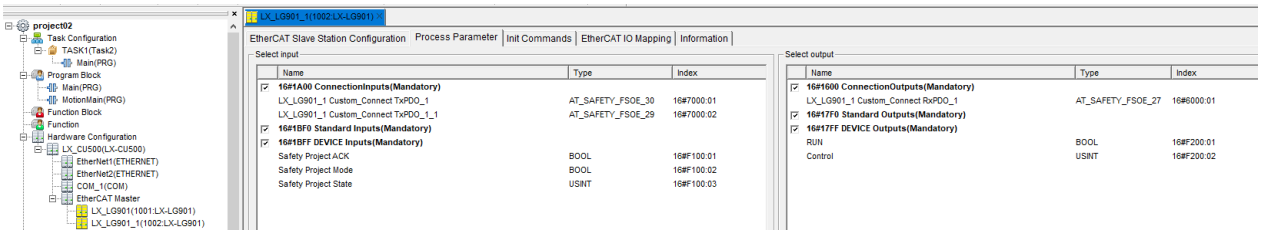
Click "Export" to export and save the configured input/output data to your local machine (.sfd file). This file can be directly imported when configuring the slave LX-LG901 to communicate with it later.

- Click the "EtherCAT IO Mapping" tab. The input/output data configured in the "Safety Data" tab is automatically added to this tab. Parameters with a white background can be modified according to actual needs to facilitate identification and management.

11.3.2.3.1.3 Configure the network variable for sending on Controller No. 1

(1) Return to the non-safety side interface of No. 1 AutoThink. Double-click No. 1.2 LX-LG901 to open the configuration interface.

- Click the "Process Parameters" tab and check if the input/output parameters match the safety data configured in the safety side interface.



- Click the "EtherCAT IO Mapping" tab. AutoThink automatically generates the mapping relationship between process parameters and I/O areas. Check if the process parameters corresponding to the safety data have been mapped correctly.

(2) In the left navigation tree, right-click "Network Variables" and select "Add Network Variable Group".

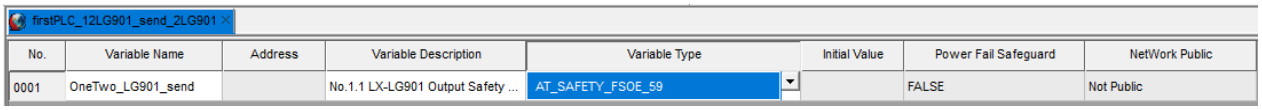
(3) In the "Add Network Variable Group" dialog box that appears, configure the following parameters and then click "OK".

GroupSet	
Enable	Check
Sender/Receiver	select "Sender"
GroupName	Enter the network variable group name, for example: firstPLC_12LG901_send_2LG901.
AssociateFile	Select the local .nvg file associated with the network variable group. If there is no associated file, you can manually enter the name, and the system will automatically create it locally , for example: firstPLC_12LG901_send_2LG901.
Export/Import	Select "Export" to export the configuration and variable list of the currently created network variable group to the associated file. Note: When selecting "Import," it indicates that the sender network variable group will be created based on the local associated file. After creation, editing is not supported.
NetworkSet	
NetworkType	Currently, only UDP transmission is supported.
BroadcastAddr	The destination IP address and port number for the UDP packet sent by the network variable sender default to 255.255.255.255:48040. When the target subnet is known, it is recommended to change the broadcast address to the subnet broadcast address of the specified segment (for example, when communicating with all controllers in the 192.168.0.x subnet through the P1 network interface, input 192.168.0.255 as the broadcast address). Note: If the broadcast address is in a different subnet from the controller's IP address, you need to configure routing information for the broadcast address in the "Network Configuration" tab under "Tool - Assistant Tool - Controller Operation" Otherwise, the network variable group may fail to send.
Port	
TransferSet	
AssociateTask	Network variable groups do not support association with Motiontask tasks.
Identifier	The identifier for the network variable group must be unique and cannot duplicate the identifiers of any existing network variable groups within the controller.
PackageVar	When checked, the network variable sender will send all variables within the group as a single package to minimize the number of packets sent and improve performance. When unchecked, the controller will generate and send a separate packet for each variable. It is recommended to check this option.
TransferCheck	It is recommended to check this option, which indicates that the sender provides a checksum for each packet, and the receiver only accepts packets with matching checksums.
Confirm	It is recommended to check this option, which indicates that the receiver sends an acknowledgment message to the sender for each received data packet.
CycleTransfer	When checked, the sender cyclically sends network variable data.
ChangeTransfer	When checked, the sender only sends variable data when the value of the variable changes.
EventTransfer	When checked, the sender only sends variable data when the specified variable's value is TRUE. This option only supports BOOL variables or BOOL member variables in complex data types.

(4) Double-click the network variable group created in the previous step to open the configuration interface. In the editing area, right-click and select "Add Variable" to add a network variable.

(5) Choose "AT_SAFETY_FSOE" as the variable type. In the "Set Safety Type Length" dialog box that appears, select the length of the safety data output from No. 1.2 LX-LG901, which corresponds to the input process parameter of 1 LX-CU500.

(6) Modify the variable name and description to help identify the purpose of the network variable.

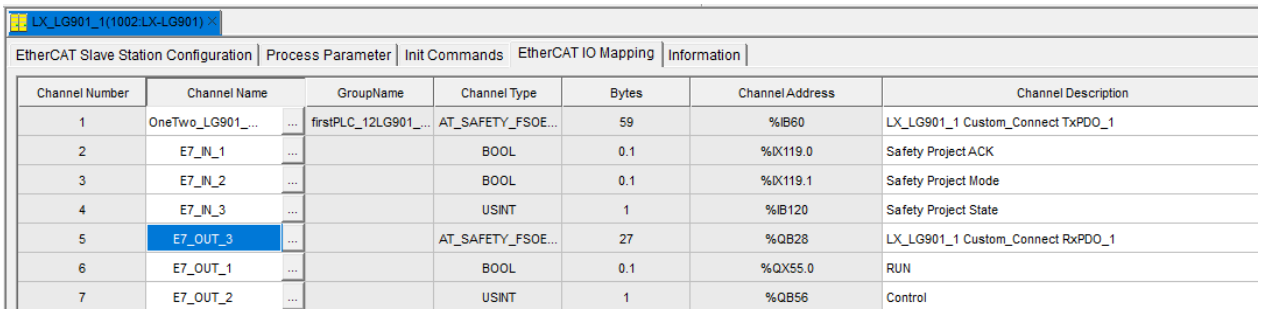


(7) Right-click the network variable group and select "Synchronize File" to ensure that the parameter information in the software matches the parameter information in the locally saved file.

(8) Double-click No. 1.2 LX-LG901 to open the configuration interface, click the "EtherCAT IO Mapping" tab, and click the channel corresponding to the relevant input process parameter. Select

"Associate IEC Variables" Choose the network variable created earlier and click "OK".

(9) After associating the IEC variable, the one-to-one correspondence between the network variable (transmitter) of Controller No. 1 - the IEC variable (input process parameter) - and the safety data (output) of No. 1.2 LX-LG901 is established.



Channel Number	Channel Name	GroupName	Channel Type	Bytes	Channel Address	Channel Description
1	OneTwo_LG901_...	firstPLC_12LG901_...	AT_SAFETY_FSOE...	59	%IB60	LX_LG901_1 Custom_Connect TxPDO_1
2	E7_IN_1		BOOL	0.1	%IX119.0	Safety Project ACK
3	E7_IN_2		BOOL	0.1	%IX119.1	Safety Project Mode
4	E7_IN_3		USINT	1	%IB120	Safety Project State
5	E7_OUT_3		AT_SAFETY_FSOE...	27	%QB28	LX_LG901_1 Custom_Connect RxPDO_1
6	E7_OUT_1		BOOL	0.1	%QX55.0	RUN
7	E7_OUT_2		USINT	1	%QB56	Control

11.3.2.3.2 Create Network Variables for Controller No. 2

11.3.2.3.2.1 Add No. 2 LX-LG901

(1) In the navigation tree on the left side of AutoThink No. 2, right-click “Hardware Configuration - LX-CU500 - EtherCAT Master,” and select “Add Slave Station”.

(2) In the “Add Slave Station” dialog box that appears, select LX-LG901 and add 1 LX-LG901 module, which will be the No. 2 LX-LG901.

11.3.2.3.2.2 Configure Custom Connection for No. 2 LX-LG901 Slave

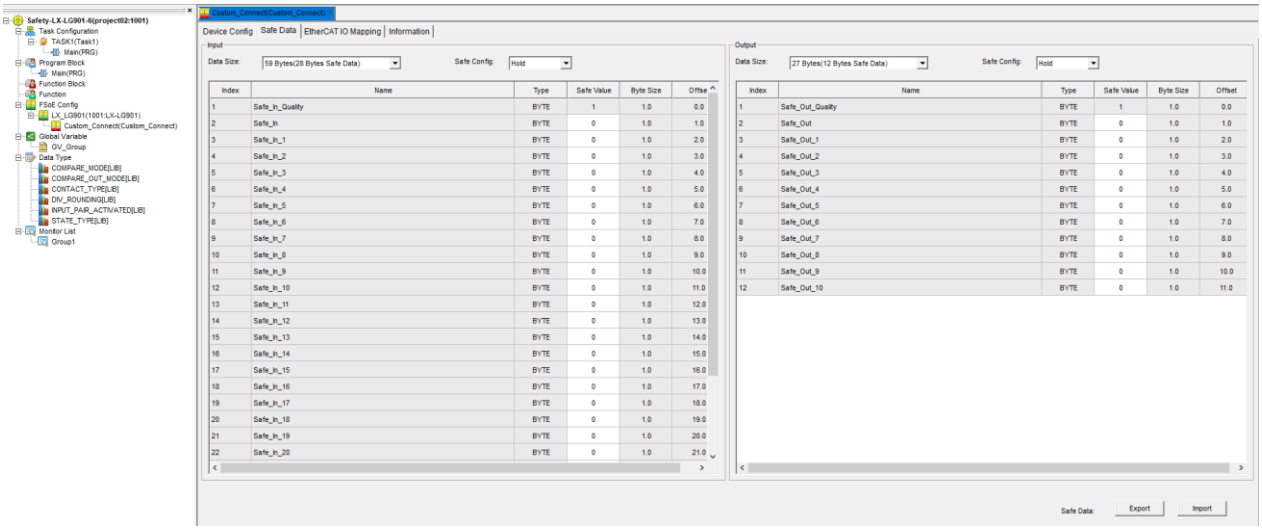
(1) Right-click the No. 2 LX-LG901 and select “Open Safety Project”.

(2) In the safety-side AutoThink interface that pops up, right-click “FSoE Configuration”—“LX_LG901,” and select “Add”. In the “Add” dialog box that appears, add 1 custom connection. Double-click the custom connection added in the previous step to open the configuration window.

□ Click the “Device Configuration” tab and configure the following parameters.

Base Info	
FSoEAddr	Represents the FSoE address of the custom connection. The value range is 1 to 65535. During configuration, it must not conflict with any existing FSoE addresses in the entire project.
Connection Setting	
Conn-ID	Configurable when the master/slave mode is set to "FSoE Master." It represents the ID of the custom connection. The value range is 1 to 65535. The system automatically assigns a unique and valid connection ID, so no modification is required. This parameter is not configurable when the Master/Slave mode is set to "FSoE Slave."
Mode	This indicates the master-slave mode of the connection. Since this LX-LG901 communicates with the No. 1.2 LX-LG901 as a slave through this custom connection, select “FSoE Slave.”
Watchdog(ms)	Represents the watchdog time for the connection. The default value is 300, and the value range is 1 to 65535. This parameter is not configurable when the Master/Slave mode is set to "FSoE Slave." Watchdog timeout indicates that the FSoE message reception interval at the slave exceeds the watchdog timeout time.
Safe Param	
Length(Byte)	CConfigure the safety parameter length for the master station . The maximum supported configuration size is 128 bytes.

□ Click the “Safety Data” tab, select “Import,” and follow the prompts to import the .sfd file saved locally from the creation of network variables for Controller No. 1.



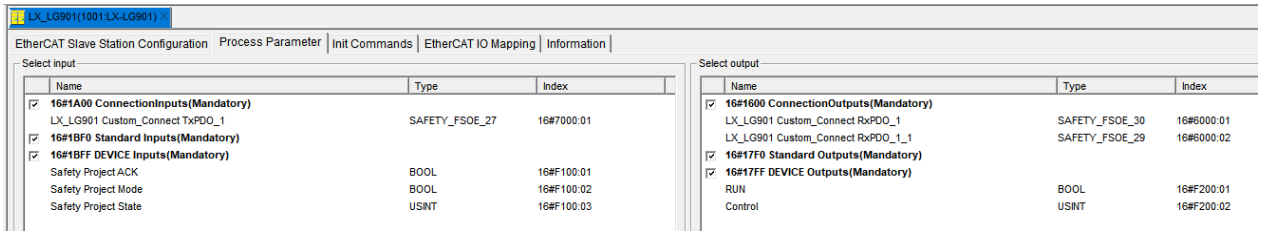
- Note: You can also manually select the input/output data length. When configuring manually, note that the output of the master station is the input of the slave station, and the input of the master station is the output of the slave station. The length and type of the data must correspond and match; otherwise, normal communication cannot be established.

□ Click the “EtherCAT IO Mapping” Tab, The input/output data configured in the “Safety Data” tab is automatically added to this tab. Parameters with a white background can be modified. Modify them according to actual needs to facilitate identification and management.

11.3.2.3.2.3 Configure Network Variables for Controller No. 2

(1) Return to the non-safety side interface of AutoThink No. 2, double-click the No. 2 LX-LG901 to open the configuration interface.

□ Click the “Process Parameters” tab and check if the input/output parameters match the safety data configured in the safety side interface.



□ Click the “EtherCAT IO Mapping ” tab, the AutoThink automatically generates the mapping relationship between process parameters and I/O areas. Check if the process parameters corresponding to the safety data have been mapped correctly.

(2) In the navigation tree on the left, Right-click “Network Variables,” select “Add Network Variable Group.” In the “Add Network Variable Group” dialog box that appears, configure the following parameters and then click “OK”.

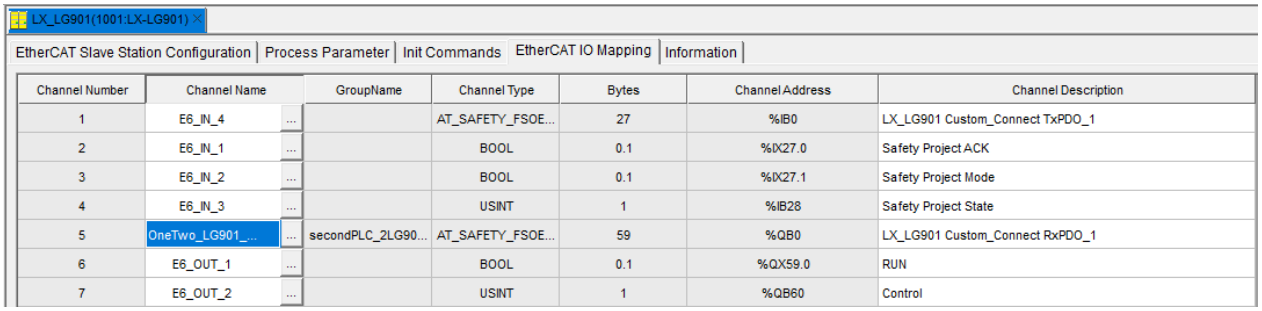
GroupSet	
Enable	Check
Sender/Receiver	select "Receiver"
GroupName	Enter the network variable group name, for example: secondPLC_2LG901_receive_12LG901.

AssociateFile	Select the associated file from Configure Network Variables for Controller No.1 (firstPLC_12LG901_send_2LG901.nvg)
AssociateTask	The task associated with the network variable group does not support selecting the Motiontask task.

System automatically imports variable information from associated files.The imported information does not support addition, deletion, or modification.

Double-Click No. 2 LX-LG901:Open the configuration interface, click the “EtherCAT IO Mapping” tab, and click the channel corresponding to the relevant output process parameter . . . . Select “Associate IEC Variable.” Choose the network variable created above and click “OK”.

After associating the IEC variable, this completes the one-to-one correspondence between the network variables of Controller 2 (receiver) - IEC variables (output process parameters) - safety data of No.2 LX-LG901 (input).



11.3.2.3.3 Create Network Variables for Sending from Controller No. 2

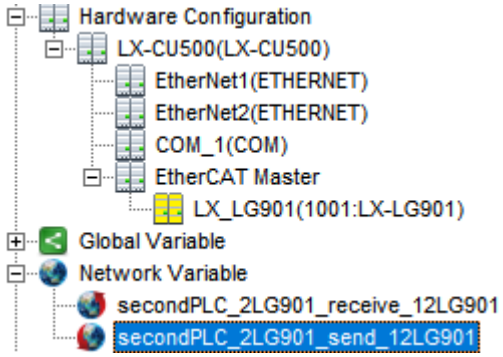
(1) In the navigation tree on the left side of AutoThink 2, right-click “Network Variables” and select “Add Network Variable Group”.

(2) In the “Add Network Variable Group” dialog box that appears, configure the following parameters and then click “OK.” After adding, Controller No. 2 will have two network variable groups, one for sending and one for receiving. Please note the distinction.

GroupSet	
Enable	Check
Sender/Receiver	select "Sender"
GroupName	Enter the network variable group name, for example: secondPLC_2LG901_send_12LG901.
AssociateFile	Select the local .nvg file associated with the network variable group. If there is no associated file, you can manually enter the name, and the system will automatically create it locally , for example: secondPLC_2LG901_send_12LG901.
Export/Import	Select "Export" to export the configuration and variable list of the currently created network variable group to the associated file.
NetworkSet	
NetworkType	Currently, only UDP transmission is supported.
BroadcastAddr	The destination IP address and port number for the UDP packet sent by the network variable sender default to 255.255.255.255:48040. When the target subnet is known, it is recommended to change the broadcast address to the subnet broadcast address of the specified segment (for example, when communicating with all controllers in the 192.168.0.x subnet through the P1 network interface, input 192.168.0.255 as the broadcast address). Note: If the broadcast address is in a different subnet from the controller's IP address, you need to configure routing information for the broadcast address in the "Network Configuration" tab under "Tool - Assistant Tool - Controller Operation" Otherwise, the network variable group may fail to send.
Port	
TransferSet	
AssociateTask	Network variable groups do not support association with Motiontask tasks.

Identifier	The identifier for the network variable group must be unique and cannot duplicate the identifiers of any existing network variable groups within the controller.
PackageVar	When checked, the network variable sender will send all variables within the group as a single package to minimize the number of packets sent and improve performance. When unchecked, the controller will generate and send a separate packet for each variable. It is recommended to check this option.
TransferCheck	It is recommended to check this option, which indicates that the sender provides a checksum for each packet, and the receiver only accepts packets with matching checksums.
Confirm	It is recommended to check this option, which indicates that the receiver sends an acknowledgment message to the sender for each received data packet.
CycleTransfer	When checked, the sender cyclically sends network variable data.
ChangeTransfer	When checked, the sender only sends variable data when the value of the variable changes.
EventTransfer	When checked, the sender only sends variable data when the specified variable's value is TRUE. This option only supports BOOL variables or BOOL member variables in complex data types.

The added sender and receiver network variable groups are shown in the figure below.

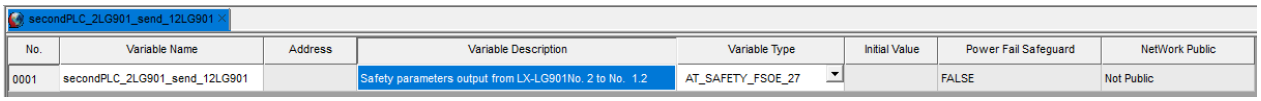


(3) Double-click the sender network variable group created in the previous step, open the configuration interface.

(4) In the editing area, right-click and select “Add Variable”, add a network variable.

(5) Choose the variable type as “AT_SAFETY_FSOE”, in the “Set Safety Type Length” dialog box that appears, select the length of the safety data output by No. 2 LX-LG901, which corresponds to the input process parameters of No. 2 LX-CU500.

(6) Modify the variable name and description,This helps in identifying the purpose of the network variable.

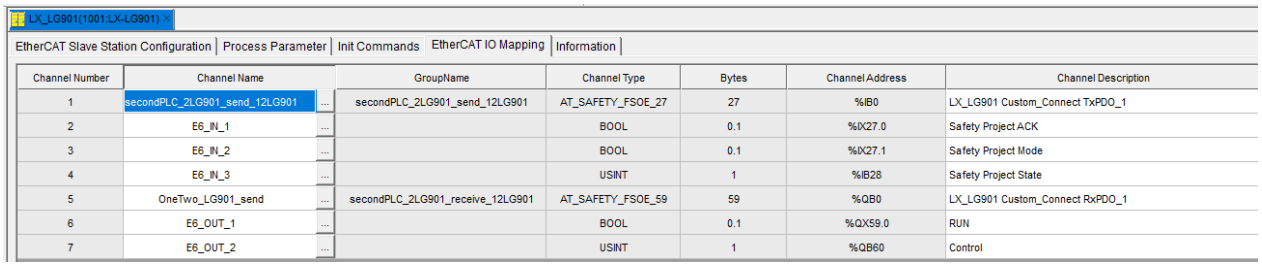

secondPLC_2LG901_send_12LG901

No.	Variable Name	Address	Variable Description	Variable Type	Initial Value	Power Fail Safeguard	NetWork Public
0001	secondPLC_2LG901_send_12LG901		Safety parameters output from LX-LG901No. 2 to No. 1.2	AT_SAFETY_FSOE_27		FALSE	Not Public

(7) Right-click the network variable group and select "Synchronize Files" to ensure that the parameter information in the software matches the parameter information in the locally saved file.

(8) Double-click No. 2 LX-LG901 to open the configuration interface, and click the "EtherCAT IO Mapping" tab. Click the channel corresponding to the relevant input process parameter  , and select "Associate IEC Variables." Choose the network variable created above and click "OK."

(9) After associating the IEC variable, the one-to-one correspondence between the network variable (sender) of Controller No. 2 - the IEC variable (input process parameter) - and the safety data output of No. 2 LX-LG901 is established.



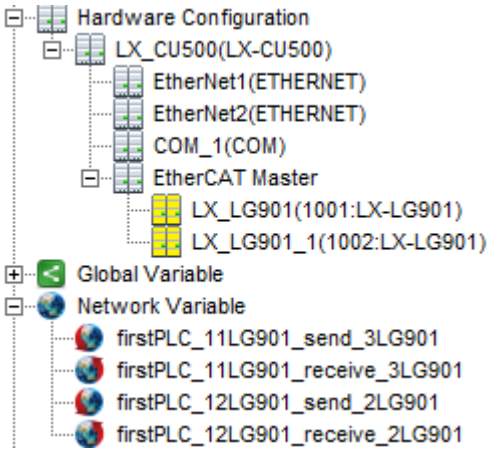
Channel Number	Channel Name	GroupName	Channel Type	Bytes	Channel Address	Channel Description
1	secondPLC_2LG901_send_12LG901	secondPLC_2LG901_send_12LG901	AT_SAFETY_FSOE_27	27	%IB0	LX_LG901 Custom_Connect TxPDO_1
2	E6_IN_1		BOOL	0.1	%IX27.0	Safety Project ACK
3	E6_IN_2		BOOL	0.1	%IX27.1	Safety Project Mode
4	E6_IN_3		USINT	1	%IB28	Safety Project State
5	OneTwo_LG901_send	secondPLC_2LG901_receive_12LG901	AT_SAFETY_FSOE_59	59	%QB0	LX_LG901 Custom_Connect RxPDO_1
6	E6_OUT_1		BOOL	0.1	%QX59.0	RUN
7	E6_OUT_2		USINT	1	%QB60	Control

11.3.2.3.4 Create a network variable group for receiving on Controller No. 1

(1) In the navigation tree on the left side of No. 1 AutoThink, right-click "Network Variables" and select "Add Network Variable Group." In the "Add Network Variable Group" dialog box that appears, configure the following parameters and then click "OK." After adding, Controller No. 1 will have two network variable groups: two for 1.1 LX-LG901 and two for 1.2 LX-LG901. Please note the distinction.

GroupSet	
Enable	Check
Sender/Receiver	select "Receiver"
GroupName	Enter the network variable group name, for example: firstPLC_12LG901_receive_2LG901.
AssociateFile	Select the associated file from Create Network Variables for Controller No. 2 (secondPLC_2LG901_send_12LG901nvg)
AssociateTask	The task associated with the network variable group does not support selecting the Motiontask task.

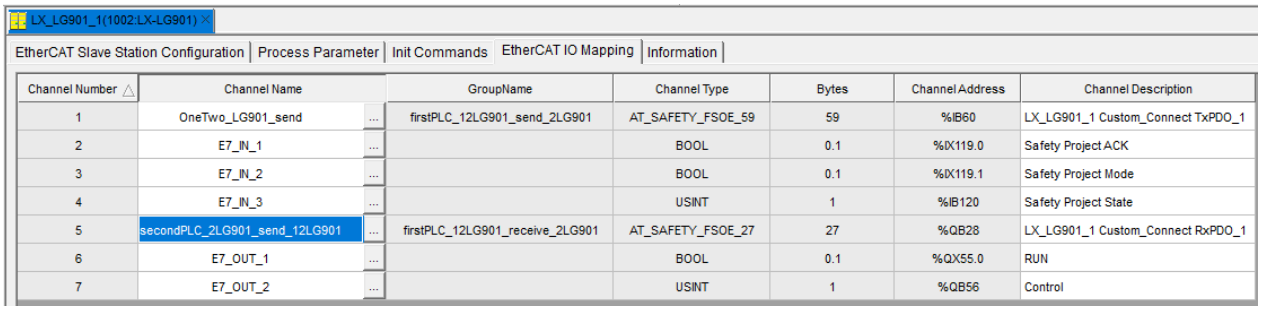
The added sender and receiver network variable groups are shown in the figure below.



The system automatically imports the variable information from the associated file, and the imported information cannot be modified, added, or deleted.

(2) Double-click No. 1.2 LX-LG901 to open the configuration interface, and click the "EtherCAT IO Mapping" tab. Click the channel corresponding to the relevant output process parameter  , and select "Associate IEC Variable." Choose the receiver network variable created above and click "OK".

(3) After associating the IEC variable, the one-to-one correspondence between the network variable (receiver) of Controller 1, the IEC variable (output process parameter), and the safety data input of No. 1.2 LX-LG901 is established.



Channel Number	Channel Name	GroupName	Channel Type	Bytes	Channel Address	Channel Description
1	OneTwo_LG901_send	firstPLC_12LG901_send_2LG901	AT_SAFETY_FSOE_59	59	%IB60	LX_LG901_1 Custom_Connect TxPDO_1
2	E7_IN_1		BOOL	0.1	%IX119.0	Safety Project ACK
3	E7_IN_2		BOOL	0.1	%IX119.1	Safety Project Mode
4	E7_IN_3		USINT	1	%IB120	Safety Project State
5	secondPLC_2LG901_send_12LG901	firstPLC_12LG901_receive_2LG901	AT_SAFETY_FSOE_27	27	%QB28	LX_LG901_1 Custom_Connect RxPDO_1
6	E7_OUT_1		BOOL	0.1	%QX55.0	RUN
7	E7_OUT_2		USINT	1	%QB56	Control

11.3.2.4 Establish Communication Between LX-LG901 No. 2 and LX-LG901 No. 3

Establishing communication between No. 2 LX-LG901 and No. 3 LX-LG901 using network variable:The communication setup follows this logical sequence: No. 2 Controller sends network variables → No. 3 Controller receives network variables → No. 3 Controller returns network variables → No. 2 Controller receives network variables.

11.3.2.4.1 Create Network Variables for Sending from Controller No. 2

11.3.2.4.1.1 Configure Custom Connection for No. 2 LX-LG901 Master Station

(1) Right-click No. 2 LX-LG901 and select “Open Safety Project.

(2) In the safety-side AutoThink interface that appears, right-click “FSoE Configuration” — “LX_LG901” and select “Add”. In the “Add” dialog box that appears, add one custom connection. After adding, this LX-LG901 will have two custom connections: one as a slave station to communicate with No. 1.2 LX-LG901, and one as a master station to communicate with No. 3 LX-LG901. Double-click the custom connection added in the previous step to open the configuration window.

- FSoE Slave (Slave Type) connection icon is red. FSoE Master (Master Type) connection icon is green.

□ Click the "Device Configuration" tab and configure the following parameters.

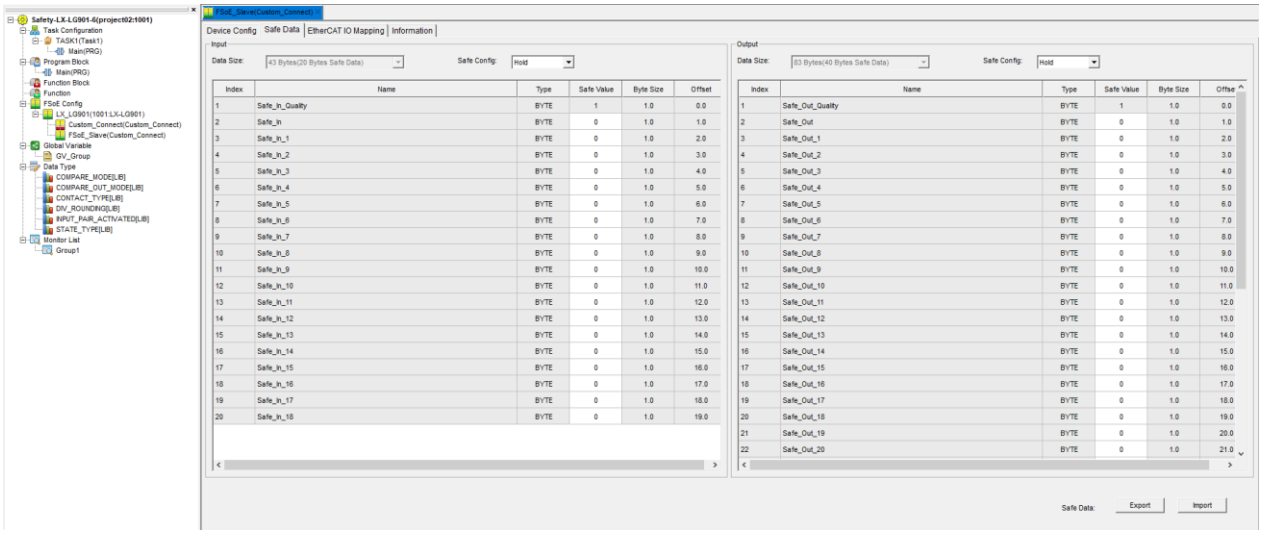
Base Info	
FSoEAddr	Represents the FSoE address of the custom connection. The value range is 1 to 65535. During configuration, it must not conflict with any existing FSoE addresses in the entire project.
Connection Setting	
Conn-ID	Configurable when the master/slave mode is set to "FSoE Master." It represents the ID of the custom connection. The value range is 1 to 65535. The system automatically assigns a unique and valid connection ID, so no modification is required. This parameter is not configurable when the Master/Slave mode is set to "FSoE Slave."
Mode	Since this LX-LG901 uses the current custom connection to communicate with other slave stations as the master station, select "FSoE Master."
Watchdog(ms)	Represents the watchdog time for the connection.The default value is 300, and the value range is 1 to 65535. This parameter is not configurable when the Master/Slave mode is set to "FSoE Slave." Watchdog timeout indicates that the FSoE message reception interval at the slave exceeds the watchdog timeout time.
Safe Param	

Param(Hex)	Configure the safety parameters in hexadecimal format. The maximum supported configuration size is 128 bytes.

□ Configure the following parameters in the input/output areas respectively.

Parameter	Instruction
DataSize	Indicates the Length of Safety Data, choose based on the actual situation. Example: If the input data is 6 bytes, select “43 Bytes(20 Bytes Safe Data)” from the data length dropdown in the input area.
Safe Config	Actions Taken on Input/Output When the Module is Abnormal: Hold: The input/output retains the value before the abnormality. Clear: The input/output values are reset to zero. Safe Value: The input/output uses a safe value.

After selecting the data length, the software automatically fills all bytes, with the default type being Byte. Data can be deleted, inserted, or added according to actual needs (the first byte is the safety quality parameter and cannot be modified, deleted, or added).



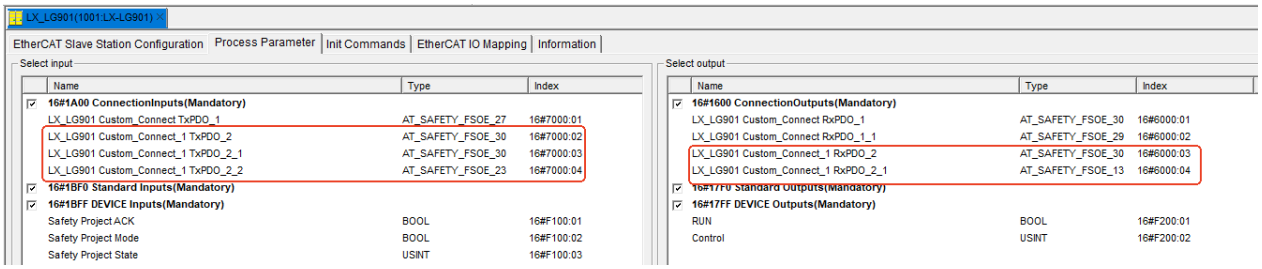
□ Click "Export" to save the configured input/output data locally (.sfd file). This file can be directly imported when configuring the slave station LX-LG901 for communication in the future.

□ Click on the "EtherCAT IO Mapping" tab. The input/output data configured in the "Safety Data" tab is automatically added to this tab. Parameters with a white background can be modified according to actual needs to facilitate identification and management.

11.3.2.4.1.2 Configure Network Variables for Controller No. 2

(1) Return to the non-safety side interface of AutoThink No. 2, double-click on LX-LG901 No. 2 to open the configuration interface.

□ Click on the "Process Parameters" tab and check if the input/output parameters match the safety data configured in the safety side interface.



- Click on the "EtherCAT IO Mapping" tab. AutoThink automatically generates the mapping relationship between process parameters and I/O areas. Check if the process parameters corresponding to the safety data have been correctly mapped.

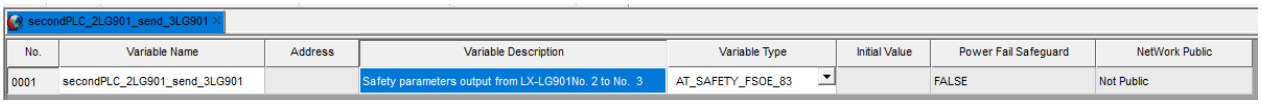
(2) In the navigation tree on the left, right-click "Network Variable" and select "Add Network Variable Group".In the "Add Network Variable Group" dialog box that appears, configure the following parameters and then click "OK".

GroupSet	
Enable	Check
Sender/Receiver	select "Sender"
GroupName	Enter the network variable group name, for example: secondPLC_2LG901_send_3LG901.
AssociateFile	Select the local .nvg file associated with the network variable group. If there is no associated file, you can manually enter the name, and the system will automatically create it locally , for example: secondPLC_2LG901_send_3LG901.
Export/Import	Select "Export" to export the configuration and variable list of the currently created network variable group to the associated file. Note: When selecting "Import," it indicates that the sender network variable group will be created based on the local associated file. After creation, editing is not supported.
NetworkSet	
NetworkType	Currently, only UDP transmission is supported.
BroadcastAddr Port	The destination IP address and port number for the UDP packet sent by the network variable sender default to 255.255.255.255:48040. When the target subnet is known, it is recommended to change the broadcast address to the subnet broadcast address of the specified segment (for example, when communicating with all controllers in the 192.168.0.x subnet through the P1 network interface, input 192.168.0.255 as the broadcast address). Note: If the broadcast address is in a different subnet from the controller's IP address, you need to configure routing information for the broadcast address in the "Network Configuration" tab under "Tool - Assistant Tool - Controller Operation" Otherwise, the network variable group may fail to send.
TransferSet	
AssociateTask	Network variable groups do not support association with Motiontask tasks.
Identifier	The identifier for the network variable group must be unique and cannot duplicate the identifiers of any existing network variable groups within the controller.
PackageVar	When checked, the network variable sender will send all variables within the group as a single package to minimize the number of packets sent and improve performance. When unchecked, the controller will generate and send a separate packet for each variable. It is recommended to check this option.
TransferCheck	It is recommended to check this option, which indicates that the sender provides a checksum for each packet, and the receiver only accepts packets with matching checksums.
Confirm	It is recommended to check this option, which indicates that the receiver sends an acknowledgment message to the sender for each received data packet.
CycleTransfer	When checked, the sender cyclically sends network variable data.
ChangeTransfer	When checked, the sender only sends variable data when the value of the variable changes.
EventTransfer	When checked, the sender only sends variable data when the specified variable's value is TRUE. This option only supports BOOL variables or BOOL member variables in complex data types.

(3) Double-click the network variable group created in the previous step to open the configuration interface. In the editing area, right-click the mouse and select "Add" to add a network variable.

(4) Select the variable type as "AT_SAFETY_FSOE". In the "Set Safety Type Length" dialog box that appears, select the Safety Data Output by No. 2 LX-LG901 to No. 3 LX-LG901 , which corresponds to the length of the input process parameters of No. 2 LX-CU501.

(5) Modify the variable name and variable description to help identify the purpose of the network variable.

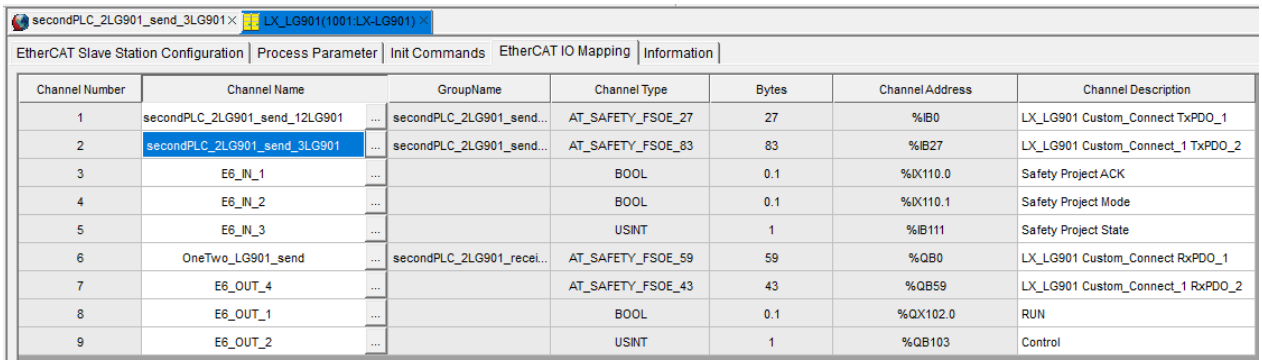

secondPLC_2LG901_send_3LG901

No.	Variable Name	Address	Variable Description	Variable Type	Initial Value	Power Fail Safeguard	NetWork Public
0001	secondPLC_2LG901_send_3LG901		Safety parameters output from LX-LG901No. 2 to No. 3	AT_SAFETY_FSOE_83		FALSE	Not Public

(6) Right-click the network variable group and select "Synchronize File" to ensure that the parameter information in the software matches the parameter information in the locally saved file.

(7) Double-click No. 2 LX-LG901 to open the configuration interface. Click the "EtherCAT IO Mapping" tab, and then click the corresponding channel for the related input process parameter  . Select "Associate IEC Variables," choose the network variable created above, and click "OK."

(8) After associating the IEC variable, the one-to-one correspondence between the network variables of Controller No. 2 (sent to Controller No. 3) - IEC variables (input process parameters) - safety data output by No. 2 LX-LG901.


secondPLC_2LG901_send_3LG901 | LX_LG901(1001:LX-LG901)

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	secondPLC_2LG901_send_12LG901	secondPLC_2LG901_send...	AT_SAFETY_FSOE_27	27	%IB0	LX_LG901 Custom_Connect TxPDO_1
2	secondPLC_2LG901_send_3LG901	secondPLC_2LG901_send...	AT_SAFETY_FSOE_83	83	%IB27	LX_LG901 Custom_Connect_1 TxPDO_2
3	E6_IN_1		BOOL	0.1	%IX110.0	Safety Project ACK
4	E6_IN_2		BOOL	0.1	%IX110.1	Safety Project Mode
5	E6_IN_3		USINT	1	%IB111	Safety Project State
6	OneTwo_LG901_send	secondPLC_2LG901_recei...	AT_SAFETY_FSOE_59	59	%QB0	LX_LG901 Custom_Connect RxPDO_1
7	E6_OUT_4		AT_SAFETY_FSOE_43	43	%QB59	LX_LG901 Custom_Connect_1 RxPDO_2
8	E6_OUT_1		BOOL	0.1	%QX102.0	RUN
9	E6_OUT_2		USINT	1	%QB103	Control

11.3.2.4.2 Create Network Variables for Receiving on Controller No. 3

11.3.2.4.2.1 Configure Custom Connection for No. 3 LX-LG901 Slave Station

(1) Right-click the No. 3 LX-LG901 and select “Open Safety Project”.

(2) In the safety-side AutoThink interface that pops up, right-click “FSoE Configuration”—“LX_LG901,” and select “Add”. In the “Add” dialog box that appears, add 1 custom connection. After adding the custom connections, LX-LG901 now has two custom connections, both as slave stations, communicating with No. 1.1 LX-LG901 and No. 2 LX-LG901, respectively. Double-click the custom connection added in the previous step to open the configuration window.

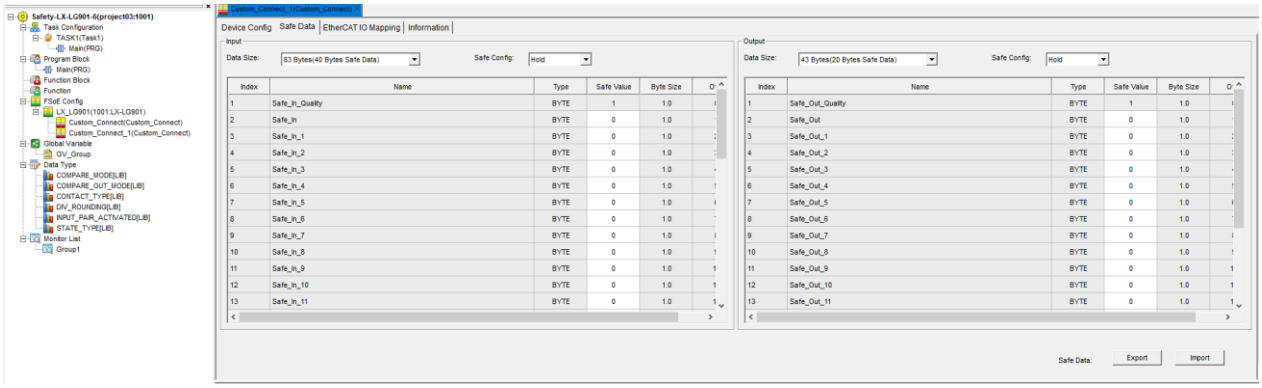
□ Click the “Device Configuration” tab and configure the following parameters.

Base Info	
FSoEAddr	Represents the FSoE address of the custom connection. The value range is 1 to 65535. During configuration, it must not conflict with any existing FSoE addresses in the entire project.
Connection Setting	
Conn-ID	Configurable when the master/slave mode is set to "FSoE Master." It represents the ID of the custom connection. The value range is 1 to 65535. The system automatically assigns a unique and valid connection ID, so no modification is required.

	This parameter is not configurable when the Master/Slave mode is set to "FSoE Slave."
Mode	This indicates the master-slave mode of the connection. Since this LX-LG901 communicates with 1.1 LX-LG901 as a slave through the current custom connection, "FSoE Slave" is selected.
Watchdog(ms)	Represents the watchdog time for the connection. The default value is 300, and the value range is 1 to 65535. This parameter is not configurable when the Master/Slave mode is set to "FSoE Slave." Watchdog timeout indicates that the FSoE message reception interval at the slave station is greater than the watchdog's timeout time.
Safe Param	
Length(Byte)	Configure the safety parameter length for communication with the master station, the value range is 1 to 128.

□ Click the “Safety Data” tab, select “Import,” aaccording to the interface prompts, import the .sfd file saved locally for the network variables created for sending from Controller No. 2.

- Note: The input/output data length can also be manually selected here. When configuring manually, note that the master station's output is the slave station's input. The master station's input is the slave station's output. The length and type of data must match; otherwise, normal communication will not be possible.

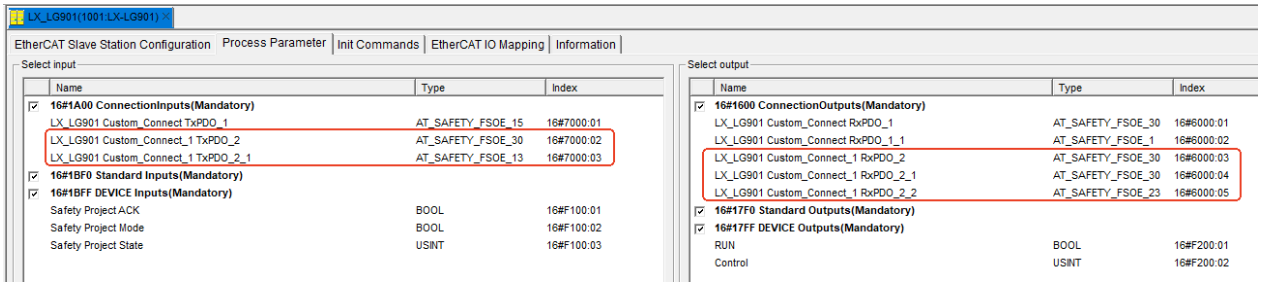


□ Click the “EtherCAT IO Mapping” tab. The input/output data configured in the “Safety Data” tab will be automatically added to this tab. Parameters with a white background can be modified according to actual needs for easier identification and management.

11.3.2.4.2.2 Configure the Receive Network Variables for Controller No. 3

(1) Return to the non-safety side interface of AutoThink No. 3, double-click on No. 3 LX-LG901 to open the configuration interface.

□ Click on the "Process Parameters" tab, check if the input/output parameters match the safety data configured in the safety side interface.



□ Click the “EtherCAT IO Mapping” tab. AutoThink automatically generates the mapping relationship between process parameters and I/O areas. Check if the process parameters

corresponding to the safety data have been mapped correctly.

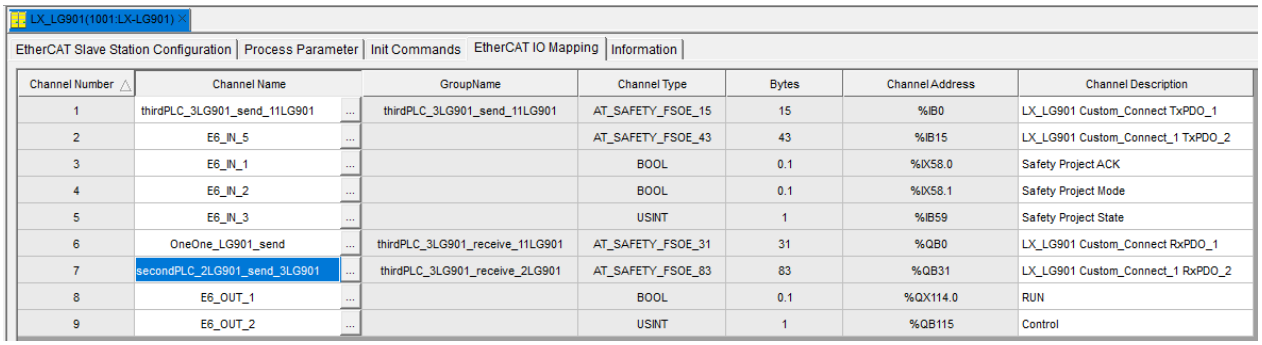
(2) In the navigation tree on the left, right-click “Network Variables,” select “Add Network Variable Group,” and in the “Add Network Variable Group” dialog box that appears, configure the following parameters, then click “OK.”

