



 Version 1.3.2

Safety FA-AutoThink

User Manual





Safety FA-AutoThink

User Manual

Version: V1.7
January, 2025

Copyright Notice

The text, illustrations, charts, marks, trademarks, product models, programs, page layout and other contents included in this manual are under protection of “Copyright Law of the People’s Republic of China”, “Trademark Law of the People’s Republic of China” , “Patent Law of the People’s Republic of China” and the laws of applicable international conventions regarding copyright, trademark right, patent right or other property ownership, and they are owned or possessed exclusively by Beijing HollySys Intelligent Technologies Co., Ltd..

Since the equipment explained in this manual has a variety of uses, the user and those responsible for applying this equipment must satisfy themselves as to the acceptability of each application and use of the equipment. Under no circumstances will Beijing HollySys Intelligent Technologies Co., Ltd. be responsible or liable for any damage, including indirect or consequential losses resulting from the use, misuse, or application of this equipment.

Due to the many variables associated with specific uses or applications, Beijing HollySys Intelligent Technologies Co., Ltd. cannot assume responsibility or liability for actual use based upon the data provided in this manual.

This manual is provided only for commercial users to read. Without prior written permission of Beijing HollySys Intelligent Technologies Co., Ltd., no part of this manual should be reproduced and transmitted in any forms by any means, including electronic, mechanical or otherwise regardless of whatever reasons and purposes. We will investigate violator’s legal liability in accordance with the relevant laws.

The text HollySys, and the logos  are registered trademarks of Beijing HollySys Intelligent Technologies Co., Ltd..

All other trademarks are the property of their respective holders.

All rights reserved for 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: 4008-111-999

Fax: +86 010-5898 1558

Web: <http://www.hollysys.com>

Email: PLC@hollysys.com

Sina weibo: <http://weibo.com/hollysysplc>

Contents

Chapter 1	About manual	1
1.1	Document update	1
1.2	Purpose	1
1.3	Intended audience	1
1.4	Document conventions	2
1.4.1	Menu	2
1.4.2	Mouse	2
1.4.3	Keyboard	2
1.4.4	Online help	2
1.4.5	Important information	2
1.5	Related documentation	3
Chapter 2	Overview	5
Chapter 3	Installation and uninstallation	7
3.1	Installation	7
3.1.1	Installation environment	7
3.1.2	Installation steps	7
3.2	Uninstall	15
Chapter 4	Start Safety FA-AutoThink	17
Chapter 5	Software interface	19
5.1	Title bar	19
5.2	Menu bar	19
5.3	Toolbar	20
5.4	Project management	20
5.4.1	Task configuration	20
5.4.2	User program	21
5.4.3	Hardware configuration	21
5.4.4	Global variable	21
5.4.5	Modbus configuration	21
5.4.6	Surveillance list	21
5.5	Work area	21
5.6	Output window	22
5.7	Status bar	23
5.8	Library manager window	23
5.8.1	Library configuration	23
5.8.2	View	26

5.8.3	Find.....	27
5.8.4	Usage	29
5.9	Device library window	29
5.9.1	Find.....	30
Chapter 6	Project management	31
6.1	Create a project.....	31
6.1.1	Project	31
6.1.2	Library.....	32
6.2	Task configuration	32
6.2.1	Program calling.....	32
6.2.2	Configure task property	34
6.3	Set project safety property.....	34
6.3.1	Set safety property.....	35
6.3.2	Configuration information.....	35
6.4	Project safety	36
6.4.1	New	37
6.4.2	Edit	39
6.4.3	Delete	39
6.4.4	Action timing	39
6.5	Library safety	40
6.6	Project setting	40
6.6.1	Configuration	41
6.6.2	Color.....	42
6.6.3	Configuration language.....	43
6.6.4	Default	45
6.7	Save project.....	45
6.7.1	Save	45
6.7.2	Save as.....	45
6.8	Close project	45
6.9	Recent projects list.....	46
6.10	Upload project file.....	46
6.11	Download project file.....	46
6.12	Window	46
6.13	Help	47
Chapter 7	Hardware configuration	49
7.1	Controller module	49
7.2	Configure DP protocol.....	49
7.2.1	Add DP master device	49
7.2.2	Add communication protocol	50
7.2.3	Add PROFI-safe/Profibus-DP slave station	51
7.2.4	Modify slave address	53
7.2.5	Configure slave station parameters	54
7.3	Configure Modbus TCP protocol.....	58
7.3.1	Configure Modbus TCP master protocol.....	58

7.3.2	Configure Modbus TCP slave protocol	64
7.4	System diagnosis.....	68
7.5	Hardware list	70
Chapter 8	Program organization unit	73
8.1	Add POU	73
8.1.1	Common	74
8.1.2	Difference	75
8.2	Edit POU	75
8.2.1	Variable declaration	76
8.2.2	Body.....	77
8.3	POU property.....	77
8.4	Rename POU	78
8.5	Copy and paste	78
8.6	Delete POU	79
8.7	Zoom POU	79
8.8	Print POU	79
8.8.1	Print preview	79
8.8.2	Print	81
8.9	Find	81
8.10	Replace	82
8.11	Call POU.....	82
8.11.1	Program calls Function Block	83
8.11.2	Program calls Function	83
8.11.3	Call between program.....	83
8.11.4	Call between Function Blocks.....	84
8.11.5	View the calling tree.....	85
8.12	Folder	85
8.12.1	Add	85
8.12.2	Delete	86
8.12.3	Rename	86
Chapter 9	Variable	87
9.1	Variable type.....	87
9.1.1	Classify according to structure.....	87
9.1.2	Classify according to the value range of variable.....	87
9.2	Variable naming rules	88
9.3	Variable declaration	88
9.4	Standard data type.....	89
9.5	Complex data type	90
9.5.1	Array type	90
9.5.2	Function block type	93
9.6	Surveillance list.....	94
9.7	General operation of variable	96

9.7.1	Add	96
9.7.2	Insert.....	96
9.7.3	Copy and paste.....	96
9.8	Customized variable group	96
9.8.1	Add	96
9.8.2	Rename	97
9.8.3	Delete	97
9.9	Export and Import Variables	97
9.9.1	Export Variables	97
9.9.2	Import Variables.....	98
9.9.3	Export hardware Variables	99
9.9.4	Import Hardware Variables	99
9.10	Variable Group Management.....	100
9.10.1	Create a File	100
9.10.2	Delete	100
9.10.3	Rename	101
9.11	Access of Variable	101
9.12	Delete unused global variables	101
9.13	Show cross reference table	102
9.14	Variable debug readback.....	103
Chapter 10	Programming language.....	107
10.1	FBD editor.....	107
10.1.1	Cursor Position	108
10.1.2	Network line	110
10.1.3	Input.....	110
10.1.4	Output.....	111
10.1.5	Block.....	111
10.1.6	Jump and Label	112
10.1.7	Return.....	113
10.1.8	Comment	114
10.1.9	Set /Reset.....	114
10.1.10	Negate	115
10.1.11	Component Align	116
10.1.12	Execute sequence and number	117
10.2	LD Editor.....	118
10.2.1	Cursor position.....	120
10.2.2	LD elements	122
10.2.3	Network	123
10.2.4	Series contact.....	124
10.2.5	Parallel contact	125
10.2.6	Coil	125
10.2.7	Block.....	126
10.2.8	Set/Reset.....	127
10.2.9	Negate	127
10.2.10	Comment	127
10.2.11	Jump and return.....	129
10.2.12	Move the element	131
Chapter 11	Controller operation	133

11.1 Controller operation.....	133
11.1.1 Connect the controller.....	133
11.1.2 Controller information.....	134
11.1.3 Controller lock.....	135
11.1.4 Modify Controller IP	137
11.1.5 Configure Route.....	138
11.1.6 Project upgrading.....	140
11.1.7 Firmware upgrading.....	141
11.1.8 Log reading.....	142
11.1.9 Scan IP	143
11.2 Reset	145
11.3 Cold reset	146
11.4 Warm reset.....	146
11.5 Clear controller	146
11.6 Controller calibration time	147
Chapter 12 Compile.....	149
12.1 Incremental compilation.....	149
12.2 Full compilation.....	149
12.3 Compilation result.....	150
12.4 Analysis for compiled information	150
12.5 Decompile.....	153
Chapter 13 Download.....	155
13.1 Communication Settings.....	155
13.2 Download.....	155
Chapter 14 Debug.....	157
14.1 Simulation mode	157
14.2 Operation online.....	157
14.2.1 Write value	157
Chapter 15 Online monitor	161
15.1 Run	161
15.2 Stop.....	161
15.3 Pause	161
15.4 Monitor.....	162
15.4.1 Variable and program	162
15.4.2 Online Value and quality of channel	163
15.5 View Operation Log	163
15.6 Enable SOE.....	164
15.6.1 Overview.....	164
15.6.2 Requirement	164
15.6.3 Step	164

15.7 View SOE	165
15.7.1 Overview.....	165
15.7.2 Requirement	166
15.7.3 Step	166
15.8 Set NTP	166
15.8.1 Overview.....	166
15.8.2 Requirements	167
15.8.3 Steps	167

Chapter 1 About manual

1.1 Document update

Version	Date	Description
V1.0	August 23, 2017	New
V1.1	February 2, 2018	Updated: 6.3 Set project safety property 7.2 Configure DP protocol 7.4 System diagnosis
V1.2	November 30, 2018	Update Executing Cycle in Chapter 6.2Task configuration Add Chapter 12.1 Incremental compilation Add hardware configuration readback option in Chapter 6.6.1 Configuration Add confirmation for hardware configuration readback in Chapter 13.2 Download
V1.0	June 14,2019	Update Copyright
V1.1	November 10, 2019	Support for importing and exporting variables
V1.2	January 17, 2020	New functions: 1.Controller as a Modbus master can configure two IP addresses 2. IP scan 3. NTP timing 4. Global variable group supports SOE function 5. Add LK630S module 6. LK610S updates filter parameter
V1.3	August 14, 2020	Update the version of Safety FA-AutoThink
V1.4	January 26, 2021	Add LK611S/LK410/LK442/LK720
V1.5	March 21,2022	Update Logo
V1.6	August 30,2024	New Input Assistant and Variable Properties
V1.7	January 20, 2025	Firmware upgradin: Add the key switch is set to PRG or REM mode, and IEC is stopping when you upgrade firmware.

1.2 Purpose

This document introduces programming software (Safety FA-AutoThink) of LKS safety control system, including menu, function and configuration.

1.3 Intended audience

The information provided in this manual meets the requirements of LKS programmers, operators, and system maintenance personnel.

The engineers who use this manual are required to have knowledge of automatic control and IEC 61131 programming.

1.4 Document conventions

1.4.1 Menu

Menu commands, the sub commands in the menu, the tag page name in this manual with the note said, such as [file], [Edit] menu.

1.4.2 Mouse

Press: press the left mouse button once and keep.

Click: Press the left mouse button once and release.

Right-click: Press the right mouse button once and release.

Double click: quickly press the left mouse button two times and release.

Drag: Press and hold the left mouse button while moving the mouse.

1.4.3 Keyboard

The keys in the manual are used with bold style, such as **Enter**.

1.4.4 Online help

The online help is a supplement to this manual. The purpose is to provide detailed support when using the software.

The online help is integrated into the software, and you can get help through following way:

- Press the **F1** key in sub-node of Menu, Device Library, Library Manager and Project Management Tree to accurately locate the relevant help instructions.
- You can also use the menu bar [Help] - [Content] to call the help, and enter keyword in [Index], [Search] to find and locate.

- ☐ [Index]: Quickly find by traversing the index topics.

- ☐ [Search]: Full-text search.

When searching keyword in online help, it only retrieves topics containing the keyword in the topic, and does not retrieve the topics containing the keywords in topic name.