GroupSet	
Enable	Check
Sender/Receiver	select "Receiver"
GroupName	Enter the network variable group name, for example: thirdPLC_3LG901_receive_2LG901.
AssociateFile	Select the associated file from Configure Network Variables for Controller No.2 (secondPLC_2LG901_send_3LG901.nvg)
AssociateTask	The task associated with the network variable group does not support selecting the Motiontask task.

The system automatically imports the variable information from the associated file, and the imported information does not support additions, deletions, or modifications.

(3) Double-click on No. 3 LX-LG901 to open the configuration interface. Click the “EtherCAT IO Mapping” tab, then click on the channel corresponding to the relevant output process parameter  , and select “Associate IEC Variables.” Choose the network variable created above, and click “OK”.

(4) After associating the IEC variable, the one-to-one correspondence between the network variables of Controller No. 3 (as the receiver from Controller No. 2) - IEC variables (output process parameters) - Safety data of No. 3 LX-LG901 (input) .



Channel Number	Channel Name	GroupName	Channel Type	Bytes	Channel Address	Channel Description
1	thirdPLC_3LG901_send_11LG901	thirdPLC_3LG901_send_11LG901	AT_SAFETY_FSOE_15	15	%IB0	LX_LG901 Custom_Connect TxPDO_1
2	E6_IN_5		AT_SAFETY_FSOE_43	43	%IB15	LX_LG901 Custom_Connect_1 TxPDO_2
3	E6_IN_1		BOOL	0.1	%IX58.0	Safety Project ACK
4	E6_IN_2		BOOL	0.1	%IX58.1	Safety Project Mode
5	E6_IN_3		USINT	1	%IB59	Safety Project State
6	OneOne_LG901_send	thirdPLC_3LG901_receive_11LG901	AT_SAFETY_FSOE_31	31	%QB0	LX_LG901 Custom_Connect RxPDO_1
7	secondPLC_2LG901_send_3LG901	thirdPLC_3LG901_receive_2LG901	AT_SAFETY_FSOE_83	83	%QB31	LX_LG901 Custom_Connect_1 RxPDO_2
8	E6_OUT_1		BOOL	0.1	%QX114.0	RUN
9	E6_OUT_2		USINT	1	%QB115	Control

11.3.2.4.3 Create Network Variables for Sending from Controller No. 3

(1) In the navigation tree on the left side of AutoThink No. 3, right-click on "Network Variables" and select "Add Network Variable Group".

(2) In the "Add Network Variable Group" dialog box, configure the following parameters and click "OK". After adding, Controller No. 3 will have four network variable groups, two for communication with Controller No. 1, two for communication with Controller No. 2. Please ensure you distinguish between them.

GroupSet	
Enable	Check
Sender/Receiver	select "Sender"
GroupName	Enter the network variable group name, for example: thirdPLC_3LG901_send_2LG901.
AssociateFile	Select the local .nvg file associated with the network variable group. If there is no associated file, you can manually enter the name, and the system will automatically create it locally , for example: thirdPLC_3LG901_send_2LG901.

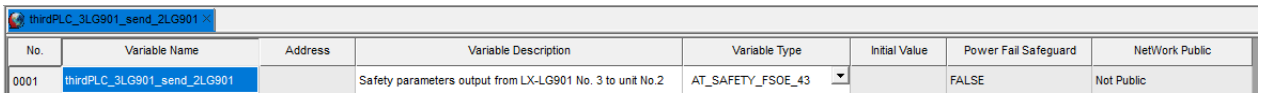
Export/Import	Select "Export" to export the configuration and variable list of the currently created network variable group to the associated file.
NetworkSet	
NetworkType	Currently, only UDP transmission is supported.
BroadcastAddr	The destination IP address and port number for the UDP packet sent by the network variable sender default to 255.255.255.255:48040. When the target subnet is known, it is recommended to change the broadcast address to the subnet broadcast address of the specified segment (for example, when communicating with all controllers in the 192.168.0.x subnet through the P1 network interface, input 192.168.0.255 as the broadcast address). Note: If the broadcast address is in a different subnet from the controller's IP address, you need to configure routing information for the broadcast address in the "Network Configuration" tab under "Tool - Assistant Tool - Controller Operation" Otherwise, the network variable group may fail to send.
Port	
TransferSet	
AssociateTask	Network variable groups do not support association with Motiontask tasks.
Identifier	The identifier for the network variable group must be unique and cannot duplicate the identifiers of any existing network variable groups within the controller.
PackageVar	When checked, the network variable sender will send all variables within the group as a single package to minimize the number of packets sent and improve performance. When unchecked, the controller will generate and send a separate packet for each variable. It is recommended to check this option.
TransferCheck	It is recommended to check this option, which indicates that the sender provides a checksum for each packet, and the receiver only accepts packets with matching checksums.
Confirm	It is recommended to check this option, which indicates that the receiver sends an acknowledgment message to the sender for each received data packet.
CycleTransfer	When checked, the sender cyclically sends network variable data.
ChangeTransfer	When checked, the sender only sends variable data when the value of the variable changes.
EventTransfer	When checked, the sender only sends variable data when the specified variable's value is TRUE. This option only supports BOOL variables or BOOL member variables in complex data types.

(3) Double-click the sender network variable group created in the previous step to open the configuration interface.

(4) In the editing area, right-click the mouse and select "Add Variable" to add a network variable.

(5) Select the variable type as "AT_SAFETY_FSOE". In the "Set Safety Type Length" dialog box that appears, select the safety data output from No. 3 LX-LG901 to No. 2 LX-LG901 , which corresponds to the input process parameter length of No. 3 LX-CU501.

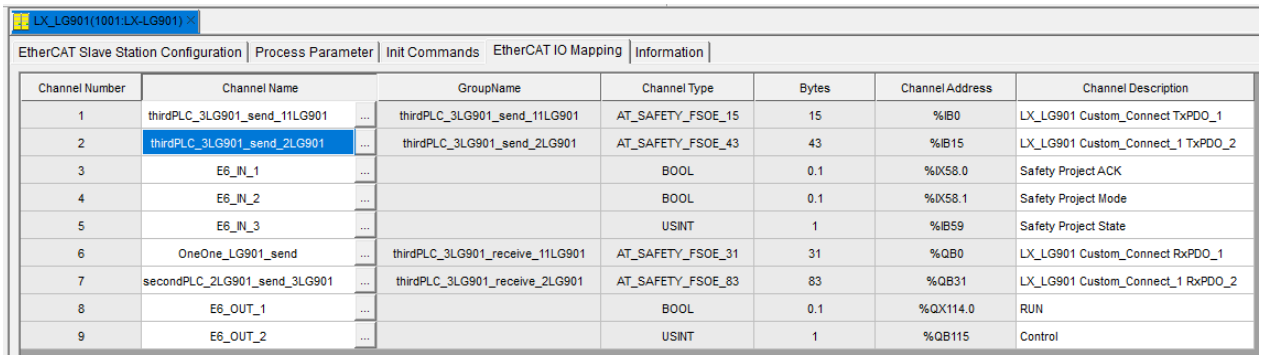
(6) Modify the variable name and variable description to help identify the purpose of the network variable.



(7) Right-click the network variable group and select "Synchronize Files" to ensure that the parameter information in the software matches the parameter information in the locally saved file.

(8) Double-click No. 3 LX-LG901 to open the configuration interface, and click the "EtherCAT IO Mapping" tab. Click the channel corresponding to the relevant input process parameter  , and select "Associate IEC Variables." Choose the network variable created above and click "OK."

(9) After associating the IEC variable, the one-to-one correspondence between the network variables of Controller No. 3 (as the sender to Controller No. 2) - IEC variables (input process parameters) - Safety data of No. 3 LX-LG901 (output) .


LX_LG901(1001:LX-LG901)

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	thirdPLC_3LG901_send_11LG901	thirdPLC_3LG901_send_11LG901	AT_SAFETY_FSOE_15	15	%IB0	LX_LG901 Custom_Connect TxPDO_1
2	thirdPLC_3LG901_send_2LG901	thirdPLC_3LG901_send_2LG901	AT_SAFETY_FSOE_43	43	%IB15	LX_LG901 Custom_Connect_1 TxPDO_2
3	E6_IN_1		BOOL	0.1	%IX58.0	Safety Project ACK
4	E6_IN_2		BOOL	0.1	%IX58.1	Safety Project Mode
5	E6_IN_3		USINT	1	%IB59	Safety Project State
6	OneOne_LG901_send	thirdPLC_3LG901_receive_11LG901	AT_SAFETY_FSOE_31	31	%QB0	LX_LG901 Custom_Connect RxPDO_1
7	secondPLC_2LG901_send_3LG901	thirdPLC_3LG901_receive_2LG901	AT_SAFETY_FSOE_83	83	%QB31	LX_LG901 Custom_Connect_1 RxPDO_2
8	E6_OUT_1		BOOL	0.1	%QX114.0	RUN
9	E6_OUT_2		USINT	1	%QB115	Control

11.3.2.4.4 Create a network variable group for receiving on Controller No. 2

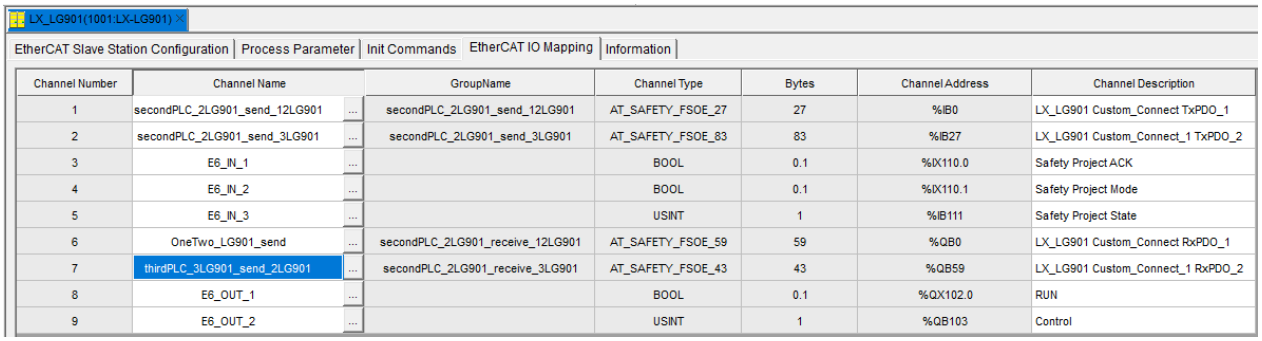
(1) In the navigation tree on the left side of AutoThink No. 2, right-click on "Network Variables" and select "Add Network Variable Group". In the "Add Network Variable Group" Dialog Box, configure the following parameters and click "OK". After adding, Controller No. 2 will have four network variable group, two for communication with Controller No. 1, two for communication with Controller No. 3. Please ensure you distinguish between them.

GroupSet	
Enable	Check
Sender/Receiver	select "Receiver"
GroupName	Enter the network variable group name, for example: secondPLC_2LG901_receive_3LG901.
AssociateFile	Select the associated file from Create Network Variables for Sending from Controller No. 3 (thirdPLC_3LG901_send_2LG901.nvg)
AssociateTask	The task associated with the network variable group does not support selecting the Motiontask task.

The system automatically imports the variable information from the associated file, and the imported information cannot be modified, added, or deleted.

(2) Double-click No. 2 LX-LG901 to open the configuration interface, and click the "EtherCAT IO Mapping" tab. Click the channel corresponding to the relevant output process parameter  , and select "Associate IEC Variable." Choose the receiver network variable created above and click "OK".

(3) After associating the IEC variable, the one-to-one correspondence between the network variables of Controller No. 2 (as the receiver from Controller No. 3) - IEC variables (output process parameters) - safety data of No. 2 LX-LG901 (input) .


LX_LG901(1001:LX-LG901)

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	secondPLC_2LG901_send_12LG901	secondPLC_2LG901_send_12LG901	AT_SAFETY_FSOE_27	27	%IB0	LX_LG901 Custom_Connect TxPDO_1
2	secondPLC_2LG901_send_3LG901	secondPLC_2LG901_send_3LG901	AT_SAFETY_FSOE_83	83	%IB27	LX_LG901 Custom_Connect_1 TxPDO_2
3	E6_IN_1		BOOL	0.1	%IX110.0	Safety Project ACK
4	E6_IN_2		BOOL	0.1	%IX110.1	Safety Project Mode
5	E6_IN_3		USINT	1	%IB111	Safety Project State
6	OneTwo_LG901_send	secondPLC_2LG901_receive_12LG901	AT_SAFETY_FSOE_59	59	%QB0	LX_LG901 Custom_Connect RxPDO_1
7	thirdPLC_3LG901_send_2LG901	secondPLC_2LG901_receive_3LG901	AT_SAFETY_FSOE_43	43	%QB59	LX_LG901 Custom_Connect_1 RxPDO_2
8	E6_OUT_1		BOOL	0.1	%QX102.0	RUN
9	E6_OUT_2		USINT	1	%QB103	Control

11.4 Safety Program Writing

11.4.1 Task Configuration

The safety logic module LG901 supports only one task. The task properties are shown in the following figure:

- Task Name

  The task name can be user-defined.

- Estimated Runtime

  After the safety program has passed the compilation, the software generates an estimated runtime for the task based on the compilation results.

- Task Cycle

  Users can set the task cycle according to the estimated runtime, with a setting range of 1-1,000 ms.

- Run Type

  The safety tasks can be set to run periodically and at full speed. Periodic running means that the task runs according to the set task cycle time, while full-speed running means that the controller executes the task according to the maximum performance.

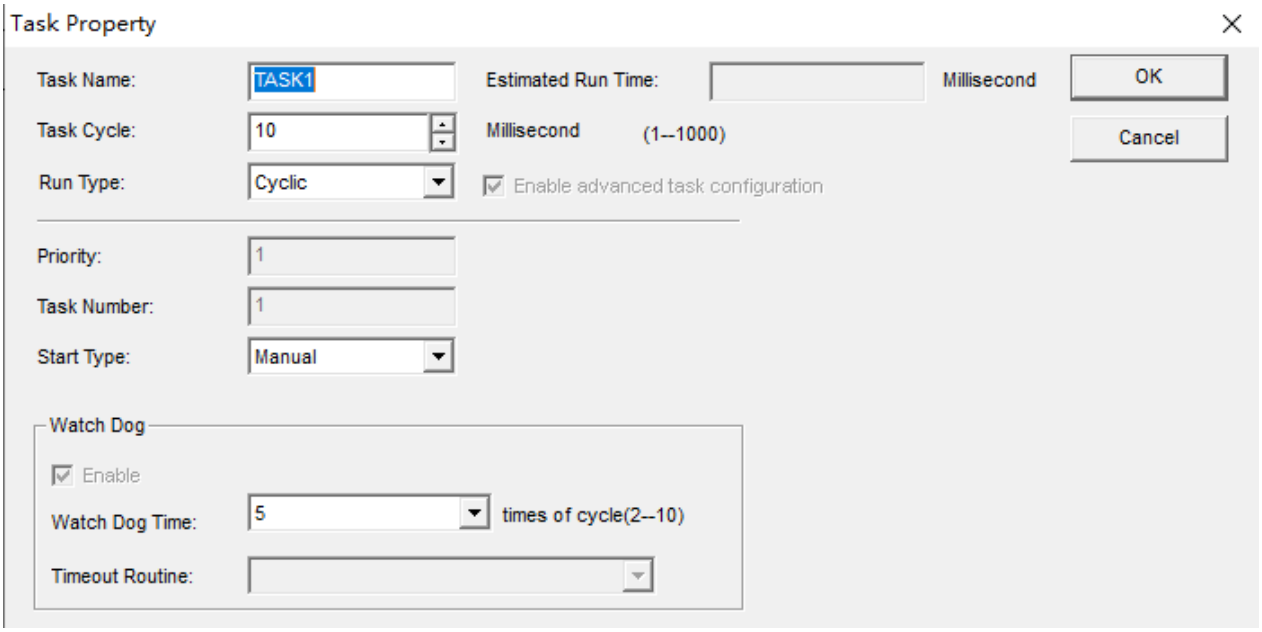
- Start-up Type

  The start-up type is divided into Manual and Automatic. For manual start-up, users need to click the Run icon in the toolbar to execute the task; And for automatic start-up, after the automatic start-up module is powered on, the task is automatically executed.

  Attention: After the full compilation and download of non security projects, security tasks will stop. Regardless of whether the security task startup type is manual or automatic, it is necessary to enter debugging mode and click the "Run" icon on the toolbar to run the task.

- Watchdog

  Users can configure the watchdog time by setting the value to an integer multiple of the task cycle.



Task Properties

11.4.2 Variable Declaration

There are two types of variables in a safety program. One is a safety variable, which is colored yellow, and the other is a non-safety variable, which is colored gray. The declaration of these two variables is shown in the figure.

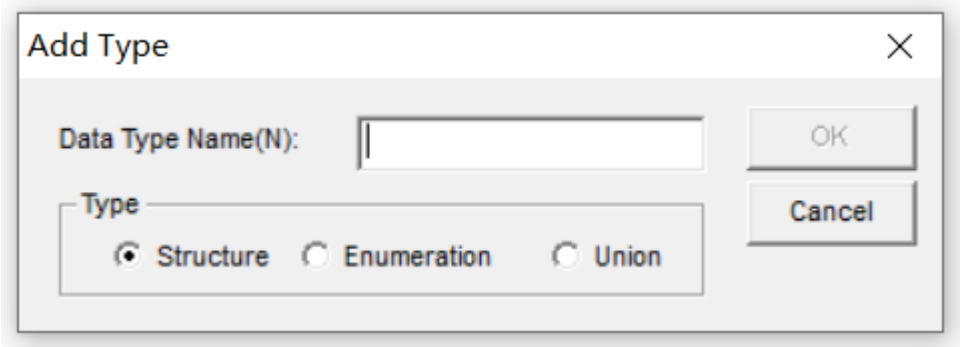
No.	Variable Name	Variable Description	Variable Type	Initial Value	Safety
0001	p1		BOOL	FALSE	TRUE
0002	p2		BOOL	FALSE	TRUE
0003	p3		BOOL	FALSE	FALSE
0004	p4		BOOL	FALSE	FALSE

Declaration of Two Types of Variables

The safety properties of a variable can be selected as either a safety or non-safety variable using the drop-down menu. The variables in a safety project do not support properties such as association with direct address, constant, network disclosure, and power failure protection, compared to those in a non-safety project. Local and global variables in a safety project are declared in the same way as in a non-safety project.

11.4.3 Data Type

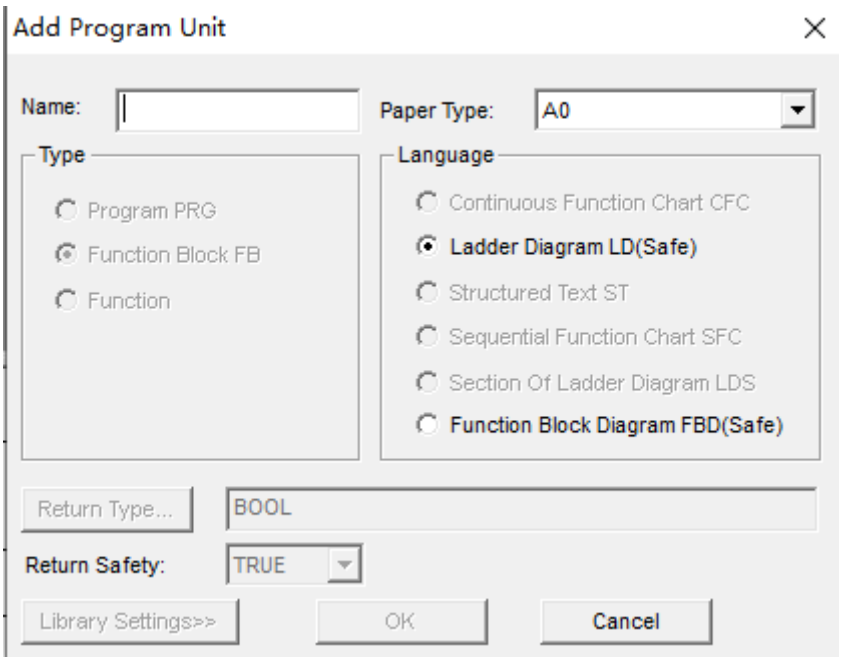
The secure side supports basic data types and three custom data types. Basic data types include ARRAY, BOOL, BYTE, DATE, DINT, DT, DWORD, INT, LREAL, REAL, SINT, STRING, TIME, TOD, UDINT, UINT, USINT, WORD, etc. Custom data types support Structure, Enumeration, and Union.



Custom Data Types

11.4.4 Program Block

The program block on the secure side supports two languages, LD and FBD, as shown in Figure 12. Both languages are written in the same way as the non-safety project. See Non-Safety Program Writing for details.

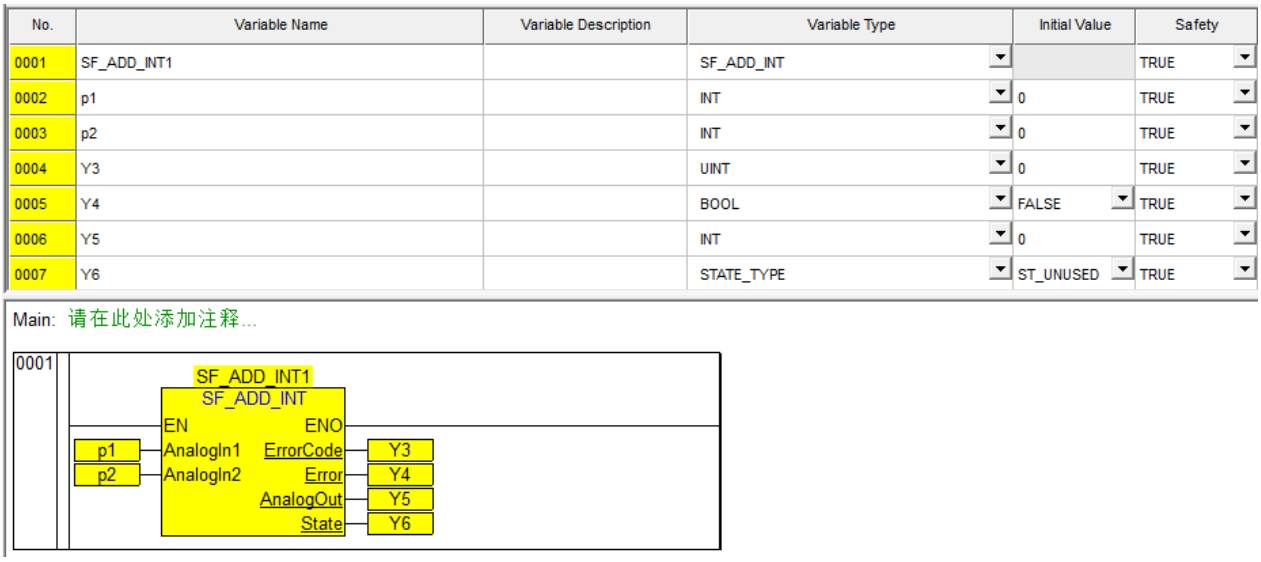


Add Program Block

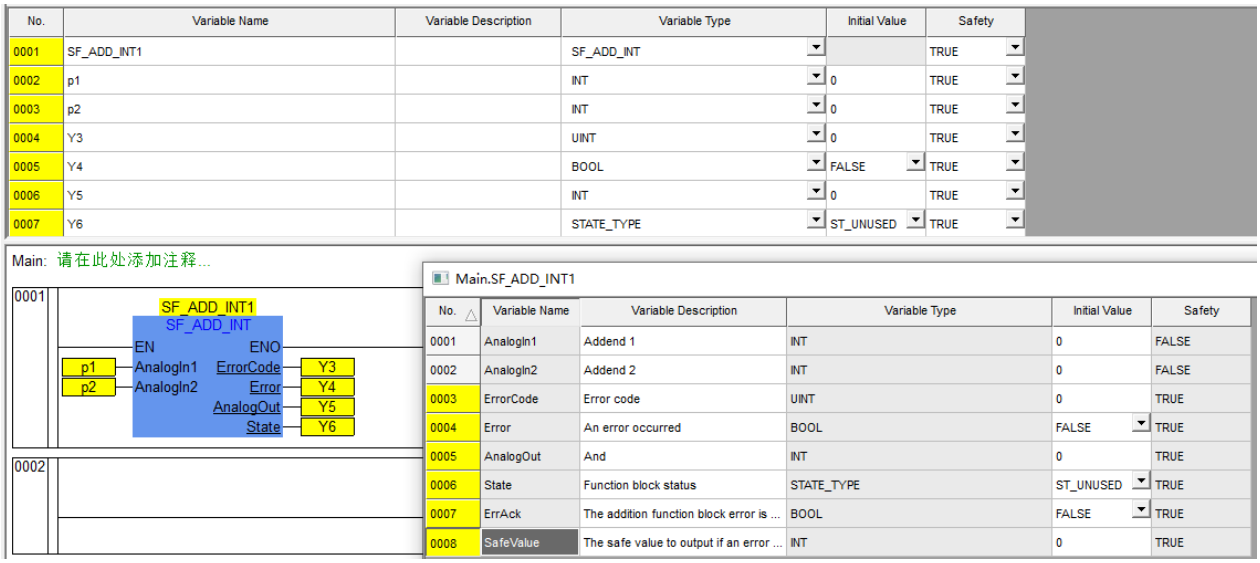
11.4.5 Functional Block

The functional block supports both LD and FBD languages. Users can write customized functional blocks as needed. In addition, the Library Manager provides safety libraries, including mathematical operations, logical operations, and basic libraries. Let's take the SF_ADD_INT functional block as an example. For the pin variables that are not underlined below, users can associate either safety or non-safety variables, such as the AnalogIn1 and AnalogIn2 pins in Figure 13; And for the pin variables that are underlined below, users can only associate safety variables, such as the ErrorCode, Error, AnalogOut, and State pins in

Figure 13. When the safety properties of the pins are not consistent, the compilation reports an error. Users can double-click a functional block to view its pin properties.



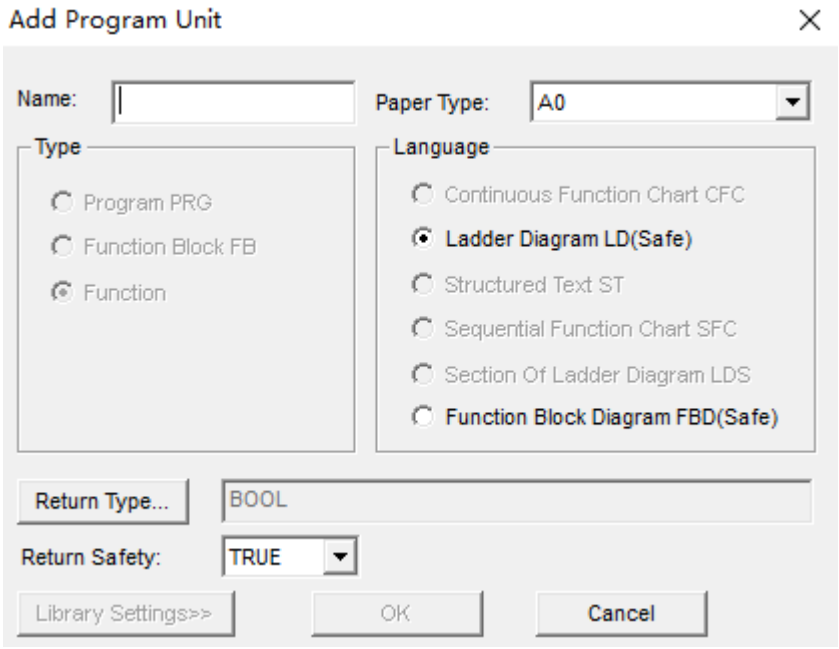
Functional Block



Properties of Functional Block Pins

11.4.6 Function

The function supports both LD and FBD languages. The return value type supports BOOL, BYTE, DATE, DINT, DT, DWORD, INT, LREAL, REAL, SINT, STRING, TIME, TOD, UDINT, UINT, USINT, WORD. And the Return Value Safety Property can be selected to be TRUE (secure) or FALSE (non-secure).

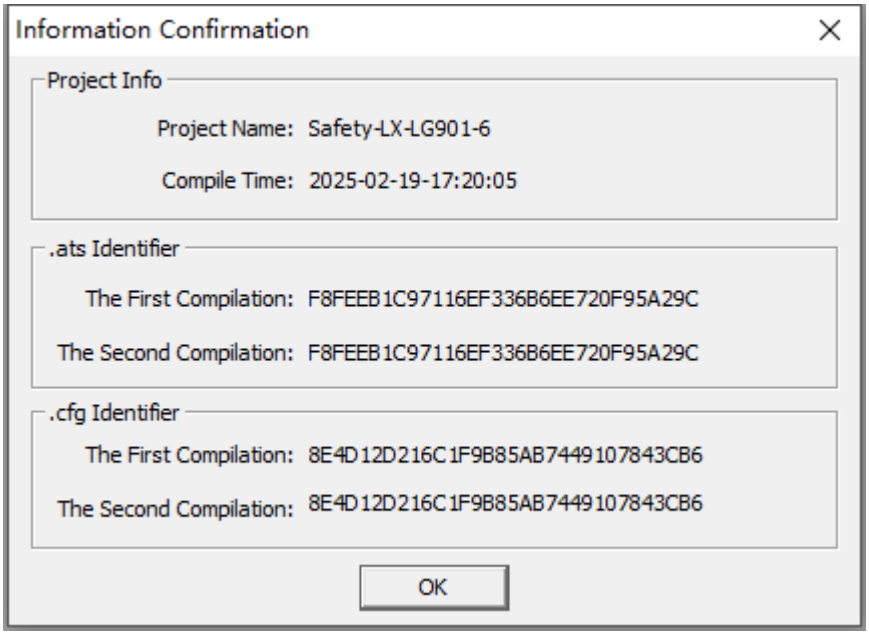


Function

11.5 Secure Compilation and Installation Commissioning

11.5.1 Compile

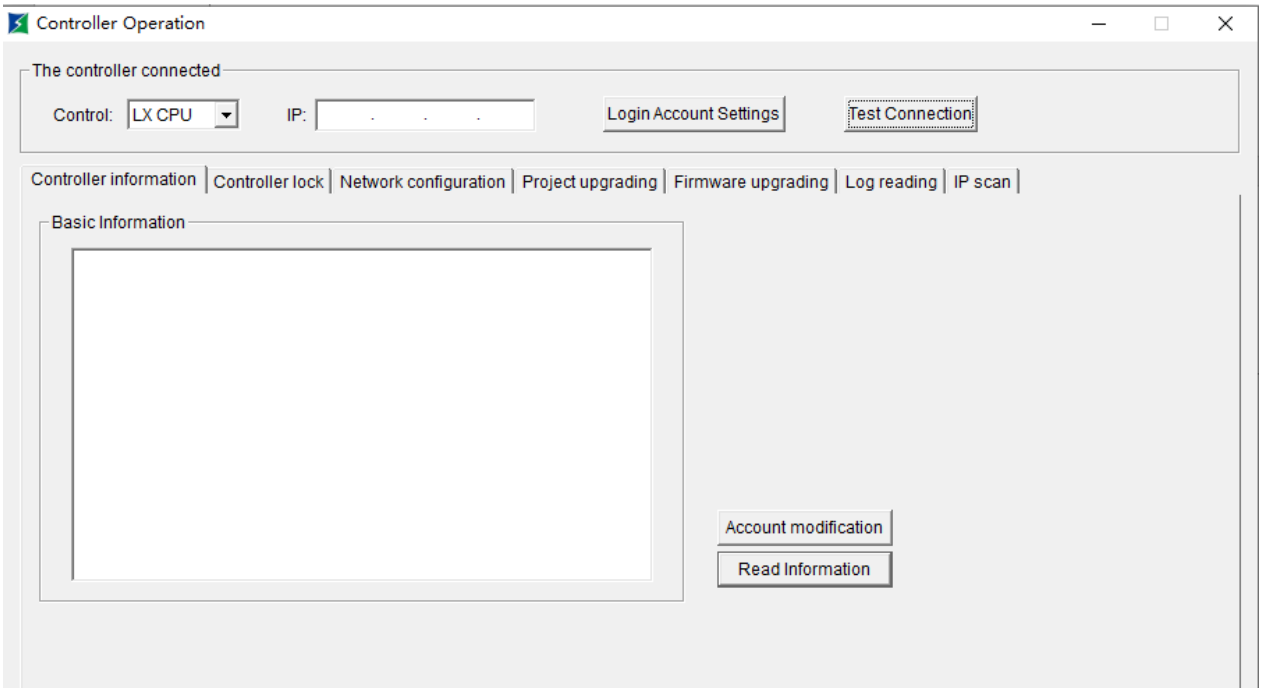
The safety program is compiled twice. The MD5 values generated by the two compilations are strictly consistent; Otherwise, a compilation error is reported.



Secure Compilation

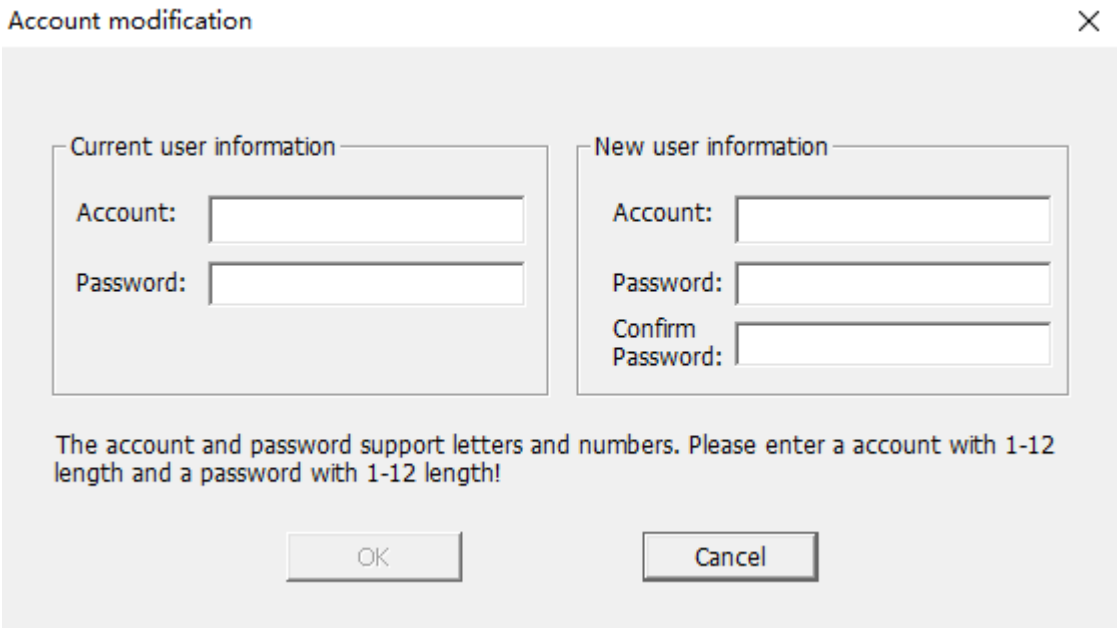
11.5.2 Download

Since the LG901 logic module is a slave station for EtherCAT on the non-secure side, the LG901 module shall be online before the safety project is installed. For the first installation, users can install the non-safety project first, and wait for the module to be online before installing the safety project. Before the first installation of the new LG901 module, it is necessary to set the user password first. The password setting shall be consistent with that on the non-secure side. It can be set with the auxiliary tool for controller operation. Users can select the controller type as "LXS CPU", and configure the "IP Address" and the "EtherCAT Slave Address of LG901".



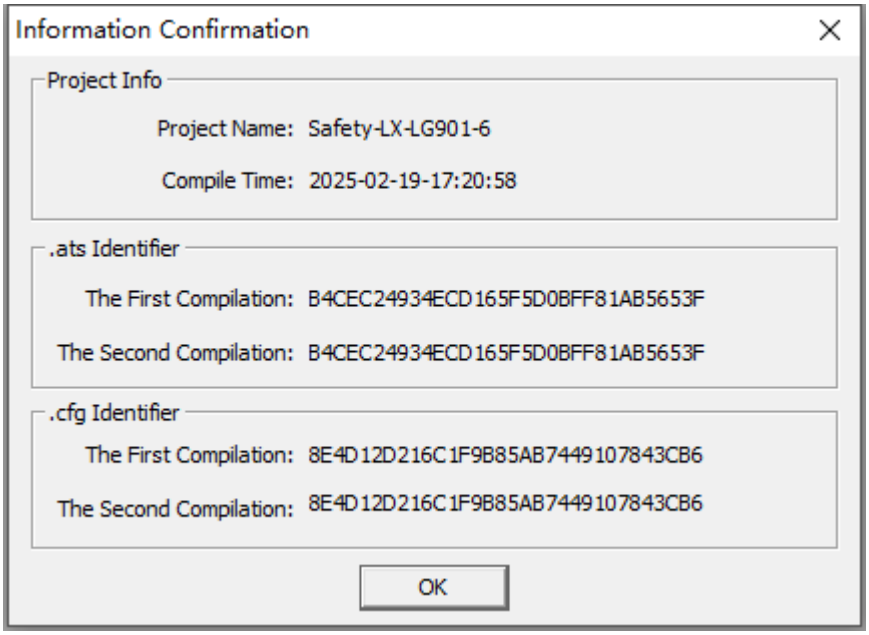
Controller Operation

The safety project supports read-only and read/write permissions. 1-12 digits in length are supported for the username and password.



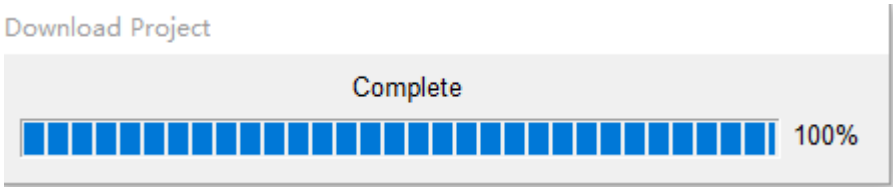
Password Setting

When the safety project is installed, the Information Confirmation window is displayed. It shows the MD5 value of the controller to be installed and the MD5 value of the local project, which can be installed only after users' confirmation.



Information Confirmation Window

After the information is confirmed, a progress bar for the project installation is displayed.



11.5.3 Debug

In the EtherCAT IO Mapping of the LG901 module on the non-secure side, there are 5 status words and control words. Their feedback and control are explained below:

- RUN: Start-up control for tasks in the safety project at power-on. After the LX-CU500 is powered on, and RUN is TRUE, the safety tasks can only run normally after the tasks on the secure side are run manually or run automatically.
- Safety Project ACK: Status feedback for task start-up on the secure side. TRUE: Start; FALSE: Stop.
- Safety Project Mode: Status feedback for online commissioning mode on the secure side. TRUE: Online commissioning mode; FALSE: Offline commissioning mode.
- Safety Project State: Running status feedback for tasks on the secure side. 0: Offline status; 1: Task running; 2: Task stopped.
- Control: Reserved.

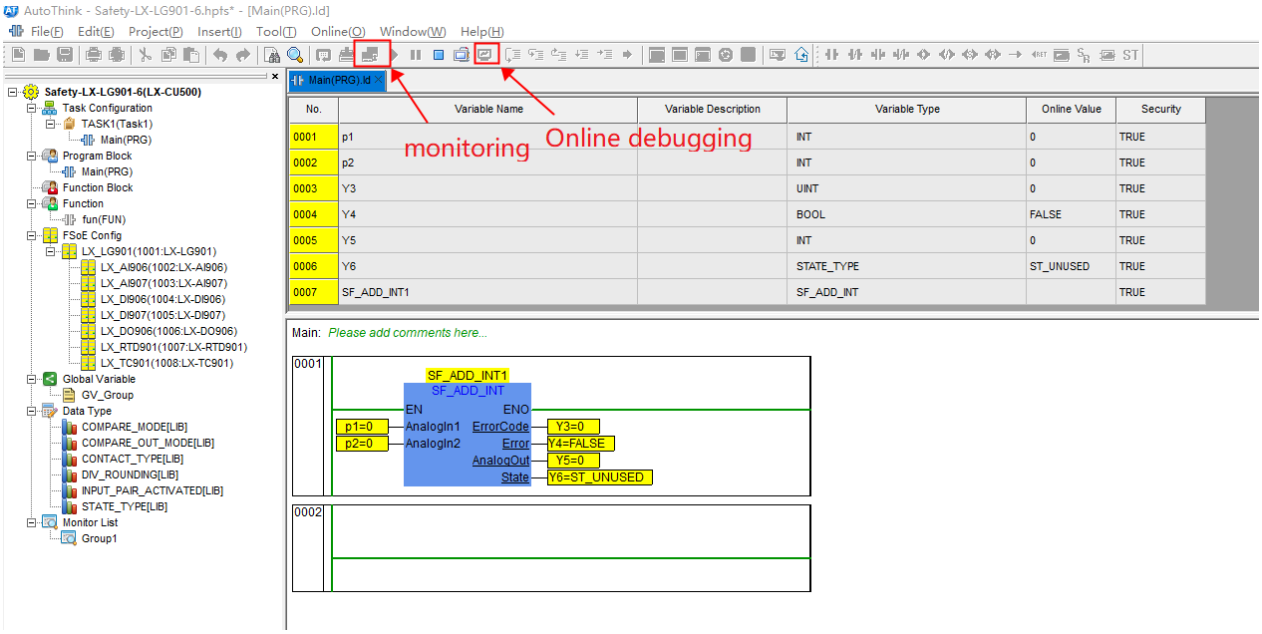
LX_LG901(1001:LX-LG901)

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	E7_IN_1		BOOL	0.1	%IX44.0	Safety Project ACK
2	E7_IN_2		BOOL	0.1	%IX44.1	Safety Project Mode
3	E7_IN_3		USINT	1	%IB45	Safety Project State
4	E7_OUT_1		BOOL	0.1	%QX44.0	RUN
5	E7_OUT_2		USINT	1	%QB45	Control

Safety Control Word

In a safety project, click the monitoring window in the toolbar, or press the shortcut key ALT + F8 to enter the online state, as shown in the figure. If commissioning is required after entering the online state, it is necessary to enter the online commissioning mode. After entering the online commissioning mode, users can perform operations such as starting and stopping tasks, writing variables, etc.



Online Debug

Chapter 12 Compilation and Installation

12.1 Compile

Before you install the project, you need to compile it, check for syntax errors, and generate a binary file that can be executed by the controller. After compilation, a .at file is generated under the installation path for project installation on the controller.

According to the compilation scope, two compilation modes are provided: full compilation and incremental compilation.

- Full compilation: The whole project is compiled to generate new configuration logic information and hardware configuration information. It is triggered only when compilation is performed for the first time, full compilation is manually performed, or the hardware configuration is modified.
- Incremental compilation: Only the modified and new content is compiled to generate incremental compilation information. After the first compilation, incremental compilation will be triggered by other project modifications except for hardware configuration modification.

12.1.1 Full Compilation

12.1.1.1 Introduction

Full compilation is triggered only when compilation is performed for the first time or full compilation is performed manually for the project. Full compilation will cause initial installation by AutoThink. Please operate with caution.

During a full compilation, the hardware configuration information is re-generated, all variables are re-assigned, and POUs are re-translated. New configuration logic information and hardware configuration information will be generated. Full compilation will be triggered in the following situations:

- Compilation is performed for the first time after a project is created.
- The .iec or .tmp file is incomplete or damaged during compilation.
- Switching to the target configuration to perform compilation.
- Compilation after switching the CPU model.
- The project file of an earlier version is opened, causing the TrgVersion file to be upgraded.
- The .iec or .tmp file is incomplete when the project is installed on the controller.
- The Full Compilation command is executed directly.

- Compilation after changing the hardware configuration.

The full compilation should be confirmed when it is triggered in other situations except for the first compilation after project creation.

The deleted variables of the G and R areas are not released when variables are modified repeatedly. After full compilation, the memory space occupied by the variables can be reduced.

12.1.1.2 Requirement

The project has been modified or configured.

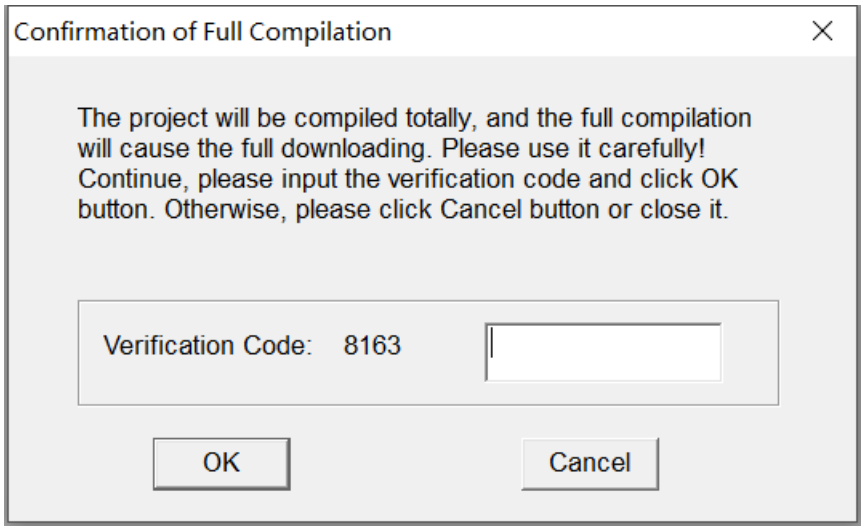
12.1.1.3 Steps

Follow the steps below to perform a full compilation:

(1) In the Project menu, select the Full Compilation command.

(2) A full compilation is performed even if the Compilation command is selected for the first project compilation.

(3) The window for confirming the full compilation will pop up when the Full Compilation command is executed, as shown in the figure below.



Full Compilation Confirmation

(4) Enter the correct verification code in the pop-up window, and click OK to perform full compilation.

12.1.2 Incremental Compilation

12.1.2.1 Introduction

After the first compilation of the project, incremental compilation will be triggered if the project configuration information is modified.

Incremental compilation only applies to modified and new content.

Incremental compilation will be triggered in any of the following situations:

- POU modification: Add or delete POUs, and modify the logic of existing POUs.
- Variable modification: Add or delete variable definitions, and modify existing variable definitions.
- Task property modification: Add called POUs, and change task property values.

The following table lists related files after a successful compilation:

File Extension	Example	Description	Format
.hpf	Project01.hpf	Controller algorithm project file	Binary file
.tmp	Project01.tmp	File containing the information about compilation and installation after incremental compilation	Binary file
.iec	Project01.iec	IEC operation logic file	Binary file
.at	Project01.at	File containing the logical configuration information generated after compilation	Binary file
.sim	Project01.sim	File containing the logical configuration information generated after compilation during simulation	Binary file

12.1.2.2 Requirement

The project has been modified or configured.

12.1.2.3 Steps

Follow the steps below to perform an incremental compilation:

In the Project menu, select the Compile command.

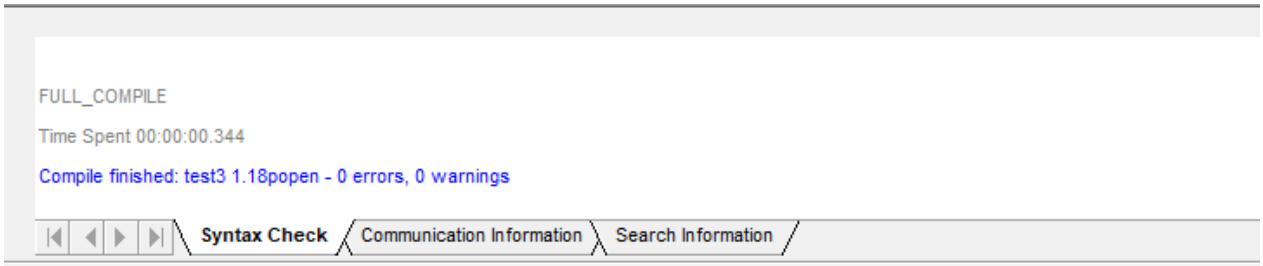
12.1.2.4 Result

"ADD_COMPILE" displayed in the Syntax Check window indicates that the compilation is completed. Compilation errors and warnings are also reported.

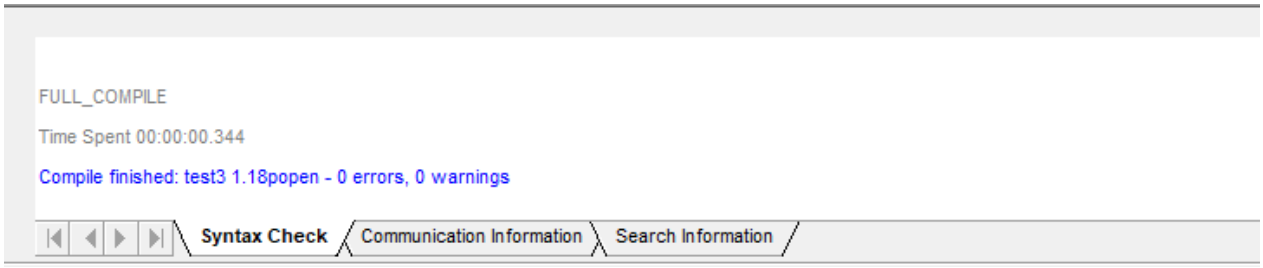
12.1.3 Compilation Result

After the compilation is completed, the results will be displayed in the information window below AutoThink. The result can be:

- The compilation reports no error, as shown in (a) in the figure below, which indicates that the project has no issue.
- Errors or warnings are reported during the compilation, as shown in (b) in the figure below. This indicates that the project has some issues. Error information is displayed in red. You can double-click the error to locate the error and make modifications.



(a)



(b)

Compilation Result

12.1.4 Compilation Error Check

12.1.4.1 Introduction

On the Syntax Check tab of the information output window, you can check whether the compilation is successful and whether any compilation error is reported. If errors are reported, you can make modifications and compile again.

12.1.4.2 Requirement

Compilation has been performed.

12.1.4.3 Steps

Follow the steps below to check for compilation errors:

(1) On the Syntax Check tab, double-click the error information line. The cursor will then automatically locate the error in the program, and the corresponding element will be selected.

(2) Fix the error.

(3) Compile again.

12.2 Set Communication

12.2.1 Introduction

Before installation, communication between the local computer and the controller should be established. You need to set the communication parameters for the local computer and controller, respectively.

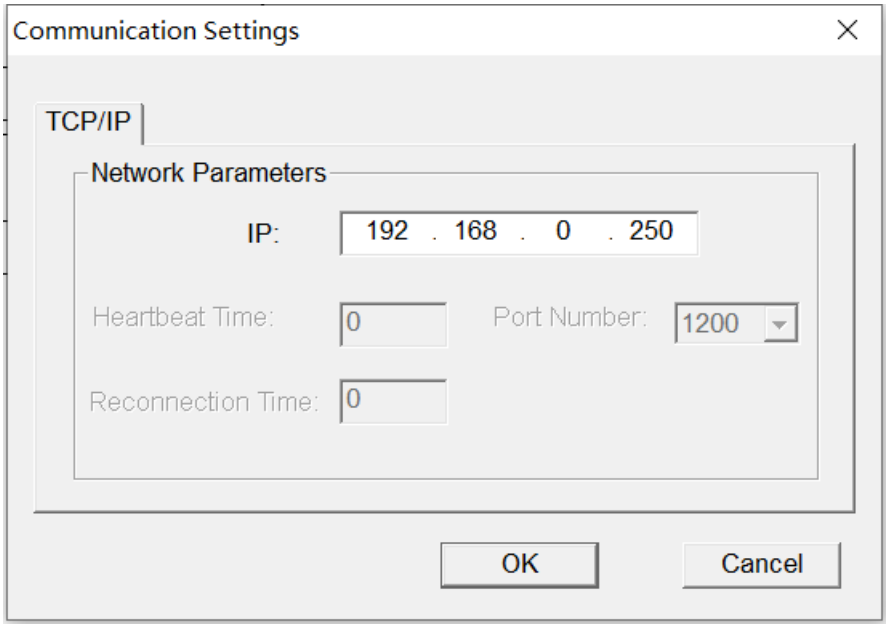
12.2.2 Requirement

The communication cable between the local computer and controller is connected properly.

12.2.3 Steps

Follow the steps below to set the controller communication parameters:

(1) In the Online menu bar, select Communication Settings to open the Communication Settings dialog box.



(2) Enter the IP address of the controller in the IP input box.

The default IP addresses of the two controller interfaces are 192.168.0.250 and 192.168.1.250, respectively.

Retain the default values of other parameters.

(3) Click OK to complete the controller communication parameter settings.

12.3 Download

12.3.1 Project Comparison before Download

12.3.1.1 Introduction

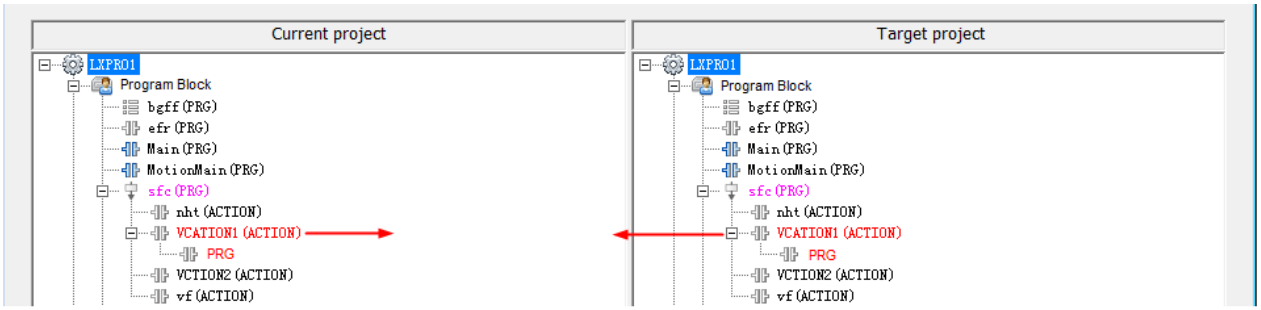
Before project download, AutoThink will read the user project in the controller, compare it with the currently opened user project, and list the POU and variable differences. Users can learn the modifications of the installed project, which ensures that the correct project is installed.

Project comparison involves comparisons of user programs, variables, and data types and can list the data differences between projects. Currently, only POUs using the ST and LD languages support detailed comparison.

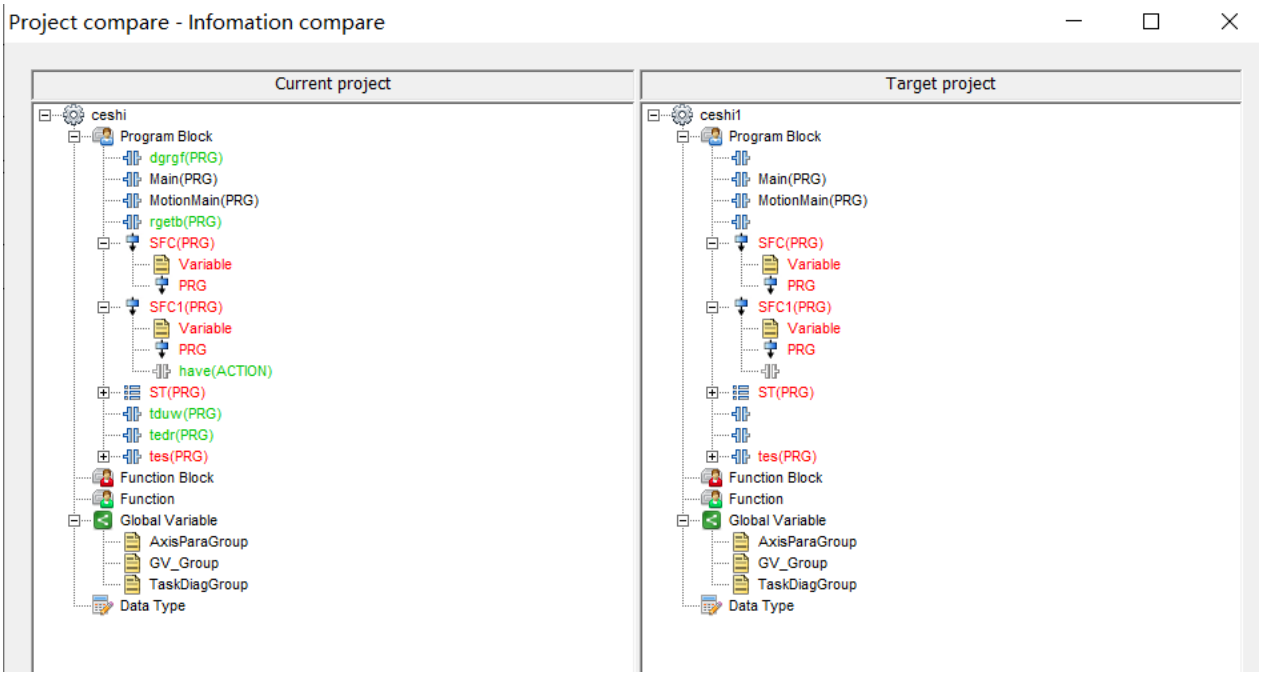
Difference data is displayed as Project Management Tree nodes. The tree nodes with differences are displayed in four different colors with the following meanings:

- Data in red: Indicates that the data in the same POU or variable group is different in the two projects.
- Data in green: Indicates that the data only exists in one of the two projects, and the data node does not exist in the other project.

- Data in purple: For SFC POUs only. Indicate that the subprograms under the same SFC program are different in the two projects, as shown in figure (a) below.
- Data in blue: For SFC POUs only. Indicate that the SFC programs in the two projects are different, and the subprograms under the SFC programs are also different, as shown in figure (b) below.



(a)



(b)

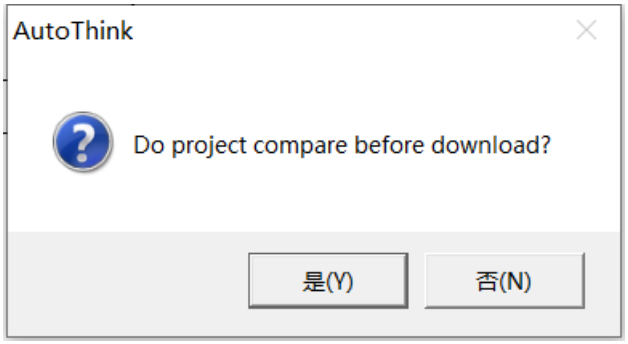
Compare Project

12.3.1.1.1 Requirement

- Compare Projects before download should be checked in the [Project] - Options dialog box.
- The user project in the current controller has been downloaded before.

12.3.1.1.2 Steps

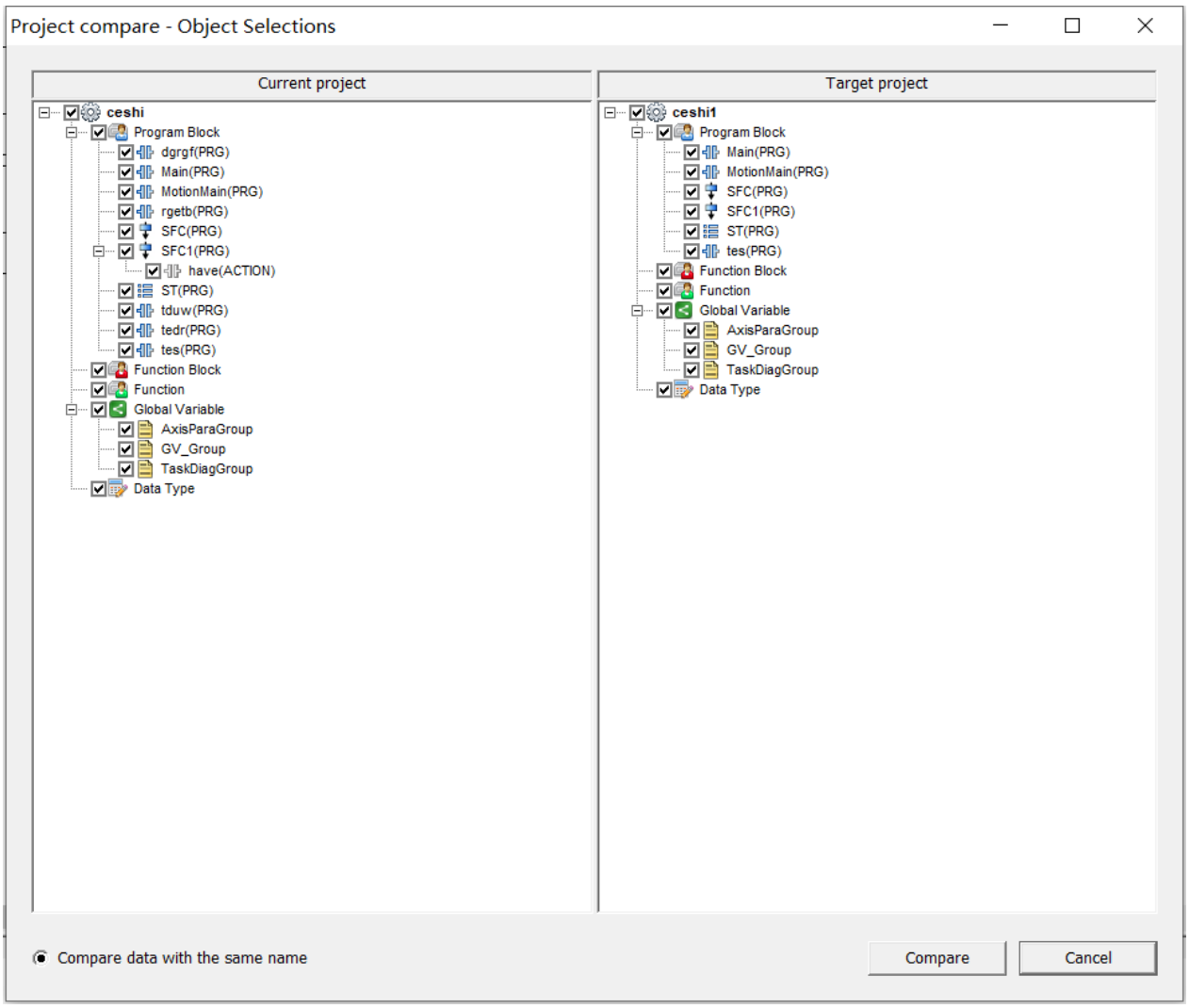
(1) Click Install on the toolbar to pop up the Project Comparison dialog box.



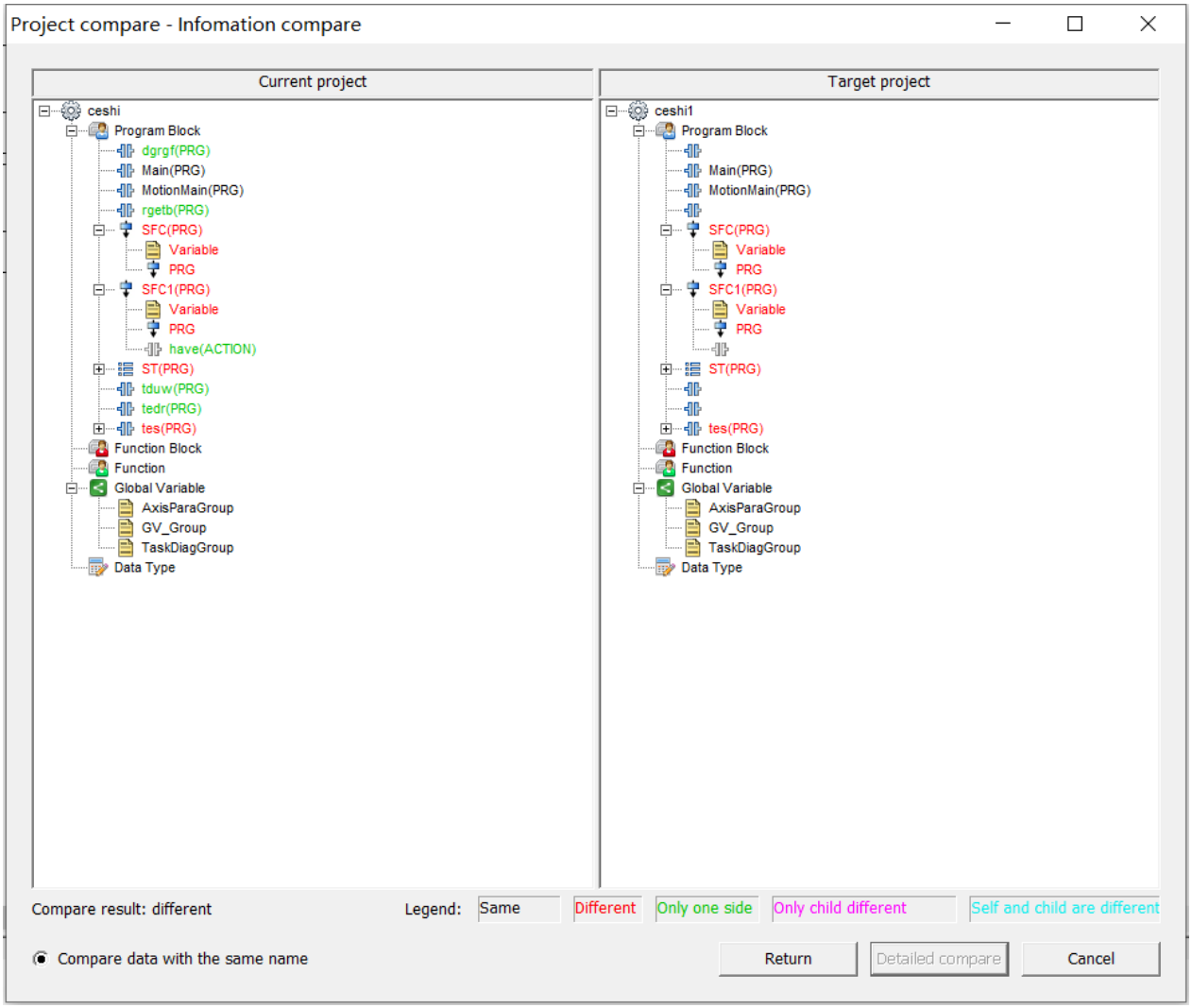
(2) Click Yes to pop up the controller login dialog box.

(3) Enter the controller account and password to log in to the controller. At this time, the user project file will be uploaded to the controller. After the information output window displays User project upload completed, the project account dialog box appears.

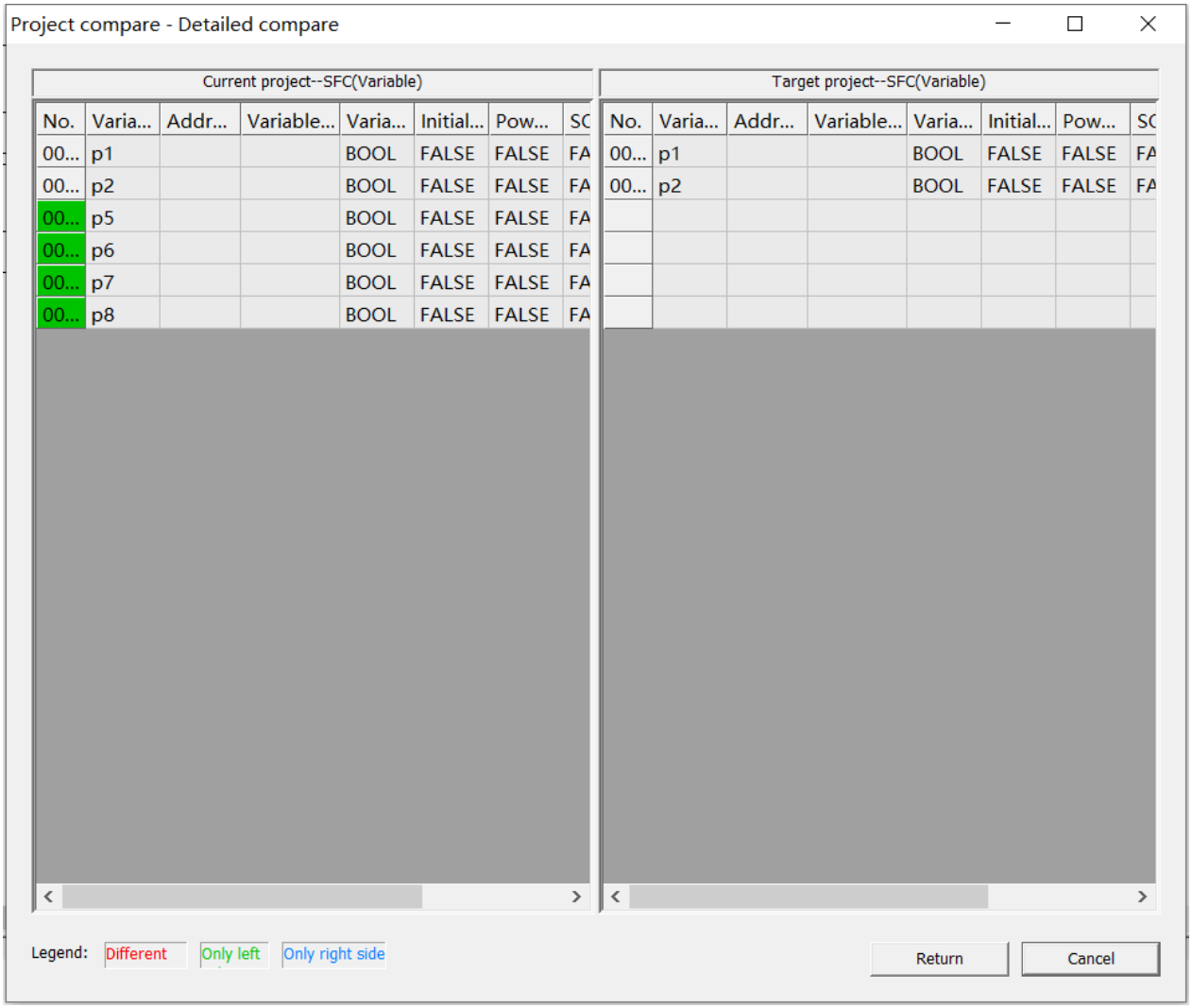
(4) Enter the project account and password to pop up the project comparison dialog box.



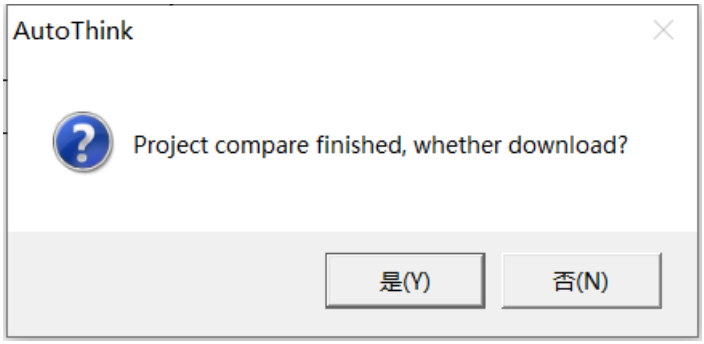
(5) Check the project nodes that need to be compared. Click Compare. The difference nodes of the two projects and the comparison result are displayed.



(6) Check the different nodes. Click Detailed Comparison to list the detailed differences.



(7) After you confirm that no project error exists, close the project comparison dialog box and continue the download.



12.3.2 Full Download

Introduction

Full download refers to initial download, which installs all configuration logic information and hardware

configuration information to the controller. During the download, the corresponding controller stops calculation and output. Initial download is triggered in any of the following situations:

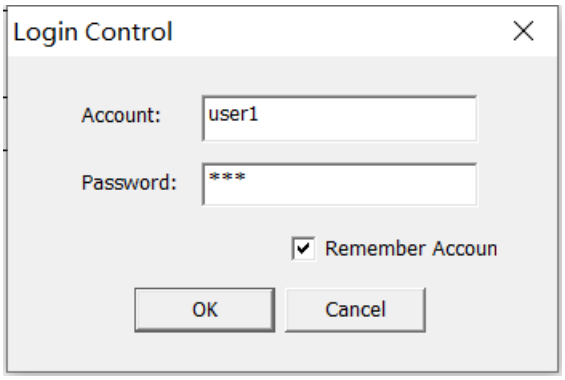
- The project is installed for the first time.
- The current project is inconsistent with the project in the controller, or the project versions are not successive.
- A full compilation is performed.

12.3.2.1 Requirement

The project has been compiled successfully.

12.3.2.2 Steps

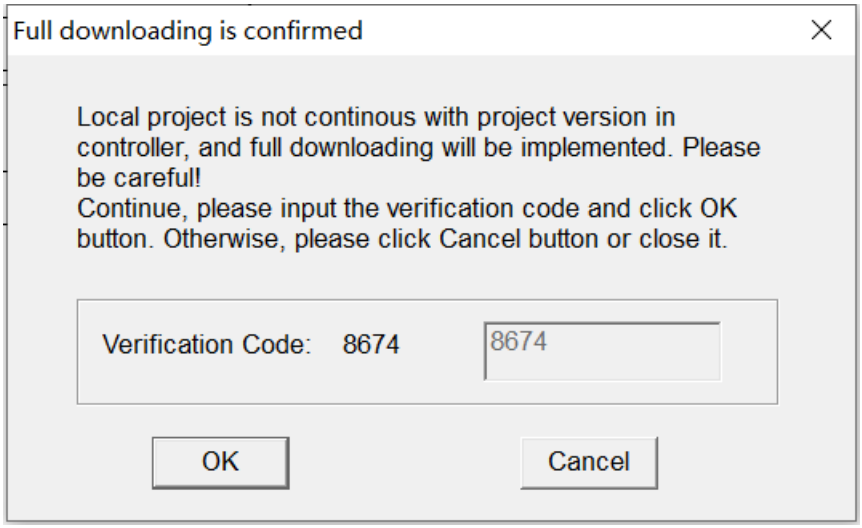
(1) In the menu bar, choose [Online] - [Download] to pop up the Log in to Controller dialog box.



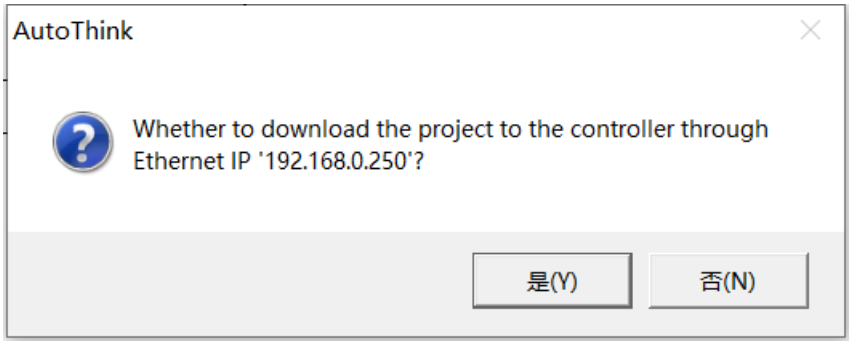
If Compare Projects before Download is checked in Project Options, the project comparison dialog box will appear. For details, see Project Comparison before Download.

If Auto Download User Project after Download is not checked in Project Options, the confirmation window of deleting user project files in the controller will appear after you click Install.

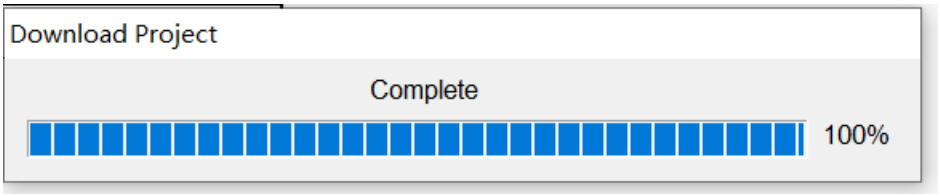
(2) Enter the controller account information and click OK to pop up the Initial Download Confirmation dialog box.



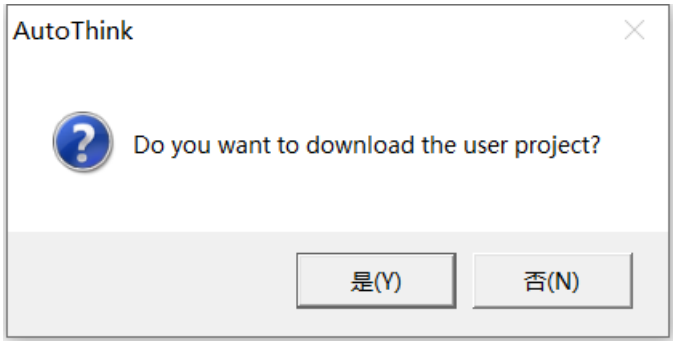
(3) Enter the correct verification code and click OK. An download prompt will appear, as shown in the figure below.



(4) Click Yes to start the download, as shown in the figure below.



If Auto Download User Project after Download is checked in Project Options, a prompt for downloading the user project will appear after the download, as shown in the figure below.



(5) Click Yes to back up the .hpf project source file to the controller. You can choose Online > Upload

User Project in the menu bar to read the project source file if needed.

12.3.3 Incremental Download

12.3.3.1 Introduction

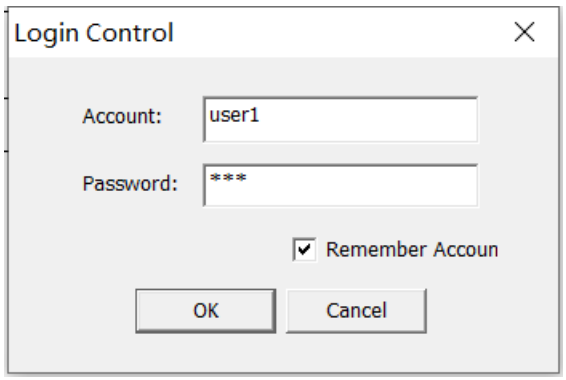
Incremental download is performed when the project is compiled and installed again after it is modified. During incremental download, tasks and algorithm operation are stopped, and the instruction cache is cleared. Variable addresses and values in the controller are not affected, and the axis parameter information of the algorithm module remains unchanged. Incremental initialization is performed on new variables.

12.3.3.2 Requirement

The project has been compiled successfully.

12.3.3.3 Steps

(1) Click Install to pop up the Log in to Controller dialog box.

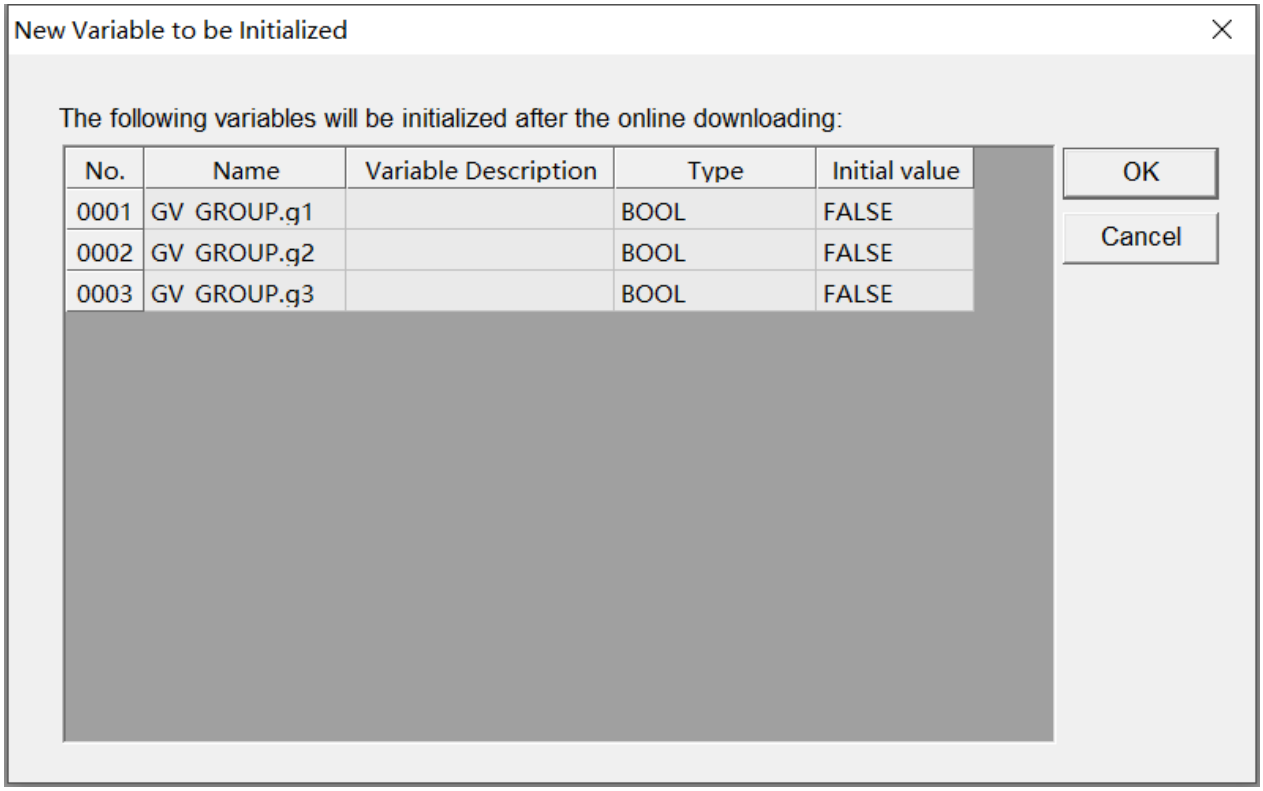


If Compare Projects before Download is checked in Project Options, the project comparison dialog box will appear. For details, see Project Comparison before Download.

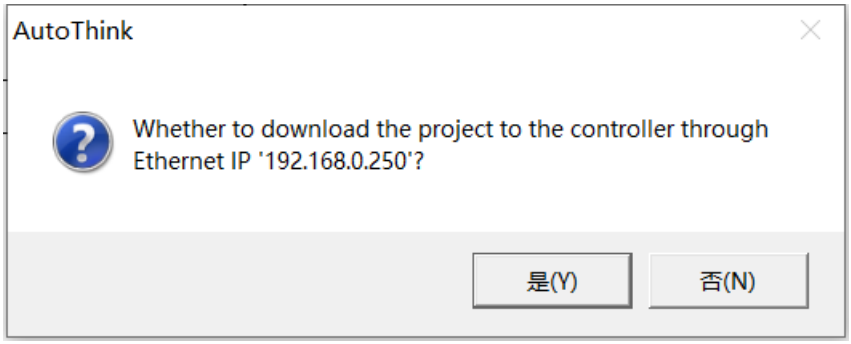
If Auto Download User Project after Download is not checked in Project Options, the confirmation window of deleting user project files in the controller will appear after you click Install.

(2) Enter the controller account and password and click OK.

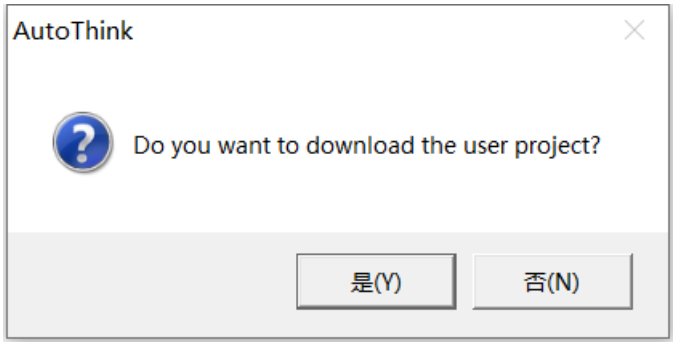
If new global variables are added, new variables are defined in PRG POUs, or the original variable types of such POUs are modified, the New Variables to Be Initialized dialog box will appear, as shown in the figure below.



(3) Click OK. After new variables are initialized, the download prompt will appear, as shown in the figure below.



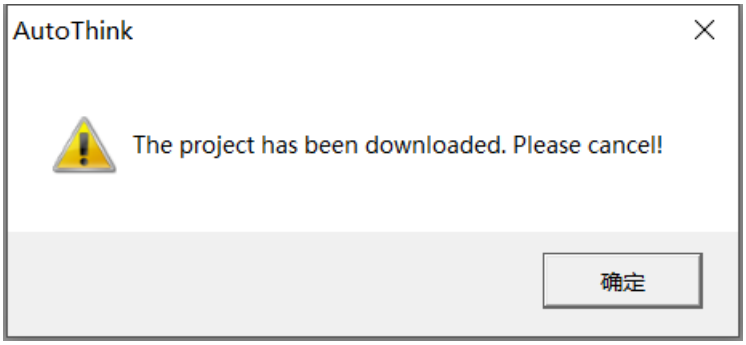
(4) Click Yes to start the download. After the project download is completed, a prompt for downloading the user project will appear, as shown in the figure below.



(5) Click Yes to back up the .hpf project source file to the controller.

You can choose [Online] - [Upload User Projec[t in the menu bar to read the project source file if needed.

If the project is not modified after the last download, a prompt will appear after you click Install, as shown in the figure below.



(6) Click OK to exit the download. A prompt message will be displayed in the Communication Information window.

After the download is completed, you can carry out online debugging to observe data changes.

12.3.4 Download/Upload User Project

12.3.4.1 Introduction

User project download will download the file (*.hpf) of the project opened in the controller from the computer of the current operator station to the CPU module with the corresponding station number for backup.

User project upload will upload the backed-up file (*.hpf) of the project opened in the controller from the CPU module to the specified path of the computer where the current operator station is located.

The two operations aim to copy the file (*.hpf) of the installed project to the corresponding CPU module for backup. When the file stored locally is lost or the file version consistency is uncertain, the consistency and accuracy of project files stored in a local path and the CPU module can be guaranteed.

12.3.4.2 Requirement

The project needs to be saved and compiled before it is downloaded.

12.3.4.3 Steps

Follow the steps below to download or upload a user project:

(1) Choose [Online] - [Download User Project/Upload User Project] in the menu to pop up the

corresponding dialog box.

12.3.5 Download User File

12.3.5.1 Introduction

Used to back up the user files to the controller SD card.

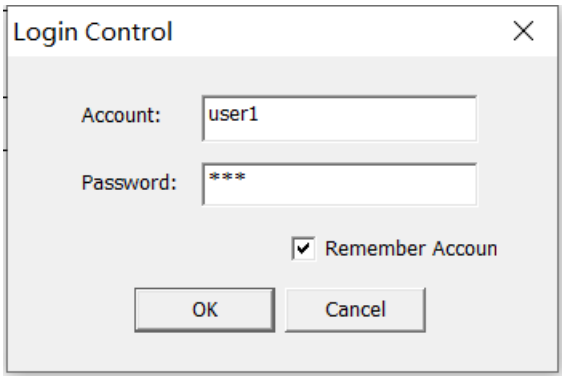
12.3.5.2 Requirement

- AutoThink is offline.
- Before you download the files, insert the SD card into the SD card slot of the controller.

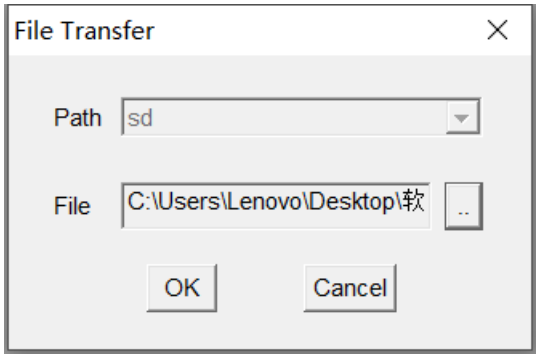
12.3.5.3 Steps

Follow the steps below to download user files:

(1) Choose [Tool] - [Download User Files] in the menu to pop up the Log in to Controller dialog box. The account information is set in Auxiliary [Tool] > [Controller Operation].



(2) Enter the controller account information and click OK to pop up the Download File dialog box. The path is the SD card by default.



(3) Click the .. button, select the files to be downloaded, and click OK.

The file names cannot contain Chinese characters. Otherwise, the files cannot be downloaded.

Start downloading.

12.3.6 Upload User File

12.3.6.1 Introduction

Upload the user files downloaded to the controller previously to a local path.

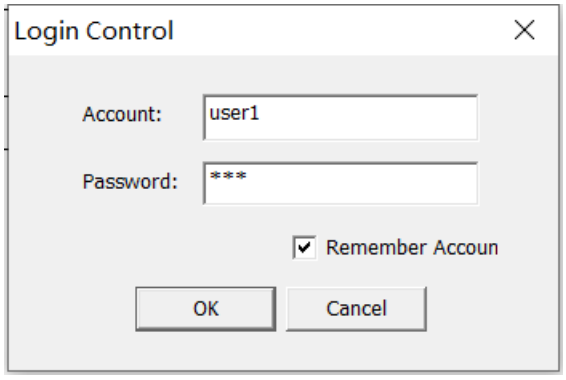
12.3.6.2 Requirement

- AutoThink is offline.
- Make sure the SD card is inserted into the controller and contains the files to be uploaded.

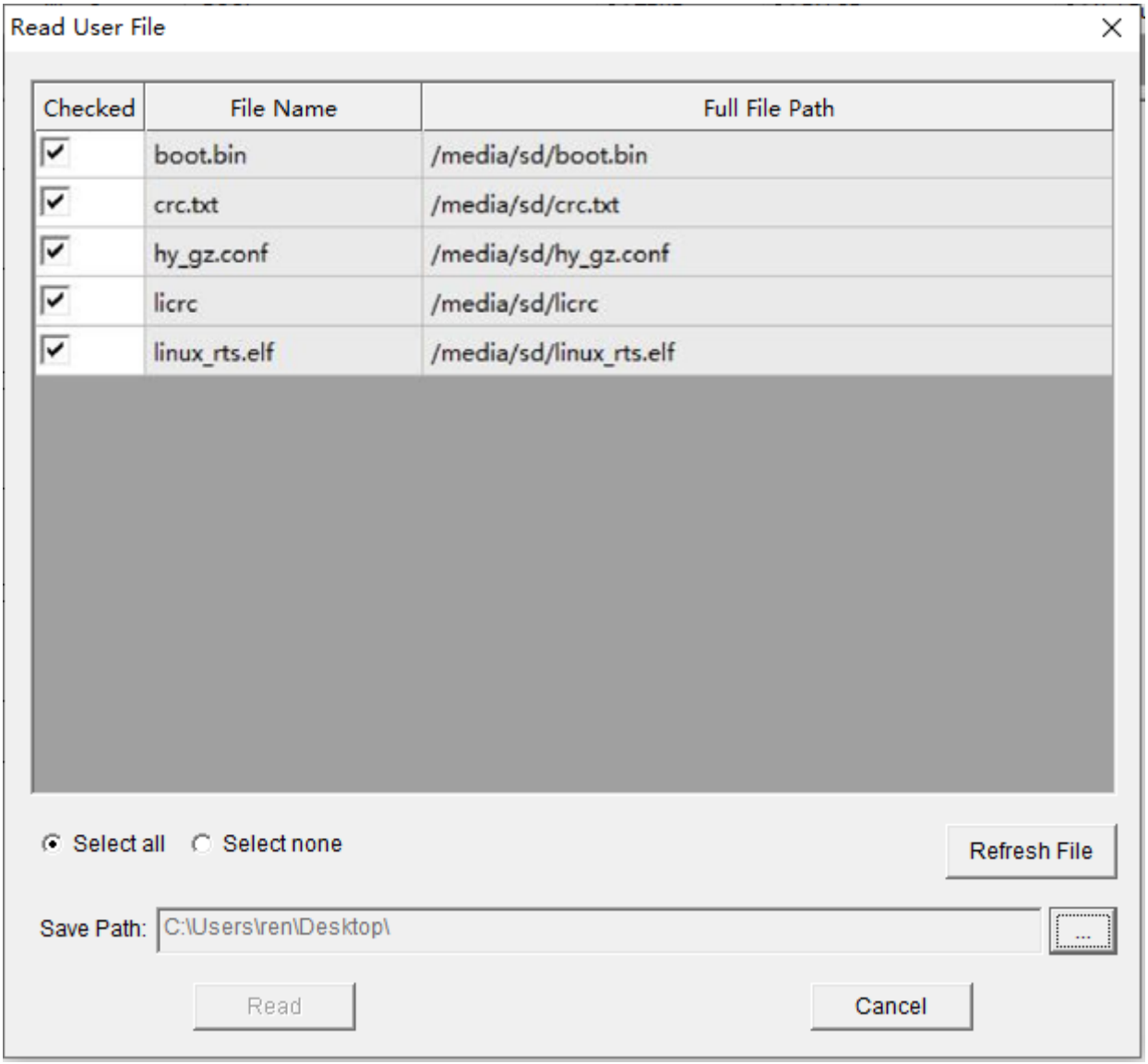
12.3.6.3 Steps

Follow the steps below to upload user files:

(1) Choose [Tool] - [Upload User File] in the menu to pop up the Log in to Controller dialog box.



(2) Enter the controller account information and click OK to pop up the Read User File dialog box.



(3) Click Refresh File to display all files that can be uploaded.

Make sure that the file names contain no Chinese characters. Otherwise, the files cannot be displayed properly after being uploaded.

(4) Select the files to be uploaded and the file upload path.

You can click the "..." button to select the path but not input the path manually.

- Select All: Select all files.
- Deselect All: Deselect all files.

(5) Click Read to start uploading.

Chapter 13 Debug and Run

13.1 Run

13.1.1 Introduction

The run instruction is used to start online or monitored task programs. After the software selects the run instruction, the PLC starts executing task programs, and the user can check the execution status of each task program and each program segment.

13.1.2 Requirement

The AT software is already in monitoring or simulation state.

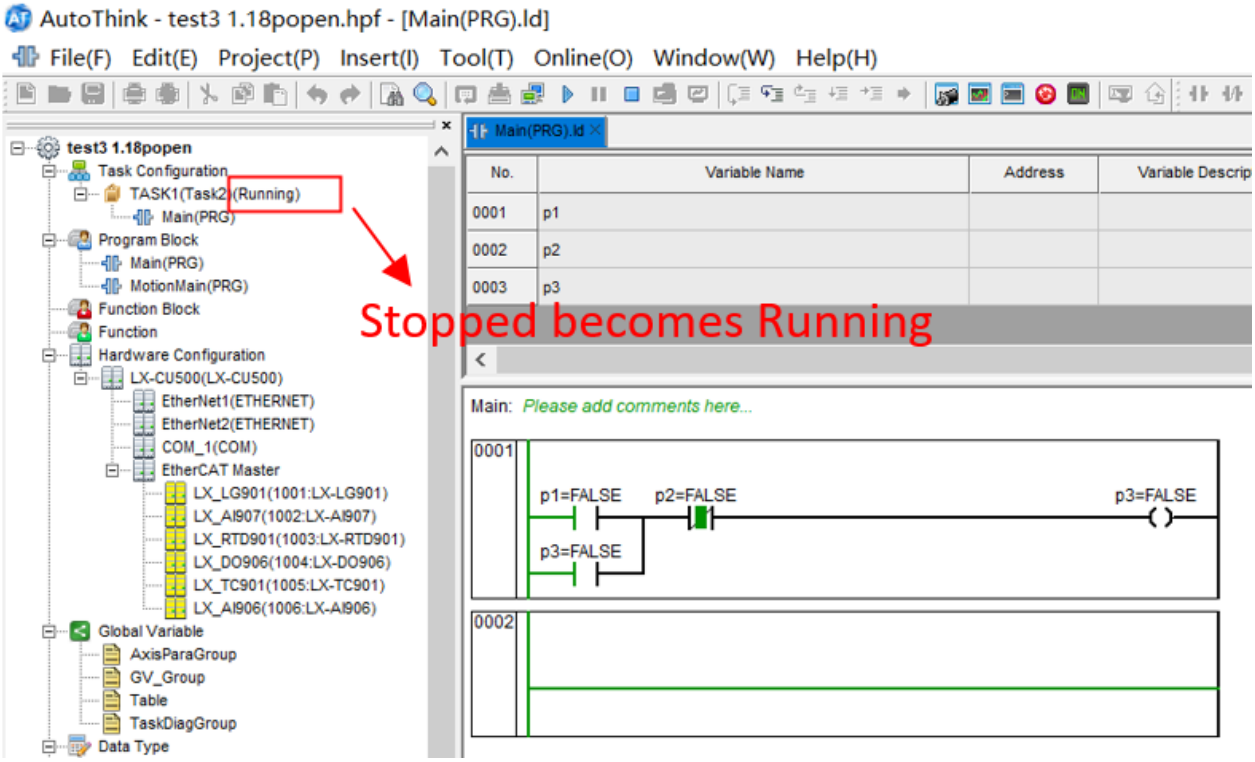
13.1.3 Steps

Follow the steps below to run the task program:

- Menu bar: Choose [Online] - [Run].
- Toolbar:  ;
- Shortcut key: F5.