When searching keyword in online help, you need to enter * keyword or keyword * to fuzzy search. Such as Target, enter *GET to search. Such as GetTickCnt, enter GET* to search.

1.4.5 Important information

The following marks are used in this manual for identifying appropriate information:



- Warning, indicating that the operation may lead to the potential threats of failure or damage to software and hardware equipment.



- Important icon, identifies important information about the operations or functions need to understand.



- Operation icon, indicating the open or operation method of the object.

For example:

The location of the operation: Mouse command + Operation sequence.



- Menu bar: click [Insert]-[Input element],
- Tool bar: ,
- Shortcut: **Ctrl + I**,
- **POU**: right click the empty area, click [input element].

As shown in the first operation method above, in the menu bar, click [Insert] menu and the sub menu [Input element] command.

SEE ALSO

- Reference icon, indicating the page or content to be referenced for getting to know the function.

1.5 Related documentation



Safety FA-AutoThink User Manual



LKS Safety Control System Instruction Manual



LKS Safety Control System Product Manual

Chapter 2 Overview

LKS is a SIL2 safety control system in industrial automation and rail automation field self-developed by HollySys, therefore, Safety FA-AutoThink is as the configuration software self-developed for LKS system. It belongs to HollySys AutoThink software family.

Safety FA-AutoThink has the following characteristics:

- It supports the international standard IEC61131-3 and a large number of applied instructions.
- It supports two programming languages. They are Ladder Diagram (LD) and Function Block Diagram (FBD).
- It supports simplified Chinese and English.
- The Find and Replace functions can be applied to a present window, or a whole project.
- It provides multi-levels password setting mechanisms.
- It supports several types of online operation such as monitoring, controller running, and controller stopping, and so on.
- It supports simulation.

Table 1 Software version information and use restrictions of some function

Software version	Problem	Elusion measure
Safety FA-AutoThink V1.0.0 / V1.0.1/V1.1.1/V1.2.1/V1.2.2/, V1.3.1/V1.3.2	Open a POU with lots of logic, when the project is online, hold down the Ctrl, scroll the mouse, scroll, zoom in and out the page, it will result in communication failure.	Online monitoring, when you zoom in and out the logic page, do not hold down the Ctrl and scroll the mouse wheel continuously for a long time. This operation will cause exit monitoring.
	Configure DP or Modbus TCP, please do not use replacement function.	Modify channel name or channel description through manual operation.
	The hardware configuration LK610S generates the intermediate variables in the global variable group Q_Profis_DPIO_Group. Do not support the right way to send an intermediate variable to [Modbus configuration].	Users can copy intermediate variables and reference member variables.
	For Safety FA-AutoThink V1.0.0 / V1.0.1, Variables in global variables and PRG do not support import and export.	Users can use variables by copying and pasting variables.
	For Safety FA-AutoThink V1.0.0, online value of the Modbus variables is not displayed.	Online, variables in Modbus can view the online values in the position where the variable is defined or the monitor list.
	Program don't have Divide-By-Zero fault protection.	Users need to pay attention to configuration, and there are warnings when compiling.
	Program don't have Cross-Border protection	Users need to pay attention

Software version	Problem	Elusion measure
	of array.	to configuration, and there are warnings when compiling.

Table 2 Tool version

Tool	Version
Safety FA-AutoThink	V1.0.0, V1.0.1, V1.1.1, V1.2.1, V1.2.2, V1.3.1,V1.3.2
Controller operation	V1.0

Chapter 3 Installation and uninstallation

3.1 Installation

3.1.1 Installation environment

Computer configuration is as shown in Table 3.

Table 3 Recommended computer configuration

Environment	Type	Model
Hardware environment	Display	Color CRT or LCD
	Input/output	Standard keyboard and mouse
	USB interface	At least with 1 USB 2.0 interface
	VGA	Resolution supports 1280×720
	CPU	Above Intel Pentium 2.4GHz
	Memory	Above 512MB
	Hard disk	Above 10GB
Software environment	Operating System	Windows 7 professional 32 Windows 7 professional 64 Windows 10 professional 64
	Software tool	MS Excel 2003, 2007, 2010

The computer configuration used for the project should be equal to or above the above configuration.

3.1.2 Installation steps

Major installation steps:

Step 1. Select the Installation Language

Insert the installation disk of the software into optical disk drive to pop up the dialog of **Select Setup Language**, as shown in Figure 1.

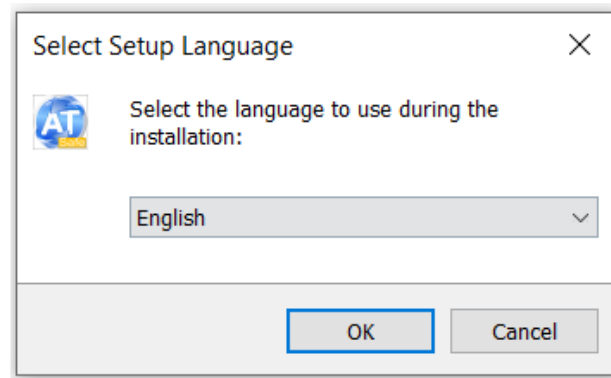


Figure 1 Select installation language

Step 2. Start installation wizard

Select the **English** as language and click the **OK** button to pop up the window of **Setup—Safety FA-AutoThink**, as shown in Figure 2.

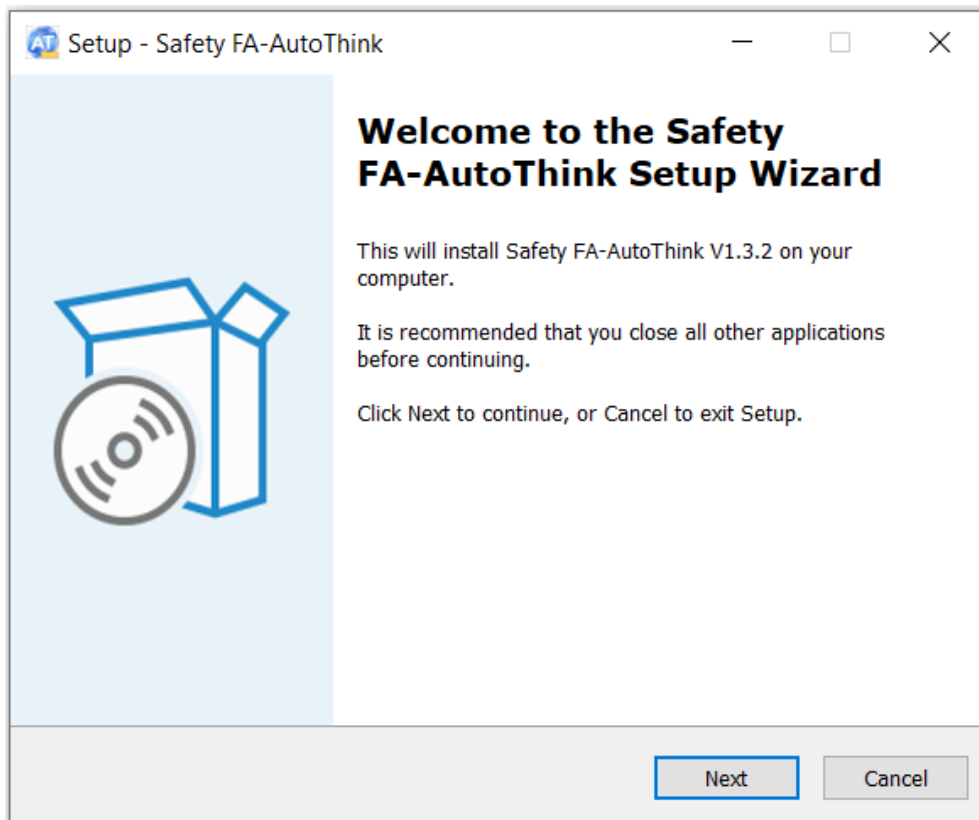


Figure 2 Installation wizard

Step 3. License agreement

Click **Next** to pop up the window of **License Agreement**, select **I accept the agreement** and click **Next**, as shown in Figure 3.

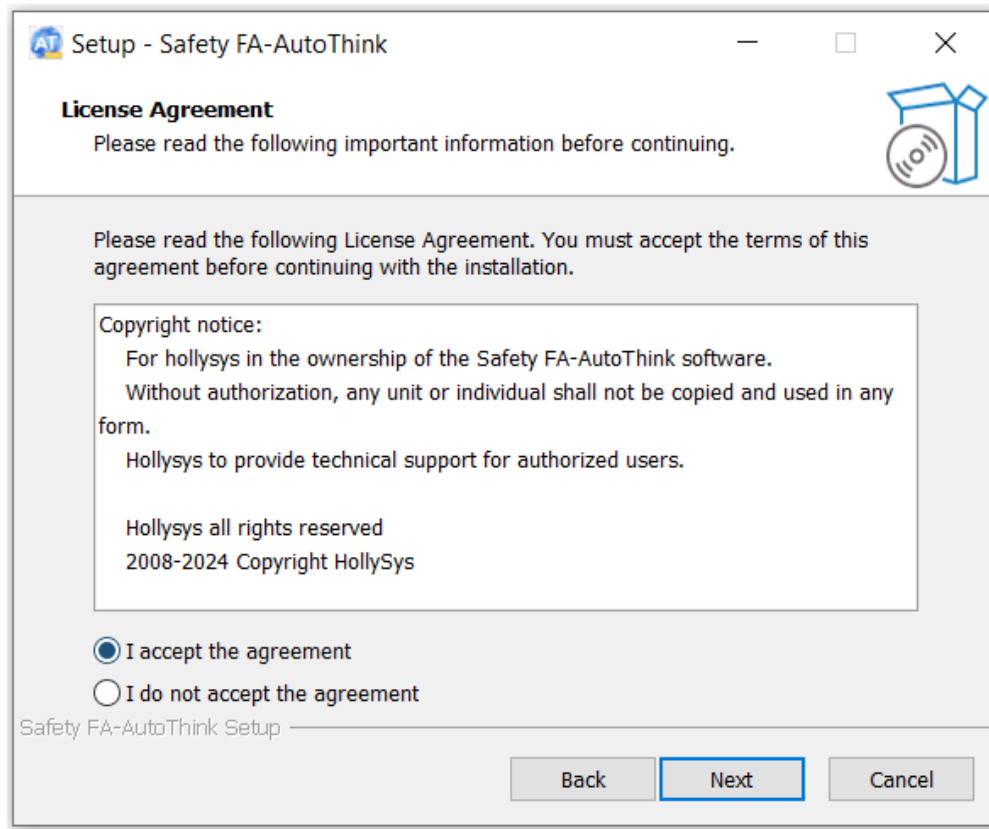


Figure 3 License agreement

Step 4. Select installation directory

The window of **Select Destination Location** pops up, as shown in Figure 4. Click **Browse...** to select installation directory, then click **Next**.