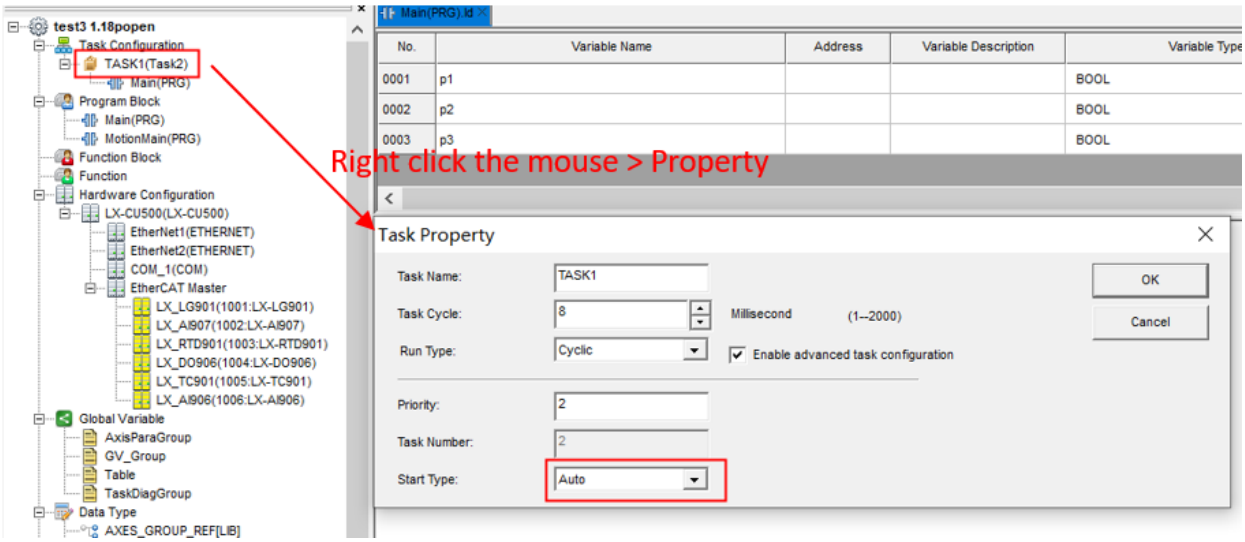
13.1.4 Result

The task program changes from the stopped state to the running state, and the program segments are executed. The RUN indicator of the PLC starts flashing.



13.1.5 Reference Message

If Startup Type in Task Properties is set to Auto, the task program will run automatically after compilation and download. You do not need to click Run.



13.2 Stop

13.2.1 Introduction

The stop instruction is used to stop monitored or simulated task programs. After the software selects the stop instruction, the PLC stops executing task programs.

13.2.2 Requirement

- AutoThink is already in monitoring or simulation state.
- The task program is running.

13.2.3 Steps

Follow the steps below to stop a task program:

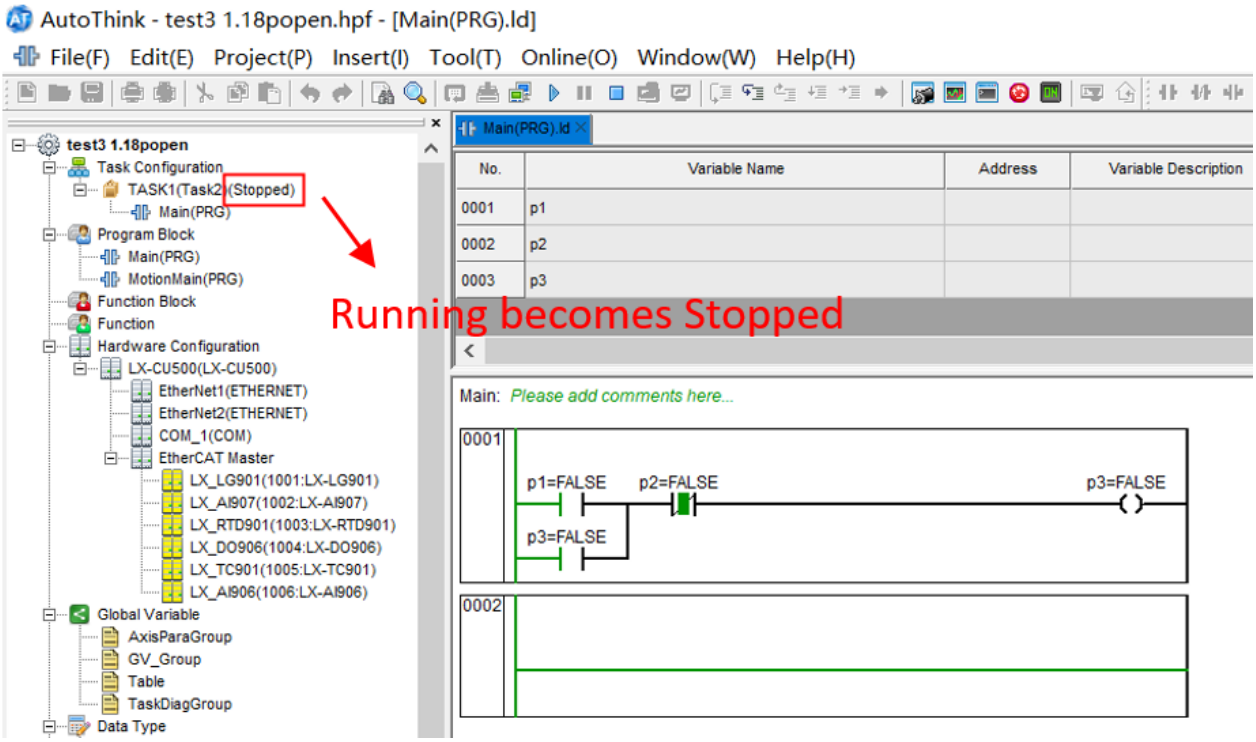
Menu bar: Choose Online > Stop.

Toolbar:  ;

Shortcut keys: Shift + F8.

13.2.4 Result

The task program changes from the running state to the stopped state, and program segment execution stops. The RUN indicator of the PLC is always on.



13.3 Write

13.3.1 Introduction

The write command is used to write variables to be debugged. To write a variable, you can double-click the variable in the program segment or the variable table at the top of the program and then click Write in the Debug Variable dialog box.

13.3.2 Requirement

The AT software is already in monitoring or simulation state.

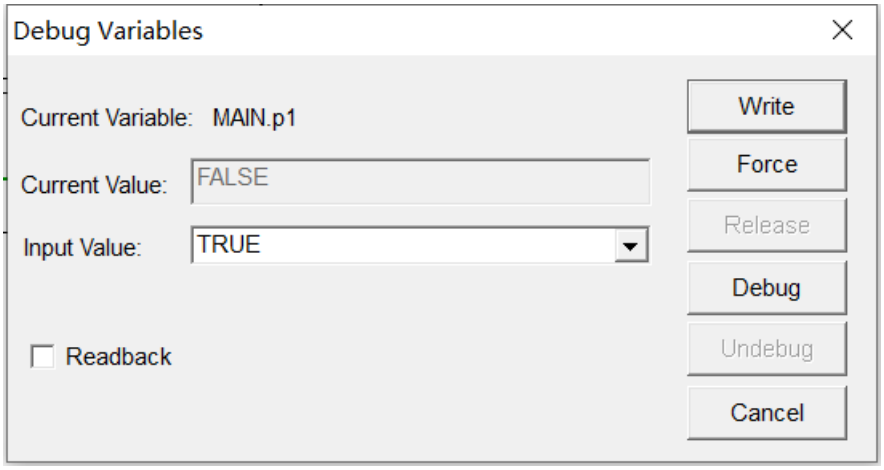
The task program is running.

13.3.3 Steps

Follow the steps below to write a variable:

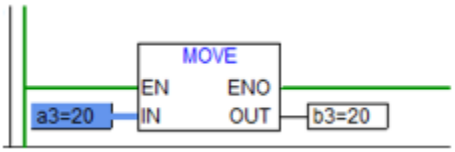
(1) Double-click the variable to be written to pop up the Debug Variable dialog box.

(2) Enter the value in the New Value field, and click Write to change the variable value.



13.3.4 Result

After the writing is completed, the monitored variable value is changed, and the relevant program logic is executed based on the new value.



13.3.5 Reference Message

The variable value can be a binary or hexadecimal number. Value format:

- Value type#number system#value. Example: BYTE#2#1010. Indicates that the value written to the variable is the binary number 10.
- Number system#value. Example: 16#A. Indicates that the value written to the variable is the hexadecimal number 10.

- Note:
- For BOOL variables, the values can only be TRUE or FALSE.
- For enumeration variables, the values can only be the enumeration values when the variables are defined.

13.4 Force

13.4.1 Introduction

Used to assign values to variables during debugging. After each loop ends, the forced variables are written with the forced values until they are released.

13.4.2 Requirement

AutoThink is already in monitoring or simulation state.

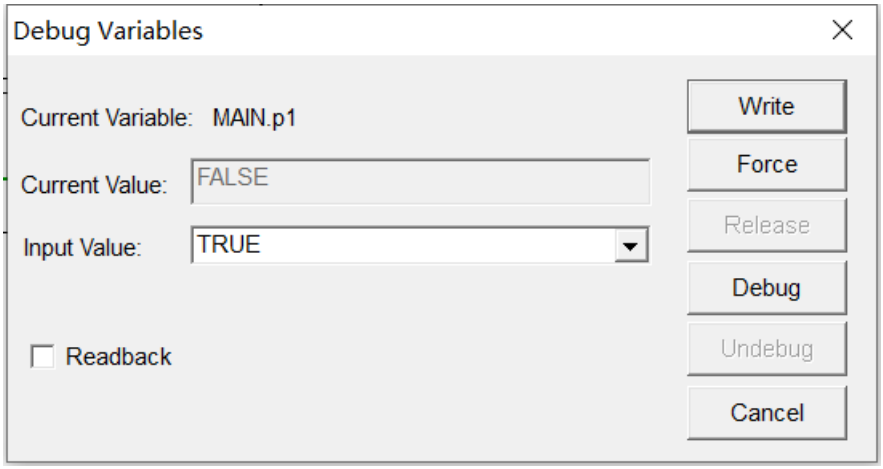
The task program is running.

13.4.3 Steps

Follow the steps below to force a variable:

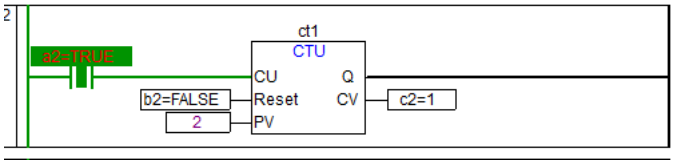
(1) Double-click the variable to be forced to pop up the Debug Variables dialog box.

(2) Enter the value in the New Value field, and click Force to change the variable value.

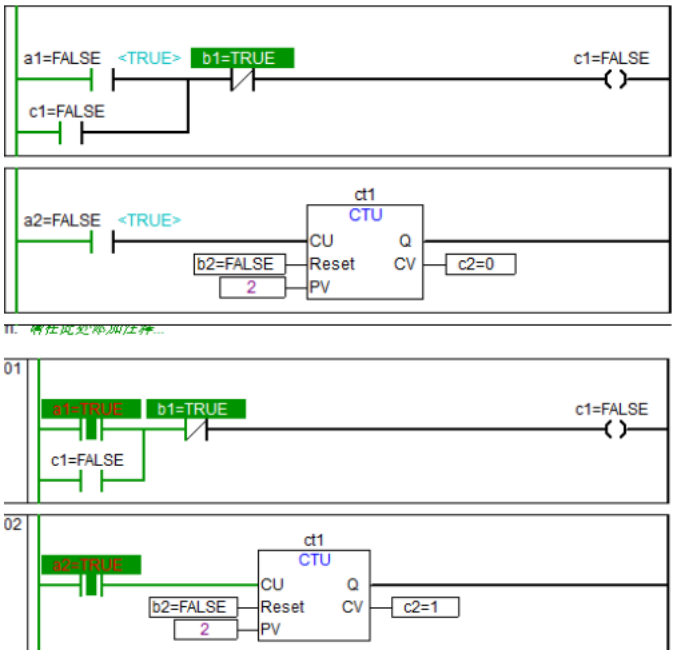


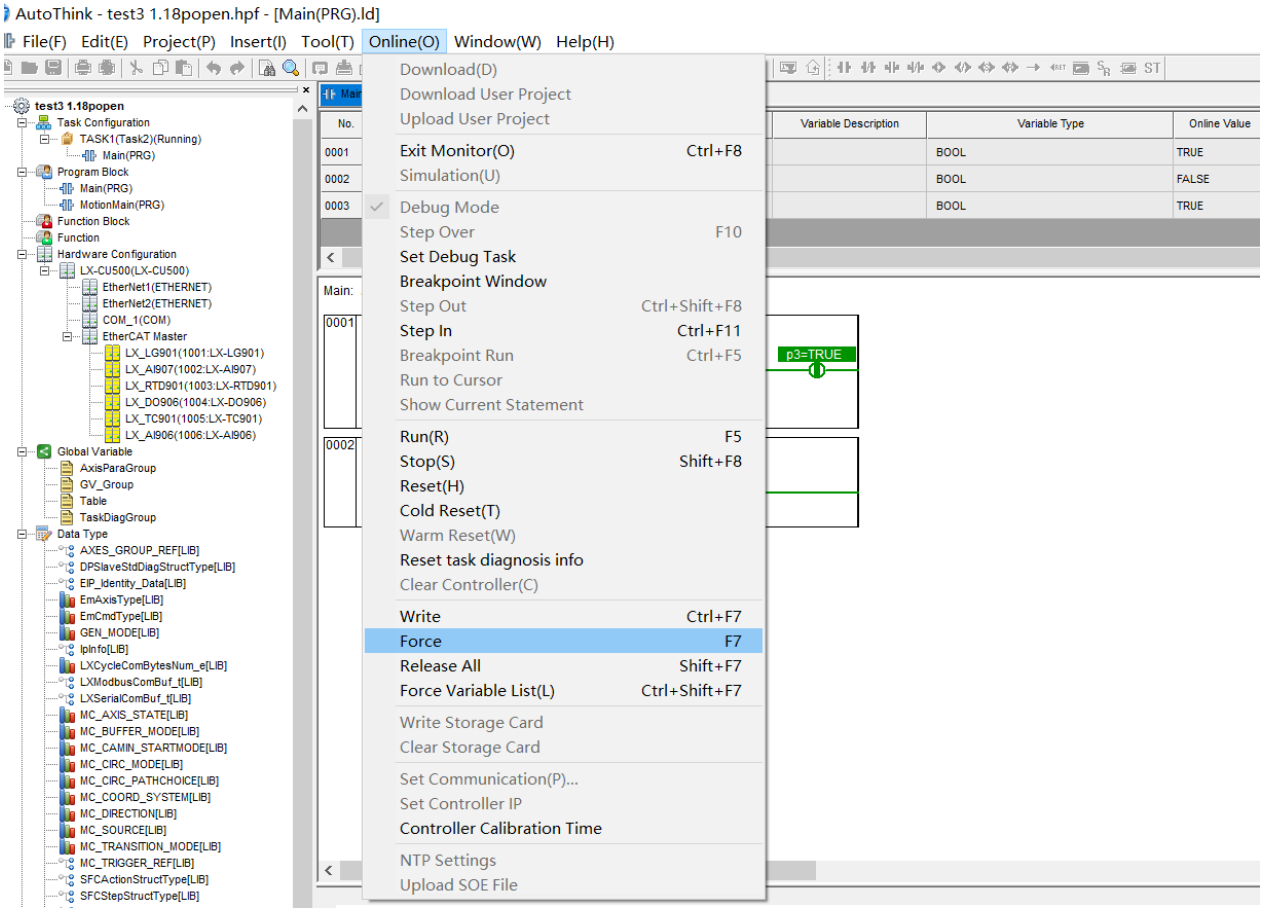
13.4.4 Result

After the operation is completed, the new value of the monitored variable is in red, and the relevant program logic is executed based on the new value.



13.4.5 Reference Message





Force multiple variables: After you modify the variable value and click To Be Debugged in the Debug Variable dialog box, the dialog box is closed, and the variable value is displayed as p1=TRUE . The value on the left is the current value, and the value on the right is the new value. If you perform the above operations on multiple variables, they will all be in the To Be Debugged state. At this time, you can choose Online > Force in the menu bar to assign the new values to selected variables.

13.5 Forced Variable Table

13.5.1 Introduction

Values can be forcibly assigned to multiple variables when the software is online or in the simulation state. The assigned value will be removed only after the variable is released.

13.5.2 Requirement

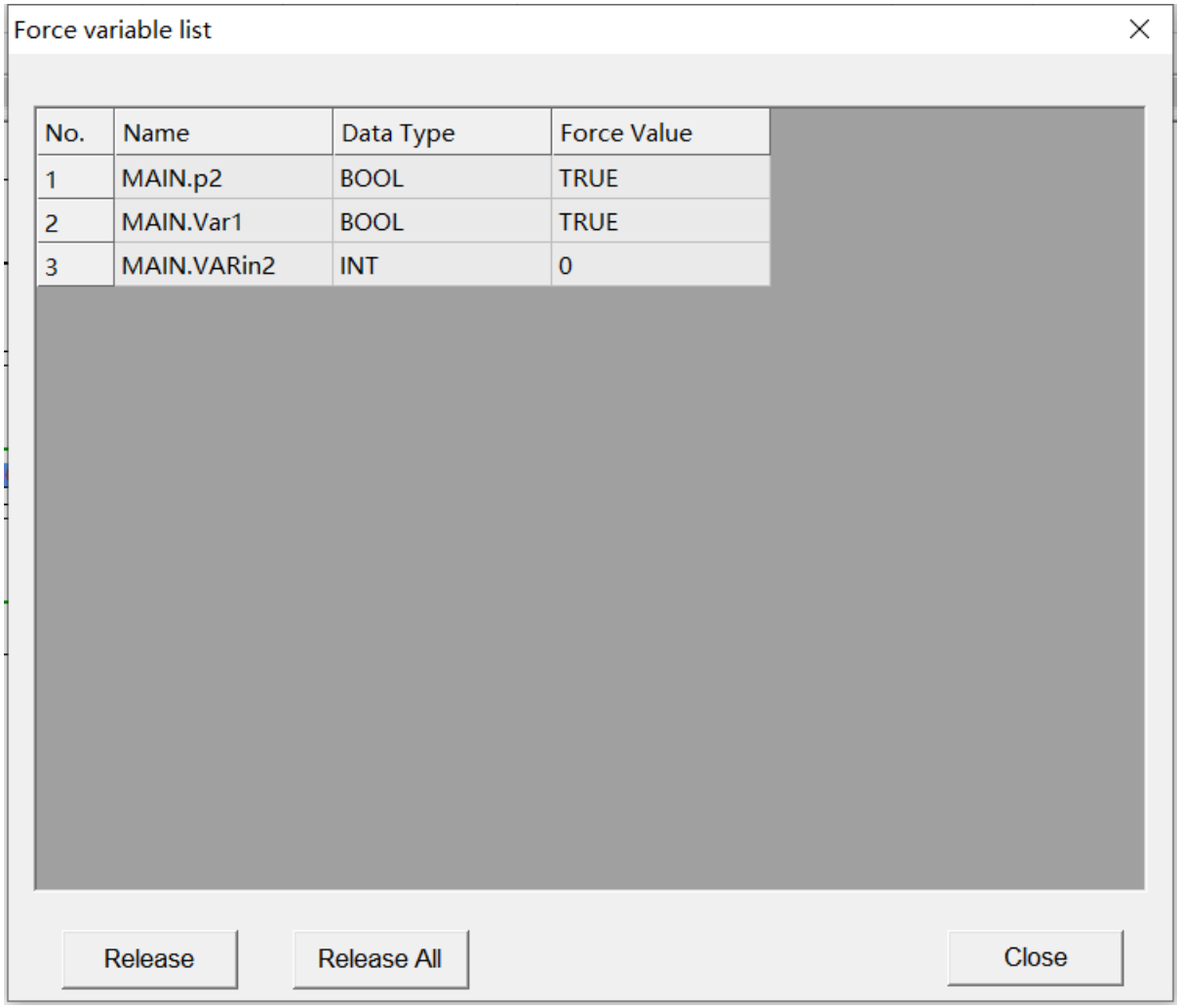
AutoThink is already in monitoring or simulation state.

The task program is running.

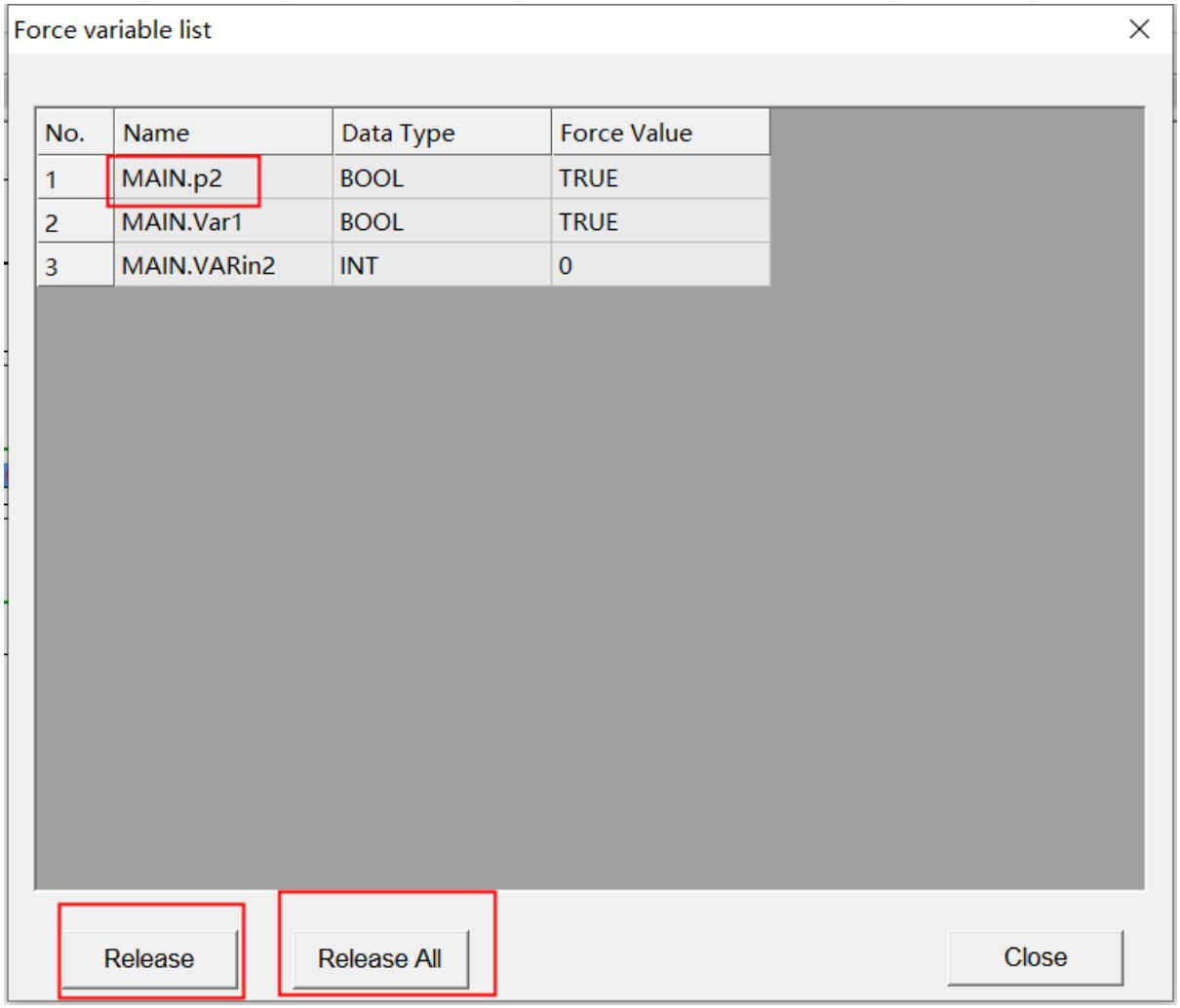
13.5.3 Steps

Follow the steps below to operate on the list:

(1) Menu bar: Choose [Online] - [Force Variable Table] to view the variables that have been forced.



(2) Select the variables to be released in the forced variable list, and click Release. You can also click Release All directly to release all forced variables.



13.5.4 Result

After a force variable is released, it is removed from the forced variable list. If Release All is clicked, all variables are removed from the forced variable table.

13.5.5 Reference Message

The Forced Variable List window lists all forced variables. You can release multiple variables in this window at a time. If you select variables to be released and click Release, only the selected variables are released. If you click Release All, all forced variables displayed in the list will be released. Released variables are removed from the list.

13.6 Release All

13.6.1 Introduction

Used to release all forced variables in the project.

13.6.2 Requirement

AutoThink is already in monitoring or simulation state.

The task program is running and contains forced variables.

13.6.3 Steps

Follow the steps below to release all forced variables:

(1) Menu bar: Choose [Online] - [Release All] or use the shortcut key Shift + F7.

13.6.4 Result

All forced variables in the program will be released.

13.7 Debug with Breakpoints

13.7.1 Enable Debug with Breakpoints

13.7.1.1 Introduction

Debugging with breakpoints is enabled by default and does not need to be set. If the debugging mode is disabled for the current project, enable debugging with breakpoints before debugging. After it is enabled, you can set breakpoints in POUs and perform debugging with the breakpoints. Currently, only breakpoints in ST and LD POUs are supported for debugging.

13.7.1.2 Requirement

AutoThink is offline.

13.7.1.3 Steps

Follow the steps below to enable debugging with breakpoints:

(1) Choose [Online] - [Debug Mode] in the menu bar.

(2) Compile and install the project again, and choose Online > Breakpoint in the menu bar.

13.7.2 Set Debug Task

13.7.2.1 Introduction

Before debug, you need to select a debugging task. Currently, only a single task is supported. TASK1 is the debugging task by default.

13.7.2.2 Requirement

AutoThink is in the monitoring state.

13.7.2.3 Steps

Follow the steps below to set a debugging task:

(1) Choose [Online] - [Set Debug] Task in the menu bar to pop up the Set Debugging Task dialog box.

(2) Click Set Debugging Task, and select the task from the drop-down task list.

(3) Click OK.

13.7.2.4 Result

Breakpoints of LD POUs are in an orange box.

13.7.3 Set Breakpoint

13.7.3.1 Introduction

Set breakpoints for the POU program to be debugged, regardless of the program online status.

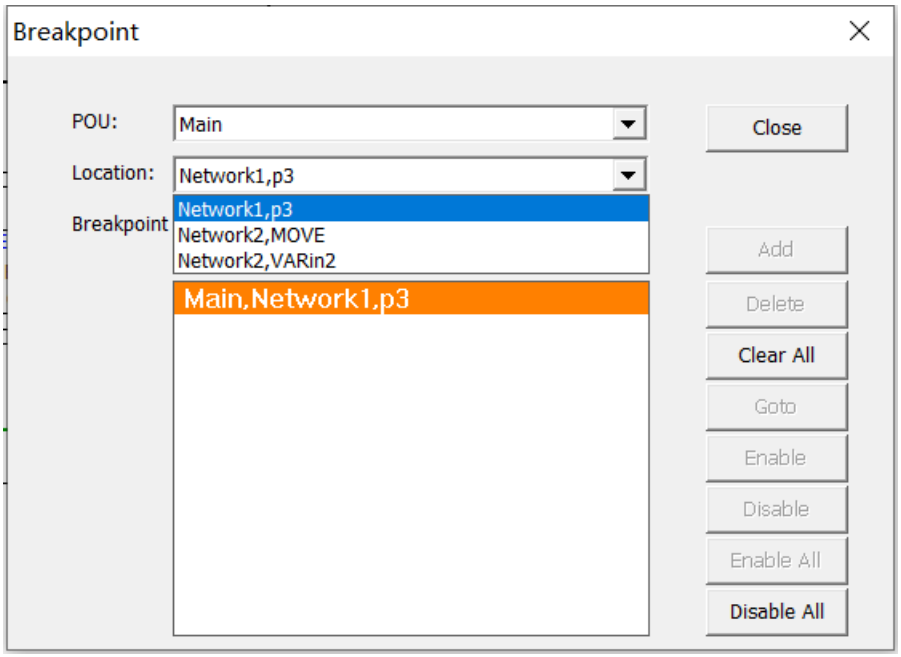
13.7.3.2 Requirement

The POU program can be either online or offline.

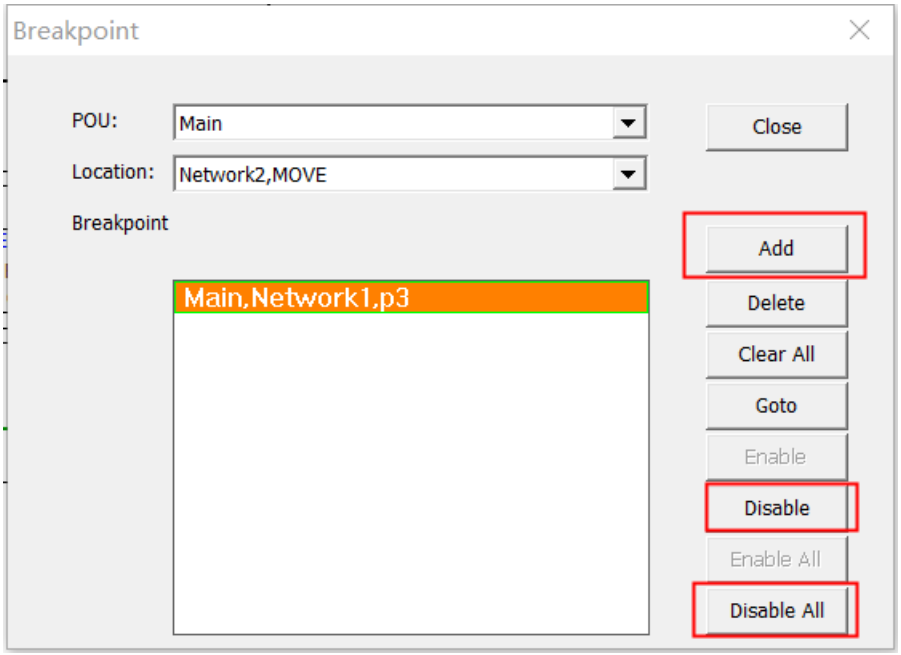
13.7.3.3 Steps

Follow the steps below to set a breakpoint:

(1) Choose [Online] - [Breakpoint window] in the menu bar to open the Breakpoint dialog box. POU: The drop-down list will display all ST and LD POUs. Location: The drop-down list will display all allowed breakpoint locations in the selected POU. For ST POUs, the locations are line numbers. For LD POUs, the locations are block components, output pins, and coils.

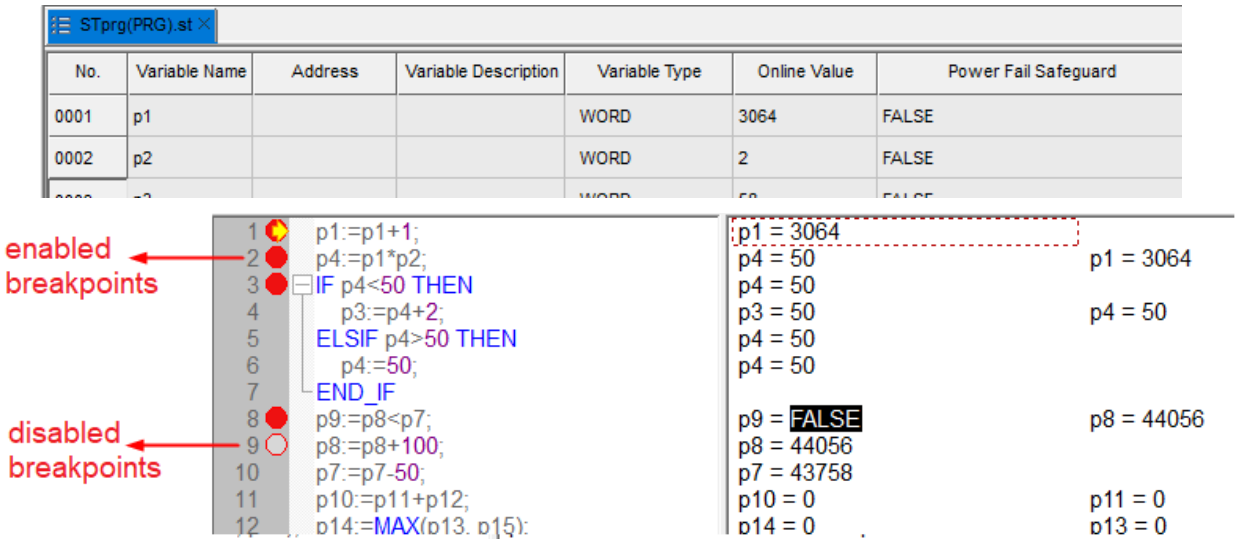


(2) After you select the POU and location, click Add to add a breakpoint. You can set the enabling status of breakpoints using the corresponding buttons.

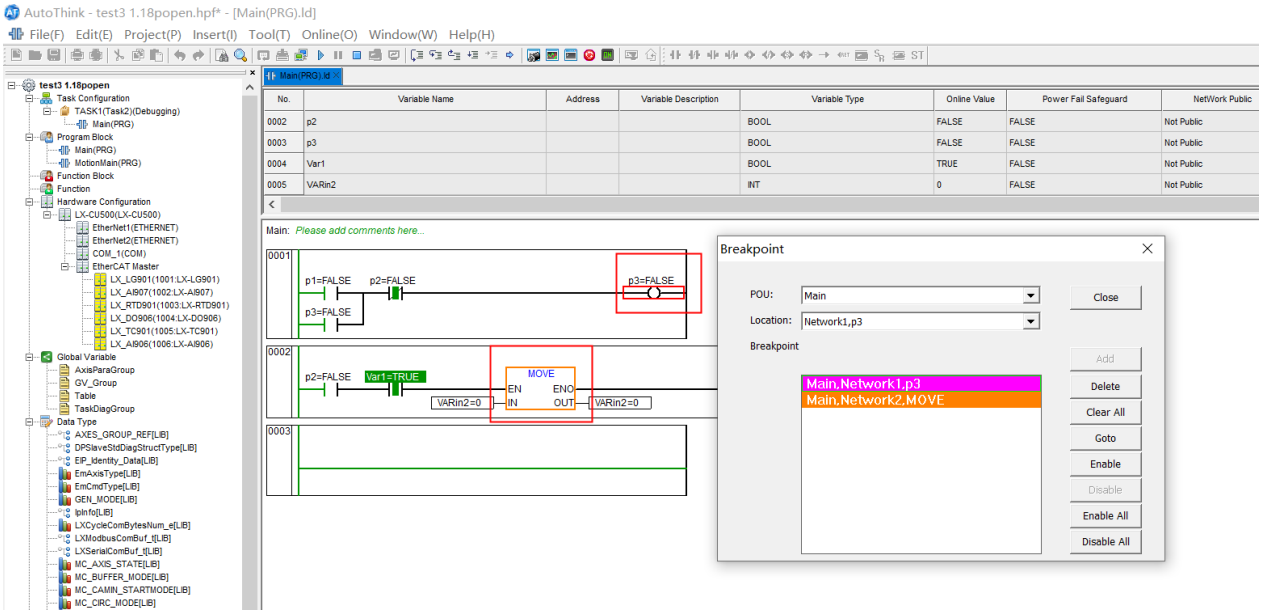


13.7.3.4 Result

In an ST POU, lines with enabled breakpoints have a red solid circle, and lines with disabled breakpoints have a red hollow circle, as shown in the figure below.



In an LD POU, enabled breakpoints are in a red box, and disabled breakpoints are in a purple box, as shown in the figure below.



13.7.3.5 Reference message

You can add breakpoints, delete breakpoints, set breakpoint enabling status, and perform other operations in the Breakpoint dialog box. A maximum of 1,024 breakpoints can be set in a project.

If the project is offline and modified with the Breakpoint dialog box open, no operation can be performed in the dialog box. You need to recompile the project before you can set breakpoints.

Button	Description
Add	Add a breakpoint. When an ST POU is online and monitored, you can click the line number of a certain statement line to add a breakpoint to this line. When an LD POU is online and monitored, you can right-click a block component, output pin, or coil and choose Debug > Set or Clear Breakpoint to add a breakpoint.
Delete	Select a breakpoint and click this button to delete it from the Breakpoint dialog box. When an ST POU is online and monitored, you can click the line number of the statement line with a breakpoint to delete the breakpoint. The breakpoint is deleted from the Breakpoint dialog box at the same time. When an LD POU is online and monitored, you can right-click a block component, output pin, or coil and choose Debug > Set or Clear Breakpoint to delete the breakpoint. The breakpoint is deleted from the Breakpoint dialog box at the same time.
Go To	Select a breakpoint and click Go To to jump to the breakpoint position and select the corresponding statement line or element.
Enable/Disable	Enable: Select a disabled breakpoint and click Enable to enable the breakpoint. Newly created breakpoints are enabled by default. Disable: Select an enabled breakpoint and click Disable to disable the breakpoint. Disabled breakpoints will be skipped during running with breakpoints.
Clear All	Click Clear All to delete all breakpoints in the Breakpoint dialog box.
Enable All	Enable all added breakpoints.
Disable All	Disable all added breakpoints.

13.7.4 Step Over

13.7.4.1 Introduction

This command is used for stepping of an online program. In an ST program, single-step execution uses statements as steps to execute the program. In an LD program, single-step execution uses elements with breakpoints as steps to execute the program.

For ordinary statement lines (not functions, functional blocks, or other called ST programs), single-step execution has the same effect as the "jump into" command.

For called functions, functional blocks, and ST/LD POUs, the POU will be executed as a complete step if no breakpoint is set in the POU or the set breakpoints are not enabled. The "jump into" command can be used to enter the POU for debugging.

13.7.4.2 Requirement

This command is used for stepping of an online program.

13.7.4.3 Steps

Follow the steps below to perform Step Over execution:

- Menu bar: [Choose Online] - [Step Over].
- Toolbar:  .
- Shortcut key: F10.

13.7.4.4 Result

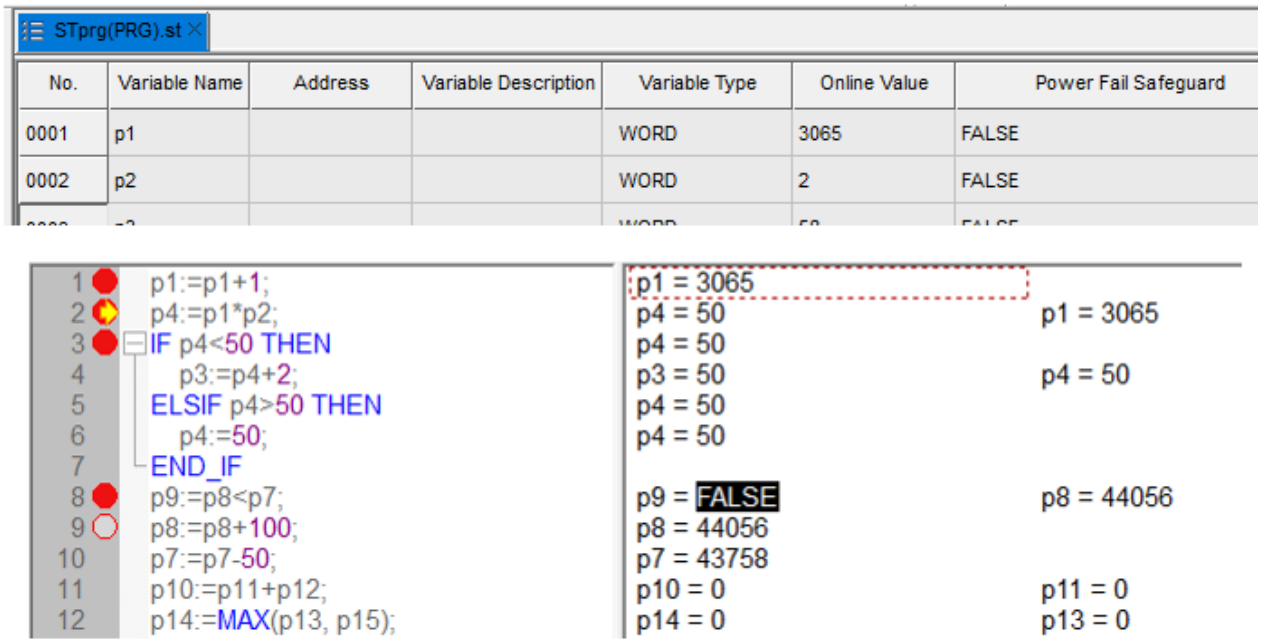
The result of single-step execution at a certain breakpoint is as follows:

- In an ST program, the execution stops at the next line after it is completed in the current line.
- In an LD program, the execution stops at the next breakpoint after it is completed in the current element.
- When multiple PRG POUs are involved, the execution process jumps to the breakpoint in the next POU after single-step execution is performed in the last line of the current POU. If only one POU is involved, the execution process jumps to the first step of the POU.
- If a function, functional block, or ST/LD program is called, the execution starts from the breakpoint in the called program if single-step execution is performed.

13.7.4.5 Reference message

- Example of ST Program

In Figure (a) below, the current line is Line 2. This line is executed after a single-step execution, and the execution process jumps to Line 3, as shown in Figure (b). After the second single-step execution, the process jumps to Line 4 if the IF condition is met. Otherwise, the process jumps to Line 5. The result is shown in Figure (c).



(a)

STprg(PRG).st

No.	Variable Name	Address	Variable Description	Variable Type	Online Value	Power Fail Safeguard
0001	p1			WORD	3065	FALSE
0002	p2			WORD	2	FALSE

```
1  p1:=p1+1;
2  p4:=p1*p2;
3  IF p4<50 THEN
4     p3:=p4+2;
5  ELSIF p4>50 THEN
6     p4:=50;
7  END_IF
8  p9:=p8<p7;
9  p8:=p8+100;
10 p7:=p7-50;
11 p10:=p11+p12;
12 p14:=MAX(p13, p15);
```

```
p1 = 3065
p4 = 6130
p4 = 6130
p3 = 50
p4 = 6130
p4 = 6130

p9 = FALSE
p8 = 44056
p7 = 43758
p10 = 0
p14 = 0
```

```
p1 = 3065
p4 = 6130
p3 = 50
p8 = 44056
p11 = 0
p13 = 0
```

(b)

STprg(PRG).st

No.	Variable Name	Address	Variable Description	Variable Type	Online Value	Power Fail Safeguard
0001	p1			WORD	3065	FALSE
0002	p2			WORD	2	FALSE
0003	p3			WORD	50	FALSE

```
1  p1:=p1+1;
2  p4:=p1*p2;
3  IF p4<50 THEN
4     p3:=p4+2;
5  ELSIF p4>50 THEN
6     p4:=50;
7  END_IF
8  p9:=p8<p7;
9  p8:=p8+100;
10 p7:=p7-50;
11 p10:=p11+p12;
12 p14:=MAX(p13, p15);
```

```
p1 = 3065
p4 = 6130
p4 = 6130
p3 = 50
p4 = 6130
p4 = 6130

p9 = FALSE
p8 = 44056
p7 = 43758
p10 = 0
p14 = 0
```

```
p1 = 3065
p4 = 6130
p3 = 50
p8 = 4405
p11 = 0
p13 = 0
```

(c)

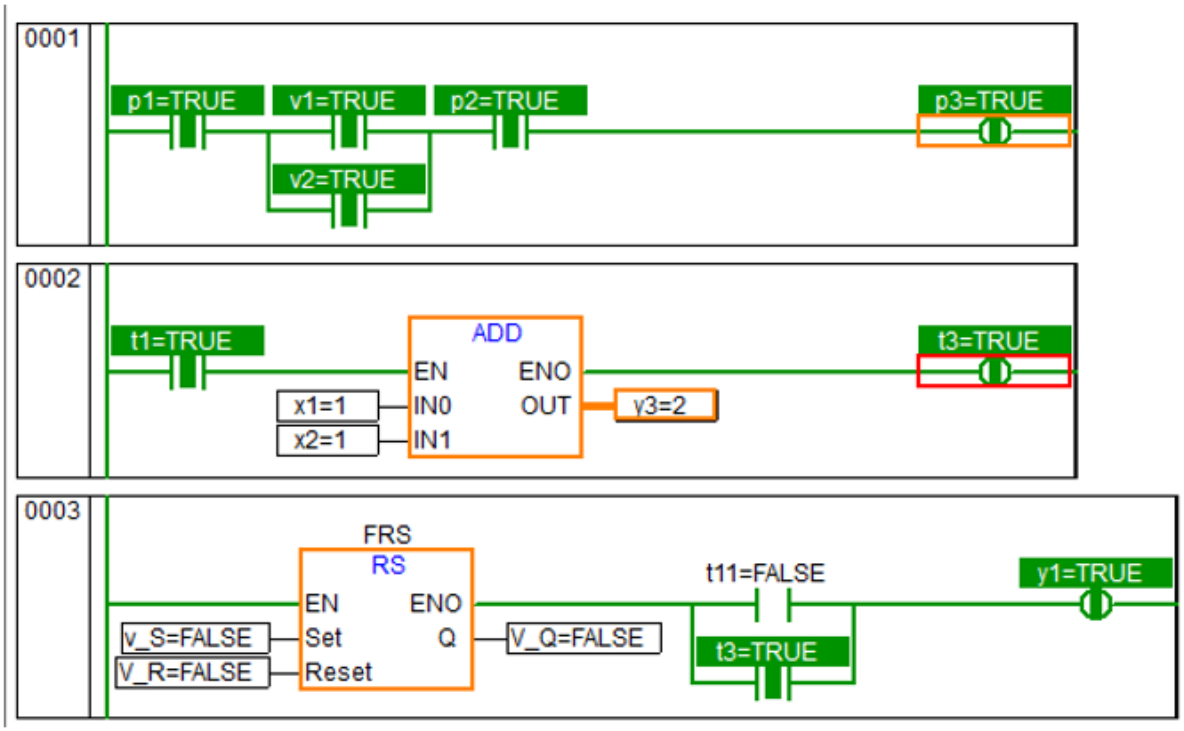
Example of ST Program

- Example of LD Program

In Figure (a) below, the current element is the coil t3. This coil is executed after a single-step execution, and the execution process jumps to the functional block FRS of the network 0003, as shown in Figure (b). After the second single-step execution, the process jumps to the output pin V_Q of the functional block FRS. The result is shown in Figure (c).