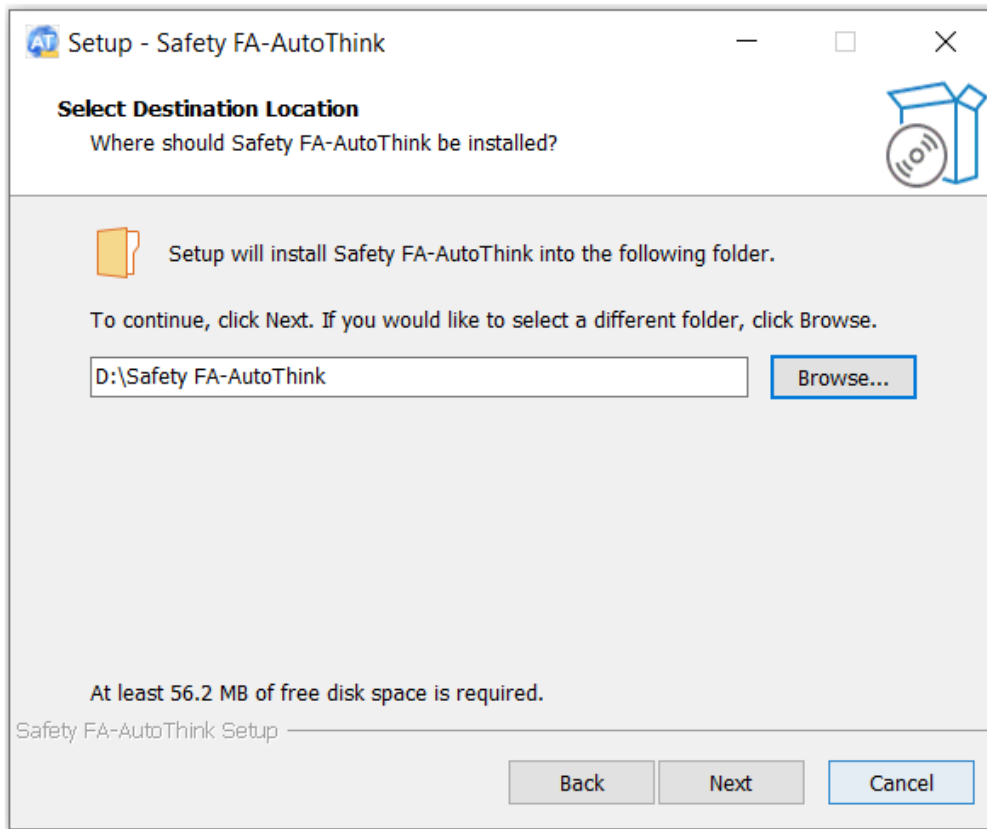


Figure 4 Select installation directory

Step 5. Select start menu folder

The window of **Select Start Menu Folder** pops up. Click **Browse...** to select folder, then click **Next**, as show in Figure 5.

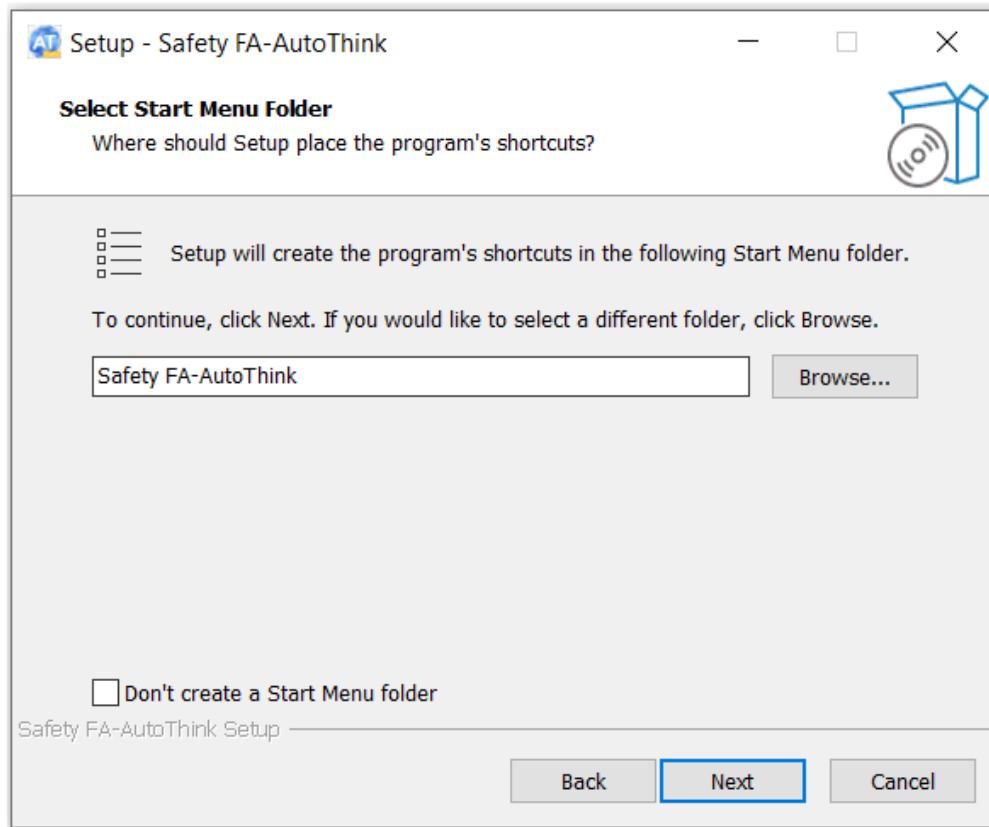


Figure 5 Name of start menu folder

Step 6. Select additional tasks

The window of **Select Additional Tasks** pops up, as shown in Figure 6. Check icon to be created below **Additional Icons**, and click **Next**.

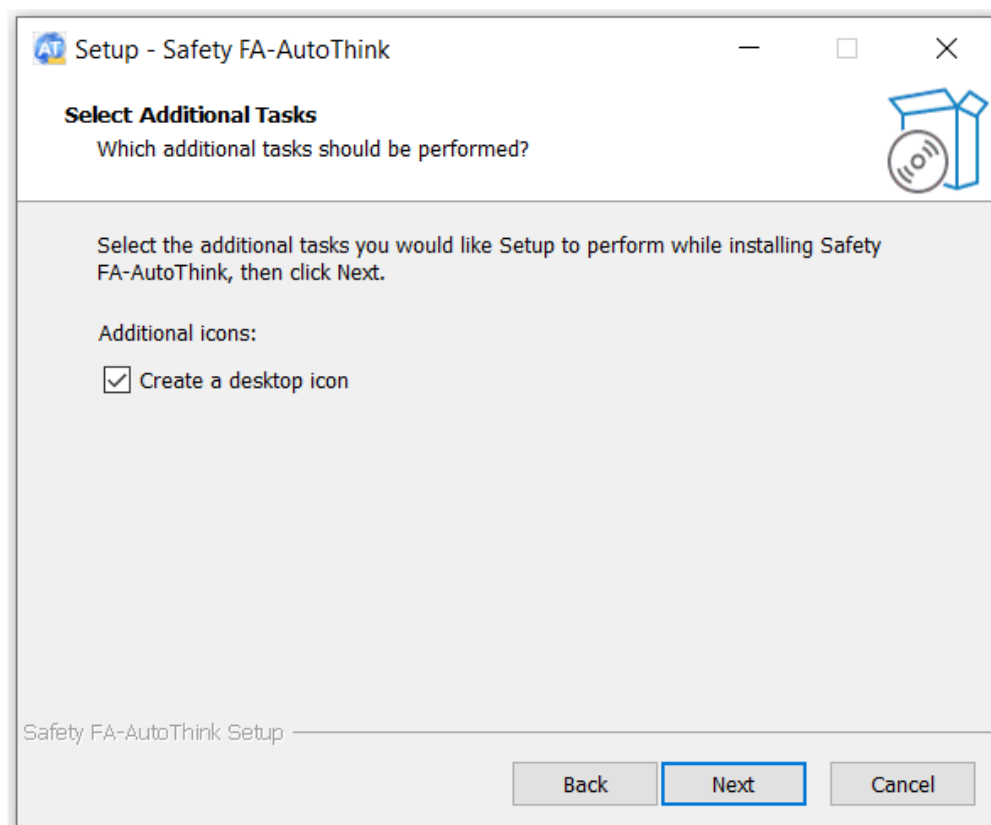


Figure 6 Create shortcut

Step 7. Prepare to install

The window of **Ready to Install** pops up, as shown in Figure 7, please confirm the installation information. Click **Back** if you want to modify. Click **Install** if no errors exist to begin installation.

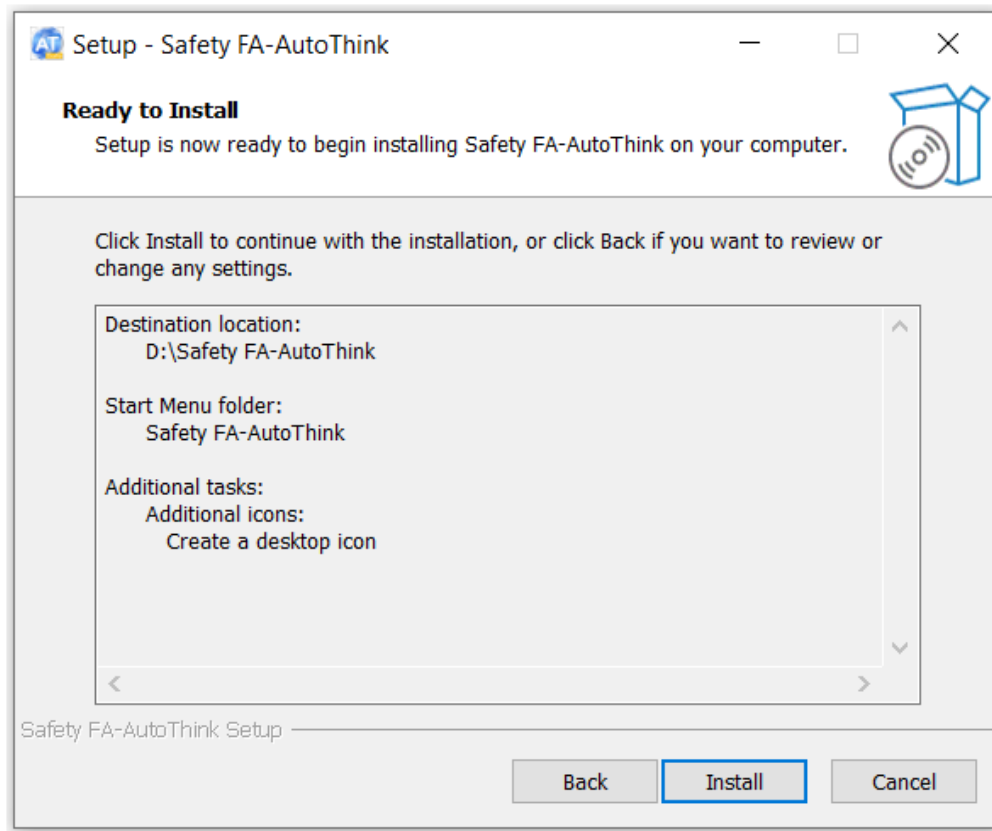


Figure 7 Prepare to install

Step 8. Installation process

The window of **Installing** pops up in the installation process with progress, as shown in Figure 8. Click **Cancel** if the user wants to cancel installation.

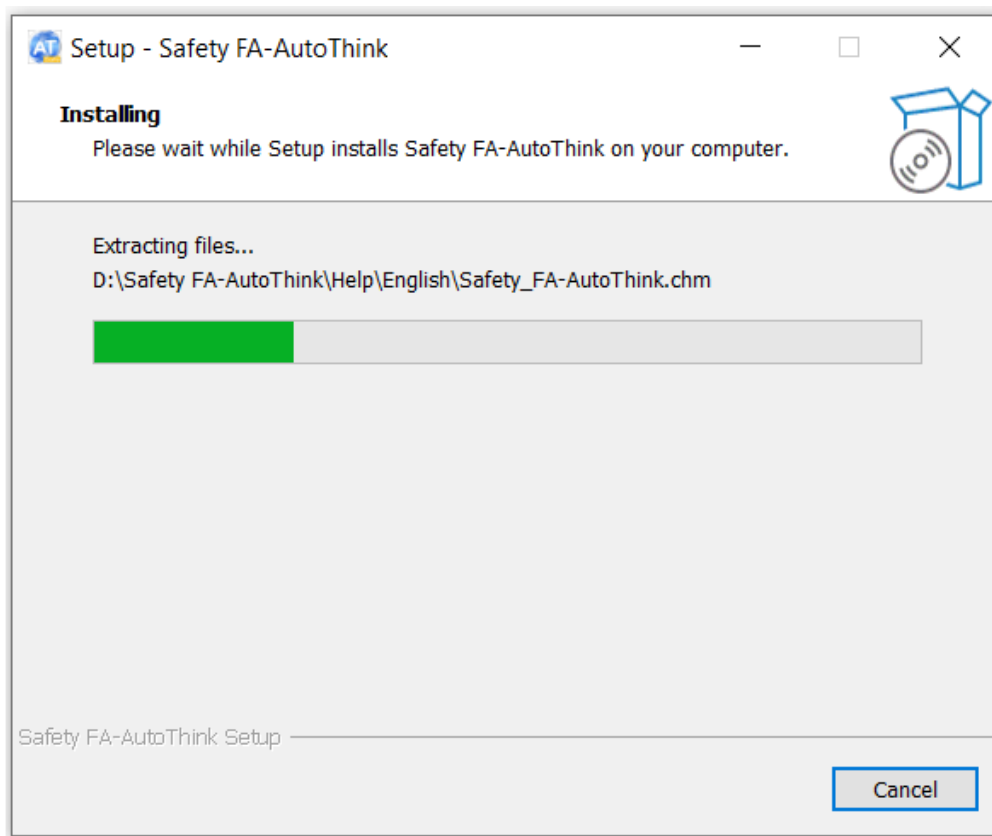


Figure 8 Installation process

Step 9. Prompt of installation complete

The window of **Completing the Safety FA-AutoThink Setup Wizard** pops up after the installation is completed, as shown in Figure 9.