Main(PRG).ld

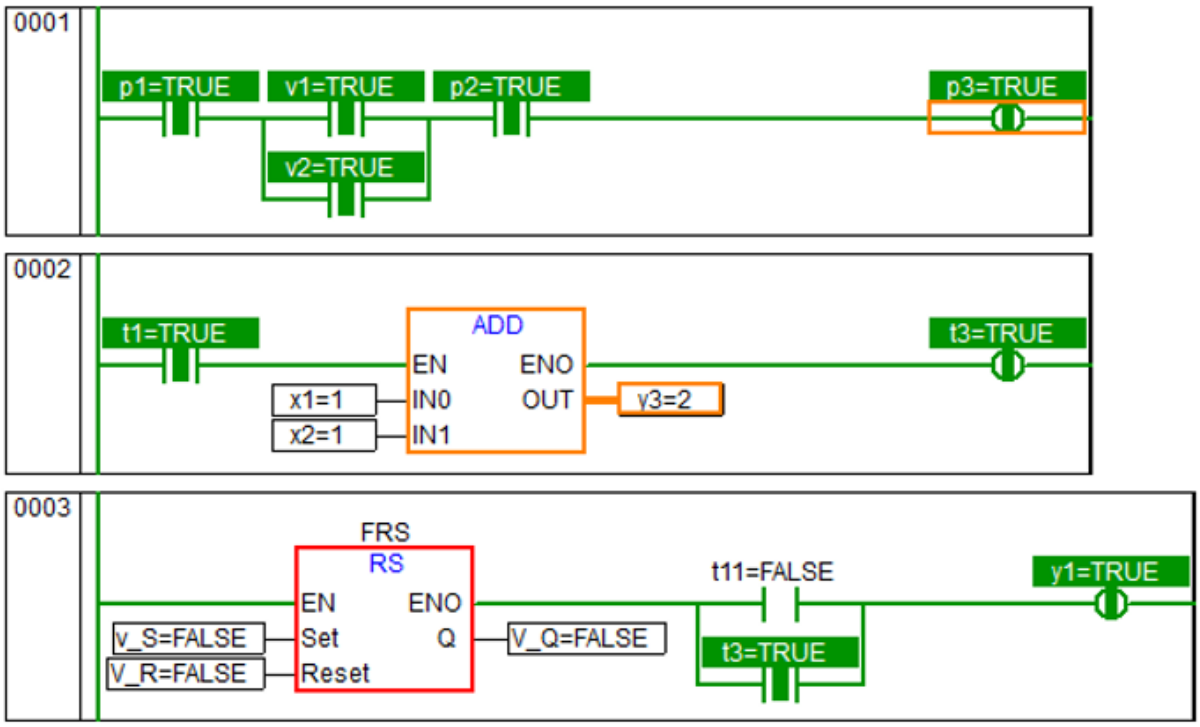
No.	Variable Name	Address	Variable Description	Variable Type
0009	v2			BOOL
0010	t1			BOOL
0011	t3			BOOL



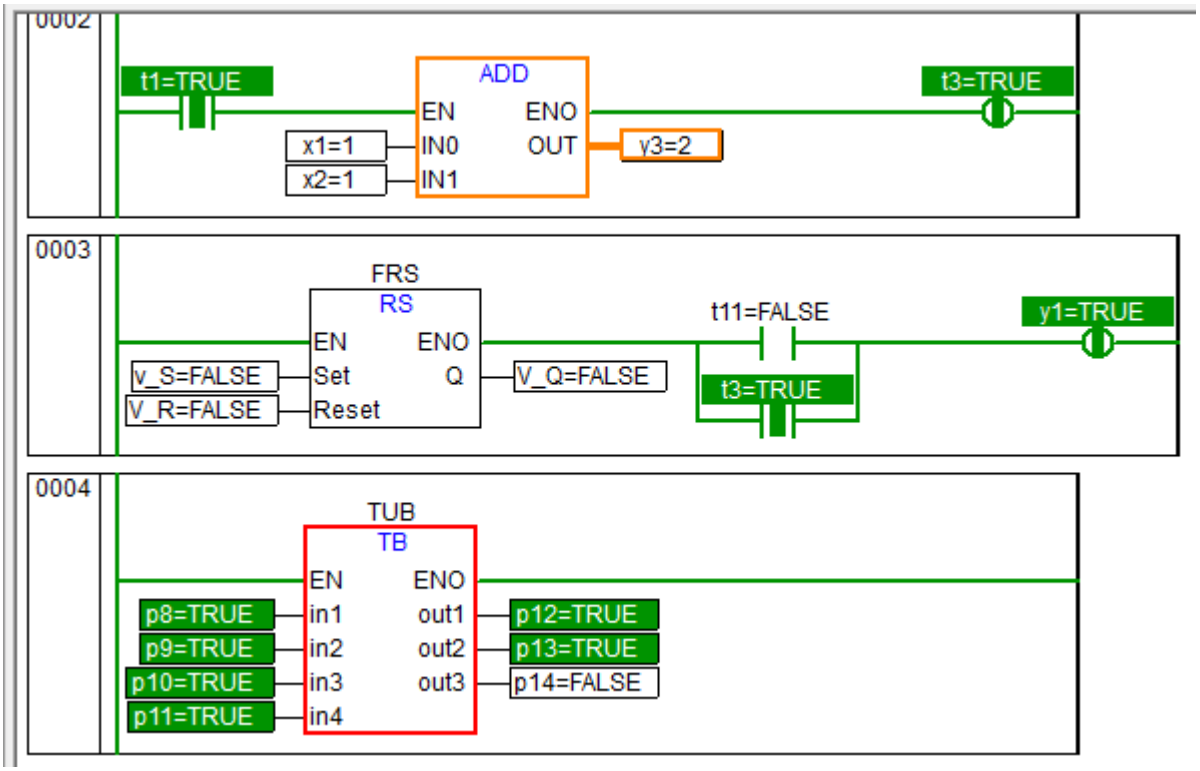
(a)

Main(PRG).ld

No.	Variable Name	Address	Variable Description	Variable Type
0009	v2			BOOL
0010	t1			BOOL
0011	t3			BOOL



(b)



(c)

Example of LD Program

13.7.5 Breakpoint Run

13.7.5.1 Introduction

Running with breakpoints use enabled breakpoints as steps to execute the program. During running with breakpoints, the program starts execution from the current position and stops at the next enabled breakpoint.

If breakpoints are not set in the currently debugged POU and other POUs it calls, the POU exits the debugging mode when running with breakpoints is performed.

A statement line may have called a function, functional block, or ST program but have no enabled breakpoint. If an enabled breakpoint is set in the function, functional block, or ST program, the execution process will jump to the corresponding enabled breakpoint when running with breakpoints is performed.

13.7.5.2 Requirement

AutoThink is in debugging mode and the monitoring state.

13.7.5.3 Steps

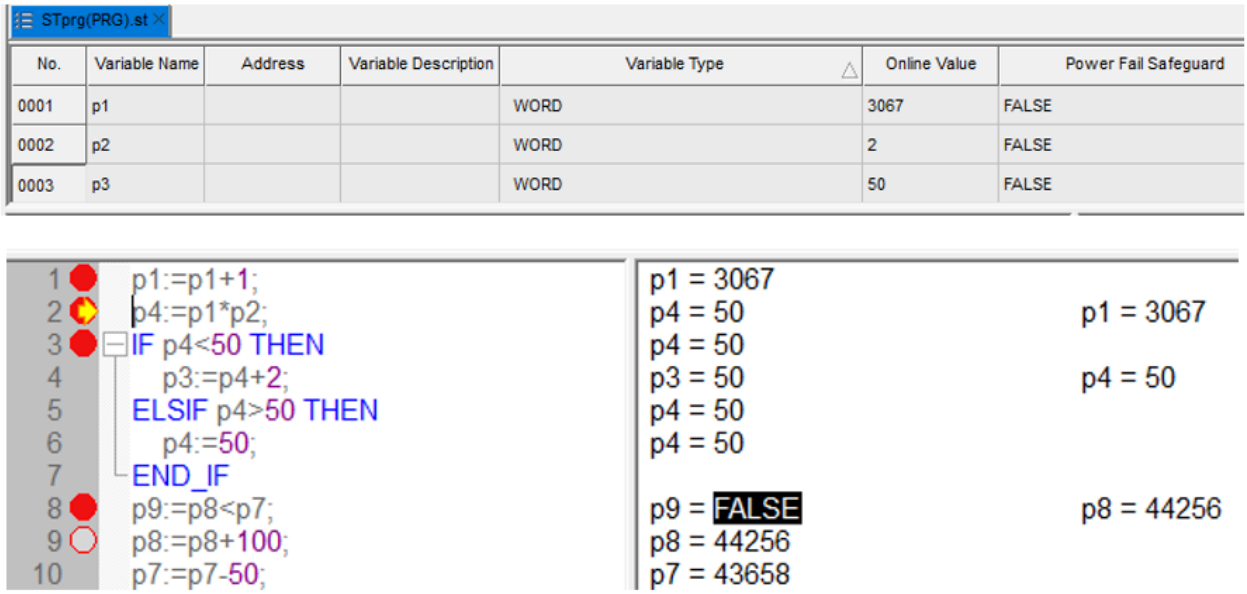
Follow the steps below to run with breakpoints:

- Menu bar: Choose [Online] > [Breakpoint Run].
- Toolbar:  .
- Shortcut key: Ctrl + F5.

13.7.5.4 Reference message

- Example of ST Program

  In Figure (a) below, the current line is Line 2. When running with breakpoints is performed, the program starts execution from this line and stops at Line 3 with the next enabled breakpoint, as shown in Figure (b) below. When running with breakpoints is performed again, the program starts execution from Line 3 and stops at Line 8 with the next enabled breakpoint, as shown in Figure (c) below.



(a)

STprg(PRG).st

No.	Variable Name	Address	Variable Description	Variable Type	Online Value	Power Fail Safeguard
0001	p1			WORD	3067	FALSE
0002	p2			WORD	2	FALSE
0003	p3			WORD	50	FALSE

```
1  p1:=p1+1;
2  p4:=p1*p2;
3  IF p4<50 THEN
4     p3:=p4+2;
5  ELSIF p4>50 THEN
6     p4:=50;
7  END_IF
8  p9:=p8<p7;
9  p8:=p8+100;
10 p7:=p7-50;
```

```
p1 = 3067
p4 = 6134
p4 = 6134
p3 = 50
p4 = 6134
p4 = 6134

p9 = FALSE
p8 = 44256
p7 = 43658
```

p1 = 3067

p4 = 6134

p8 = 44256

(b)

STprg(PRG).st

No.	Variable Name	Address	Variable Description	Variable Type	Online Value	Power Fail Safeguard
0001	p1			WORD	3067	FALSE
0002	p2			WORD	2	FALSE
0003	p3			WORD	50	FALSE

```
1  p1:=p1+1;
2  p4:=p1*p2;
3  IF p4<50 THEN
4     p3:=p4+2;
5  ELSIF p4>50 THEN
6     p4:=50;
7  END_IF
8  p9:=p8<p7;
9  p8:=p8+100;
```

```
p1 = 3067
p4 = 50
p4 = 50
p3 = 50
p4 = 50
p4 = 50

p9 = FALSE
p8 = 44256
```

p1 = 3067

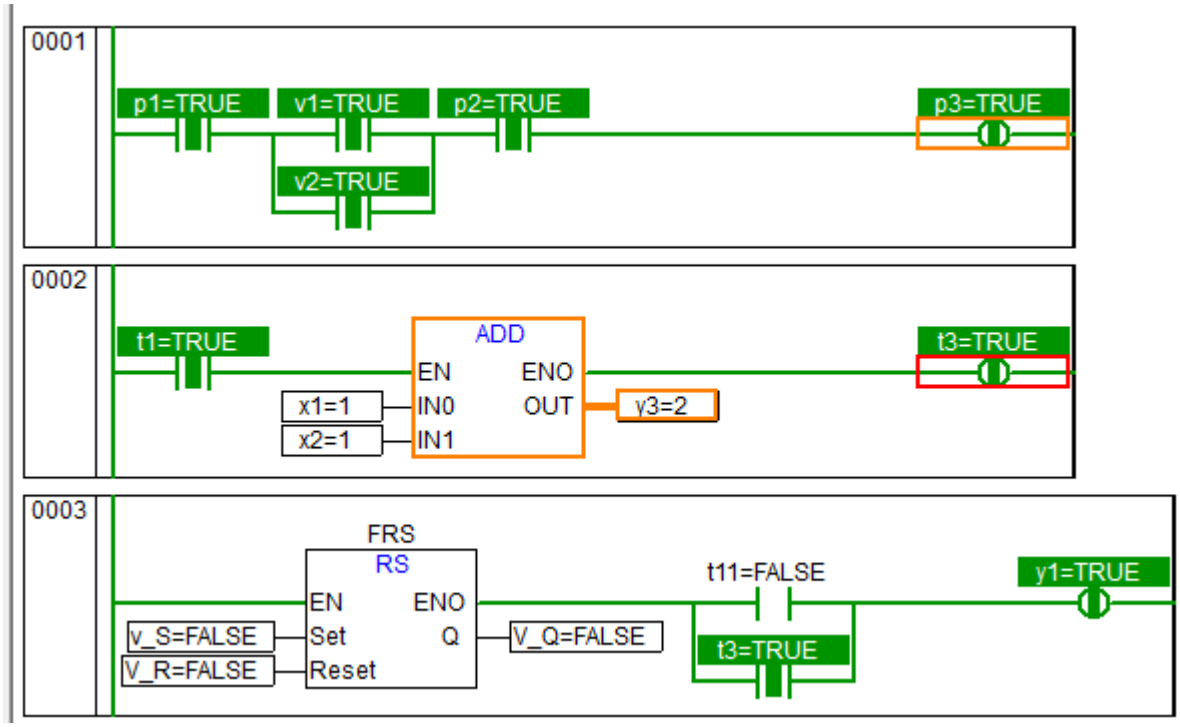
p4 = 50

p8 = 44256

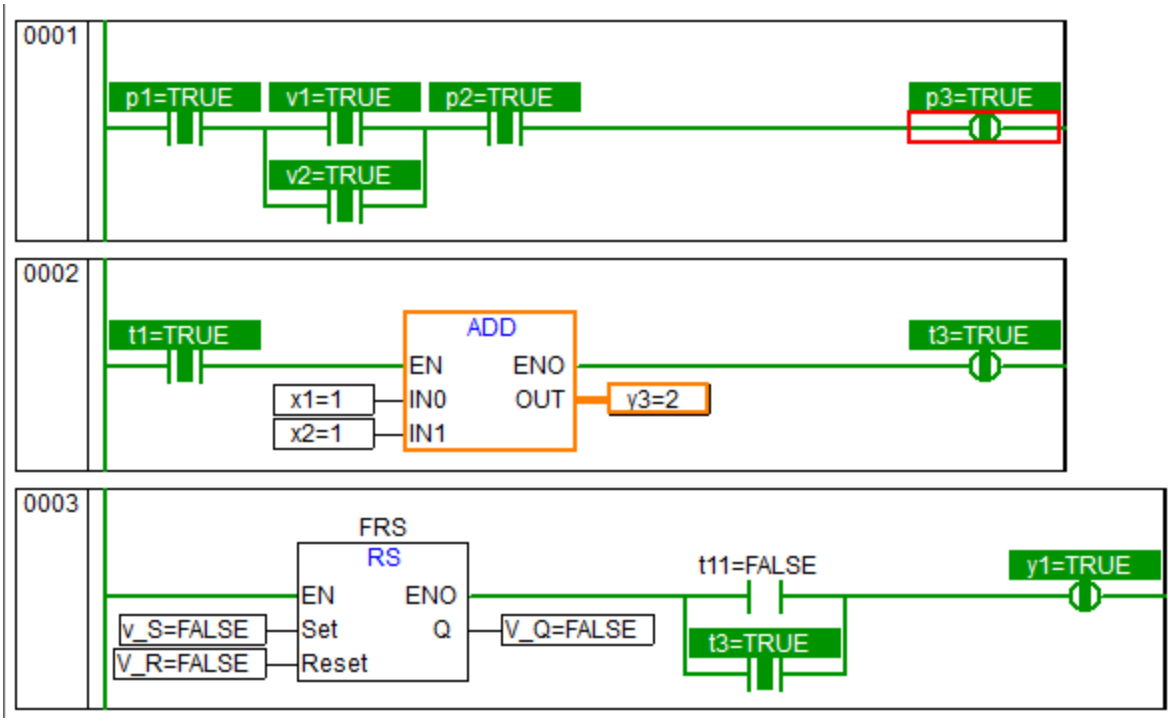
(c)

■ Example of LD Program

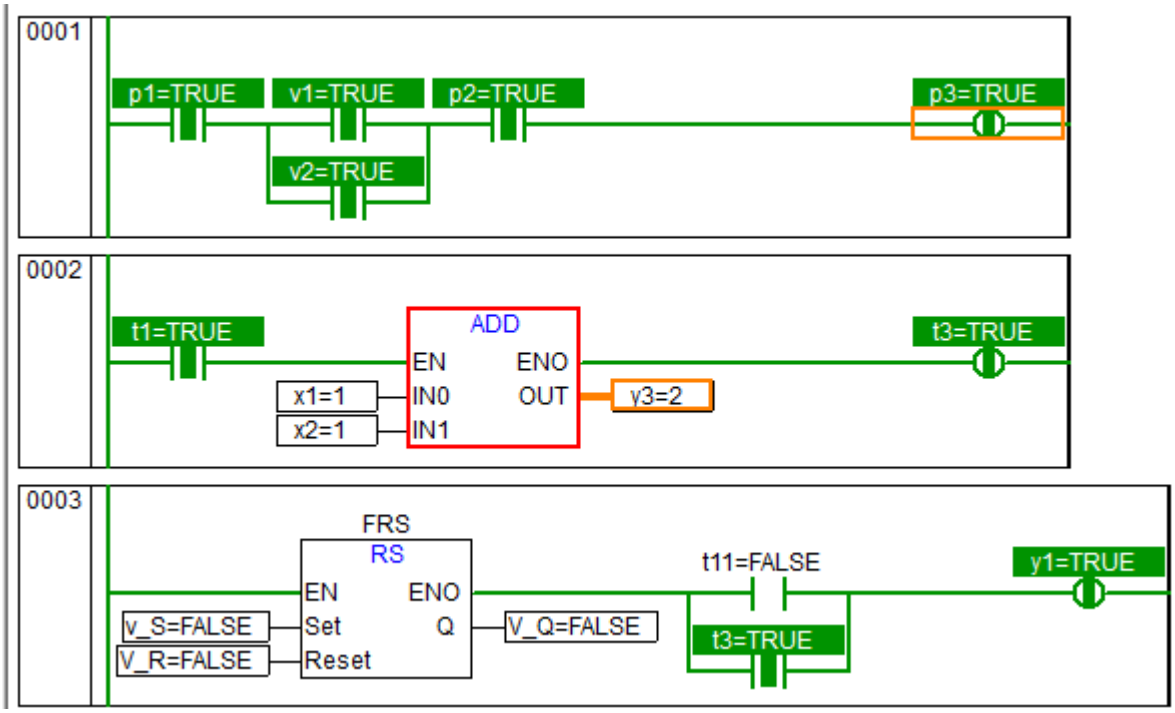
In Figure (a) below, the current element is the coil t3. When running with breakpoints is performed, the program starts execution from the coil t3 and stops at the coil p3 with the next breakpoint, as shown in Figure (b). When running with breakpoints is performed again, the program starts execution from p3 and stops at the functional block ADD with the next breakpoint. The result is shown in Figure (c).


0001
p1=TRUE
v1=TRUE
p2=TRUE
p3=TRUE
v2=TRUE
0002
t1=TRUE
ADD
EN
ENO
x1=1
IN0
OUT
y3=2
x2=1
IN1
t3=TRUE
0003
FRS
RS
EN
ENO
v_S=FALSE
Set
Q
V_Q=FALSE
V_R=FALSE
Reset
t11=FALSE
t3=TRUE
y1=TRUE

(a)


0001
p1=TRUE
v1=TRUE
p2=TRUE
p3=TRUE
v2=TRUE
0002
t1=TRUE
ADD
EN
ENO
x1=1
IN0
OUT
y3=2
x2=1
IN1
t3=TRUE
0003
FRS
RS
EN
ENO
v_S=FALSE
Set
Q
V_Q=FALSE
V_R=FALSE
Reset
t11=FALSE
t3=TRUE
y1=TRUE

(b)



(c)

Example of LD Program

13.7.6 Step In

13.7.6.1 Introduction

This command is used for stepping of an online program. When a function, functional block, or another program is called in the program, the "jump into" command can be used to enter the function, functional block, or program for execution.

If no breakpoint is set in an online POU being debugged, only the "jump into" command can be used. The POU enters the debugging mode using this command. At this time, the "single-step execution", "jump out", and "running with breakpoints" commands can be used. After the "jump into" command is executed, the execution process jumps to the corresponding position.

13.7.6.2 Requirement

This command is used for stepping of an online program.

13.7.6.3 Steps

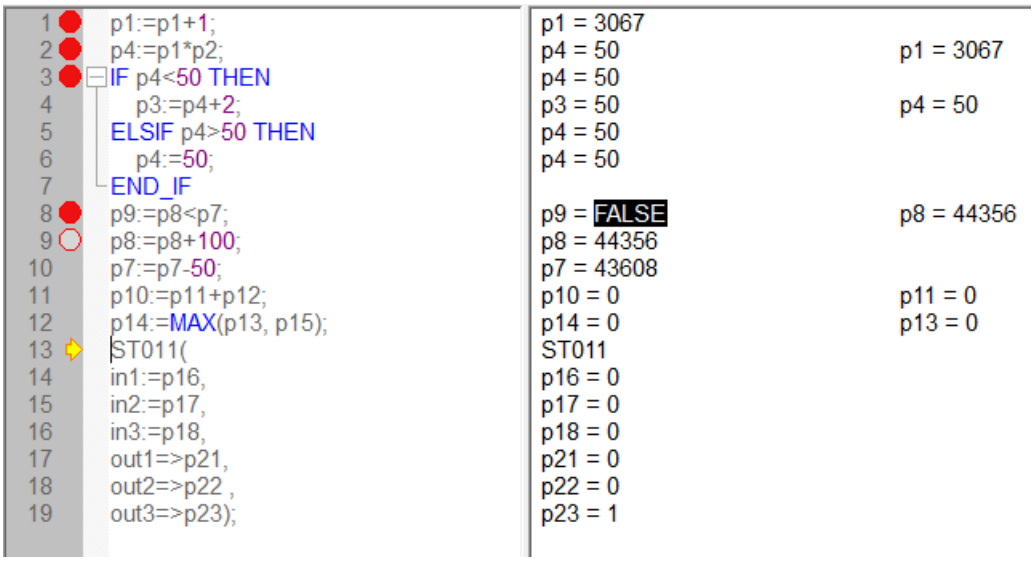
Follow the steps below to perform the jump into operation:

- Menu bar: Choose [Online] - [Step In].
- Toolbar:  .
- Shortcut key: Ctrl + F11.

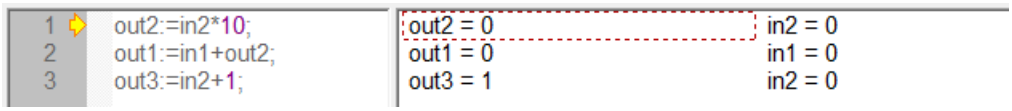
13.7.6.4 Reference message

- Example of ST Program

Functional block instance ST011 is called in the program STprg(PRG), and the current line is Line 13, as shown in Figure (a). After executing the "jump into" command, the program enters the called functional block, as shown in Figure (b). If "jump into" is executed at the last step of the functional block, the program will execute this line and return to the position where the functional block is called, as shown in Figure (a).



(a)

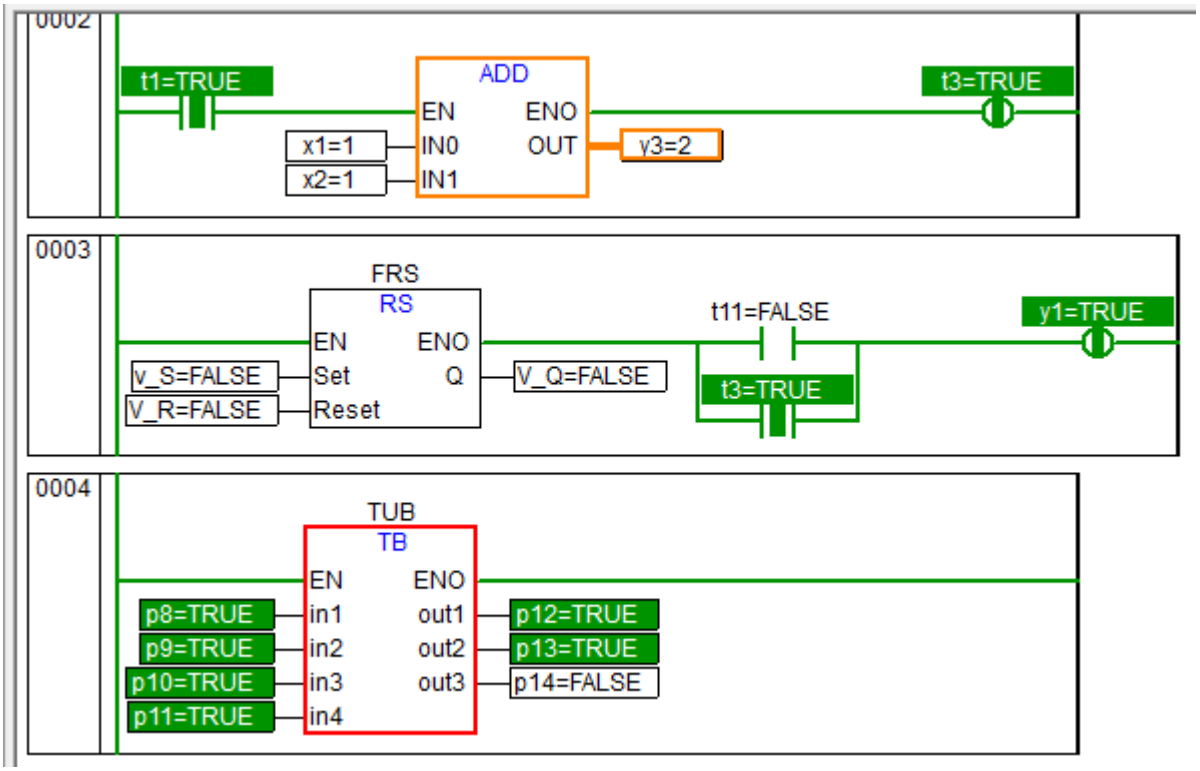


(c)

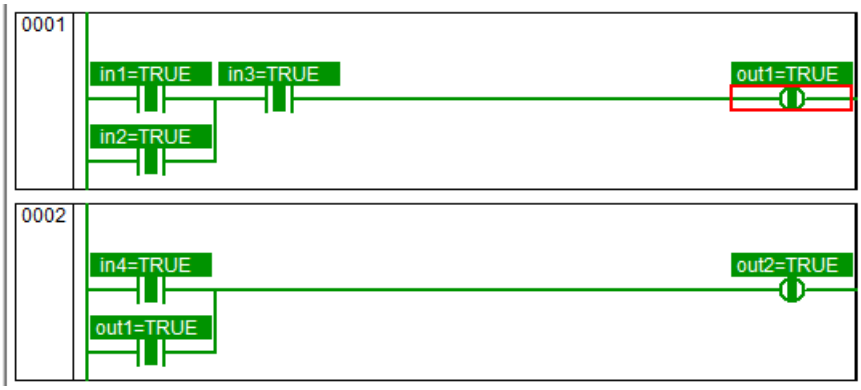
ST Example

- Example of LD Program

Functional block instance TUB is called in the program Main(PRG), and the current element is TUB, as shown in Figure (a). After executing the "jump into" command, the program enters the called functional block TUB, as shown in Figure (b). If "jump into" is executed at the last breakpoint of the functional block, the program will execute the corresponding logic and return to the position where the functional block is called, as shown in Figure (a).



(a)



(b)

LD Example

13.7.7 Step Out

13.7.7.1 Introduction

A POU being debugged (function, functional block, or other called ST/LD programs) may have no breakpoint or have only disabled breakpoints. When the "jump out" command is executed, the remaining statement lines of this POU will be executed, and the execution process returns to the position where the POU is called. If the POU being debugged has enabled breakpoints, the execution process jumps to the next breakpoint and stops.

For ordinary statement lines (not functions, functional blocks, or other called ST/LD programs), the jump out command has the same effect as the run with breakpoints command.

13.7.7.2 Requirement

This command is used for stepping of an online program.

13.7.7.3 Steps

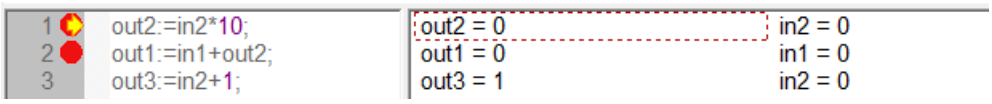
Follow the steps below to perform the jump out operation:

- Menu bar: Choose [Online] - [Step Out].
- Toolbar:  .
- Shortcut key: Ctrl + Shift + F8

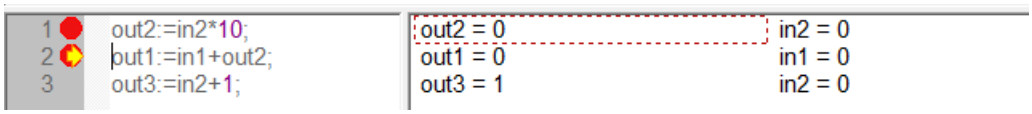
13.7.7.4 Reference message

- Example of ST Program

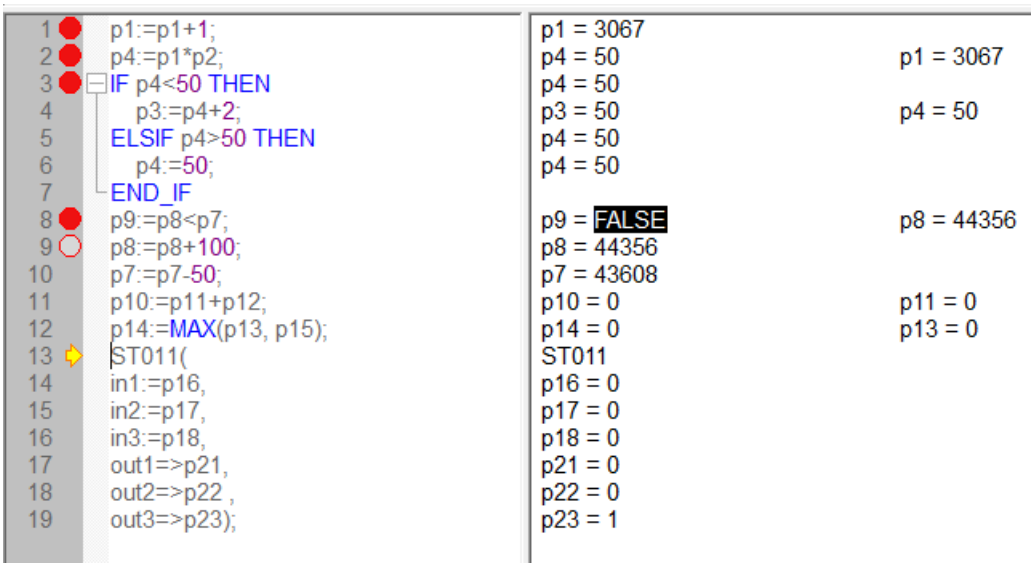
  The functional block instance ST011 is called in the program STprg(PRG), and breakpoints are set in Line 1 and Line 2 of ST011. The current line in ST011 is Line 1, as shown in Figure (a). When the "jump out" command is executed, the program executes Line 1 and jumps to Line 2 with the next breakpoint, as shown in Figure (b). When the "jump out" command is executed again, the program returns to the position where the functional block is called, as shown in Figure (c).



(a)



(b)

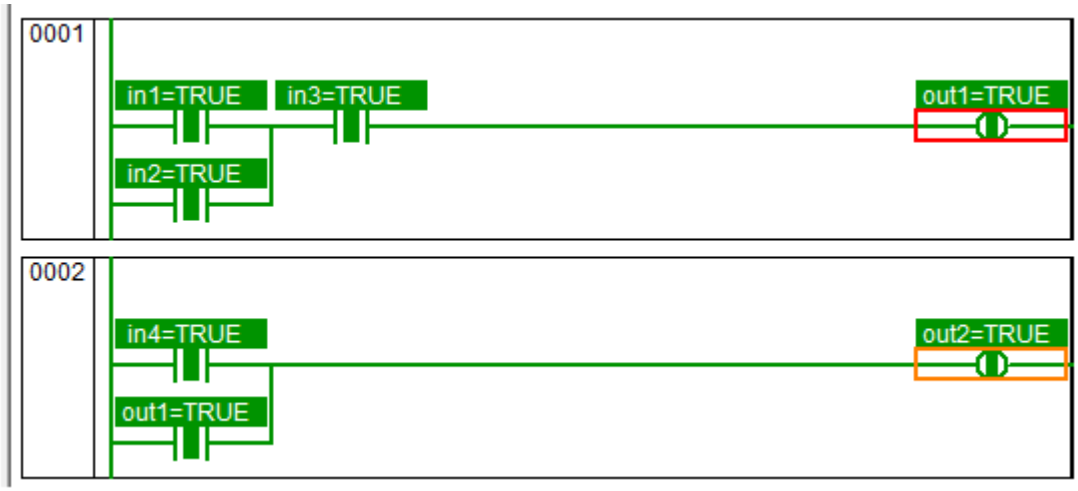


(c)

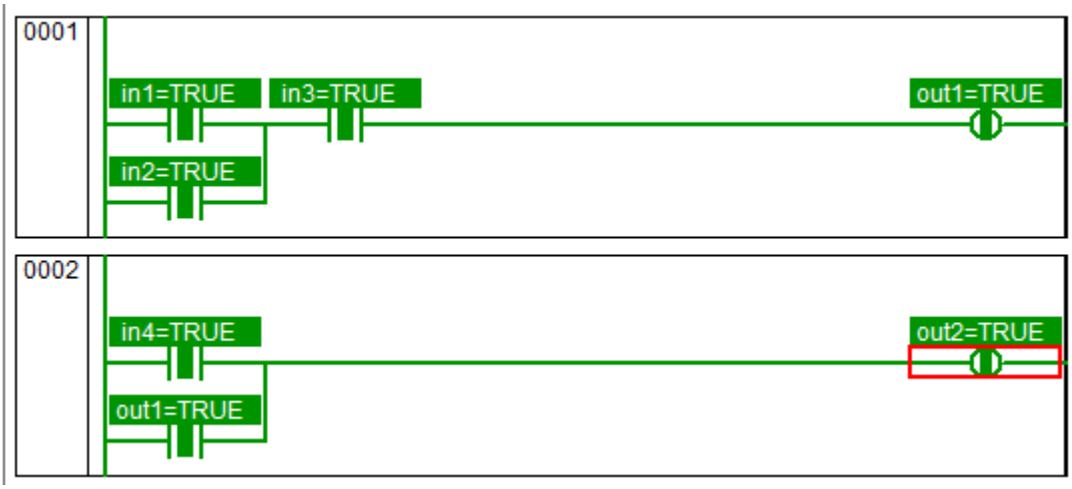
ST Example

■ Example of LD Program

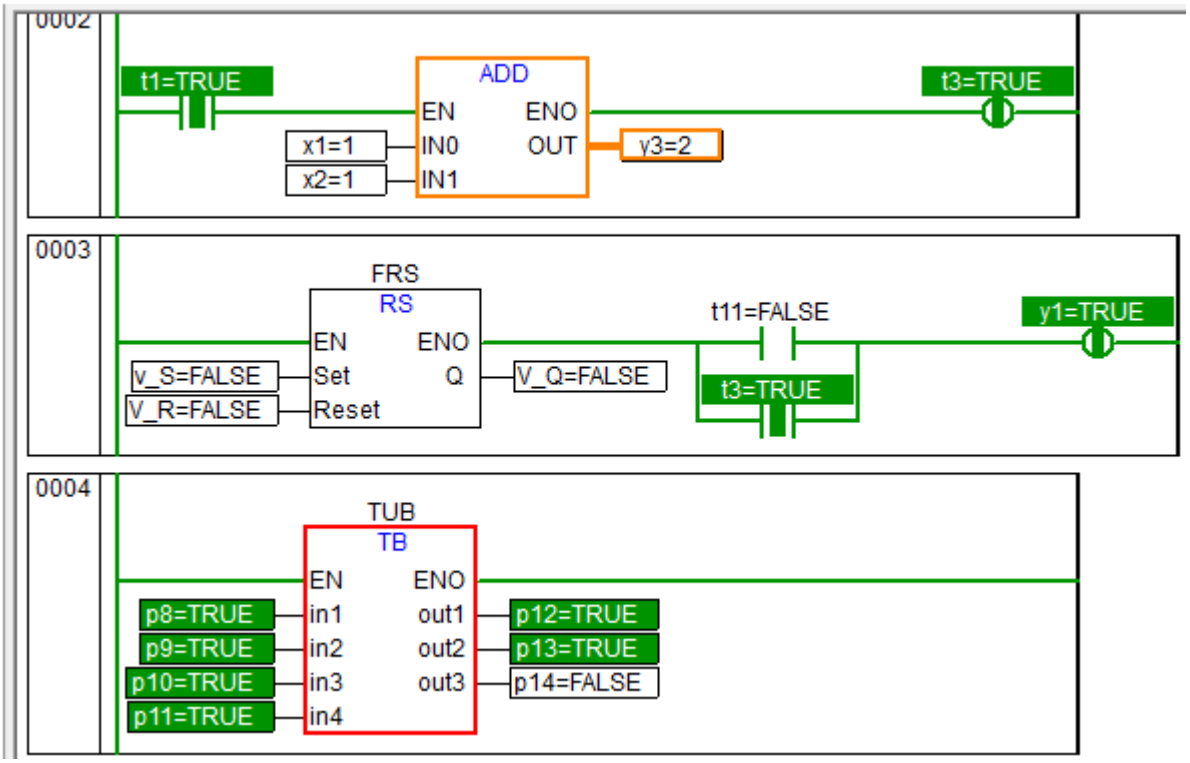
The functional block instance TUB is called in the program Main(PRG), and the coils out1 and out2 of TUB are set as breakpoints. The current position in TUB is out1, as shown in Figure (a). When the jump out command is executed, the program executes the logic in Network 0001 and jumps to out2 with the next breakpoint, as shown in Figure (b). When the "jump out" command is executed again, the program returns to the position where the functional block is called, as shown in Figure (c).



(a)



(b)



(c)

LD Example

13.7.8 Run to Cursor

13.7.8.1 Introduction

When debugging a program, you can use this command to execute from the current position continuously and stop at another position.

13.7.8.2 Requirement

This command is used for stepping of an online program.

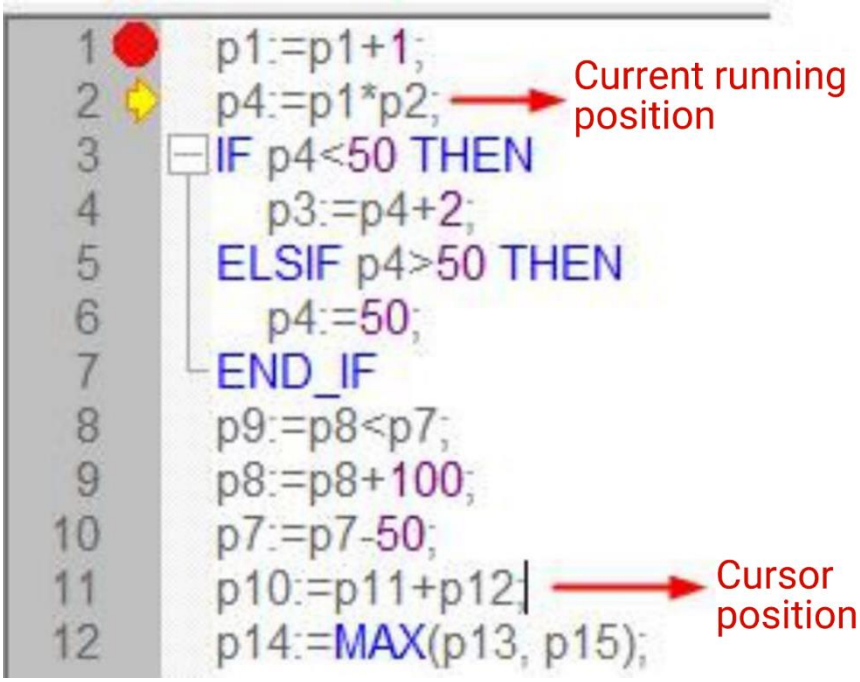
13.7.8.3 Steps

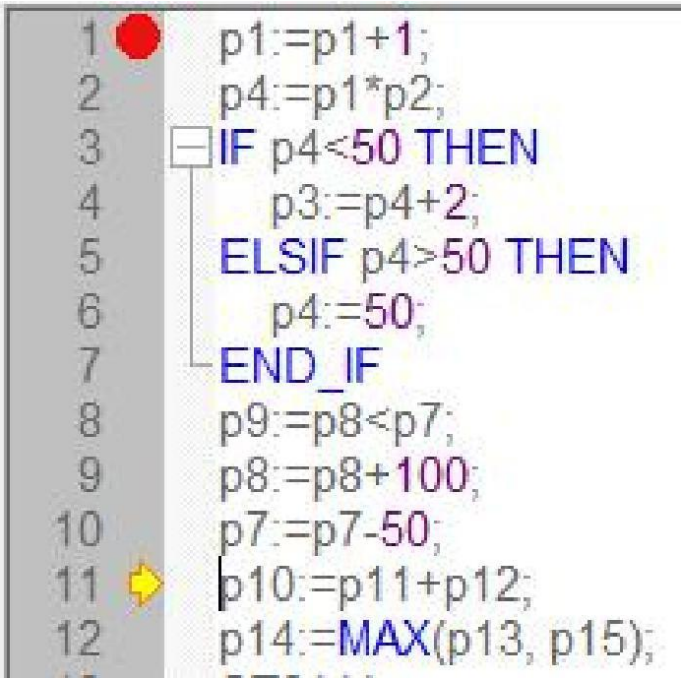
Follow the steps below to run to the cursor:

- Menu bar: Choose [Online] > [Run to Cursor].
- Toolbar:  .

13.7.8.4 Reference message

A ST program is taken as an example here. As shown in the figure, place the cursor on Line 11 where you want to stop, and execute "run to cursor". The program will start execution from Line 2 and stop at Line 11.





13.7.9 Show Current Statement

13.7.9.1 Introduction

When debugging a program, you can use this command to open the debugging window if the window is closed or to show the current statement if other statements are shown currently and then select the current execution position.

13.7.9.2 Requirement

This command is used during online debugging.

13.7.9.3 Steps

Follow the steps below to show the current statement:

- Menu bar: Choose [Online] - [Show Current Statement].
- Toolbar:  .

13.7.10 Call Stack

13.7.10.1 Introduction

This feature is used during debugging to view the list of historical call stacks of the currently debugged POU, function, or functional block.

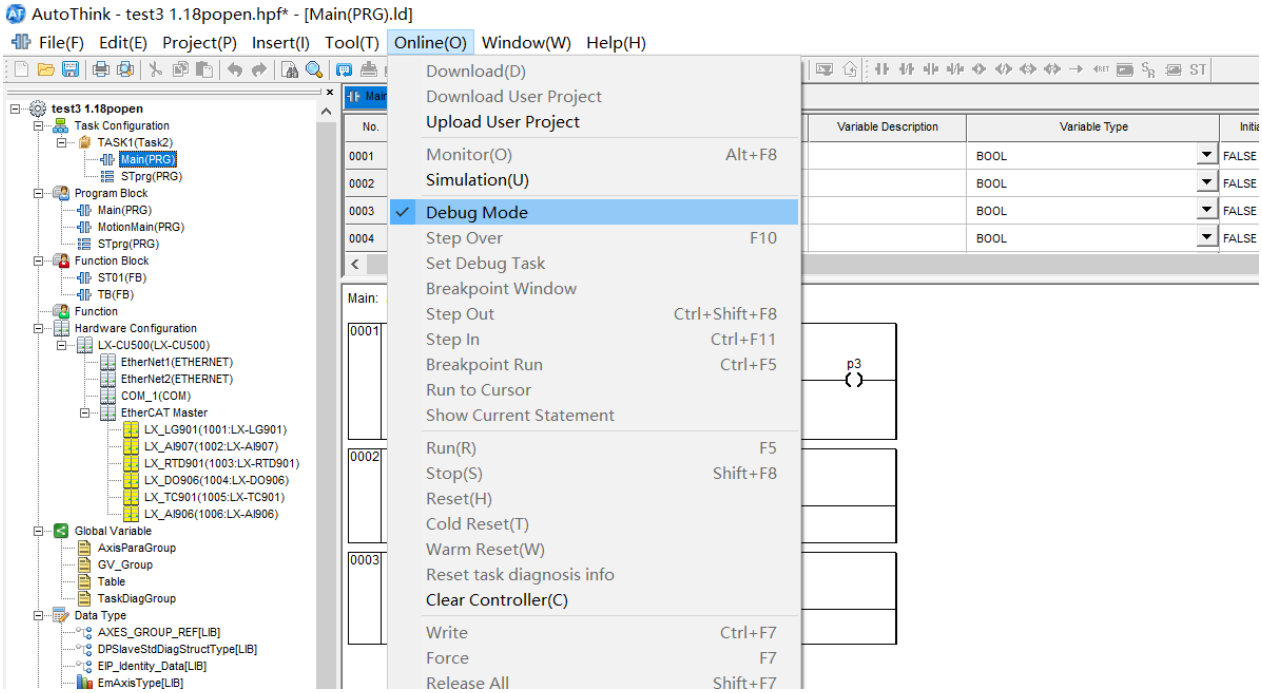
13.7.10.1.1 Requirement

This command is used during online debugging.

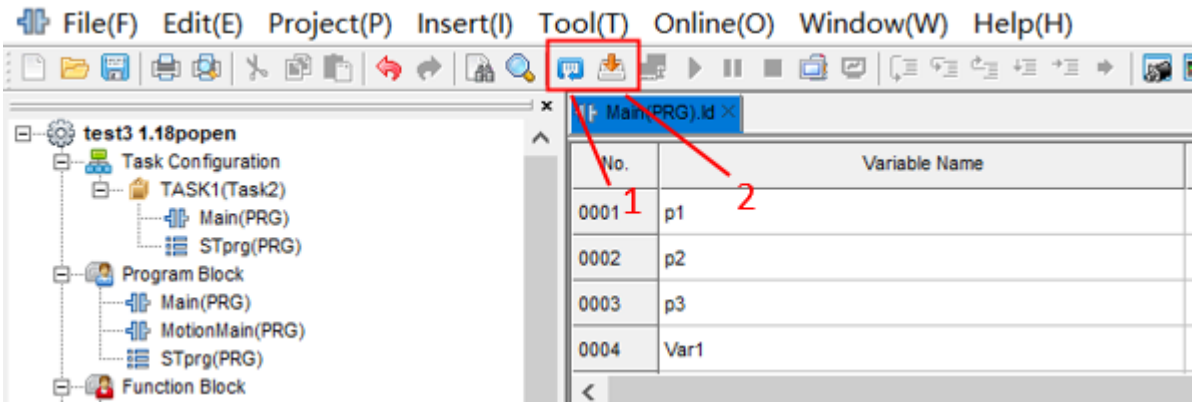
13.7.10.2 Steps

Follow the steps below to view the call stack:

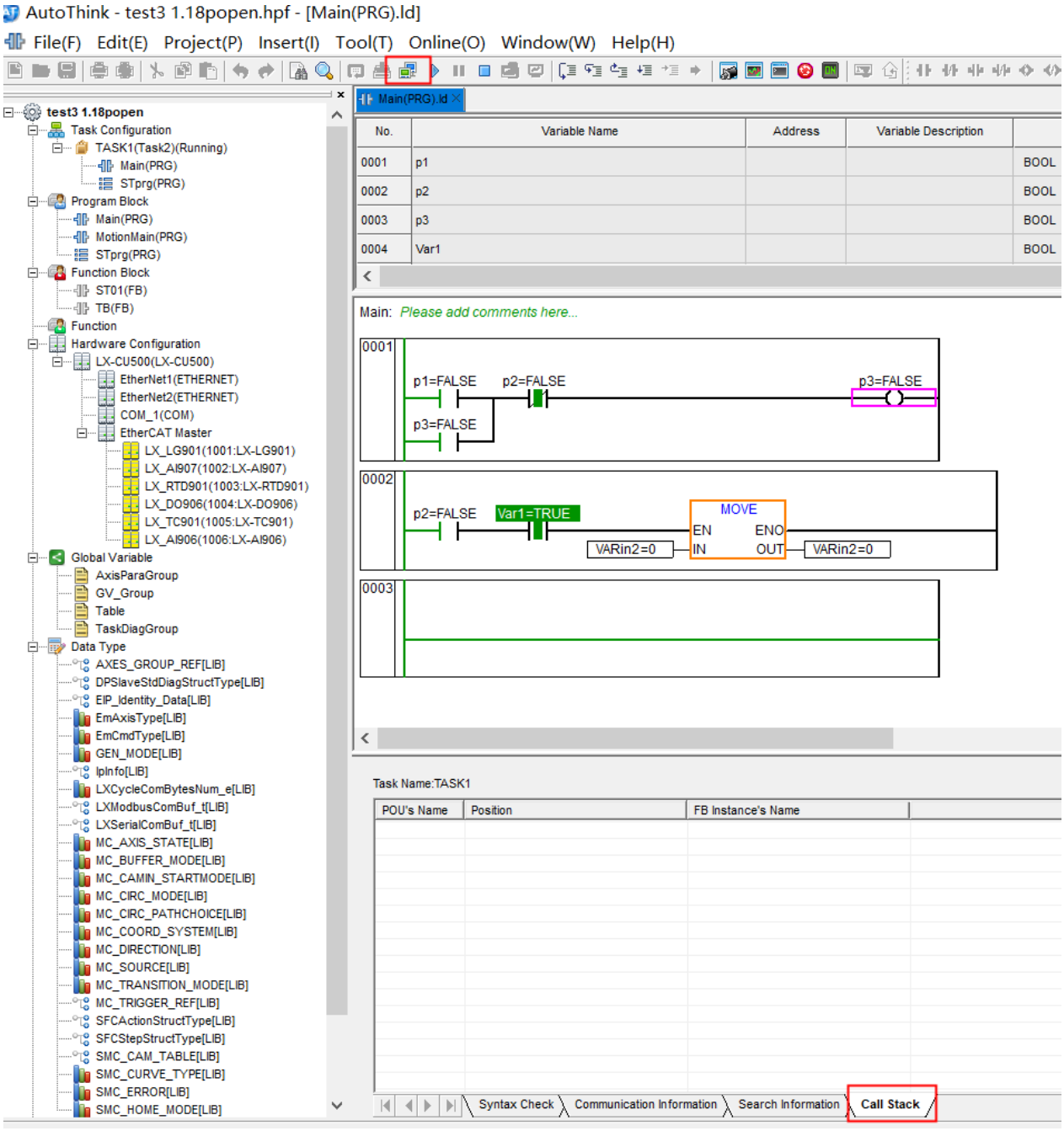
(1) Choose [Online] > [Debug Mode] in the menu bar.