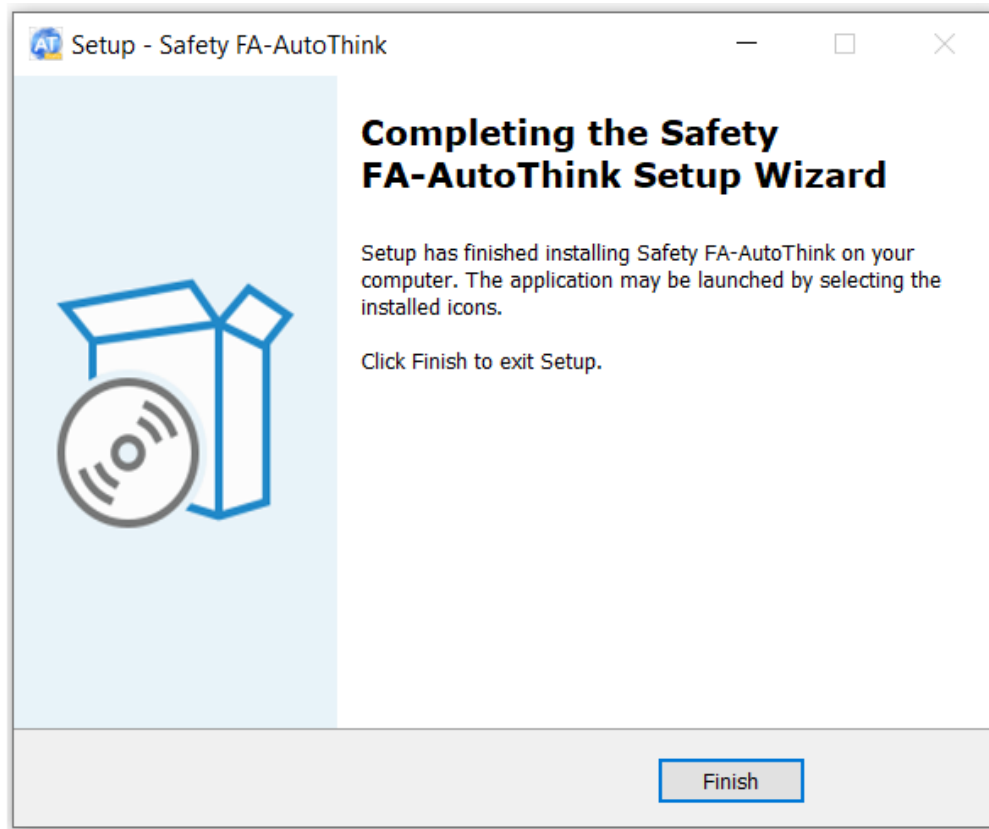


Figure 9 Installation complete

3.2 Uninstall

You can uninstall Safety FA-AutoThink through control panel or starting uninstallation procedure.

Starting uninstallation procedure:



- Desktop: Click [Start] Menu-[All Programs]-[Safety FA-AutoThink]-[Uninstall Safety FA-AutoThink].

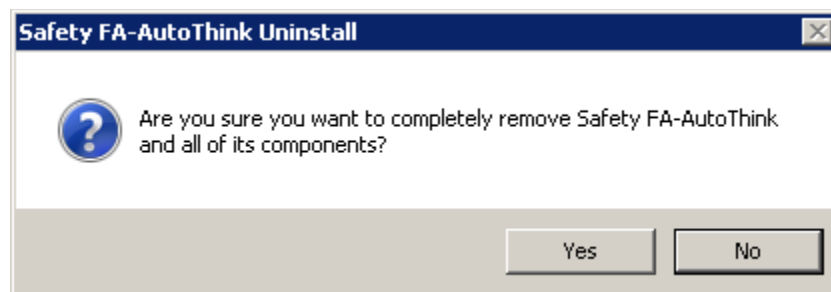


Figure 10 Confirm uninstallation

Click **Yes** to uninstall, as shown in Figure 11.

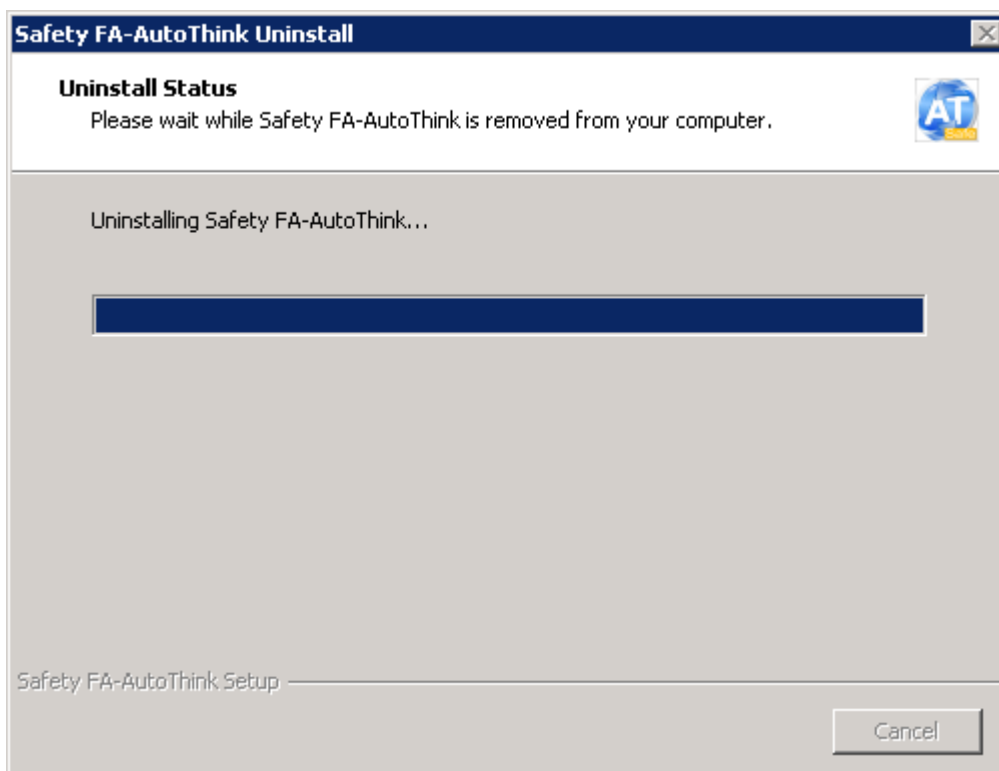


Figure 11 *Begin to uninstall*

Prompt window pops up after uninstallation is completed, as shown in Figure 12. Click **OK** to complete uninstallation.

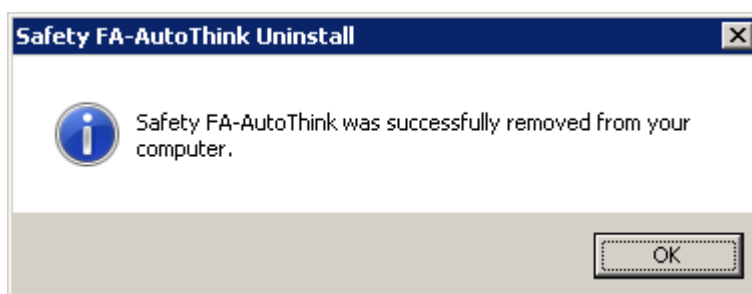


Figure 12 *Uninstallation complete*

Chapter 4 Start Safety FA-AutoThink



- Desktop: Click [Start] menu-[All Programs]-[Safety FA-AutoThink] folder-[Safety FA-AutoThink] software.

- Shortcut icon: Double click Safety FA-AutoThink icon on desktop.

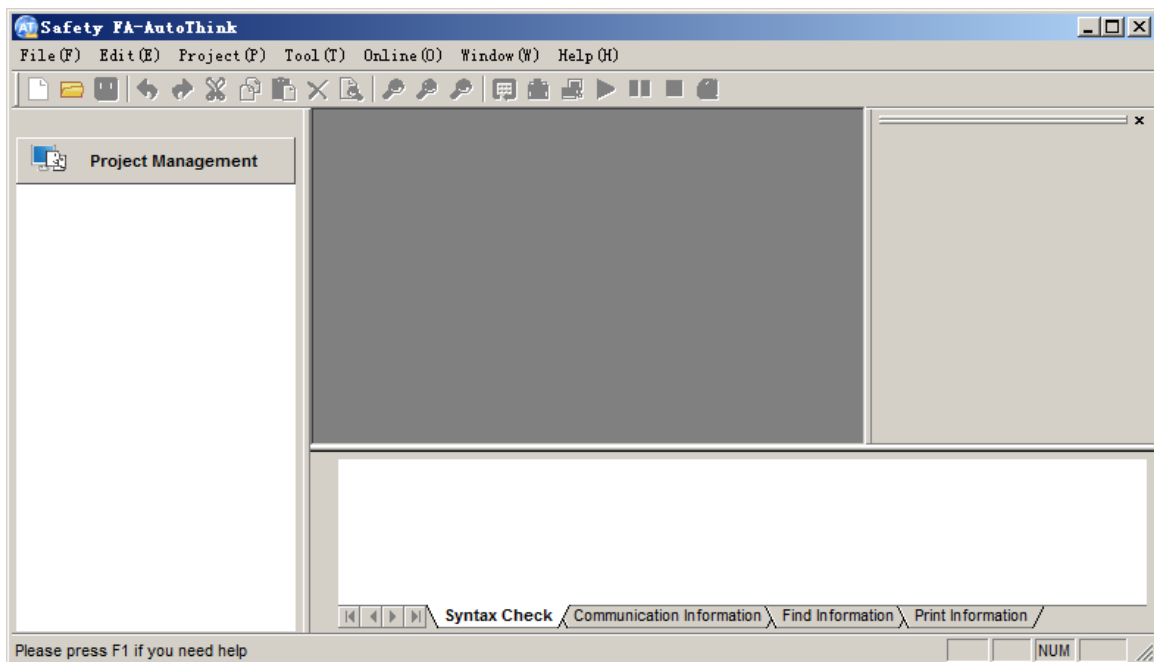


Figure 13 Safety FA-AutoThink window

You need to check the permissions before you open the created project.

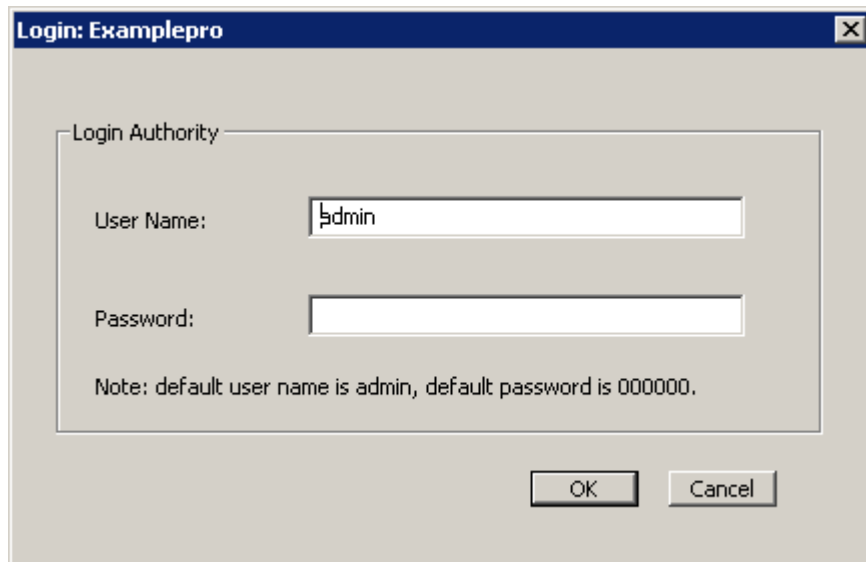


Figure 14 Login dialog

After permission check is passed, the Safety FA-AutoThink software window will appear.

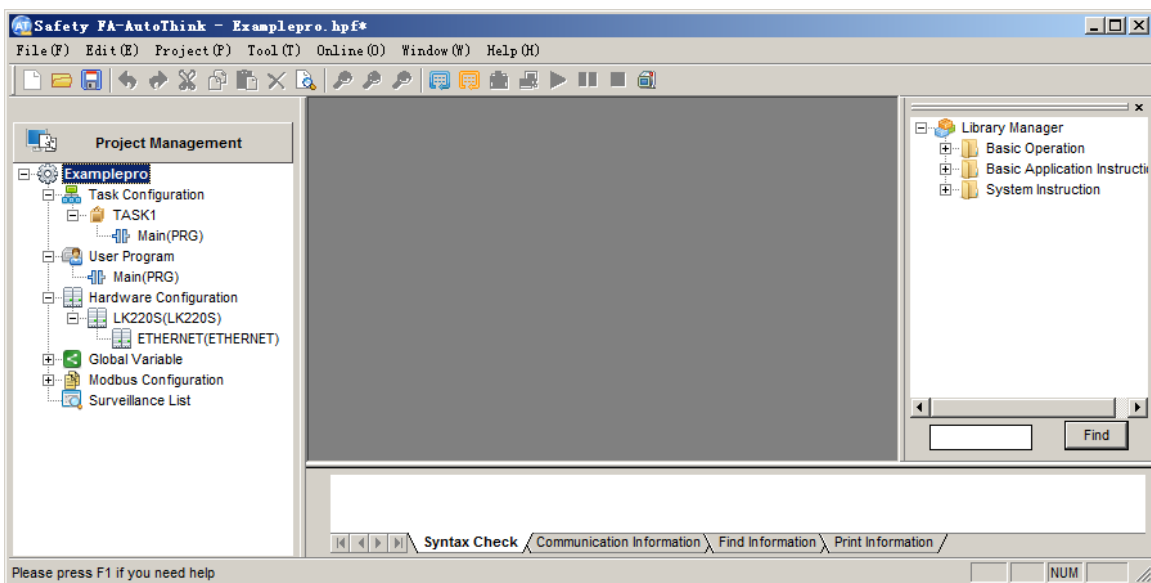


Figure 15 Configuration interface

After the project is opened, you can change the user information through the [File] - [Switch User] command.

Chapter 5 Software interface

Configuration interface is a core consisting part of the whole software and applied to configure project algorithm, compile and download project to a controller. This paragraph briefly introduces the configuration interface of the system.

5.1 Title bar

At the top of the window, it used to display the name of the related content. It is as shown in Figure 16.

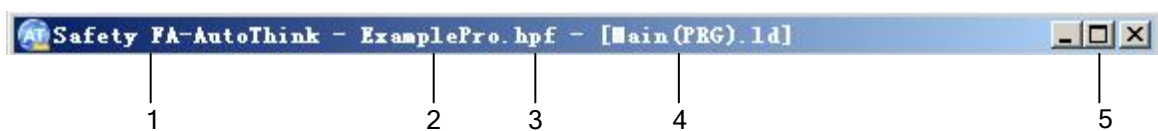


Figure 16 Title bar

- 1: Software Logo
- 2: Project name
- 3: The project change status (display * when project is not saved)
- 4: The current edit content
- 5: Window control button

5.2 Menu bar

You can click sub-menu in the drop-down menu to execute related operations. The menus bar displayed are slightly different according to editing contents in work area, generally, including the following menus.

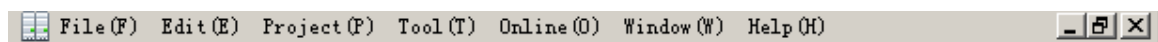


Figure 17 Menu bar

- Icons: Displaying types of interfaces currently opened, including:  (FBD language),  (LD language),  (hardware configuration),  (Global variable), etc.
- File: Execute operations related to project file, such as open, save, close, etc.
- Edit: Edit, replace and search existing content.
- Project: Compile and set the project being edited currently, and check its log, etc.
- Insert: Displayed when program block is configured, with components of graphical programming language able to be inserted.
- Tool: Provide wizards for controller operation.
- Online: Provide orders related to operations in actual system or simulation system, etc., such as download, monitoring, simulate, etc.

- Window: Provide orders for operations such as arrangement, switch display or close for the windows opened currently.
- Help: Provide help information of the software.

5.3 Toolbar

You can use not only the menu bar, but also the toolbar buttons to perform actions. Each icon is a command. There is a relevant tip as the cursor hovers over an icon.

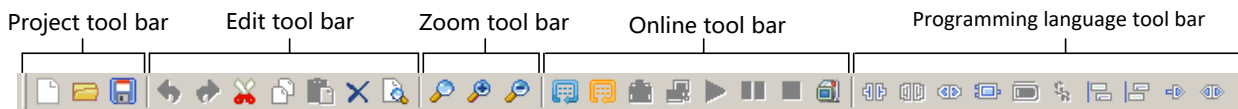


Figure 18 Toolbar

Table 4 Toolbar description

Tool bar	Description
Project tool bar	It is used to new, open and save the project files
Edit tool bar	It is used for editing the configuration content in project
Zoom tool bar	It is used to adjust zoom of the POU page
Online tool bar	It is used for online operation, including compile, download, debug
Programming language tool bar	It is used for offline configuration, this operation object is POU, its content changes according to the programming language

5.4 Project management

The project management adopts tree diagram to manage current project, and project name as the name of root node, including 6 sub-nodes: task configuration, user program, hardware configuration, global variable, Modbus configuration, surveillance list.

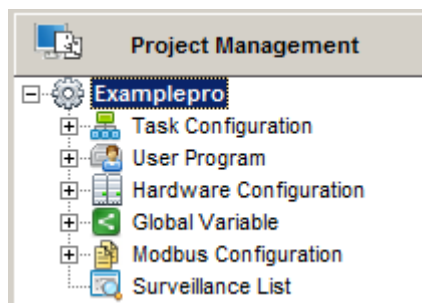


Figure 19 Project management

5.4.1 Task configuration

In Safety FA-AutoThink, a task is corresponding to the user programs' management of a project. Task configuration guides the execution of user programs, and you should define the user programs included in the task and configure task properties.

Task configuration adopts tree structure to display task name and programs called in task. The TASK1 as default task is added, and this task calls Main (PRG) by default.

5.4.2 User program

Below User Program node, users can use basic program unit - POU which contains Program, Function Block and Function meeting the IEC61131-3 standard to realize the control strategy programming.

User Program node contains the Main (PRG) by default. POU can be added, deleted, modified and classified by corresponding menus in [User Program].

5.4.3 Hardware configuration

Hardware Configuration is used to configure the hardware structure to realize connection with project data.

In hardware configuration window, you can configure project hardware, device parameters and communication parameters. **Device library** window provides various available modules used to configuration hardware.

5.4.4 Global variable

You can add global variable groups, define variables, view diagnosis information under [Global Variable], which includes 5 variable groups: DP diagnosis variable group, global variable group, Modbus diagnosis variable group, channel information variable group and system diagnosis group.

5.4.5 Modbus configuration

Variables used for Modbus communication are defined in Modbus variable group, details refers to Chapter [7.3.2.3 Modbus variable configuration](#).

5.4.6 Surveillance list

Surveillance list is a group of important variables, it is used to track and observe the changes of variables at runtime.

You can create multiple surveillance lists in a project to classify the important variables to organize them conveniently. You can delete or rename a surveillance list, and edit the variables in surveillance list. When adding a variable into a surveillance list, make sure you fill with the correct variable name.

5.5 Work area

This area used to complete the algorithms configuration, hardware configuration, and other configuration, the location is boxed in red as Figure 20.

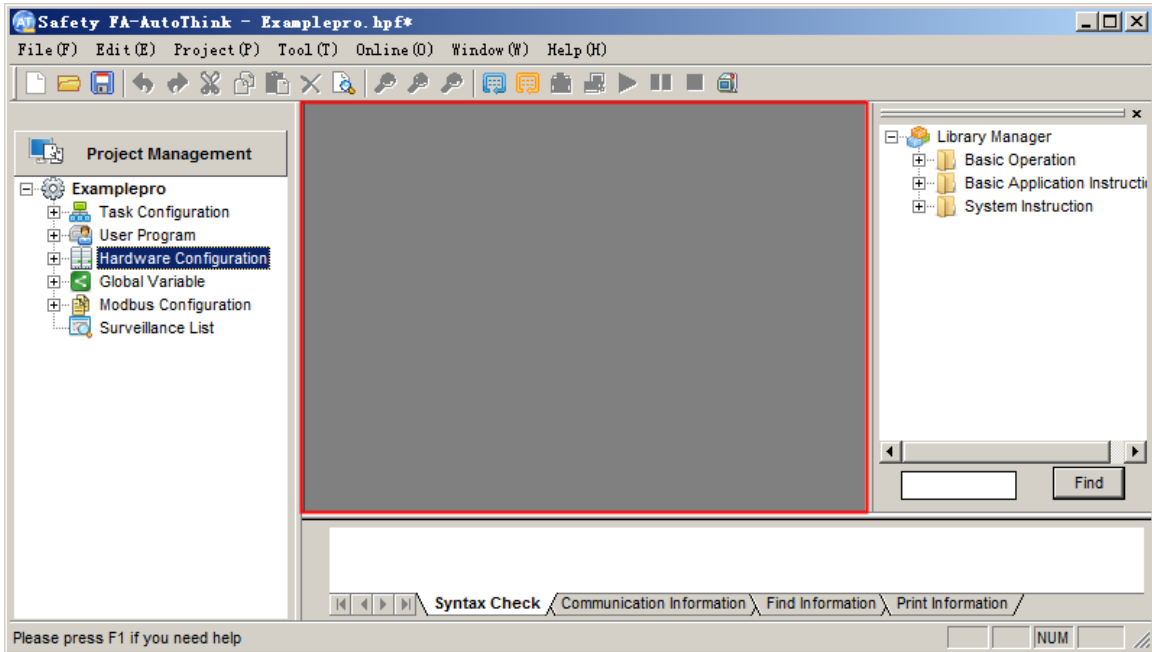


Figure 20 Work area

Any sub-nodes window under the engineering management tree can be displayed in the work area. Current opened editing windows are displayed by tab page style on the top of work area. Click relevant tab page to activate the window to edit and the identification will change to bright blue. You also can switch the active window in the opened editing window list in [Window] menu. If many editor windows have been opened, click the page turning button  in the top right corner to switch to the next or previous editing window.