(2) Click the compilation and download icons.



(3) Click the monitoring icon to view the call information on the call stack tab at the bottom.



13.8 Online Track

13.8.1 Introduction

The online tracking feature is used to obtain the values of a specified variable at a certain interval. When the tracking start condition is met, the system periodically collects the values of the specified variable at a set tracking interval until it receives the instruction to stop tracking. Tracking data can be uploaded to a

local path for data analysis.

Up to 10,000 tracking records are allowed for each variable. If this upper limit is reached, the records will be overwritten. Tracking is stopped immediately when any of the following conditions are met:

- The number of tracking operations reaches the set value.
- Stopping conditions are met.
- The Stop Tracking button is clicked.
- The Upload Tracking button is clicked.

13.8.2 Requirement

The current project has been installed.

FA-AutoThink is online.

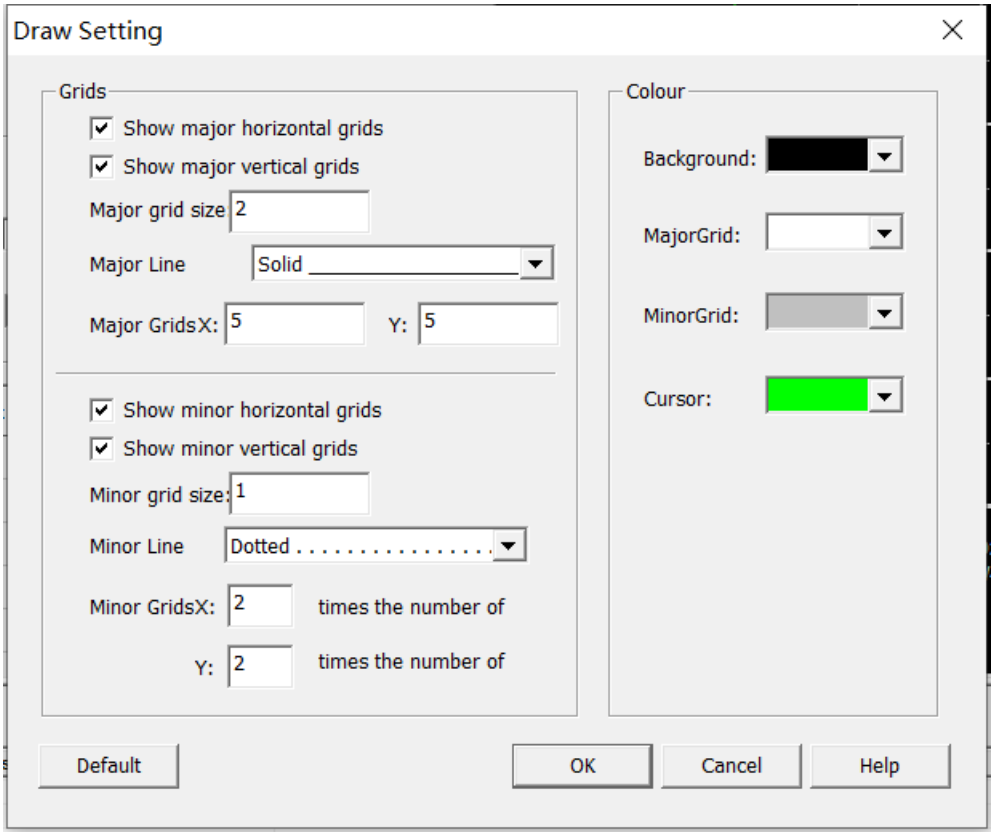
13.8.3 Steps

Follow the steps below to perform online tracking:

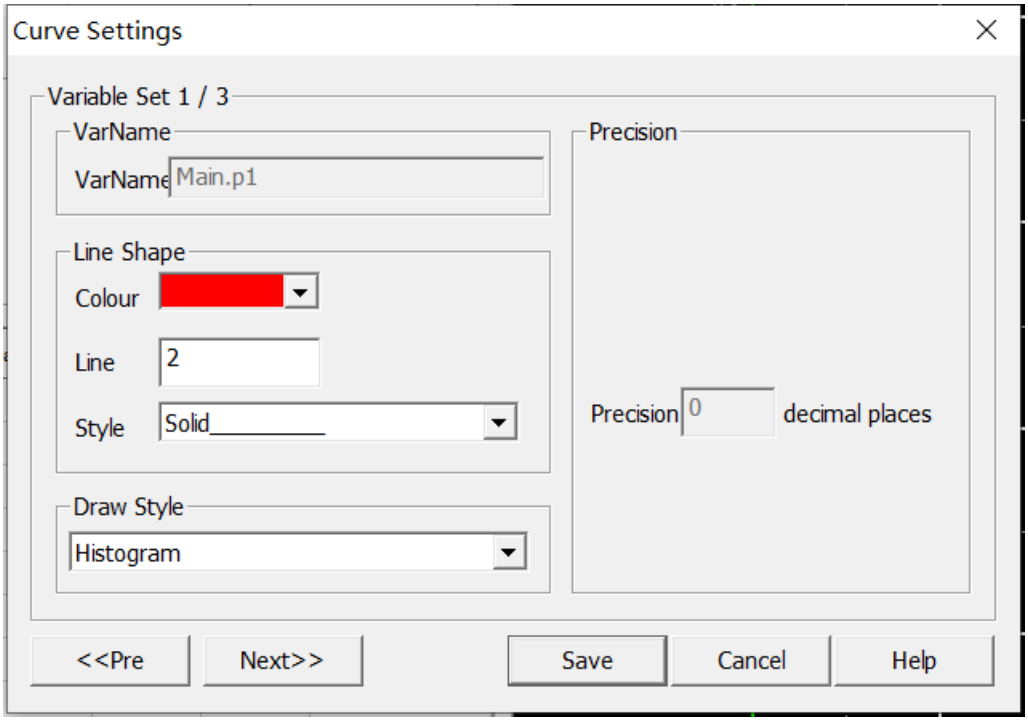
(1) Double-click "Online Track" to pop up the Online Track dialog box.

(2) Waveform area settings:

(a) Click Drawing Settings to change the main grid settings, as shown in the figure below:



(b) Click Curve Settings to set the curve specification and drawing method of the variable, as shown in the figure below:

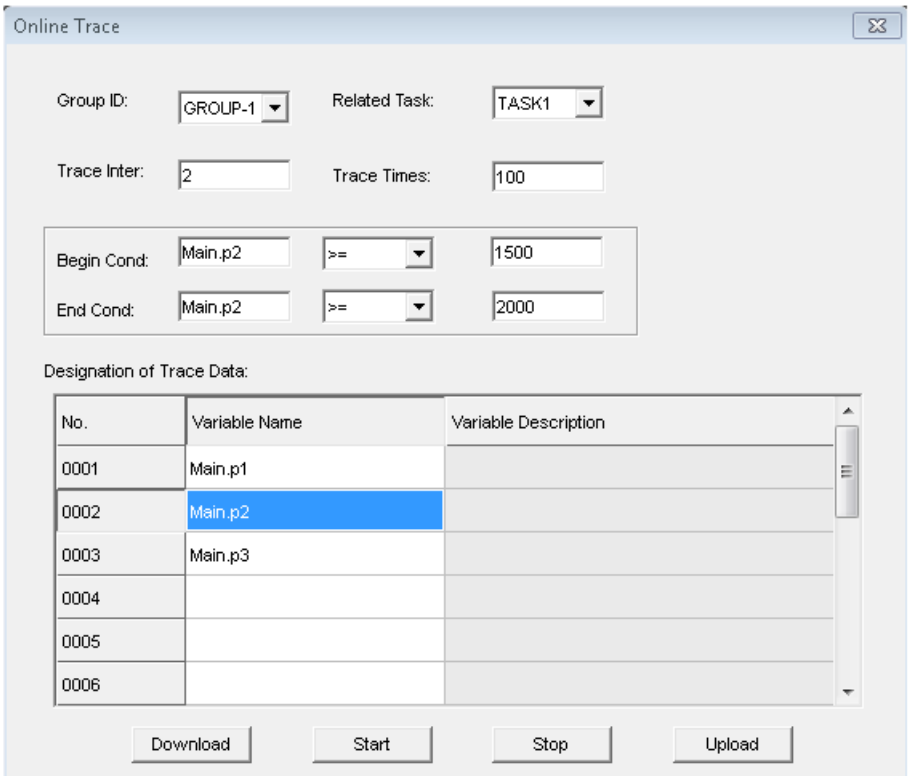


(3) Tracking type: The type can be Single (default type) or Continuous.

(4) Configure tracking:

As shown in the figure, the tracking interval is set to 2, and the task cycle is 50 ms, which means that tracking is performed every 100 ms. Set the number of tracking operations to 100. The tracking type is set to the default type Single. Tracking is started when the value of the variable Main.q2 is greater than or equal to 1000 and stopped when the value is greater than or equal to 1500.

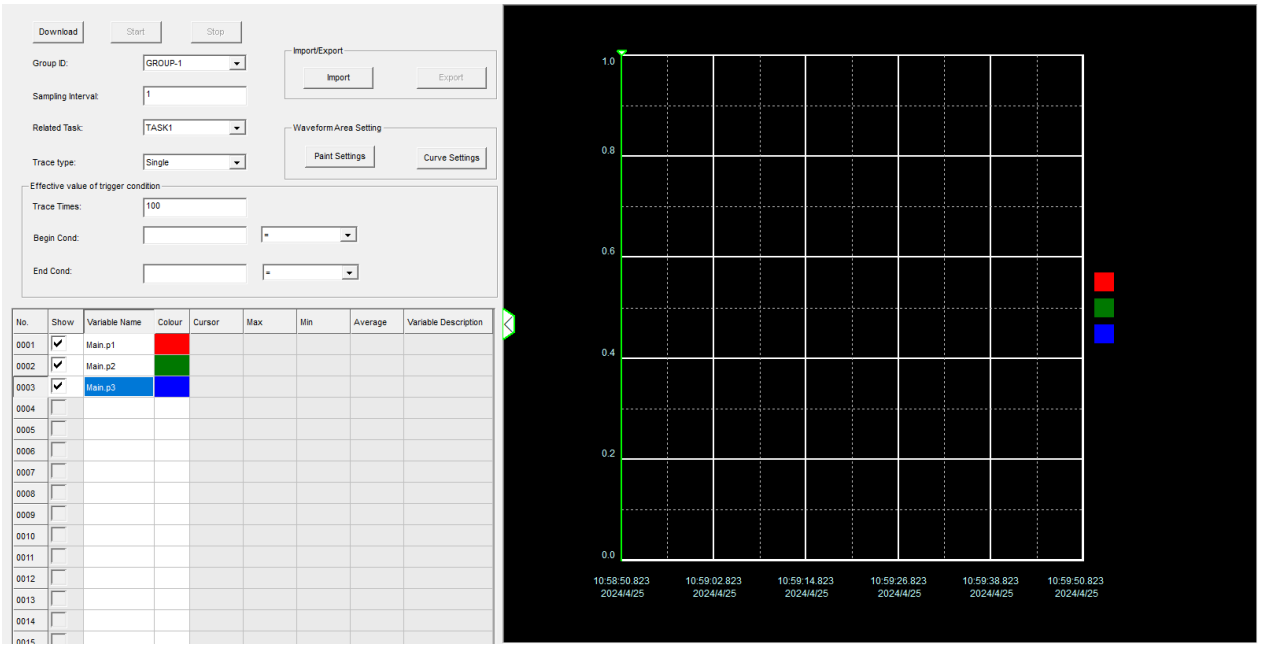
In the list below, press F2 to select and add the variable.



(5) Click Download Configuration to apply the configuration of the group GROUP-1.

(6) Click Start Tracking to start tracking.

(7) Click Export to pop up the data saving dialog box, and select a path to save the data.



13.9 Simulation

13.9.1 Introduction

In the simulation mode, the program block runs in the corresponding simulation CPU module of the local computer on the operating system. This mode can be used to check the project.

13.9.2 Requirement

AutoThink is running on the computer.

Configurations are completed according to project requirements.

13.9.3 Steps

Follow the steps below to perform simulation:

Menu bar: Choose [Online] - [Simulation].

Toolbar:  .

- Shortcut key: **Alt+F8**.

13.10 Write to SD Card

This feature is not supported.

13.11 Clear SD Card

This feature is not supported.

13.12 Clear Controller

13.12.1 Introduction

The feature can be used to clear the project in the PLC and re-initialize the PLC system when AutoThink is offline.

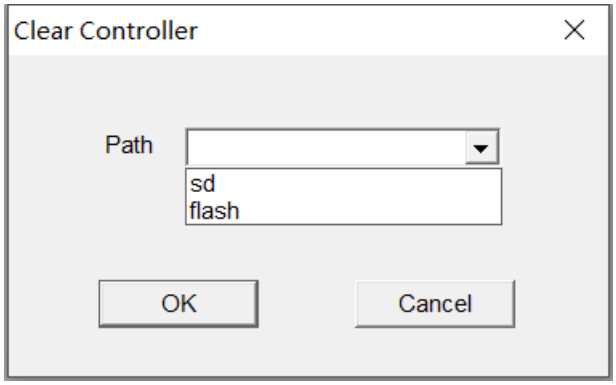
13.12.2 Requirement

AutoThink is offline.

13.12.3 Steps

Follow the steps below to clear the controller:

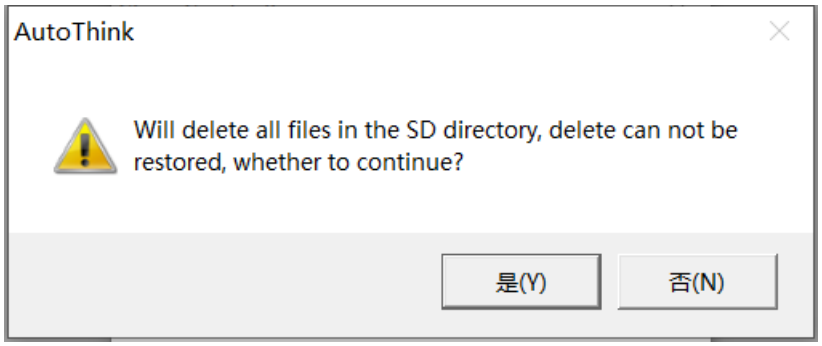
(1) Menu bar: Choose [Online] - [Clear Controller] to pop up the dialog box, as shown in the figure below.



(2) Select /media/sd/ to delete all files in the SD card, which cannot be restored.

(3) Select /media/flash/ to delete the running program, user program, and user files, which cannot be restored.

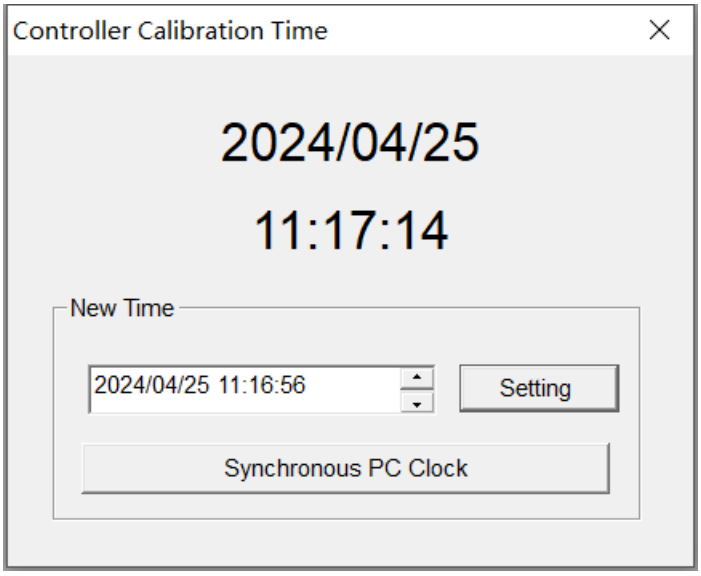
(4) Click OK after the path is selected to pop up the confirmation window, as shown in the figure below. Click Yes to perform deletion, which cannot be restored.



13.13 Controller Calibration Time

13.13.1 Introduction

This feature is available when AutoThink is online. In the pop-up window, the current controller time is displayed at the top, and the new time can be set at the bottom. You can enter the new time manually and then click Set to apply the time. You can also click Sync with PC Clock to apply the time on the PC. The window for setting the controller time is shown in the figure below.



13.13.2 Requirement

AutoThink is online.

13.13.3 Steps

Follow the steps below to set the controller time:

(1) Menu bar: Choose [Online] - [Controller Calibration Time].

13.13.4 Reference Message

The supported value ranges are:

- Year: 1971-2036
- Month: 1-12
- Day: 1-31
- Hour: 0-23
- Minute: 0-59
- Second: 0-59

13.14 Set NTP

13.14.1 Overview

When the LK controller is calibrated by NTP server, you need to enable NTP, set the IP address of the server and the calibration cycle.

Check NTP Enable to enable the timing function. The controller establishes timing connection through the IP address of the server and sends timing request periodically to keep it in synchronization with the server time.

13.14.2 Requirements

FA-AutoThink is offline

13.14.3 Steps

To set NTP calibration, follow the following steps:

(1) Click Online menu.

(2) Select [NTP settings] command.

The settings dialog box will pop up.

(3) Check the "NTP Enable" option.

(4) Enter the IP address of the NTP timing server.

(5) Please select a time value of hour, minute or second, and click the adjust button to increase or decrease the value.

You can also enter the time value manually. It can be set from 00:01:00 to 23:59:59.

(6) After setting NTP parameters, please download project to the controller.

At this time, NTP timing function takes effect.

13.15 View Operation Log

13.15.1 Introduction

You can use this command to open the Operation Logs window. The window displays the total number of logs and online operation information of the program, including the operation time and operation content. You can click the "+" icon to show the log details and click "-" to hide the details.

13.15.2 Requirement

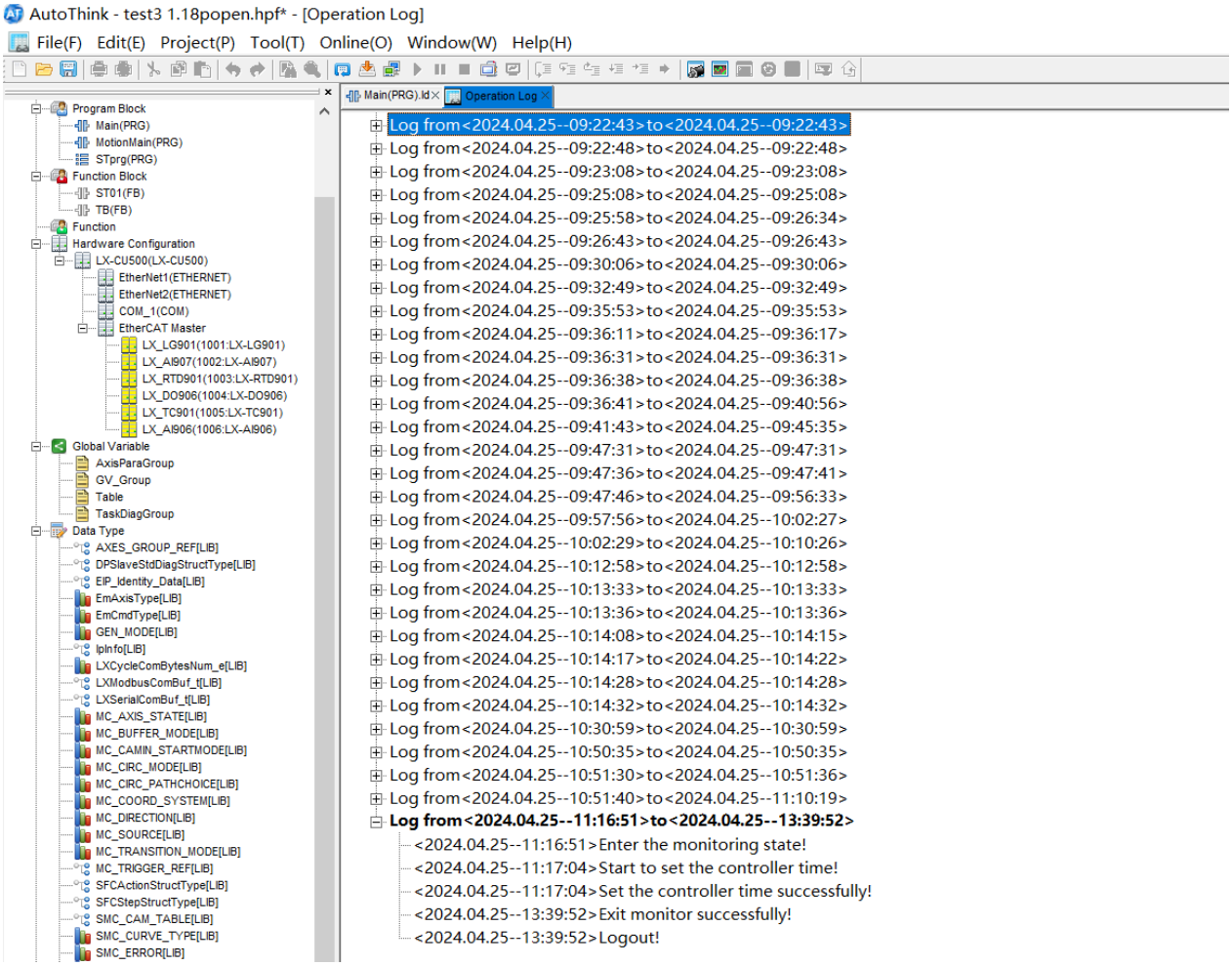
AutoThink is opened.

13.15.3 Steps

Follow the steps below to view operation logs:

(1) Choose [Project] - [View Operation Log] in the menu bar.

13.15.4 Result



13.15.5 Reference Message

The upper limit of operation logs is 5,000. When the limit is reached, a backup file will be generated. New logs can be recorded after original files are cleared. When the limit is reached again, a new backup file will be generated and will overwrite the previous backup file.

The operation log file is under the download directory and is named "project name.log", and the backup file is named "project name_bak.log".

13.16 Reset

13.16.1 Introduction

Reset simulates the state of the controller after it is powered off and then powered on again. The controller needs to restore the variables in the memory area for retaining data upon power-off to the values before the power-off and the variables in other memory areas to their initialized values. Forced variables will be released after the reset and restored to the corresponding values according to the storage area.

13.16.2 Requirement

AutoThink is in the monitoring or simulation mode.

13.16.3 Steps

Follow the steps below to perform the reset operation:

(1) Menu bar: Choose [Online] - [Reset].

13.17 Cold Reset

13.17.1 Introduction

Cold reset simulates the state after initial download is performed on the project.

13.17.2 Requirement

AutoThink is in the monitoring or simulation mode.

13.17.2.1 Steps

Follow the steps below to perform the cold reset operation:

Menu bar: Choose [Online] - [Cold Reset].

13.17.3 Result

Cold reset can be performed when variables in each memory area need to be restored to the initial values after the controller installs the program. Clod reset can be performed to restore values of forced variables to the initial values and release the forced variables.

13.18 Warm Reset

- Menu bar: Click [Online] - [Warm Reset].

Warm reset is the simulation status remained before reset in each data area after reset project. After start the warm reset function, re-load the user project logic, and data of each memory area remain unchanged.

13.19 Reset Task Diagnosis Info

13.19.1 Overview

Clear the data of the TASK task diagnostic variable group.

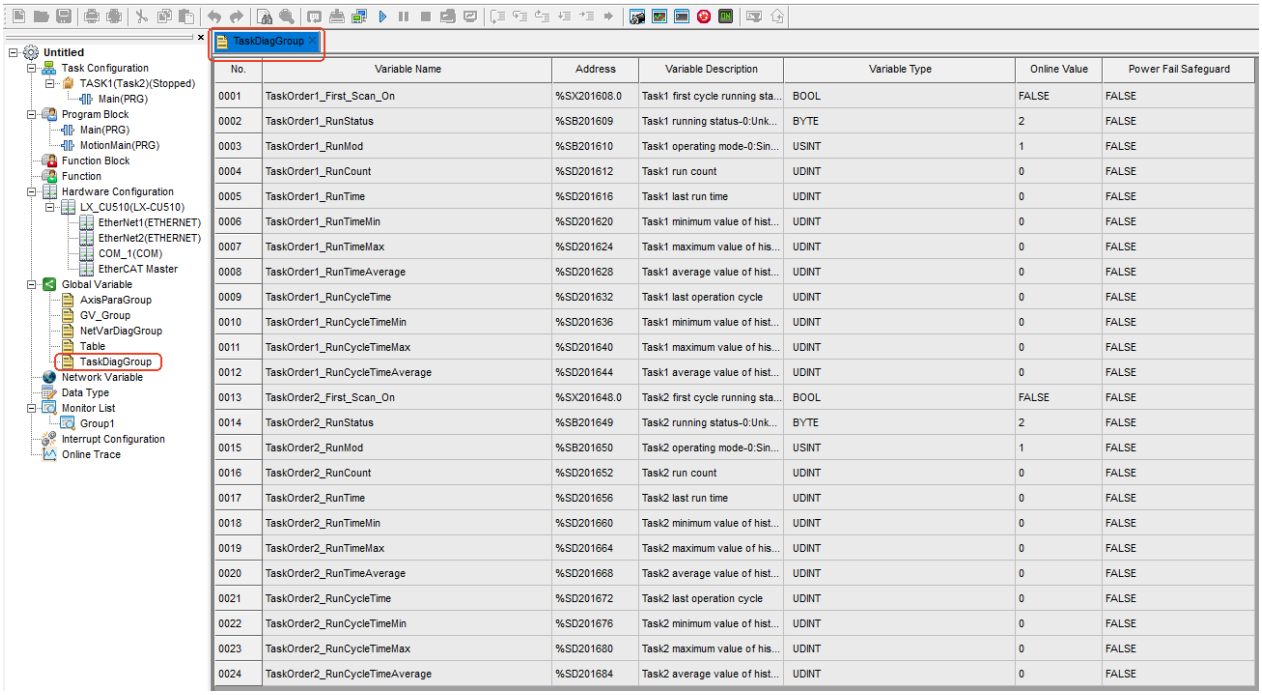
13.19.2 Requirement

- AT software is in "Monitoring" mode.
- The controller is connected.

13.19.3 Steps

Execute by following these methods:

Menu Bar: Click [Online]-[Reset Task Diagnosis Info].



No.	Variable Name	Address	Variable Description	Variable Type	Online Value	Power Fail Safeguard
0001	TaskOrder1_First_Scan_On	%SX201608.0	Task1 first cycle running sta...	BOOL	FALSE	FALSE
0002	TaskOrder1_RunStatus	%SB201609	Task1 running status-0:Unk...	BYTE	2	FALSE
0003	TaskOrder1_RunMod	%SB201610	Task1 operating mode-0:Sin...	USINT	1	FALSE
0004	TaskOrder1_RunCount	%SD201612	Task1 run count	UDINT	0	FALSE
0005	TaskOrder1_RunTime	%SD201616	Task1 last run time	UDINT	0	FALSE
0006	TaskOrder1_RunTimeMin	%SD201620	Task1 minimum value of hist...	UDINT	0	FALSE
0007	TaskOrder1_RunTimeMax	%SD201624	Task1 maximum value of his...	UDINT	0	FALSE
0008	TaskOrder1_RunTimeAverage	%SD201628	Task1 average value of hist...	UDINT	0	FALSE
0009	TaskOrder1_RunCycleTime	%SD201632	Task1 last operation cycle	UDINT	0	FALSE
0010	TaskOrder1_RunCycleTimeMin	%SD201636	Task1 minimum value of hist...	UDINT	0	FALSE
0011	TaskOrder1_RunCycleTimeMax	%SD201640	Task1 maximum value of his...	UDINT	0	FALSE
0012	TaskOrder1_RunCycleTimeAverage	%SD201644	Task1 average value of hist...	UDINT	0	FALSE
0013	TaskOrder2_First_Scan_On	%SX201648.0	Task2 first cycle running sta...	BOOL	FALSE	FALSE
0014	TaskOrder2_RunStatus	%SB201649	Task2 running status-0:Unk...	BYTE	2	FALSE
0015	TaskOrder2_RunMod	%SB201650	Task2 operating mode-0:Sin...	USINT	1	FALSE
0016	TaskOrder2_RunCount	%SD201652	Task2 run count	UDINT	0	FALSE
0017	TaskOrder2_RunTime	%SD201656	Task2 last run time	UDINT	0	FALSE
0018	TaskOrder2_RunTimeMin	%SD201660	Task2 minimum value of hist...	UDINT	0	FALSE
0019	TaskOrder2_RunTimeMax	%SD201664	Task2 maximum value of his...	UDINT	0	FALSE
0020	TaskOrder2_RunTimeAverage	%SD201668	Task2 average value of hist...	UDINT	0	FALSE
0021	TaskOrder2_RunCycleTime	%SD201672	Task2 last operation cycle	UDINT	0	FALSE
0022	TaskOrder2_RunCycleTimeMin	%SD201676	Task2 minimum value of hist...	UDINT	0	FALSE
0023	TaskOrder2_RunCycleTimeMax	%SD201680	Task2 maximum value of his...	UDINT	0	FALSE
0024	TaskOrder2_RunCycleTimeAverage	%SD201684	Task2 average value of hist...	UDINT	0	FALSE

Chapter 14 Assistant Tool

14.1 Assistant Tool

14.1.1 Connect to Controller

14.1.1.1 Introduction

The Controller Operations tool is used independently of AutoThink to establish a communication connection with the controller and perform relevant online operations on the controller. This tool and AutoThink cannot be used to connect to the controller at the same time. Therefore, make sure AutoThink is offline before you use this tool.

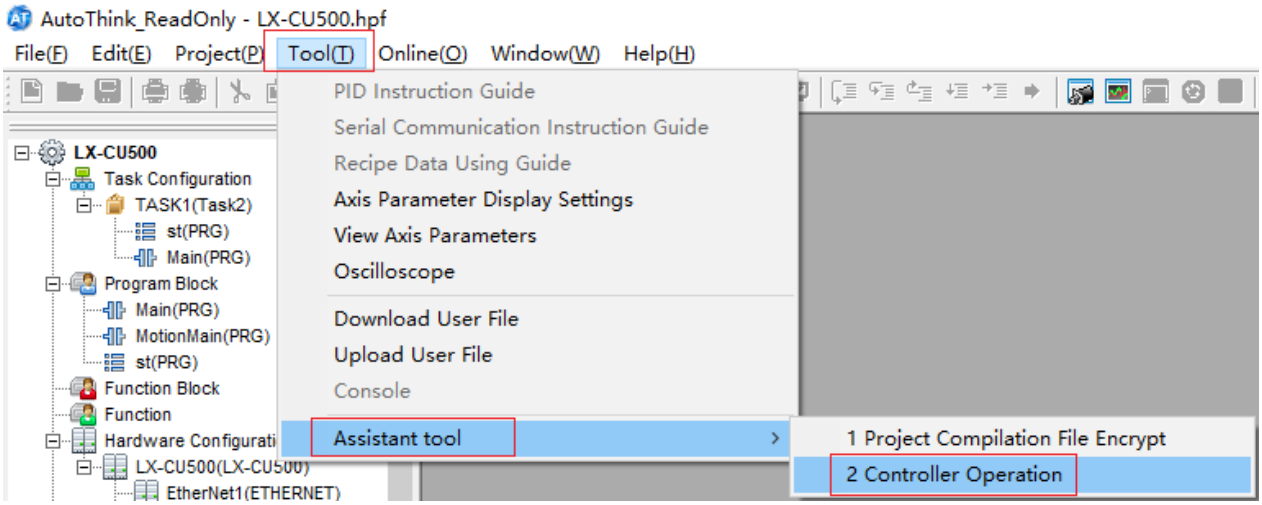
14.1.1.2 Requirement

AutoThink is offline.

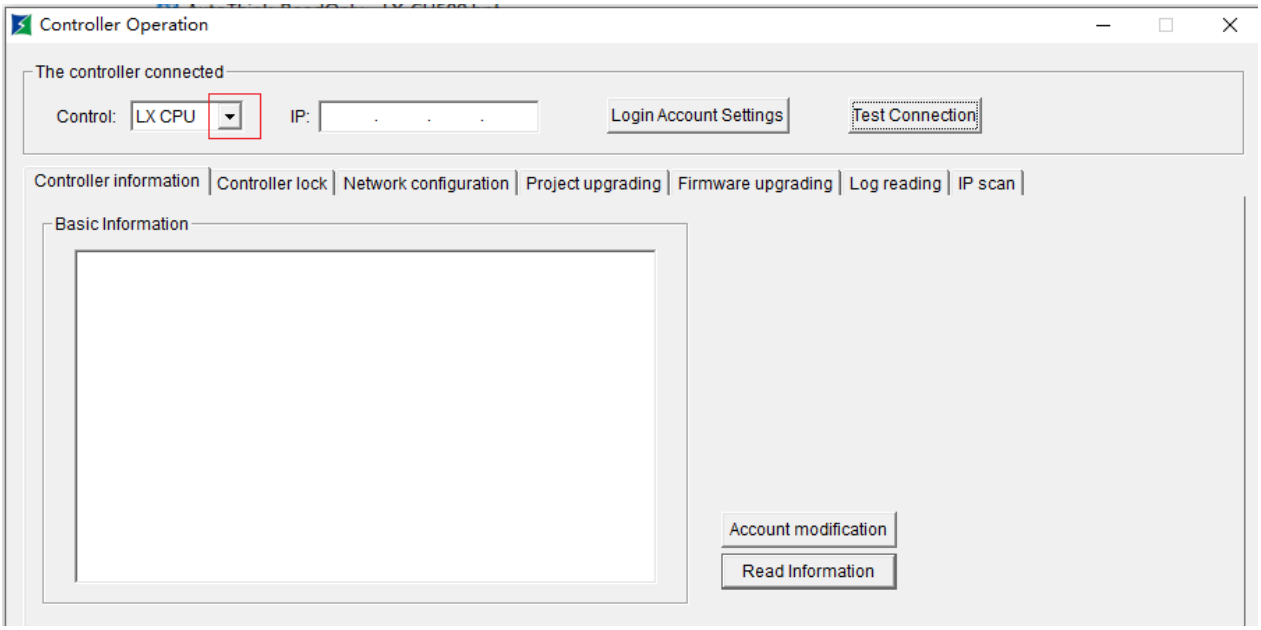
14.1.1.3 Steps

Follow the steps below to connect to the controller:

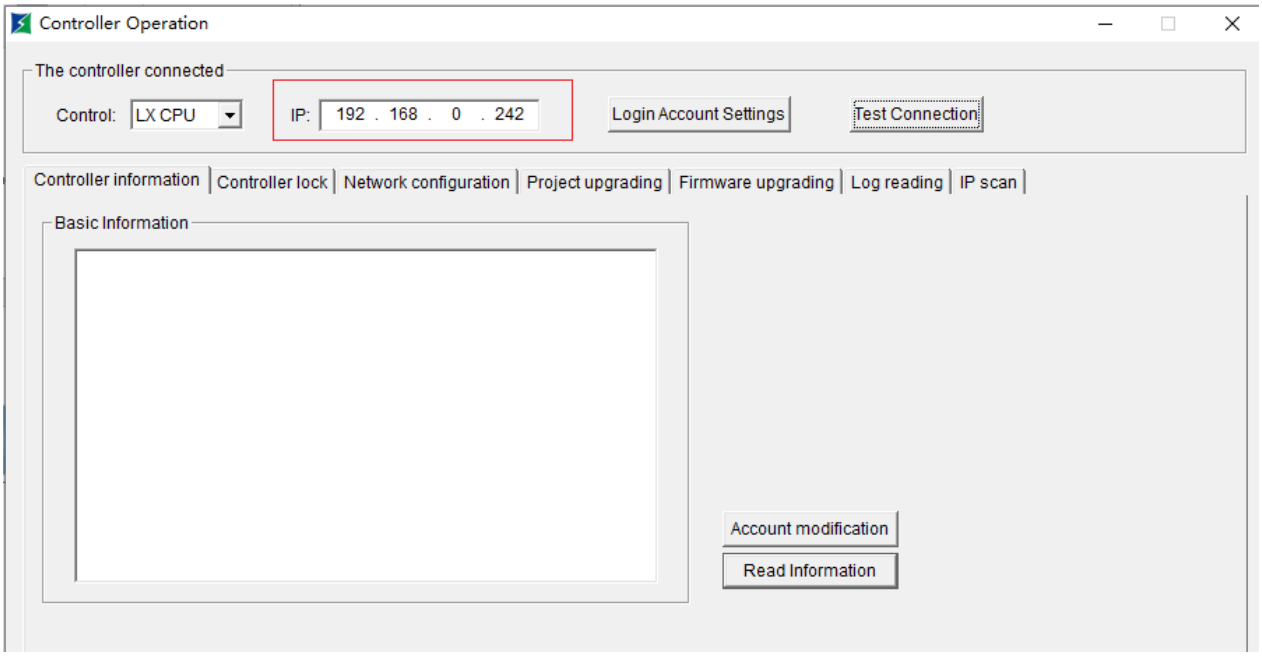
(1) Choose [Tool] > [Assistant tool] - [Controller Operation] in the menu bar;



(2) Select the controller type;

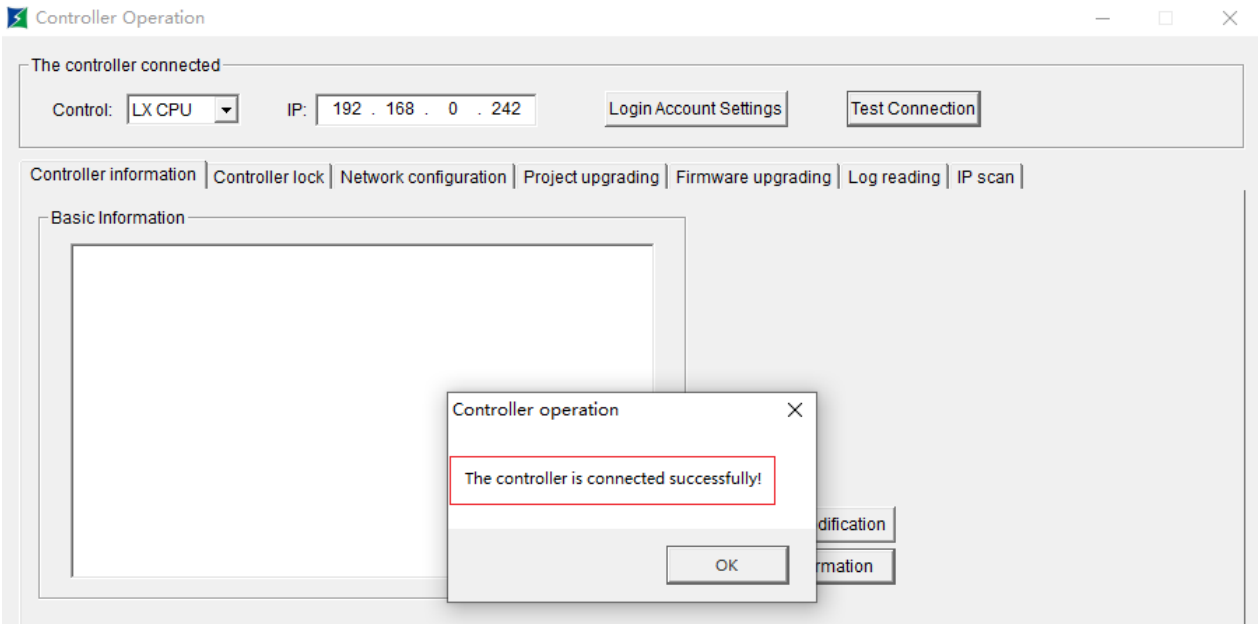


(3) Enter the IP address;



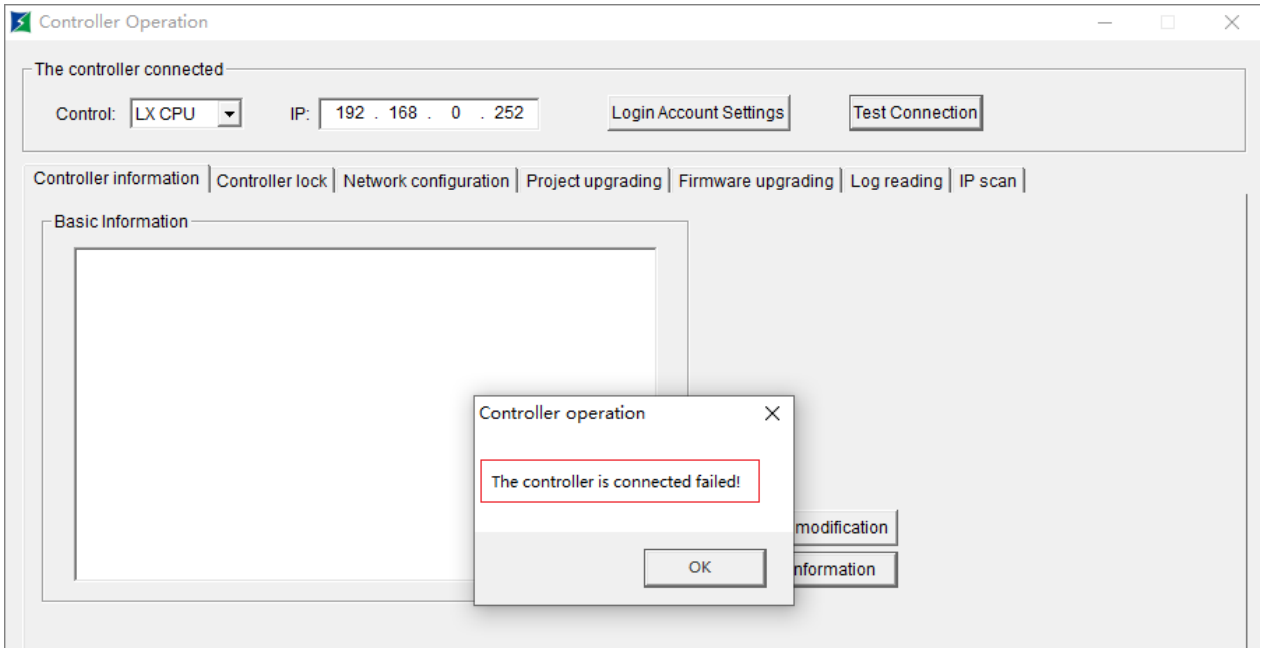
(4) Set the IP address of the local PC. Make sure that the computer and the controller are in the same network segment;

(5) Test the connection.



14.1.1.4 Result

If the connection is successful, the dialog box indicating that the controller connection is successful appears. If the connection fails, check whether the IP address is correct and whether the computer is in the same network segment as the controller.



14.1.2 Log in to Controller

14.1.2.1 Introduction

You need to log in to the controller before you can perform online operations if you connect to the controller using AutoThink or the Controller Operations tool.

14.1.2.2 Requirement

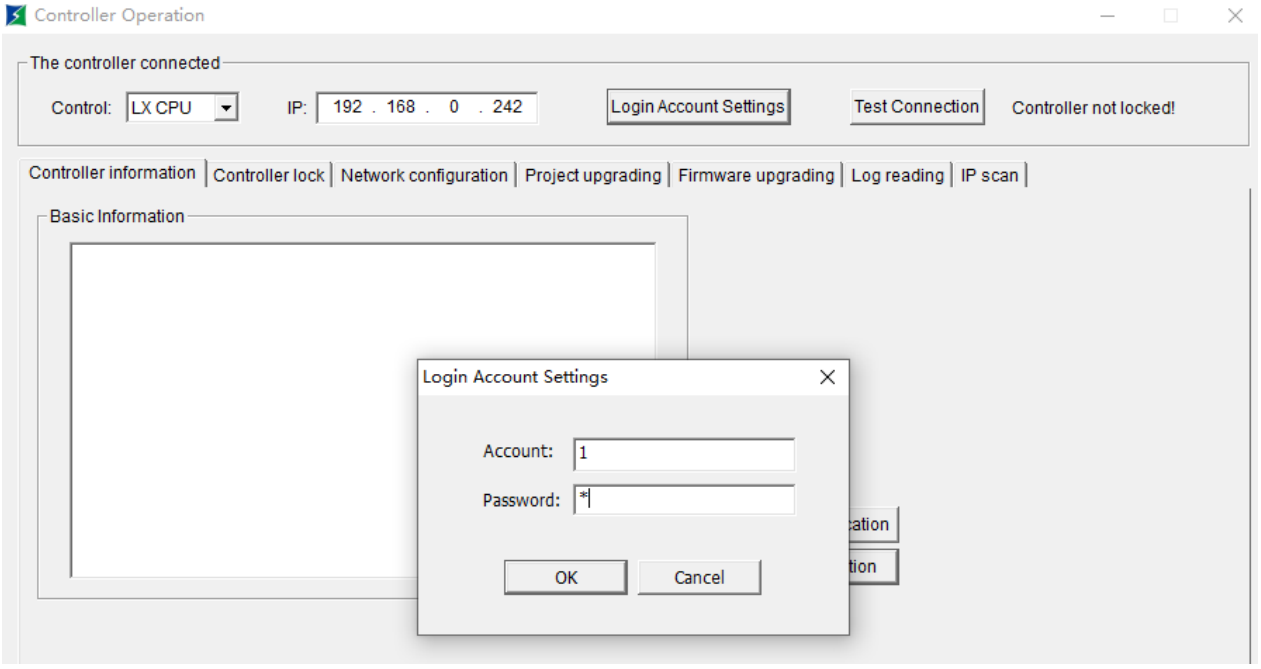
The Controller Operation tool is connected to the controller successfully.

14.1.2.3 Steps

If you are logging in to a new PLC for the first time, you need to set the account and password. You only need to set them once.

Follow the steps below to log in to the controller:

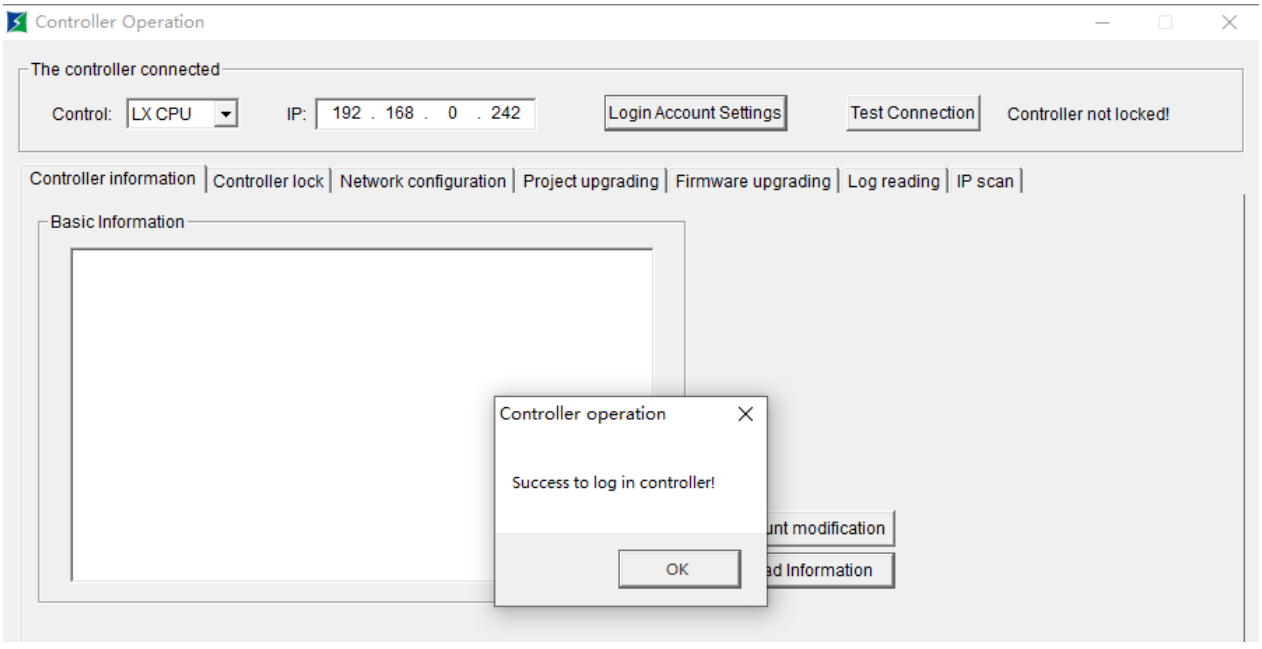
(1) Click Login Account Settings in the Controller Operations window.



(2) Enter the correct username and password in the Login Account Settings dialog box, and click OK.

14.1.2.4 Result

Log in to the controller successfully



After you log in to the controller successfully, you can click OK to close the dialog box. If login fails, check whether the username and password are correct.

14.1.2.5 Reference Message

You need to log in to the controller before you use the following features if the tool is used.

Project upgrading

Firmware upgrading

Log reading

14.1.3 Retrieve Controller Information

14.1.3.1 Introduction

You can click the corresponding button on the Controller Information tab to retrieve the version information on the current controller.

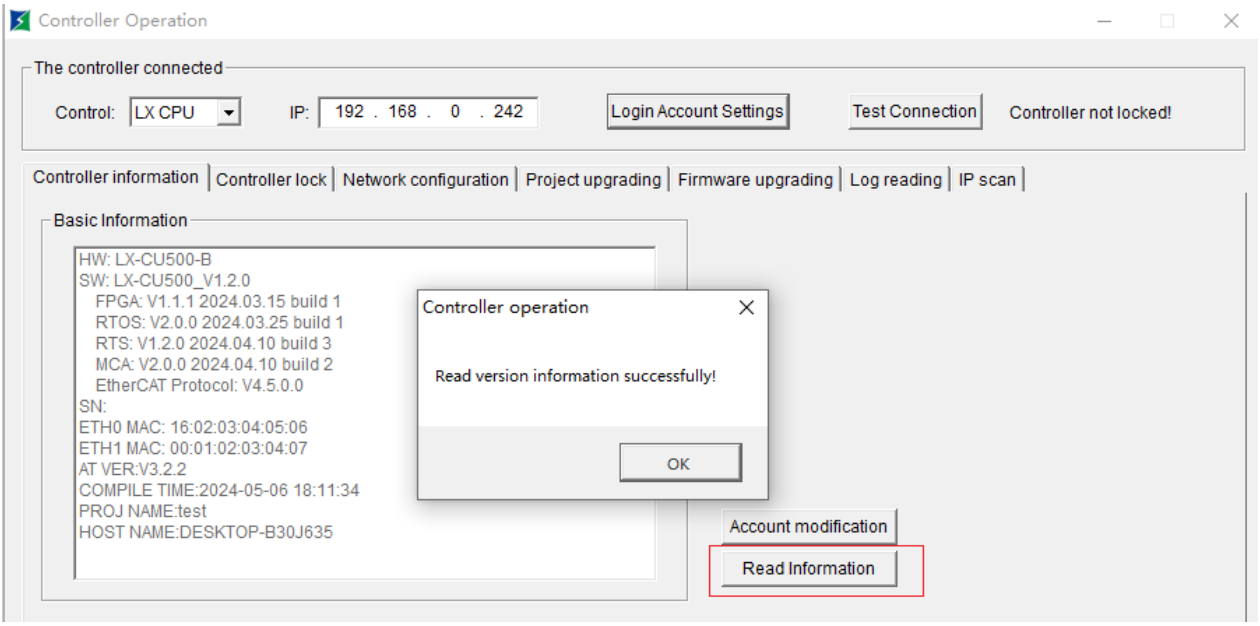
14.1.3.2 Requirement

A normal connection with the controller is established using the tool.

14.1.3.3 Steps

Follow the steps below to retrieve the controller information:

(1) Click Read Information in the Controller Operations dialog box.



14.1.3.4 Result

If the retrieval is successful, the version information of the controller will be displayed in the Controller Operations dialog box.

14.1.4 Lock/Unlock Controller

14.1.4.1 Lock Controller

14.1.4.1.1 Introduction

You can lock the controller to prevent operations on the controller when it is offline, including controller upgrade, user file download, download, user project upload, user project download, monitoring, controller clearing, IP modification, project upgrade, and firmware upgrade.

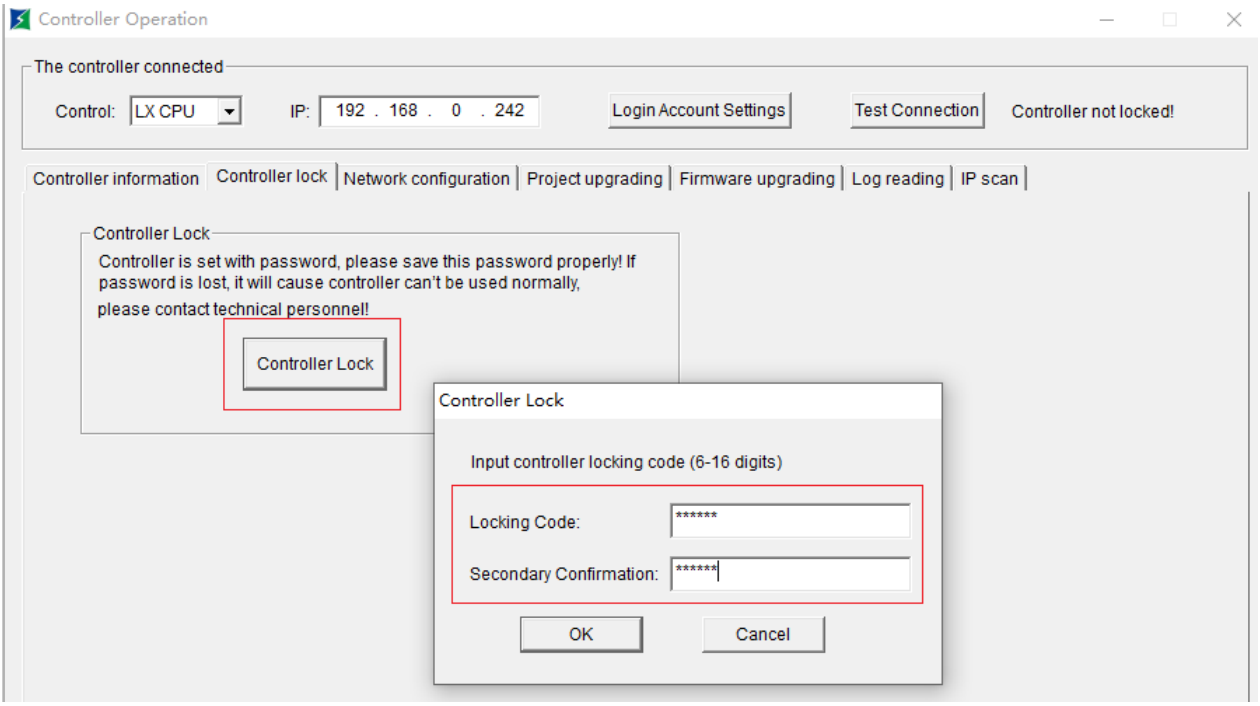
14.1.4.2 Requirement

A normal connection with the controller is established using the tool.

14.1.4.2.1 Steps

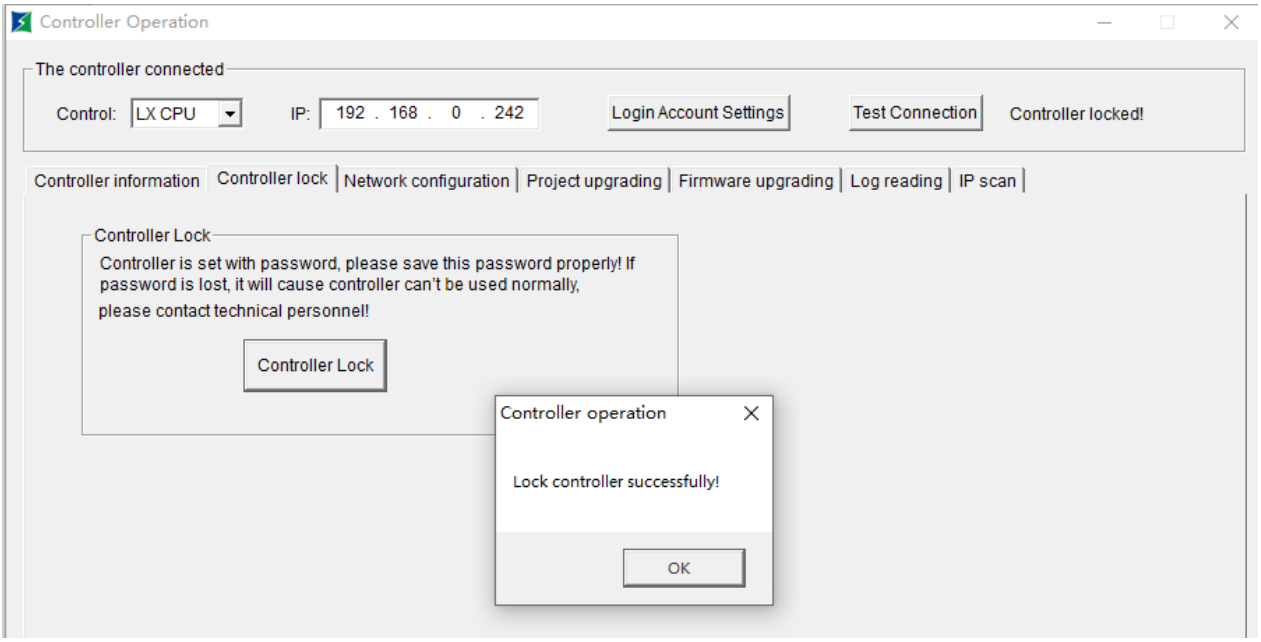
Follow the steps below to lock the controller:

(1) On the Controller Lock tab in the Controller Operations dialog box, click Controller Lock, enter the lock code with 6-16 characters twice in the dialog box that appears, and click OK.



14.1.4.2.2 Result

Lock controller successfully.



14.1.4.3 Unlock Controller

14.1.4.3.1 Introduction

Operations cannot be performed on the locked controller. You need to unlock it first.

14.1.4.3.2 Requirement

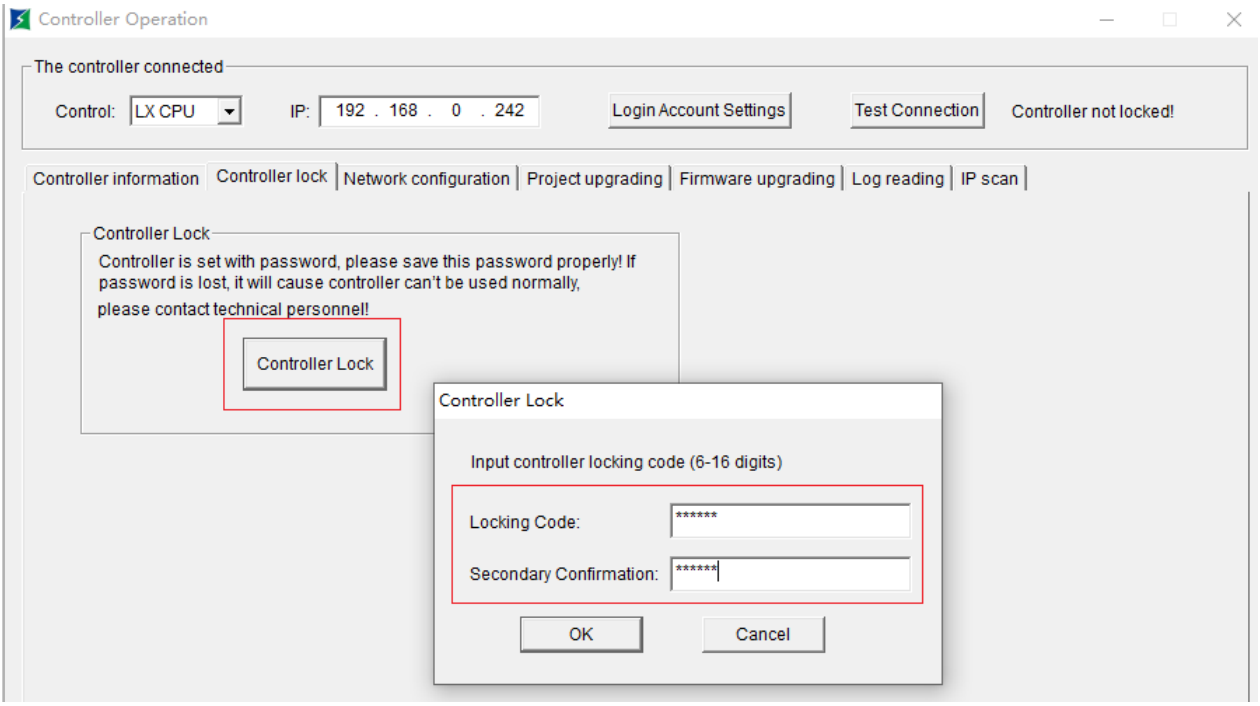
The connection with the controller is established successfully.

The controller is locked.

14.1.4.3.3 Steps

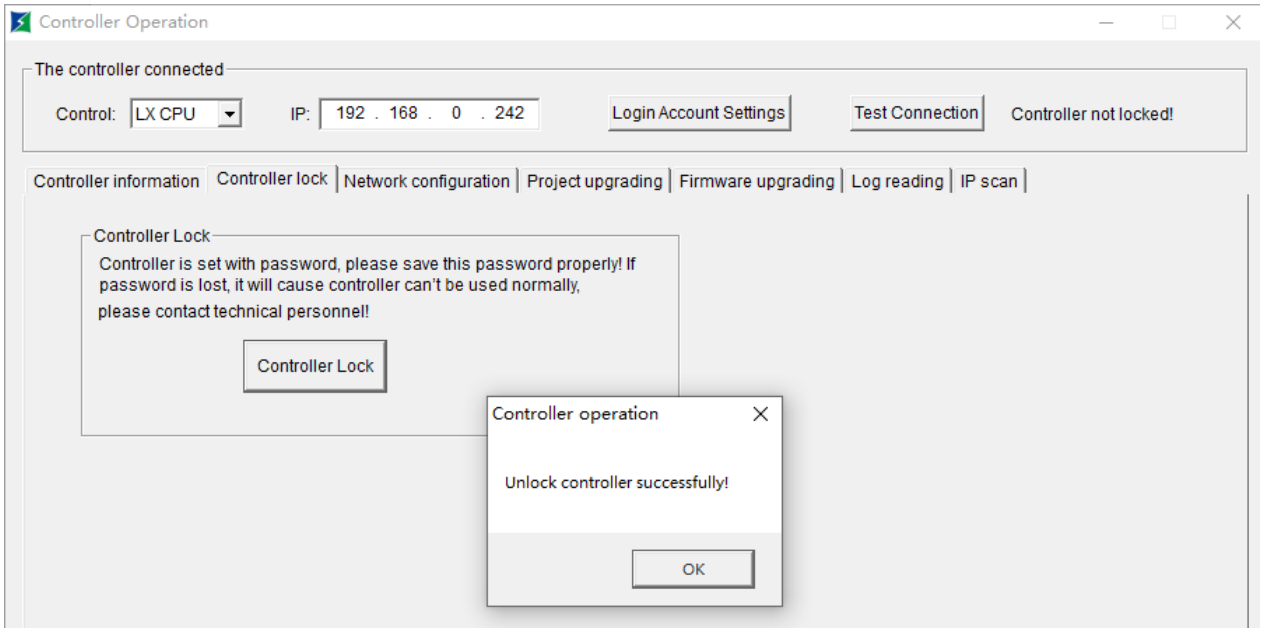
Follow the steps below to unlock the controller:

(1) On the Controller Lock tab in the Controller Operations dialog box, click Controller Lock, enter the unlock code with 6-16 characters in the dialog box that appears, and click OK.



14.1.4.4 Result

The controller is unlocked successfully. You can click OK to close the dialog box.



14.1.5 Modify Controller IP

14.1.5.1 Introduction

To avoid IP conflicts between the PLC and other devices, you can retrieve the controller IP address using the auxiliary tool, modify the address, and then write it to the controller.

14.1.5.2 Requirement

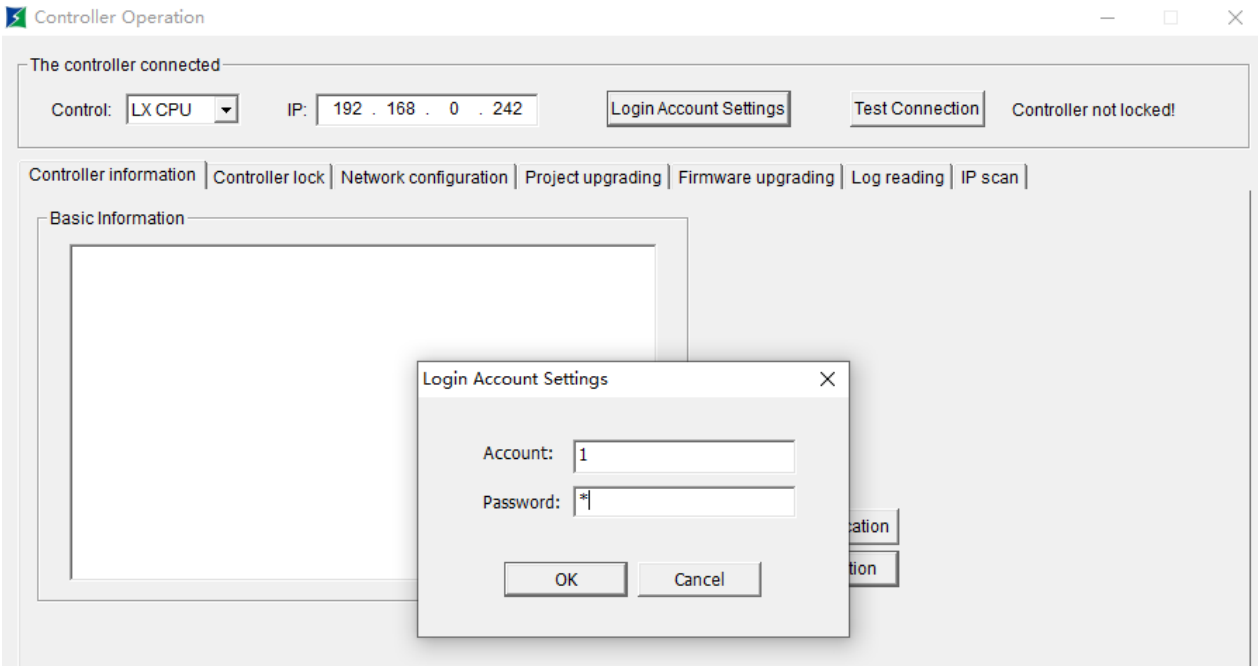
Make sure that a normal connection with the controller is established using the auxiliary tool and that you have logged in to the controller.

Make sure that the controller is in stand-alone mode and that the project is stopped before you modify the IP address.

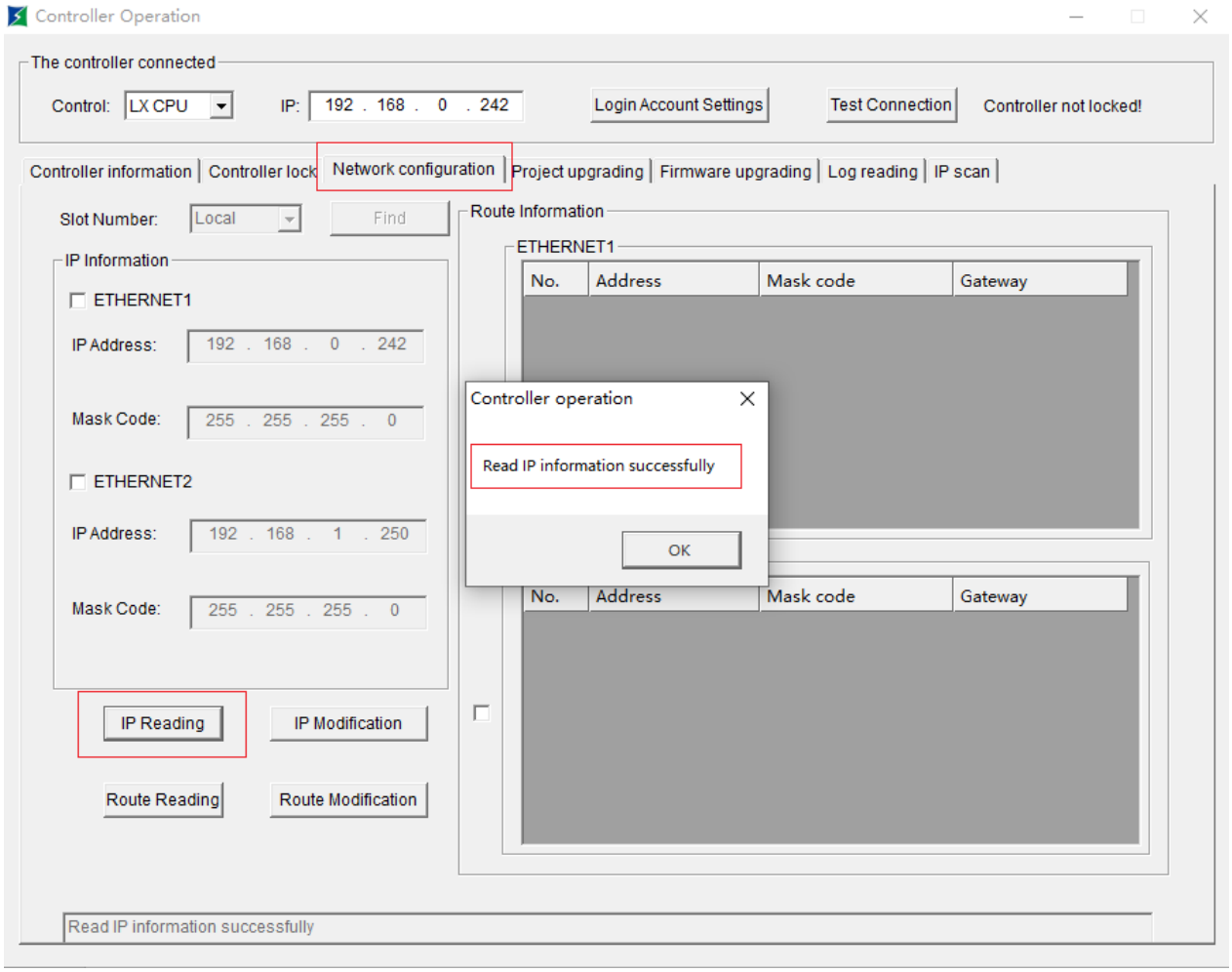
14.1.5.3 Steps

Follow the steps below to modify the controller IP address:

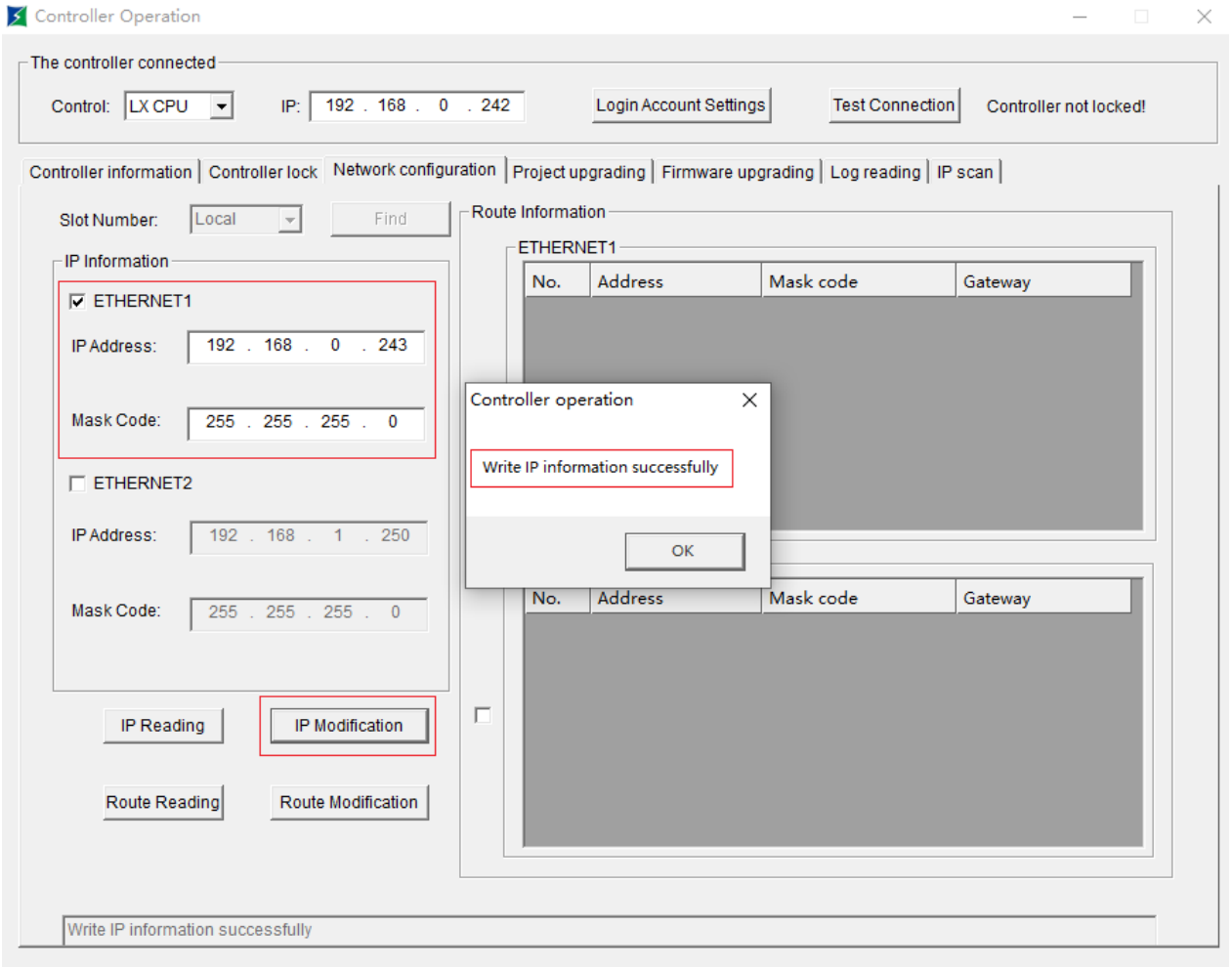
(1) In the Controller Operations dialog box, enter the correct IP address, click Login Account Settings, enter the correct username and password in the dialog box that appears, and click OK;



(2) In the Controller Operations dialog box, click the Network Configuration tab, and click IP Reading to retrieve the IP address of the controller;



(3) Check the interface requiring IP modification, modify the IP address, and click Modify IP.



14.1.5.4 Result

After the modification is completed, use the new IP address to connect to the controller.

14.1.6 Configure Route

14.1.6.1 Introduction

You need to configure the route so that the controller can communicate with other devices across network segments.

14.1.6.2 Requirement

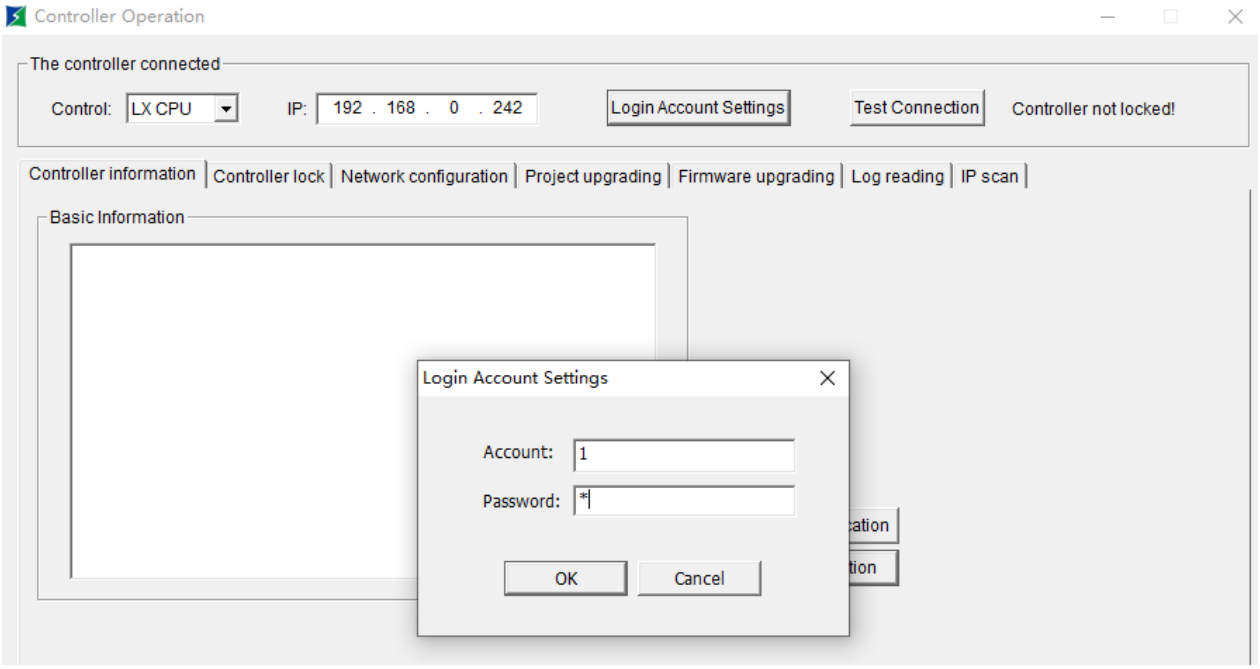
A normal connection with the controller is established using the tool, and you have logged in to the controller.

The controller is in stand-alone mode, and the project is stopped.

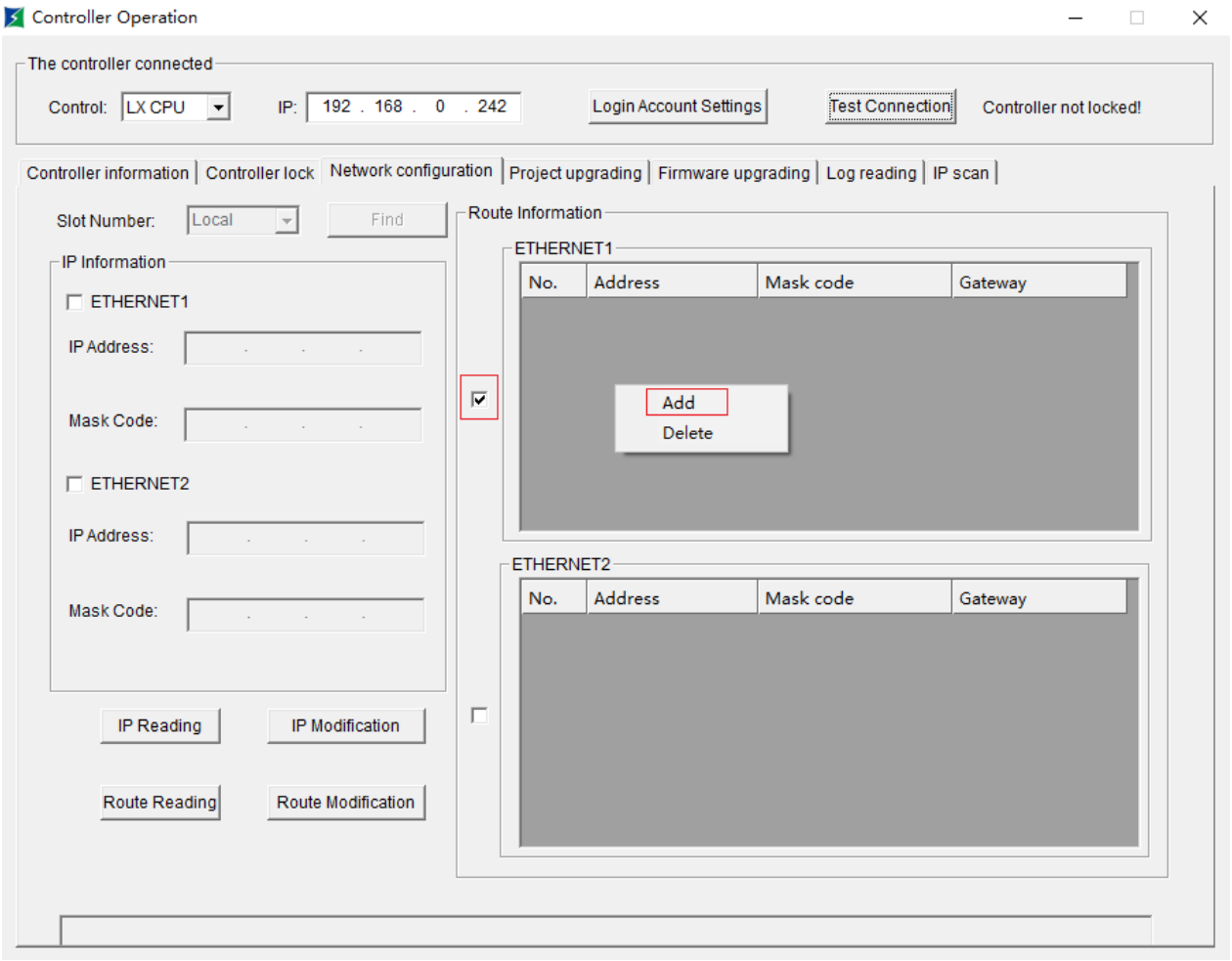
14.1.6.3 Steps

Follow the steps below to configure the route:

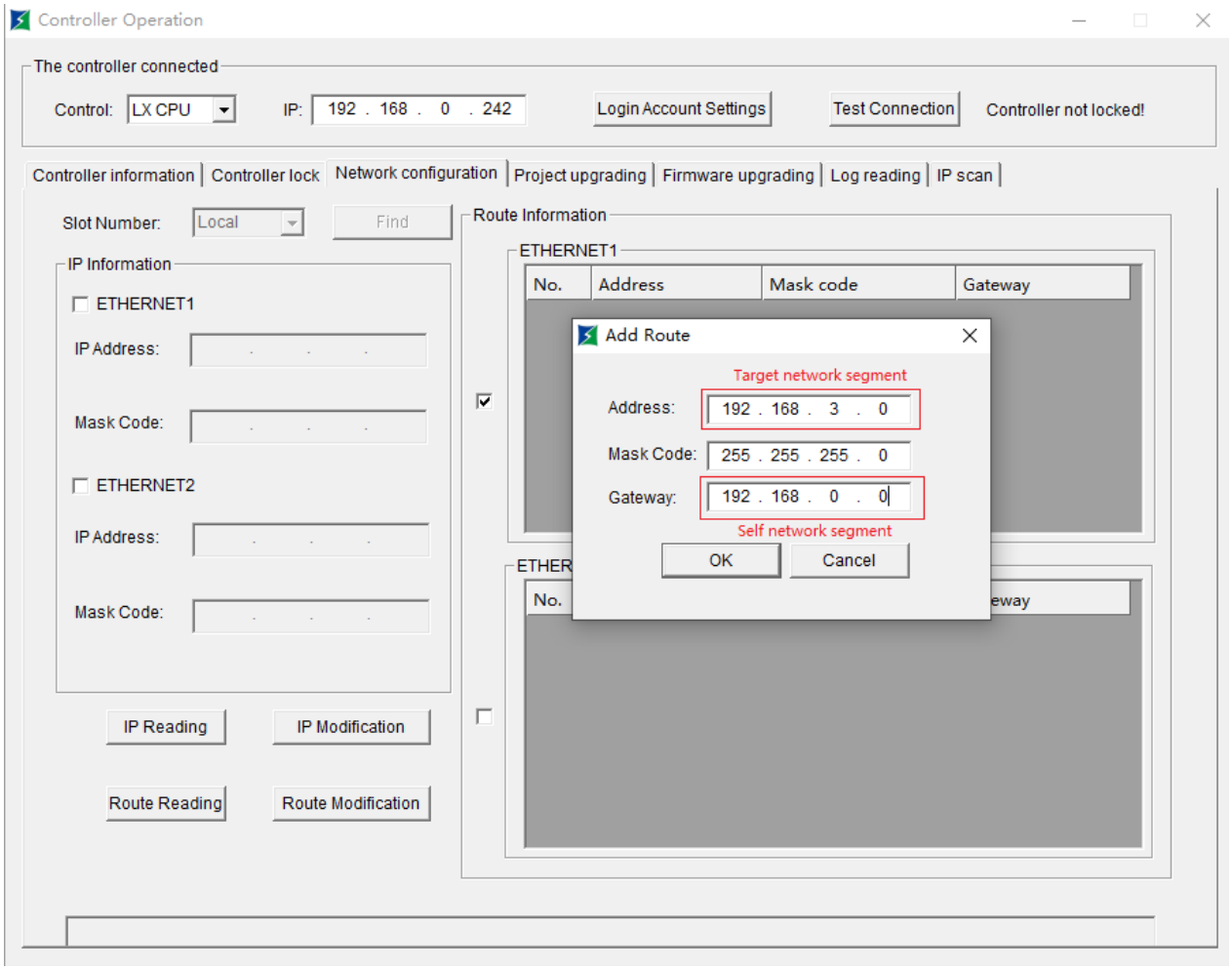
(1) In the Controller Operations dialog box, enter the correct IP address, click Login Account Settings, enter the correct username and password in the dialog box that appears, and click OK;



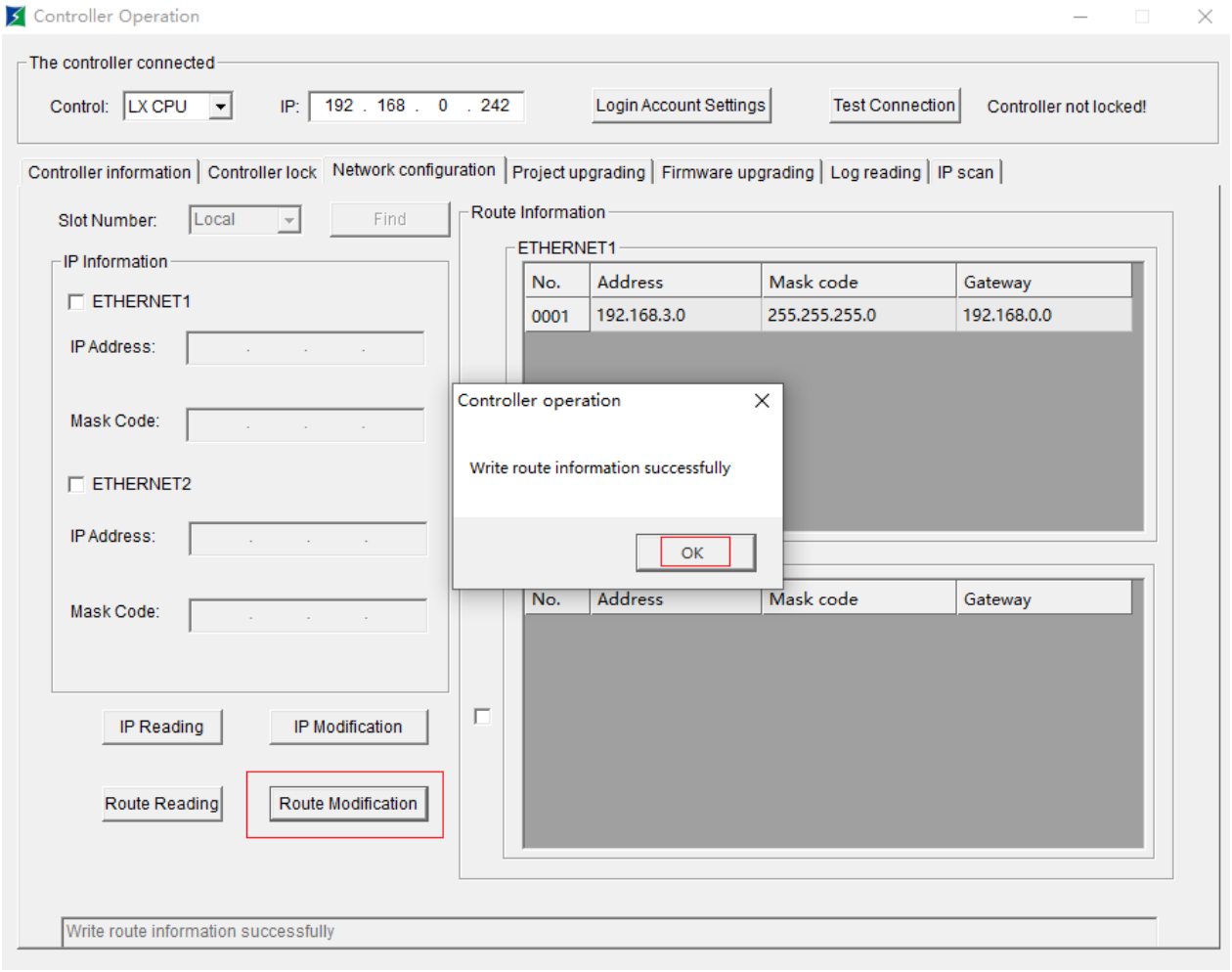
(2) In the Controller Operations dialog box, click the Network Configuration tab, check the Route Information checkbox on the right, right-click in the blank area, and click Add;



(3) In the route addition dialog box that appears, enter the IP address, netmask, and gateway, and click OK;



(4) Click Modify Route. A window indicating that the route information has been modified successfully appears.



14.1.6.4 Result

After a successful modification, you can access the PLC from the destination network segment.

14.1.7 Project Upgrading

14.1.7.1 Introduction

Project upgrade transmits the locally generated .at file to the controller to upgrade the project in the controller.

14.1.7.2 Requirement

AutoThink is offline.

The controller has successfully logged in.

14.1.7.3 Steps

Follow the steps below to upgrade the project:

- Click the button .. on the Project Upgrading page, select the .at file, and click Upgrade. In the upgrade confirmation window that appears, click Yes to upload the .at file to the controller.

14.1.8 Firmware Upgrading

14.1.8.1 Introduction

This feature can be used to upgrade the controller hardware.

14.1.8.2 Requirement

The IEC task is stopped.

A normal connection with the controller is established using the auxiliary tool, and you have logged in to the controller.

You have logged in to the controller.

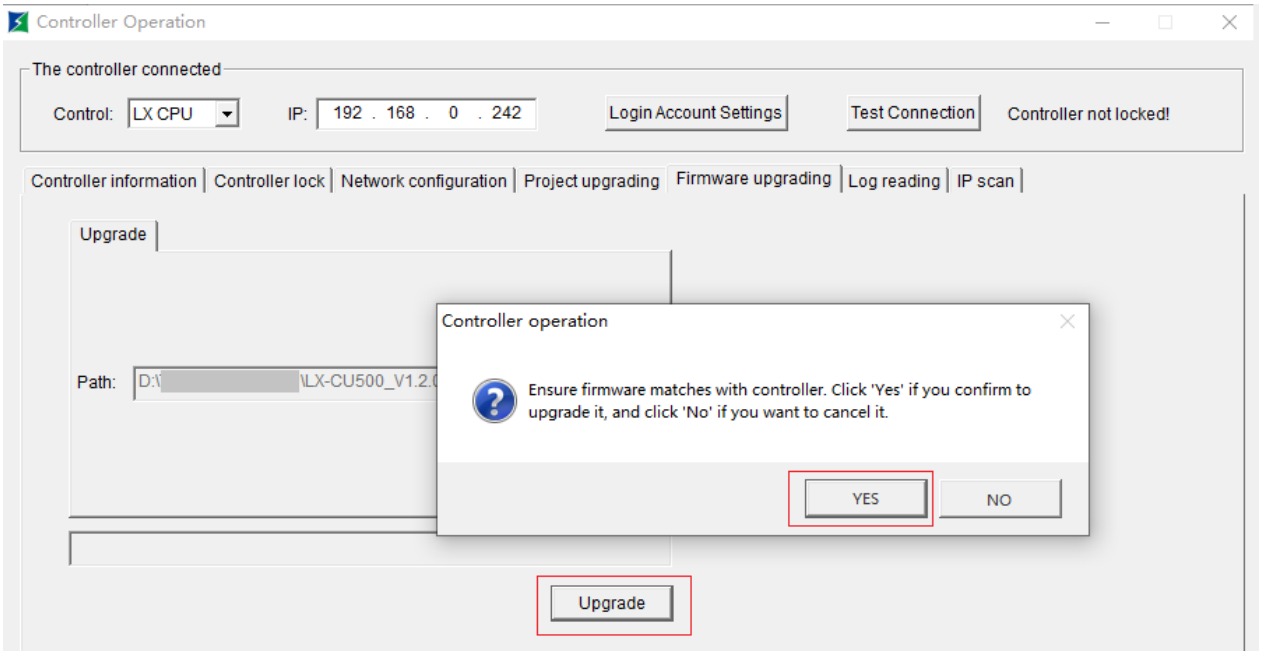
14.1.8.3 Steps

Follow the steps below to upgrade the firmware:

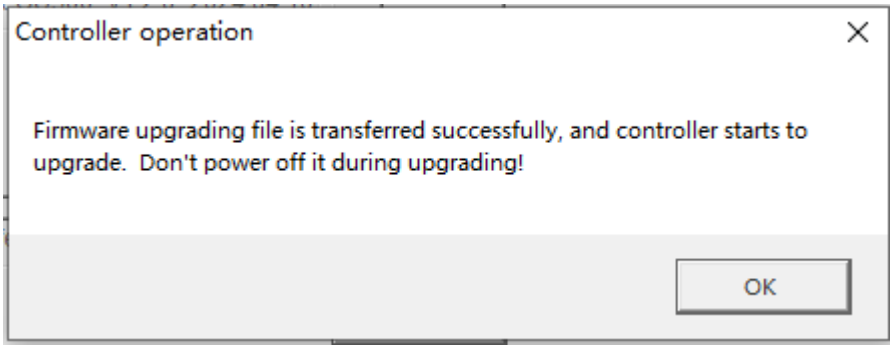
(1) Click the Firmware Upgrade tab in the Controller Operations dialog box;

(2) Click .. on the right of the path field;

(3) Select the .bin firmware file in the upgrade file path, and click Upgrade;

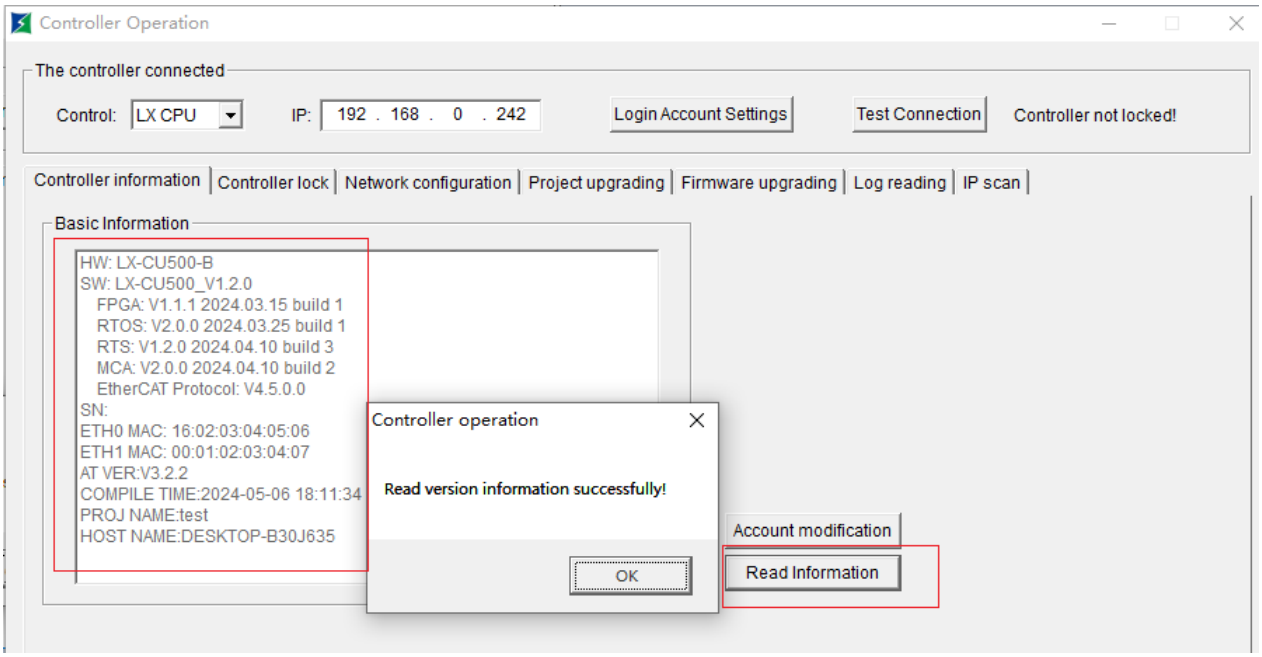


(4) Click Yes. The controller starts the upgrade after it receives the firmware file. At this time, the window as shown in the figure below will appear;



(5) Click OK, and wait for the controller to complete the upgrade;

(6) Confirm the firmware version. After the firmware upgrade is completed, you can check the new firmware version by retrieving the controller information.