5.6 Output window

Output window is used to display information related to compiling, error, warning, communication or finding.

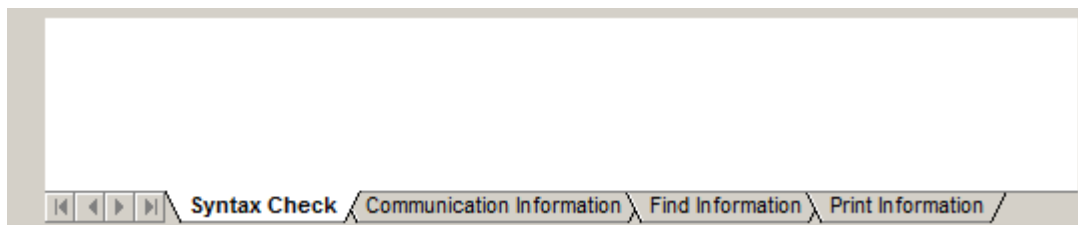


Figure 21 Output window

Double click row of compilation errors, warning information or find result displayed in information output window to automatically locate related position.

Abnormal information will be displayed in red.

You can copy and clear information through the right-click menu in output window.

5.7 Status bar

Status bar, locating at the bottom of programming interface, can display the related information of current window.



Figure 22 Status bar

Help information is displayed in the left side on the status bar, and key status on keyboard is displayed from left to right in turn: reserved, Caps Lock and Num Lock, Scroll Lock.

5.8 Library manager window

In Safety FA-AutoThink system, library contains common function block (FB) and standard function (FUN) to realize basic and common logic operations. More complex control function can be realized by library. The file format of library is "library name.hlf", such as HS_Standard.hlf (Library). " Safety FA_AutoThink\Target\Library" as the default storage path.

Libraries are taken to category and manage through "library manager" window in Safety FA-AutoThink. As shown in Figure 23.

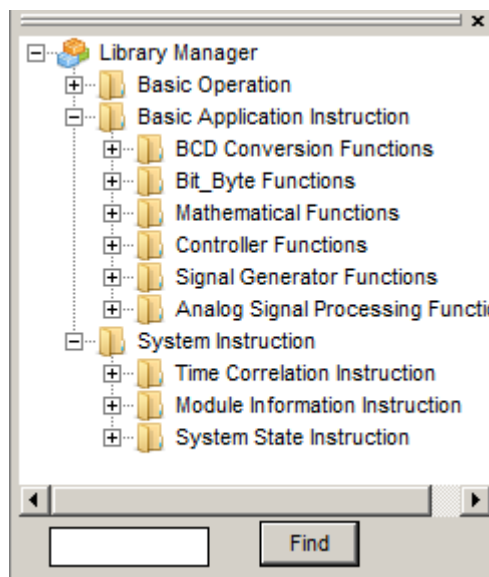


Figure 23 Library manager

5.8.1 Library configuration



- Library Manager: Right click on blank area.

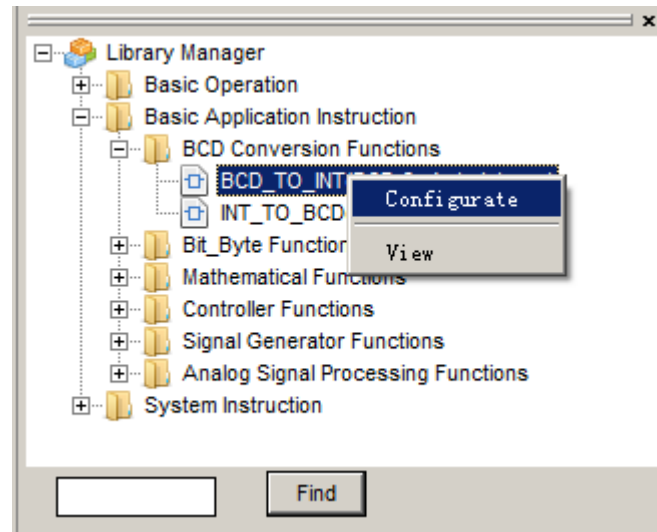


Figure 24 Shortcut menu on library manager

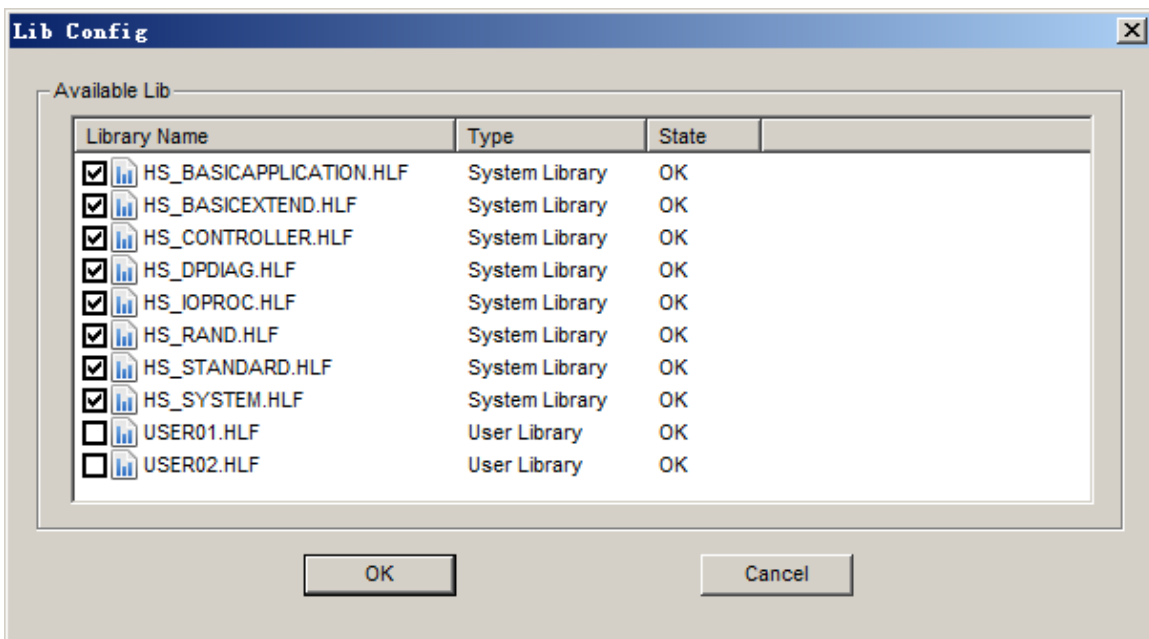


Figure 25 Library configuration

1. Related operations

(1) Sorting

You can click the list head (Library Name, Type and State) to sort according to the alphabetical order.

(2) Check box

When the check box is in the tick state, corresponding library file will be called. When a library file is not checked, the associated algorithm block is displayed in red in the library manager window.

2. Type

Library types are categorized according to implementation of libraries and no effect to use.

System library: System default libraries.

User library: User-defined libraries.

N/A: Identifier displayed when the library files are lost or broken.

3. State

The state of library is divided to two types: normal and abnormal, and libraries are red in abnormal state, including 4 types:

(1) Low version

The library file is in good condition, but the version is lower than the current software version.

- phenomenon
Display low version in state list.
- Treatment
You need to update library through compiling library and saving again.

(2) Broken

The library files are broken.

- phenomenon
Display N/A in state list.
- Treatment
The library files need to be repaired or replaced. Please touch with engineering implementers.

(3) Conflict

The libraries are repetition.

- phenomenon
Prompt information of libraries repetition will be displayed in the information window.

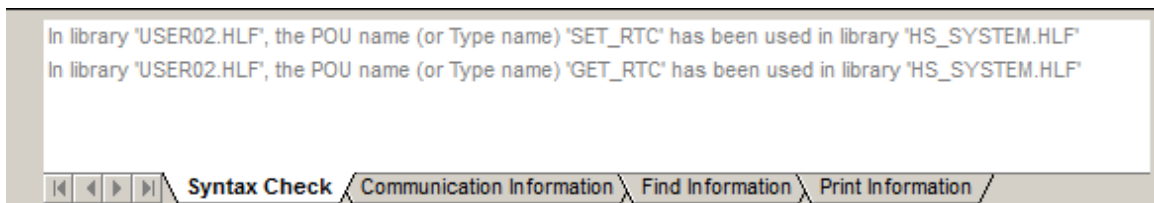


Figure 26 Prompt information of libraries repetition

- Treatment
You need to delete duplicate libraries to make sure libraries are not repeated.

(4) Lose

No library files.

- Phenomenon
 - ☐ The software cannot start when the system library is lost.
 - ☐ When the user - defined library is lost, opening project, system pops up the prompt information.

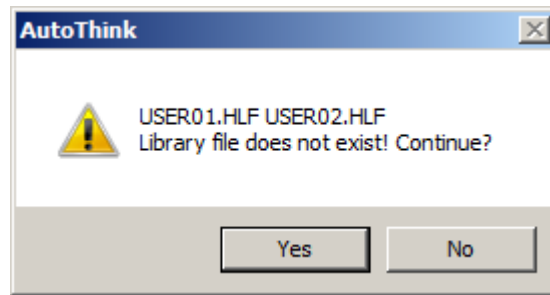


Figure 27 Prompt information

Meanwhile, library state is missing.

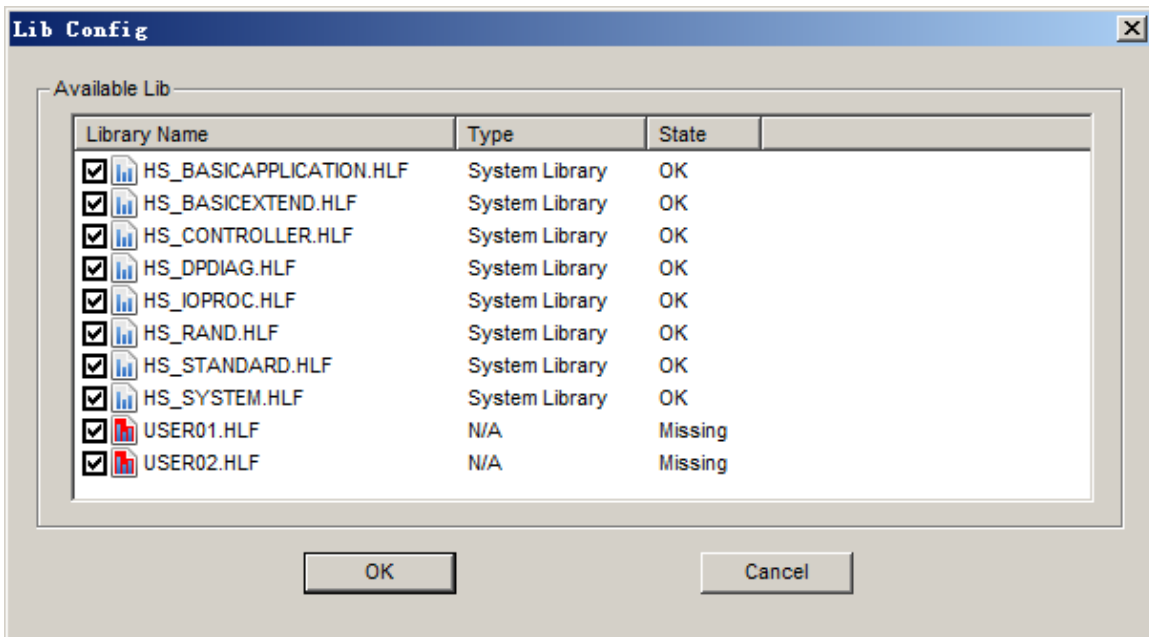


Figure 28 Abnormal library

■ Treatment

When the system library is lost, please touch with engineering implementers.

User - defined library, you need to update the valid library file to directory: Safety FA-AutoThink\Target\Library.



- Because there is a dependency relationship between individual libraries, when one of them is checked, the system will automatically check the other libraries in updating. When the only one is missed, this operation is invalid.
- If the abnormal library is checked, there will be an error prompt when the project is compiled.

5.8.2 View

You can view details of the function block via [View] menu.

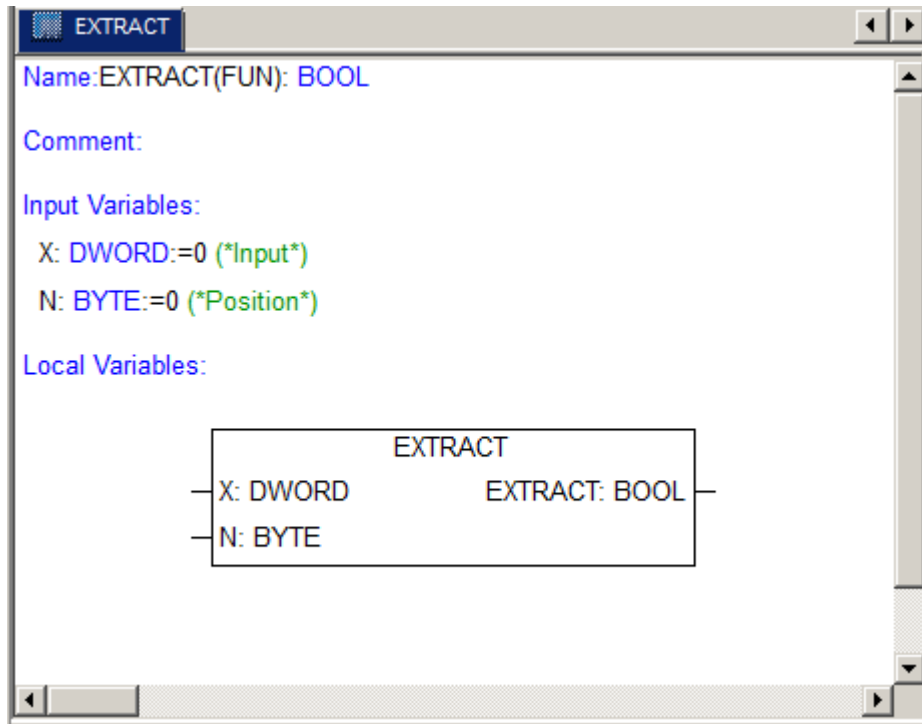


Figure 29 Details of the function block

Information window of function block consists of comment (only function blocks that support BOOL energy are displayed), variables, and function block diagram.

5.8.3 Find

Find tool in bottom of the library manager window help you find quickly.

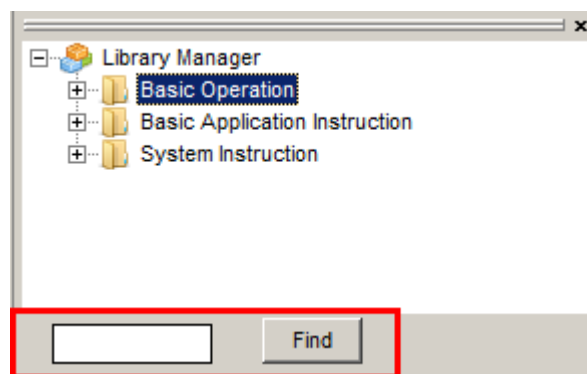


Figure 30 Find tool

Enter the lookup condition in the input box. The input contents are case-insensitive and do not restrict special symbols.

The system searches matched strings in the library manager. All instructions that name contains the matching string are displayed in list box.



- The conditions that the two strings match exactly are the same length and same character on the corresponding location.

The search results show in list box.

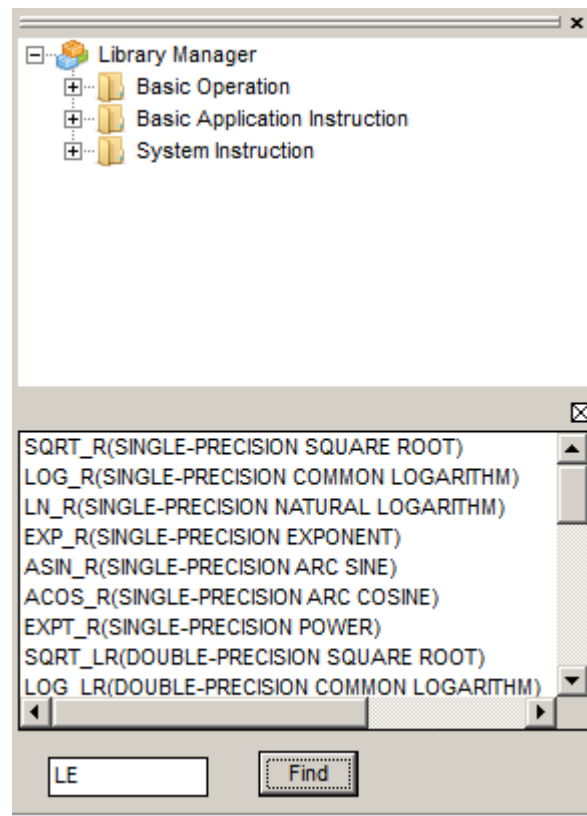


Figure 31 Find result

Double clicking on search results will automatically locate and identify in bright blue, as shown in Figure 32.

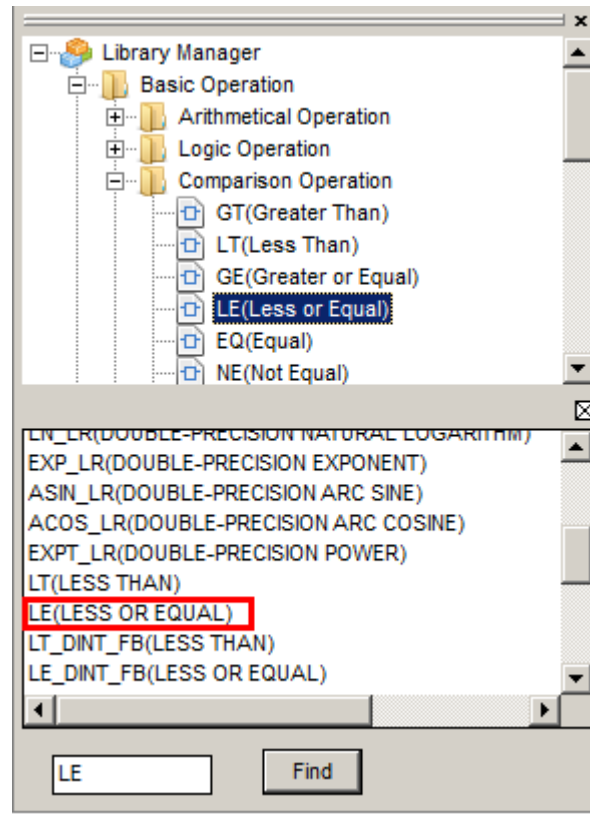


Figure 32 Locate result

5.8.4 Usage

You can drag function block or function from library manager to work area, when mouse is in  state, you need to release mouse to complete the add operation, otherwise, if mouse is in  state, you will not be able to add in the current position (existing components in current position or blank position too small).

Detailed description of the function block and function in the library, please see LKS Safety Control System Instruction Manual.

5.9 Device library window

This window is also adopts tree structure management. It mainly divided into controller, communication device and protocol.

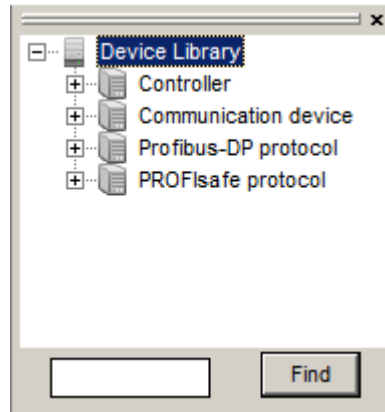


Figure 33 Device library

5.9.1 Find

Search tool in the bottom of the window help user to search quickly.

Enter the lookup condition in the input box. The input contents are case-insensitive and do not restrict special symbols.

The system searches matched strings in the device library. All modules that name contains the matching string are displayed in list box.

Double clicking on search results will automatically locate and identify in bright blue.



- The conditions that the two strings match exactly are the same length and same character on the corresponding location.

Chapter 6 Project management

6.1 Create a project

This order creates a new project or library.



- Toolbar: Click [File]-[New].
- Toolbar: .
- Shortcut: **Ctrl+N**.

6.1.1 Project

Select [New] to pop up dialog box of **New**, as shown in Figure 34.

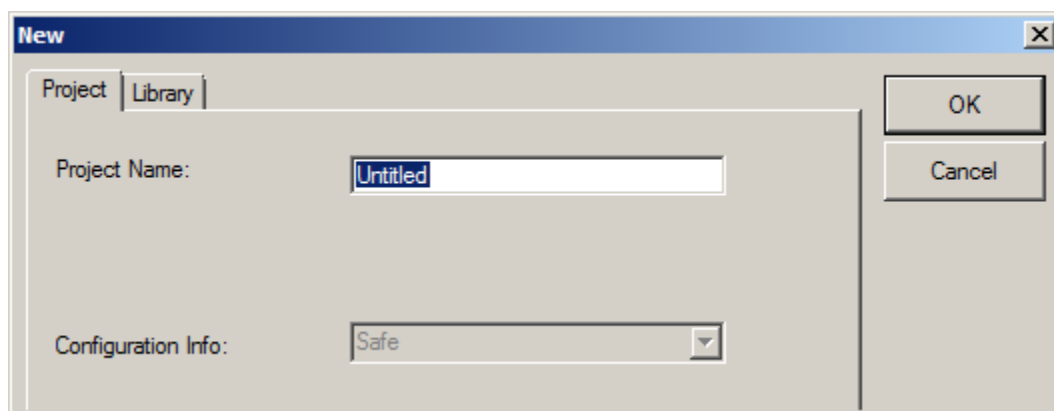


Figure 34 Dialog Box of New—Project

- Project name: **Untitled** as default.
 - ☐ The project name should not contain “\”, “/”, “:”, “*”, “?”, “””, “<”, “>”, “|”.
 - ☐ Project name should not be null.
 - ☐ The length of project name is no more than 32 bytes, and the exceed part cannot be entered.The button **OK** cannot be operated when project name is invalid.
- Configuration Info: Safe.

6.1.2 Library

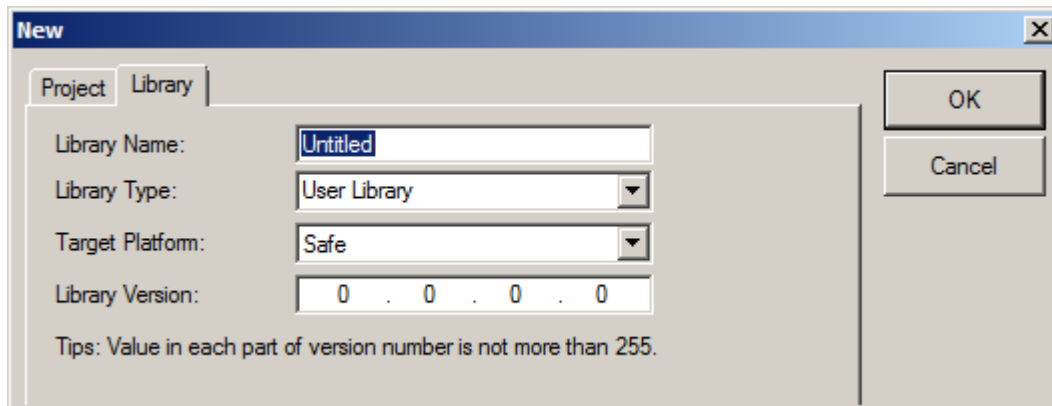


Figure 35 Dialog box of New- Library

- Library Name: **Untitled** as default name. The library name cannot start with HS_, and see project name for other rules.
- Library Type: User Library.
- Target Platform: Safe.
- Library Version: The version number takes four-segment format, with each segment's number range being 0~255. The user can record change version of library by it (recording change date, such as "20.15.01.20").