14.1.8.4 Result

During the upgrade, both the "SDIN" and "BAT" indicators flash. When they go out, the "ERR" indicator starts to flash. When the "ERR" indicator goes out, the upgrade is completed.

14.1.9 Log Reading

14.1.9.1 Introduction

Read and parse the logs in the controller for analysis by professionals.

Logs record the operating status, fault status, and other information about the controller.

14.1.9.2 Requirement

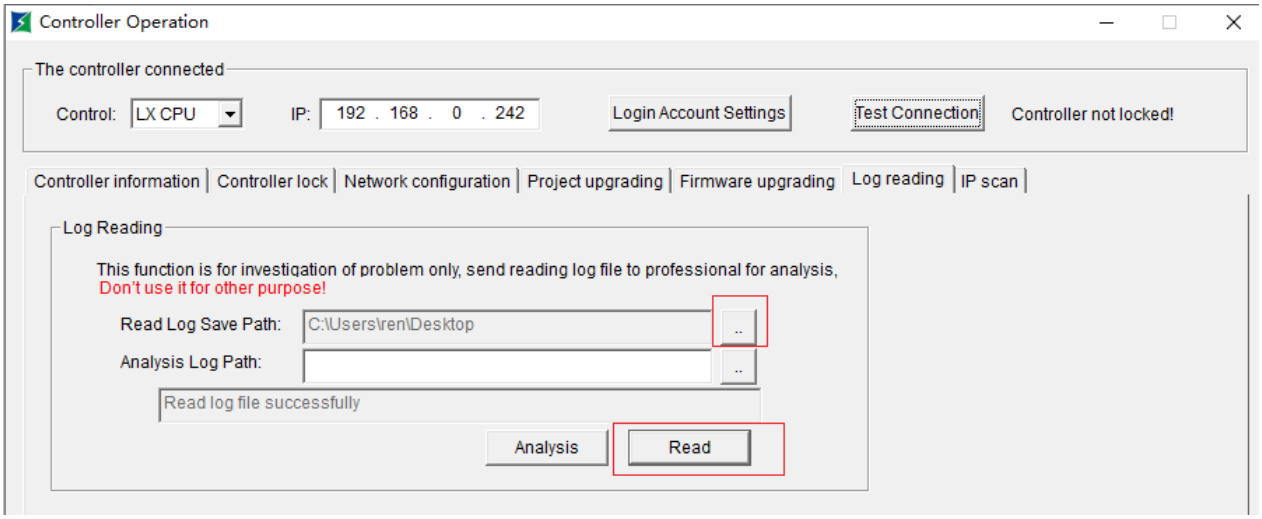
A normal connection with the controller is established using the auxiliary tool, and you have logged in to the controller.

You have logged in to the controller.

14.1.9.3 Steps

Follow the steps below to read the logs:

(1) Click the Log Read tab, and click .. on the right of the Read Log Path field.



(2) After successfully reading the log file, locate the log file read in the previous step to view the controller's operating status, fault status, and other information.

14.1.10 IP Scan

14.1.10.1 IP Scan

14.1.10.1.1 Introduction

You can use the IP scanning feature of the auxiliary tool to find the IP addresses of all controllers in the LAN. Note that IP scanning does not support cascade routing. If a router is set in the LAN, the IP address of the device connected to the router cannot be scanned.

If the PC has multiple network cards, you need to check the model of the network card connected to the network before scanning and configure the communication interface in the tool.

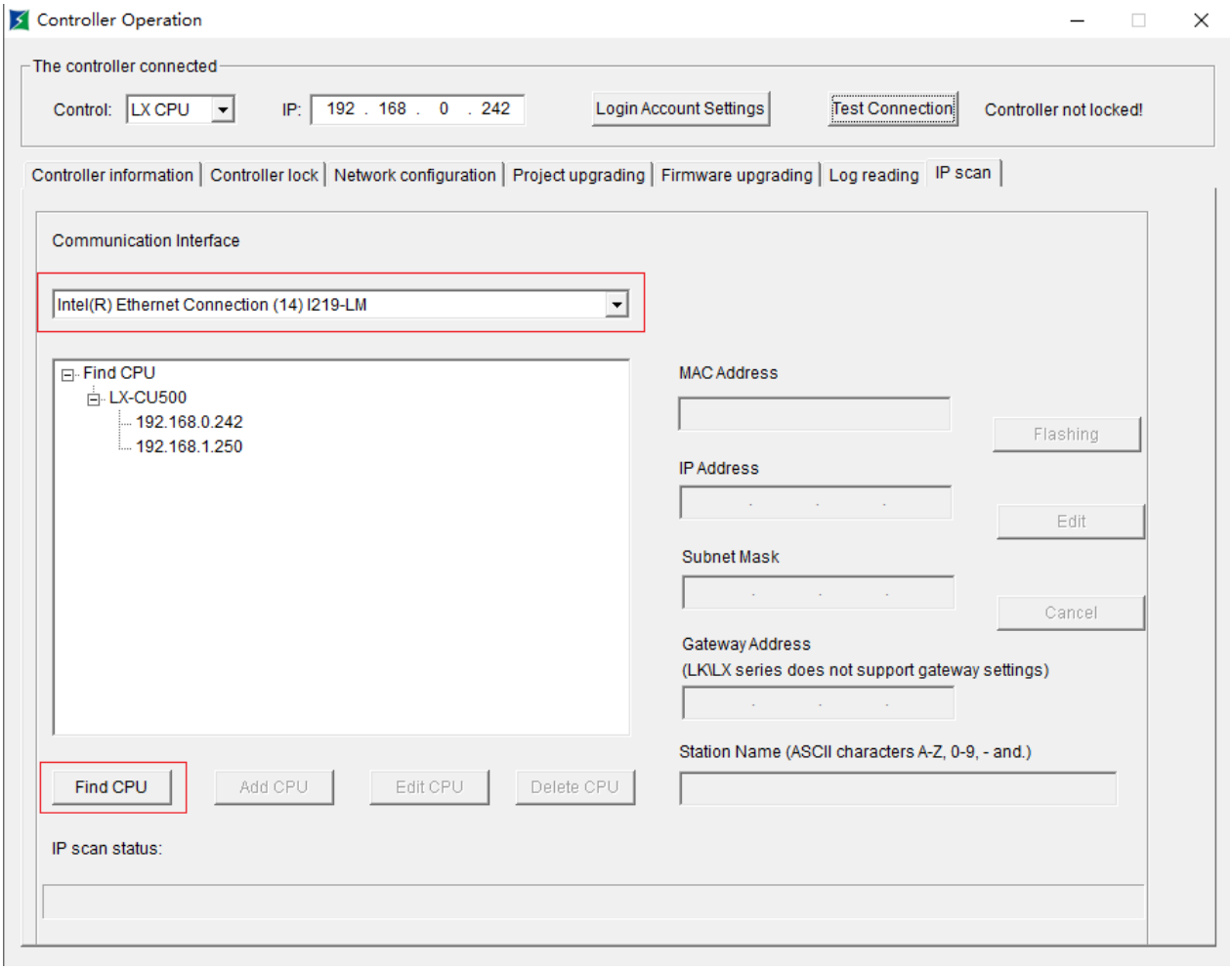
14.1.10.1.2 Requirement

AutoThink is started.

14.1.10.1.3 Steps

Follow the steps below to scan the IP address:

- In the Controller Operations dialog box, click the IP Scan tab, select the communication interface, and click Search for CPU.



14.1.10.1.4 Result

You can view the interface IP address of the corresponding PLCs in the LAN.

14.1.10.2 Modify IP

14.1.10.2.1 Introduction

To avoid IP conflicts between the PLC and other devices, you can retrieve the controller IP address using the auxiliary tool, modify the address, and then write it to the controller.

14.1.10.2.2 Requirement

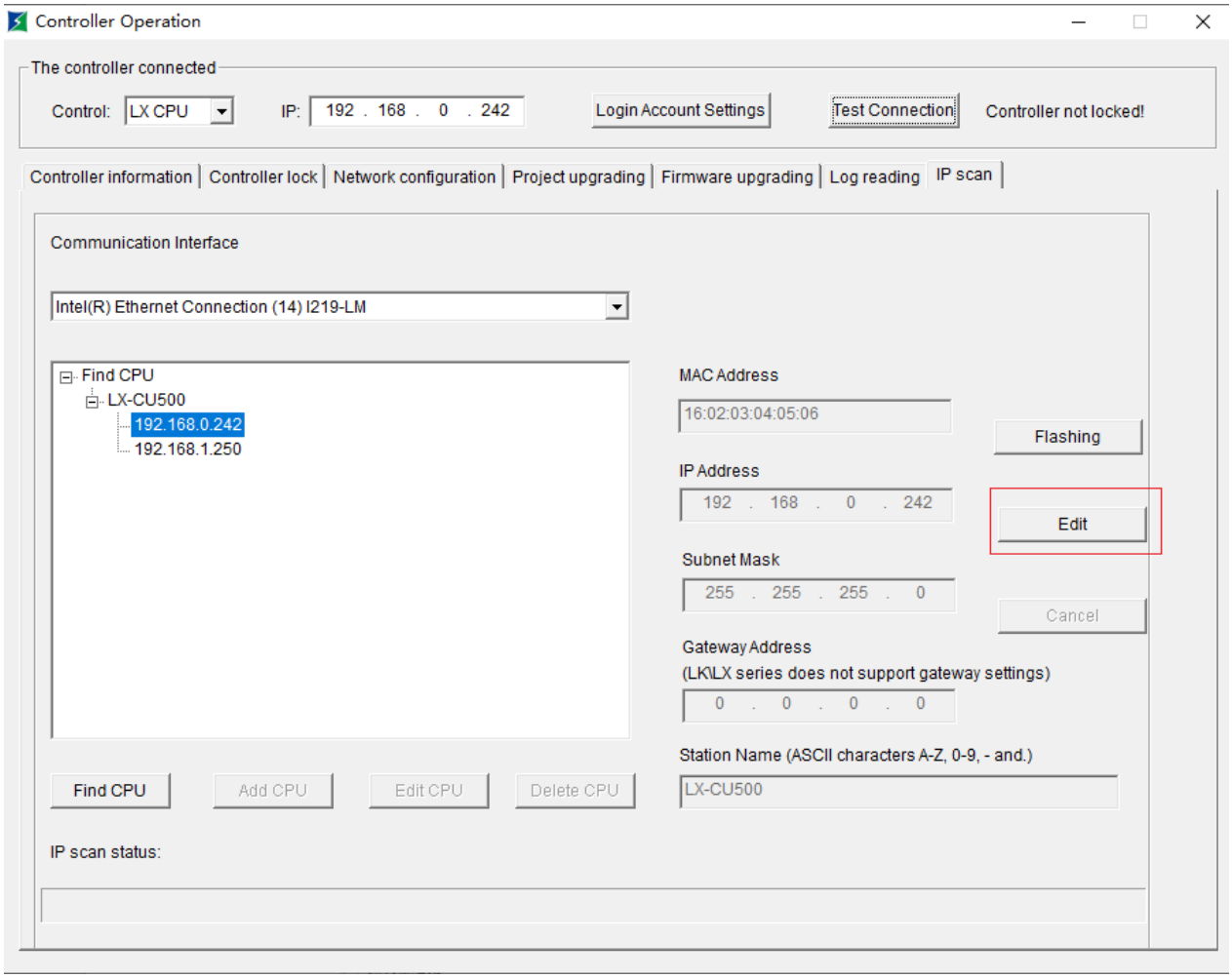
Make sure that a normal connection with the controller is established using the auxiliary tool and that you have logged in to the controller.

Make sure that the controller is in stand-alone mode and that the project is stopped before you modify the IP address.

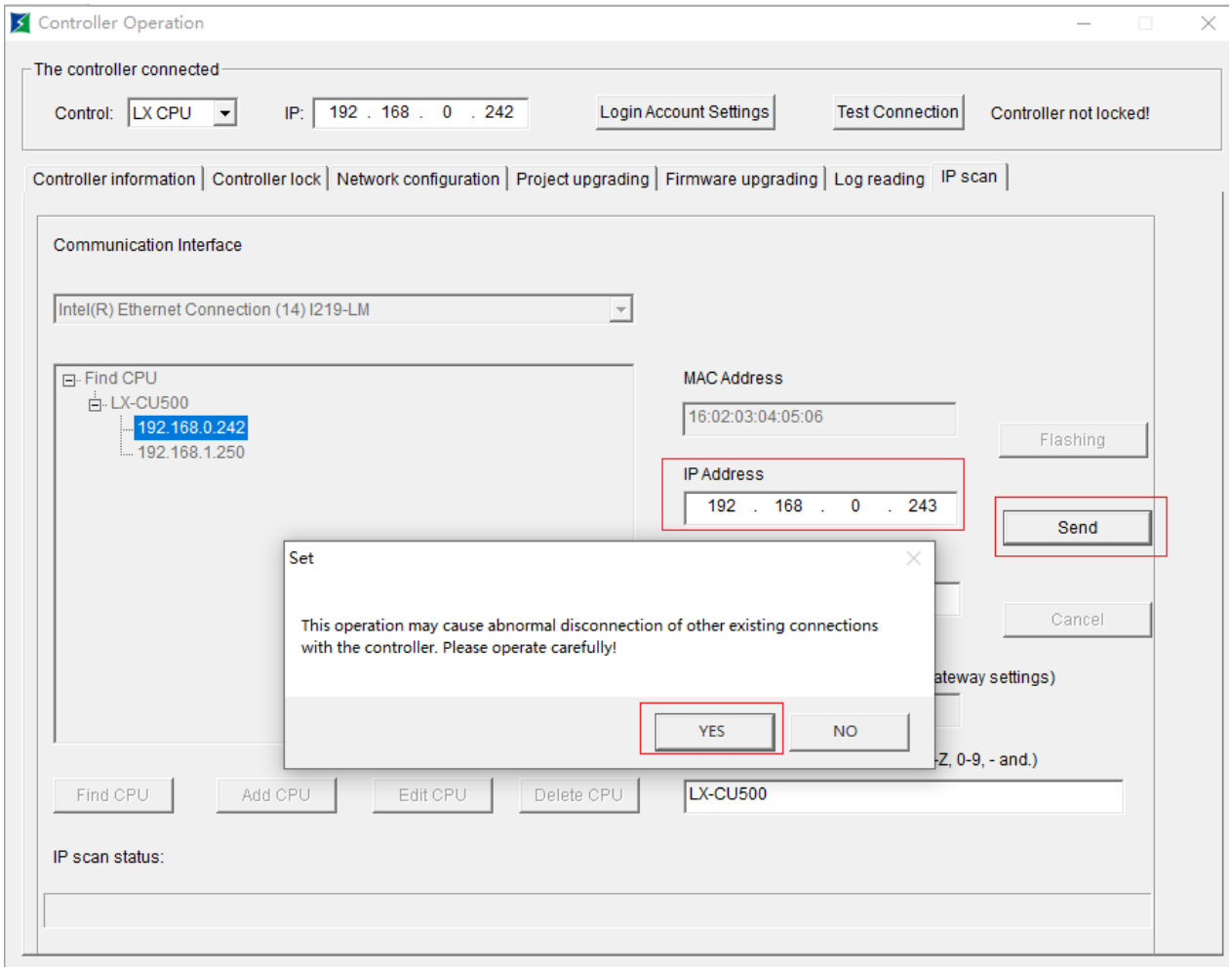
14.1.10.2.3 Steps

Follow the steps below to modify the IP address:

(1) In the Controller Operations dialog box, click the IP Scan tab, select the communication interface, and click Search for CPU. Then, click the IP address that needs to be modified, and click Edit on the right;



(2) Enter the new IP address, click Apply, and click Yes in the Settings dialog box that appears.



14.1.10.2.4 Result

After the modification is completed, use the new IP address to connect to the controller.

14.1.10.3 Detect Controller

14.1.10.3.1 Introduction

You can locate controllers using indicators.

14.1.10.3.2 Requirement

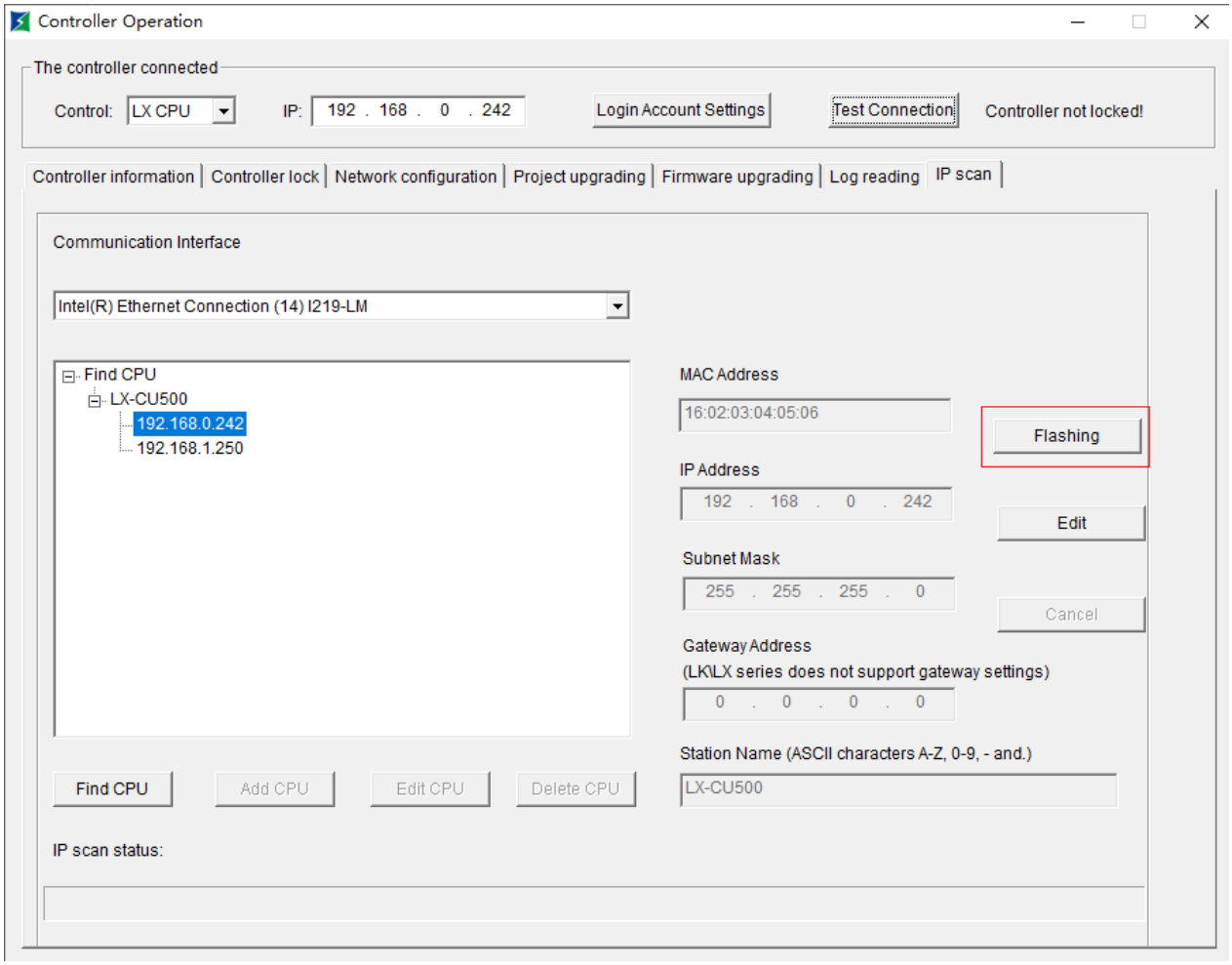
Make sure that a normal connection with the controller is established using the auxiliary tool and that you have logged in to the controller.

Make sure that the controller is in stand-alone mode and that the project is stopped.

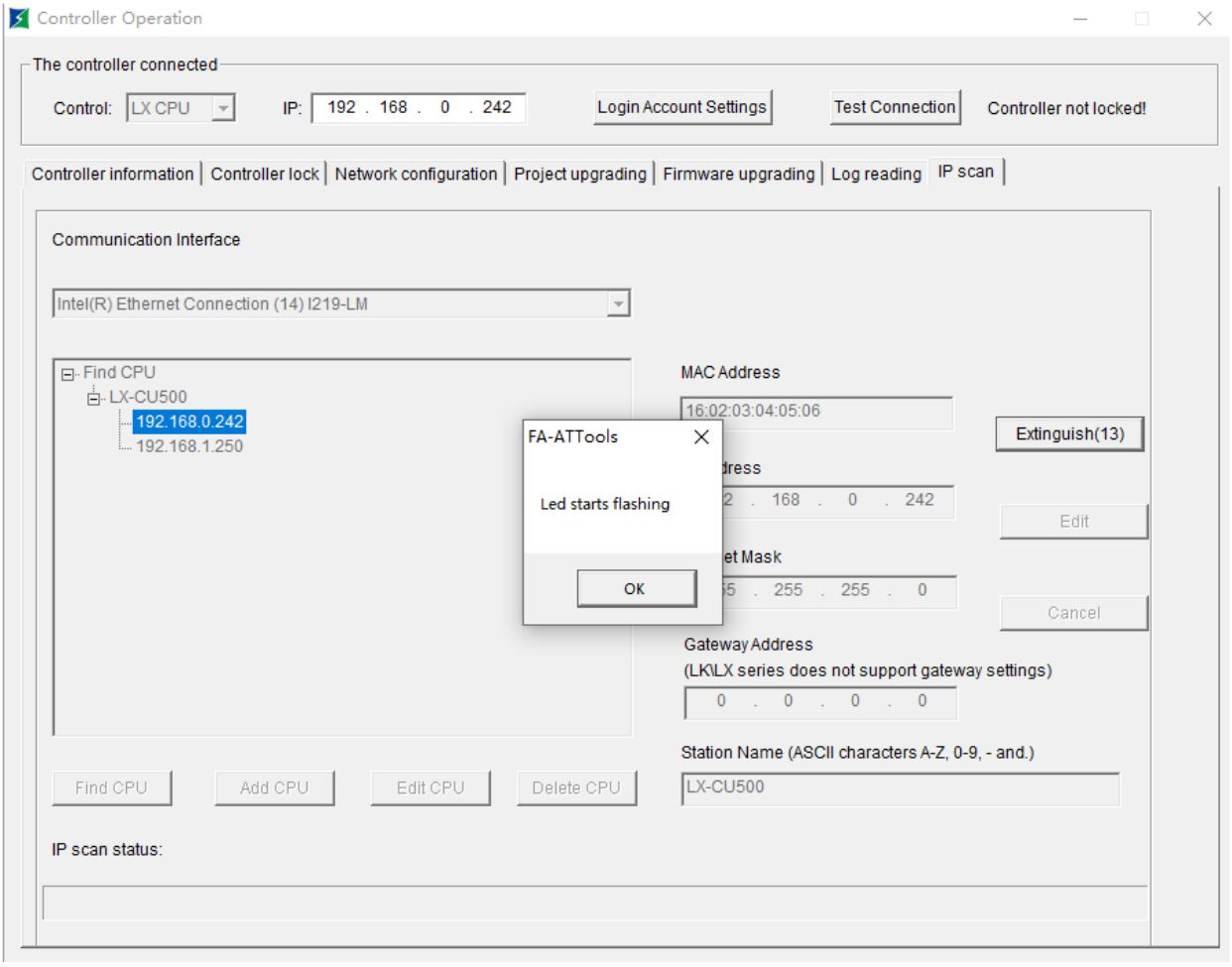
14.1.10.3.3 Steps

Follow the steps below to detect the controller:

(1) In the Controller Operations dialog box, click the IP Scan tab, select the communication interface, and click Search for CPU. Then, select the IP address that needs to be operated on, click Flash Indicators on the right;



(2) Check whether the RUN, ERR, SDIN, and BAT indicators of the controller corresponding to the IP address flash at the same time. If yes, check whether the indicators go out 30 seconds later.



14.1.10.4 Modify Controller Name

14.1.10.4.1 Introduction

You can change the controller names to facilitate the management of multiple controllers in the LAN.

14.1.10.4.2 Requirement

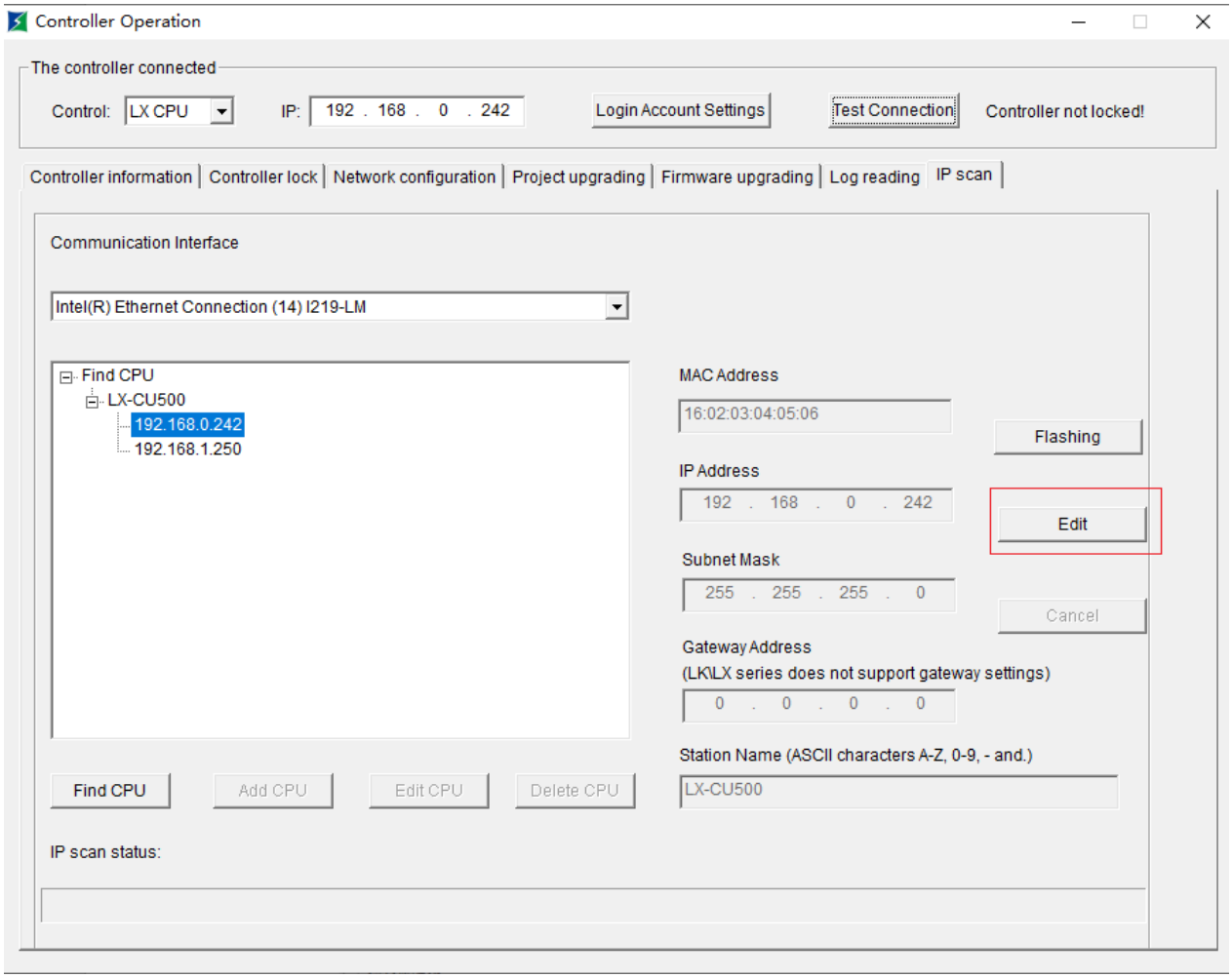
Make sure that a normal connection with the controller is established using the auxiliary tool and that you have logged in to the controller.

Make sure that the controller is in stand-alone mode and that the project is stopped before you modify the IP address.

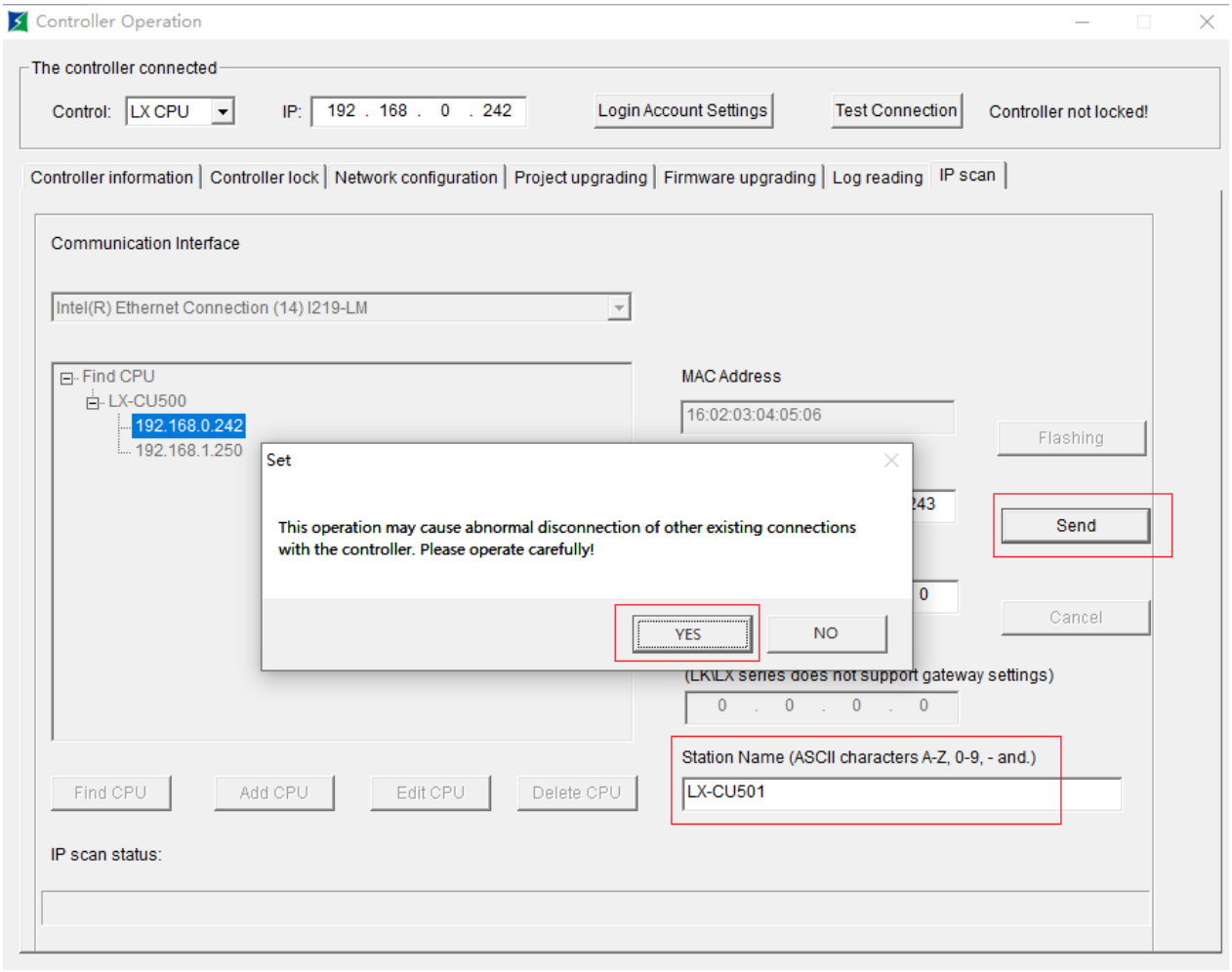
14.1.10.4.3 Steps

Follow the steps below to modify the controller name:

(1) In the Controller Operations dialog box, click the IP Scan tab, select the communication interface, and click Search for CPU. Then, select the IP address that needs to be operated on, and click Edit on the right;



(2) Enter the new controller name, click Apply, and click Yes in the Settings dialog box that appears.

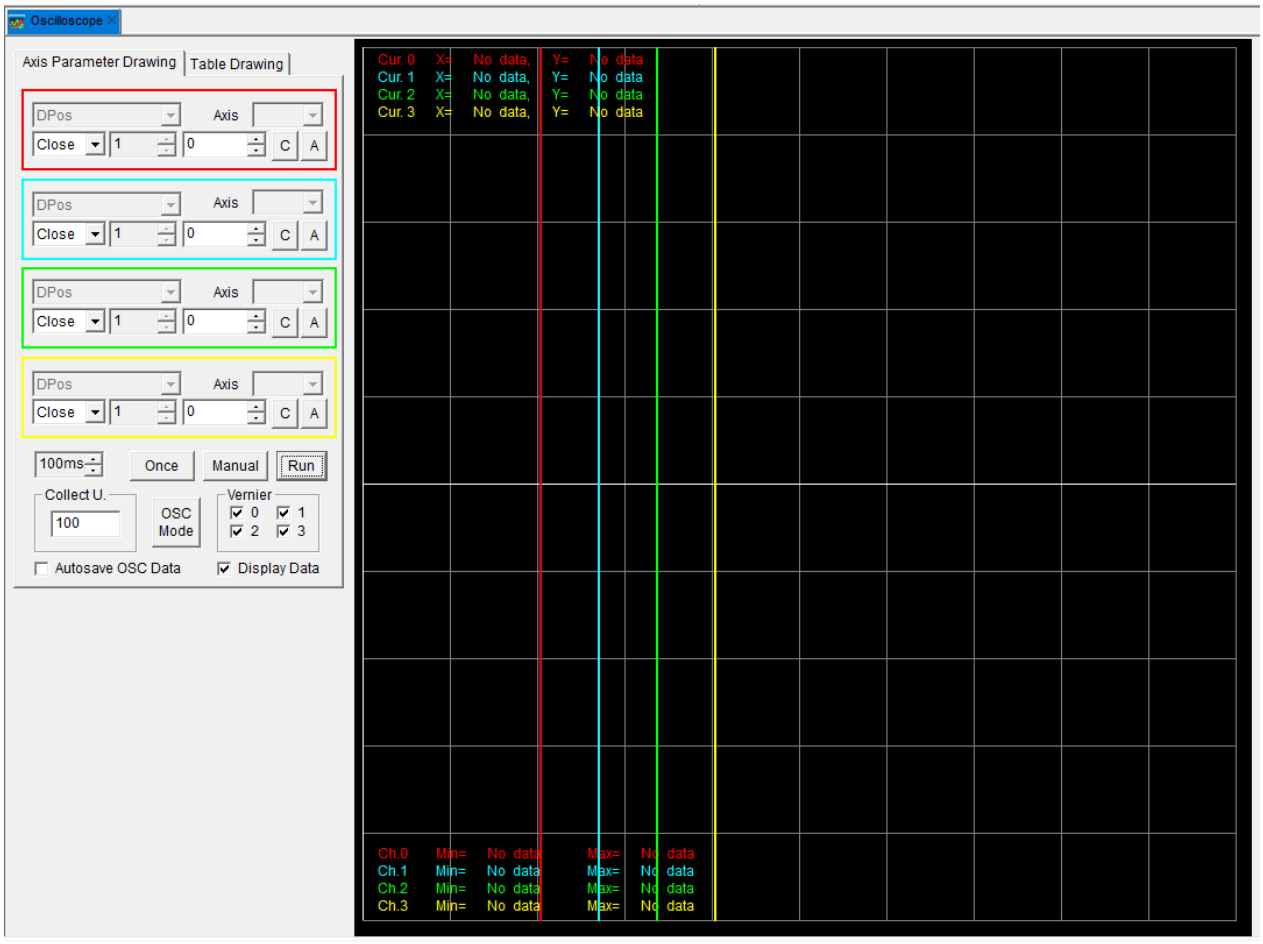


14.2 Oscilloscope

- Menu bar: Choose [Tool] - [Oscilloscope].
- Toolbar:  .

Open the Oscilloscope window, as shown in the figure below.

The configuration software provides the oscilloscope feature that is similar to a real oscilloscope. You can configure and run parameters in the oscilloscope window. The oscilloscope then draws waveforms or traces based on the channel data collected by the controller. The data on the oscilloscope can be exported as files.

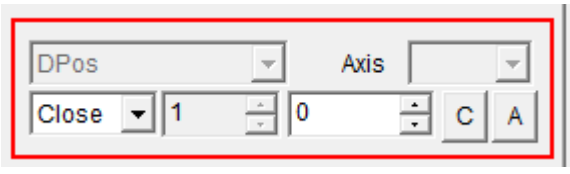


Oscilloscope Interface

14.2.1 Axis Parameter Drawing Settings

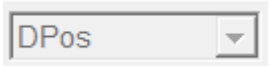
14.2.1.1 Display settings

The oscilloscope has 4 channels. Each channel has a corresponding control group, as shown in the figure below.



Oscilloscope Channel Control Group

- Axis parameters



Axis Parameter Setting

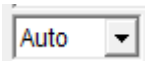
The axis parameter list box is in the upper-left corner of the control group. You can select the parameter to be recorded and displayed by the oscilloscope from the drop-down list.

- Axis number

Axis Number Setting

The axis number is selected from the drop-down list in this box.

- Running mode



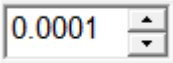
Running Mode Setting

The channel running mode controls the way the waveform (trace) is displayed and zoomed.

Mode description

Mode	Description
Closed	Collect no data and display no waveform (trace).
Auto	Collect data and zoom out to an appropriate scale for display.
Manual	Collect data and zoom out to the specified scale for display.
Frozen	Collect no data but display frozen waveform (trace).

- Vertical scale



Proportional Display Box

In Auto mode, the oscilloscope calculates the appropriate scale and displays it in the scale box.

In Manual mode, the user specifies the single-division scale. You can change the vertical scale using  on the right side of the box.

- Vertical offset

Vertical offset Setting

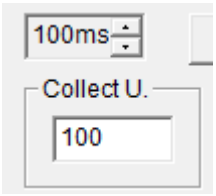
Buttons controlling the vertical offset

Button	Description
	Moves the waveform (trace) in the vertical direction. In particular, these buttons can be used to separate waveforms (traces) if they overlap and only the top

	waveform (trace) is displayed.
C	Sets the vertical offset to 0.
A	Displays the waveform (trace) in the center. When this button is used, other buttons cannot be used (in gray). The oscilloscope calculates the offset. When this button is not used, the offset set by the user applies, and other buttons can be used.

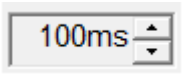
14.2.1.2 Collection parameter settings

The oscilloscope screen has 10 divisions in the horizontal direction. Collection parameters that can be set by the user include the time and the number of data points in each division, as shown in the figure below.



Collection parameter Setting

- Time base



Time BaseSetting

The time base value can be adjusted using the buttons on the right side of the box. The value determines the duration for data collection in each division.

- Number of data points in each division

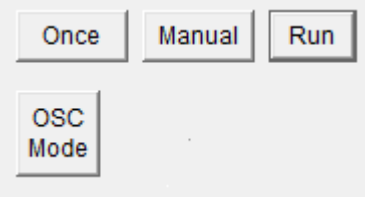
Number of Data Points in Each Division Setting

You can directly enter the value. The greater the value, the more accurate the drawing. The value range is 1 to 100.

- The cycle value must be an integer. If the number of data points in each division is not divisible by the time base value, the oscilloscope will adjust the number of data points automatically.

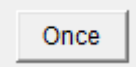
14.2.1.3 Oscilloscope configuration

Oscilloscope configuration buttons are located below the display control groups, as shown in the figure below.



Oscilloscope Configuration and Allocation

- Single/Repeat mode

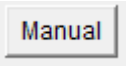


Button of Mode Setting

Click this button to switch between the Single and Repeat modes.

In Single mode, the oscilloscope starts running after being triggered and stops after one frame of data is collected. In Repeat mode, the collection continues after the oscilloscope is triggered again each time it collects one frame of data. The oscilloscope keeps working until the user clicks Stop.

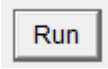
- Trigger mode



Click this button to switch between the Manual and Program modes.

In Manual mode, the controller is always triggered, and data is collected immediately after the user clicks Run. In Program mode, the oscilloscope starts running after the user clicks Run, but the controller does not start collection until the IEC program completes executing the trigger instruction "HMC_TriggerScope".

- Trigger button



The oscilloscope starts running after this button is clicked. The button then changes to Stop. If the oscilloscope is in Single mode, it will automatically stop after one frame of data is collected. When the oscilloscope is running, you can click Stop at any time to stop running.

- Display mode

Setting Button of Display Mode

You can click this button to switch between the Waveform Mode (X/T) and Trace Mode (X/Y).

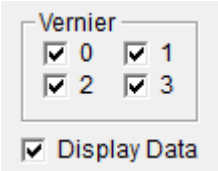
The default display mode is Waveform Mode. The horizontal axis represents time, and the vertical axis represents the collected data. You can view the value changes of axis parameters over time. In Trace Mode, you can observe the data changes of one oscilloscope channel by comparing it with another channel.

- In Trace Mode, the first two channels are in a group, and the last two channels are in another group. Do not modify the group settings.

14.2.1.4 Oscilloscope data query

The options for oscilloscope data query are located at the bottom of the oscilloscope control interface, as shown in the figure below.

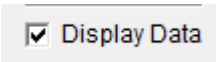
- Oscilloscope cursors



Cursor Display Setting

You can check an option to display the cursor of the corresponding channel. The cursor data is displayed at the top of the oscilloscope drawing interface.

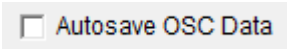
- Key data



Oscilloscope Data Display Setting

You can check this option to display the maximum and minimum values of each waveform (trace) and the current position of the oscilloscope cursor at the bottom of the oscilloscope drawing interface.

- Automatic data saving



Automatic Save Setting

You can check this option to save the oscilloscope data to the project directory each time one frame of data is collected.

14.2.2 Oscilloscope data import and export

This feature is integrated into the right-click menu of the oscilloscope drawing interface, as shown in the figure below.

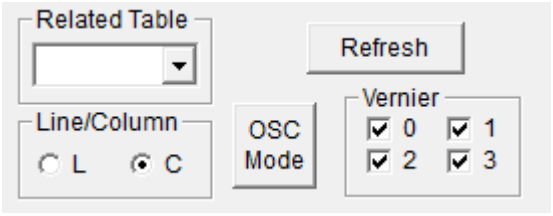
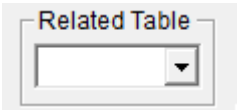


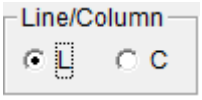
Table Drawing Setting

- Associated table



Select the associated table from the drop-down list of tables defined in the project.

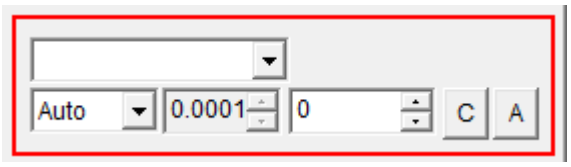
- Drawing by rows or columns



Select Rows or Columns in an Associated Table

Specify whether to draw based on rows or columns of the associated table.

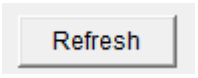
- Data source



Oscilloscope Channel Control Group

All 4 channels of the oscilloscope can be used for table drawing. Take the channel above as an example. Select the serial number from the drop-down list. If Row is checked for Select Row/Column, the row number is selected here. If Column is checked, the column number is selected here.

- Refreshing view



Refresh View Button

Click this button to draw the waveform.

- Oscilloscope status words

The oscilloscope status words are variables defined in the oscilloscope CPU and are used for communicating with the controller to obtain waveform data. When a project is created, the system predefines the oscilloscope status words as global variables.

Chapter 15 Firmware version and target configuration version compatibility table

In AutoThink, the target configuration version corresponds one-to-one with the controller firmware version.

For example: When the firmware version of LX-CU500 is B02, the target configuration version in AutoThink needs to be switched to V3.2.3. For specific methods to switch the target configuration, please refer to the project properties.

Firmware version and target configuration version compatibility table

Target Configuration Version	Firmware Version									
	LX-CU500-A01	LX-CU500-A02	LX-CU500-B01	LX-CU500-B01-SP1	LX-CU500-B02	LX-CU501-A01	LX-CU501-A02	LX-CU511-A01	LX-CU510-A01	LX-CU430-A01
V3.2.1Alpha1	√									
V3.2.1B1	√									
V3.2.2A1		√								
V3.2.2			√			√				
V3.2.2SP1				√		√				√
V3.2.2SP2				√		√				√
V3.2.3					√		√	√	√	√

- LX-CU500-B01-SP1 is a patch version and is not recommended for use.

Chapter 16 Correspondence Relationship Table of Target Configuration with AutoThink Version

Correspondence Relationship Table of Target Configuration with AutoThink Version

Target Configuration Version	AutoThink Version
V3.2.1Alpha1	FA-AT V3.2.1Alpha1
V3.2.1B1	FA-AT V3.2.1B1
V3.2.2A1	FA-AT V3.2.2A1
V3.2.2	FA-AT V3.2.2
V3.2.2SP1	FA-AT V3.2.2SP1
V3.2.2SP2	FA-AT V3.2.2SP2
V3.2.3	FA-AT V3.2.3

Beijing HollySys Intelligent Technologies Co., Ltd..

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

Hollysys Group Website: https://www.hollysys.com

Factory Automation Business Website: https://fa.hollysys.com/