6.2 Task configuration

User programs that are called by task only be executed. The following part will describe attribution of task configuration node and how to add or delete POU calling in task.

6.2.1 Program calling

The TASK1 node is generated by default under [Task Configuration], which calls program Main (PRG) configured under the [User Program] by default. User programs are called by right-click menu on TASK1.



- Project management tree: Right click TASK1, and click [Add Program Call].

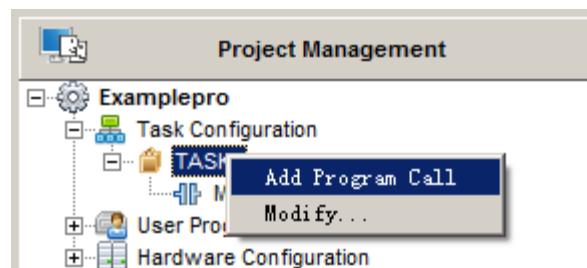


Figure 36 Add program calling

The dialog box of Add Program Call pops up, and all POU(PRG) under the node of [User Program] are displayed in Can add POU list box, all the POUs which have been called in the Task are displayed in Under the current task POU list box, as shown in Figure 37. Choose POU needing to be called, click Add to call the POU. After each addition, the cursor in the left list box is automatically moved to the next POU position.

Select the POU in the right list box and click the **Delete** button to delete the calling. You can use the Shift or **Ctrl** keys to select multiple POU when adding or deleting.

Click **Select All** to add all the POUs from **Can add POU** list box to **Under the current task POU** list box.

Click **Delete All** to delete all the POUs that have been called in the task.

Click **Reset** to restore the POU calling to the state it was in when the window was opened.

Click the button ,  to adjust the calling order of the POUs in the right list box.

When the editing is completed, click **OK** to close the Add Program Call dialog.

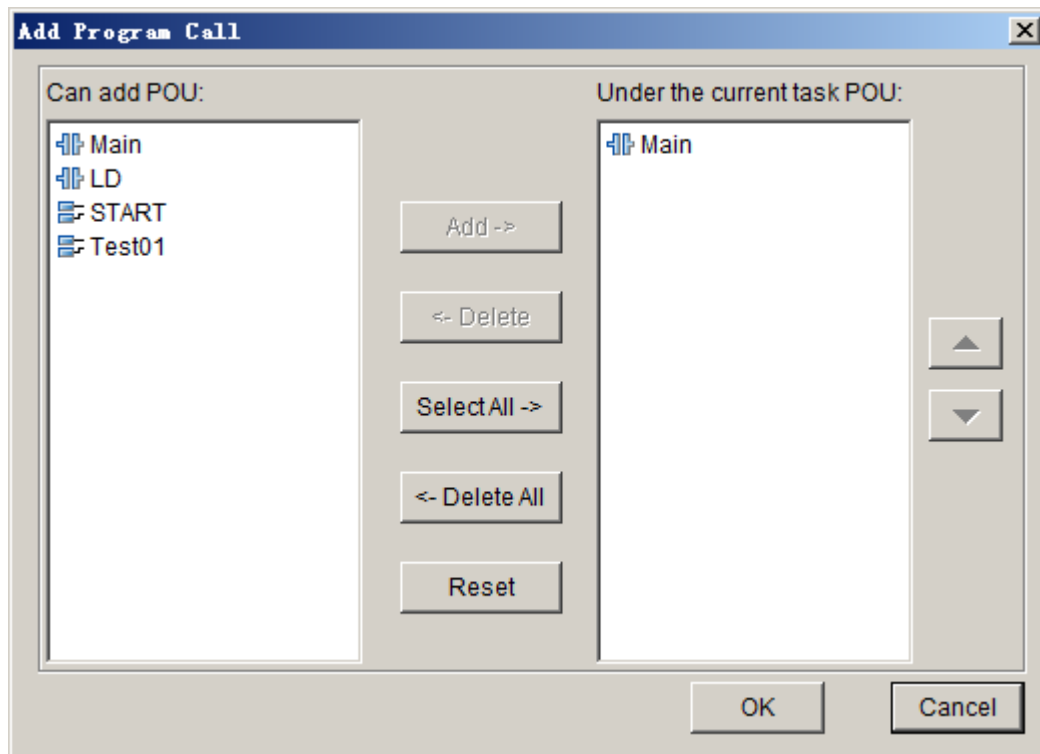


Figure 37 Select Called POU

The called POU in task is executed according to the sequence from top to bottom.

■ Rename called POU

Modify the name of POU under the node of [User Program], and the POU with the same name under task node will be updated at the same time.



- Project management tree: Right click the called POU under the node of [User Program], and click [Rename].

■ Delete called POU



- Project management tree: Right click the called POU under the task node, and click [Delete].

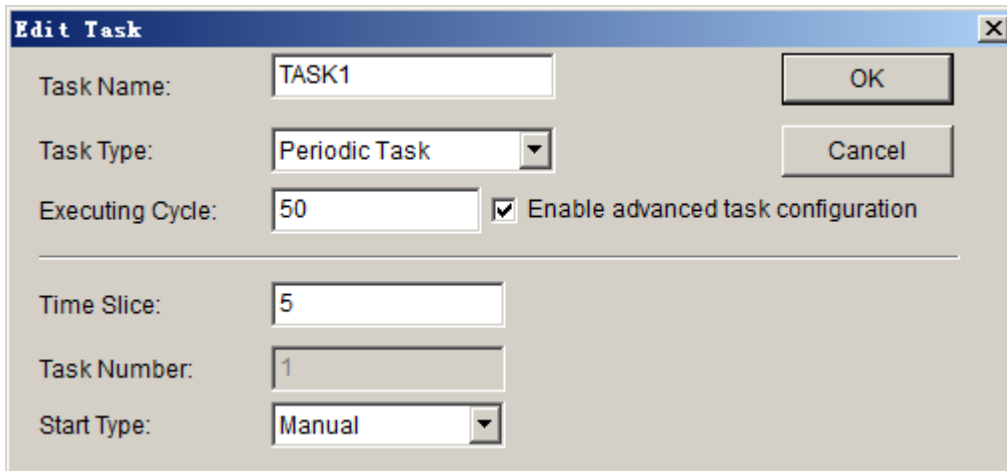
If the called POU under the node of [User Program] is deleted, then the POU of the same name under task node will be deleted at the same time.

6.2.2 Configure task property



- Project management tree: Right click TASK1, and click [Modify].

Prompt the **Edit Task** dialog, as shown in Figure 38.



The **Edit Task** dialog box contains the following fields and controls:

- Task Name:** Text field containing "TASK1".
- Task Type:** Dropdown menu set to "Periodic Task".
- Executing Cycle:** Text field containing "50".
- Enable advanced task configuration:** Checkmark is selected.
- Time Slice:** Text field containing "5".
- Task Number:** Text field containing "1".
- Start Type:** Dropdown menu set to "Manual".
- Buttons:** "OK" and "Cancel" buttons are located on the right side.

Figure 38 Edit task

- **Task Name:** The name is composed of figures, letters and underline. It should not start with figure and length should not exceed 32 characters.
- **Task Type:** Periodic task. Task runs circularly according set Time Slice and Executing Cycle.
- **Executing Cycle:** Time interval for periodic execution of task, 50ms as default value. Value is set to an integer between 15~100,000.
- **Time slice:** The minimum execution time unit which is divided by the system. You can set the number of time slice, being 1 by default, with options of integer between 1~50. Default value of each time slice is 62.5 μ s.
- **Task number:** Task ID.
- **Start type:** Being Manual by default, with options of Manual and Auto. If Manual is selected, after the project is downloaded and enters monitoring state, the task is in stop state and cannot run until manually clicking the button of Run. If Auto is selected, the task runs automatically after the project is downloaded.

6.3 Set project safety property

Right click menu of the root node (that is, the project name), click **Property** to set safety property of project and view configuration information of this project.

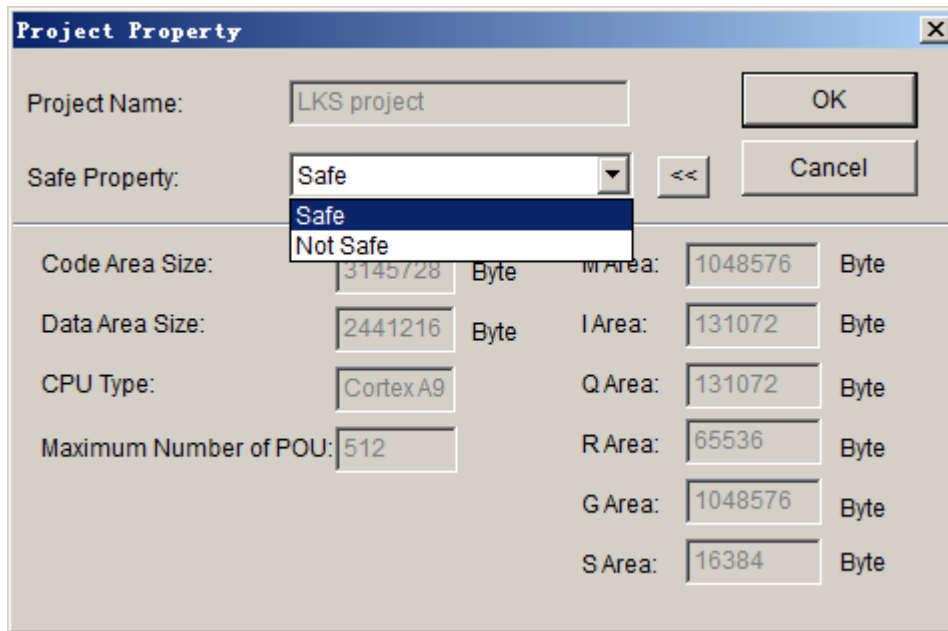


Figure 39 Project property

6.3.1 Set safety property

Project can be set as safe or unsafe, default as safe project, you can set safe property according to the actual system requirements.

In safe project, the system only as a slave station to communicate with Modbus master station, and read-only, cannot write data from master station. In safe project, all data areas will be restored to the initial value (except S area) when the key switch of controller is dialed to RUN position.

6.3.2 Configuration information

■ Code Area

The area stores you configured control logic (logic in POU of PRG type), customized function blocks (logic in POU of FB type) and function (logic in POU of FUN type).

■ Data Area

The area stores you declared variables, which include local variables and global variables.

Maximum Number of POU

It is the sum of the user POU number and system POU number.

■ M area

Parameter area. It is used to store the parameter variables or constant. In running mode, variables value of this area can't be assigned, but can be modified by Write Variable. It is synchronized in this area.

■ I area

Input area, at the beginning of each term of the scanning cycle, CPU will sample the input data, and store them into Input Area of memorizer. It will access the data as BIT, BYTE, WORD or DWORD.

■ Q area

Output Area, at the end of each term of the scanning cycle, CPU will transmit data from the Output Area of memorizer to physical output tags. It will access the data as BIT, BYTE, WORD or DWORD.

■ R area

Retentive area, the area is used to store power-loss retentive variable. If the data or variables are defined to R area, the system will automatically save the data and restore the data during power-off.

■ G area

Common variable area. It is used to store and manage the data and state produced in the intermediate process.

■ S area

Special register area is used to store Modbus TCP and system diagnosis information.

Please choose the area (G area, M area and R area) according to requirement when you define variables.

If the project is compiled successfully, you will view the usage rate and size of each area in information window.

6.4 Project safety

It is used to manage the authority of the project, to ensure the safety of the project. The operation scope varied with your authority level.



- Menu bar: Click [Project] - [Project Safety].

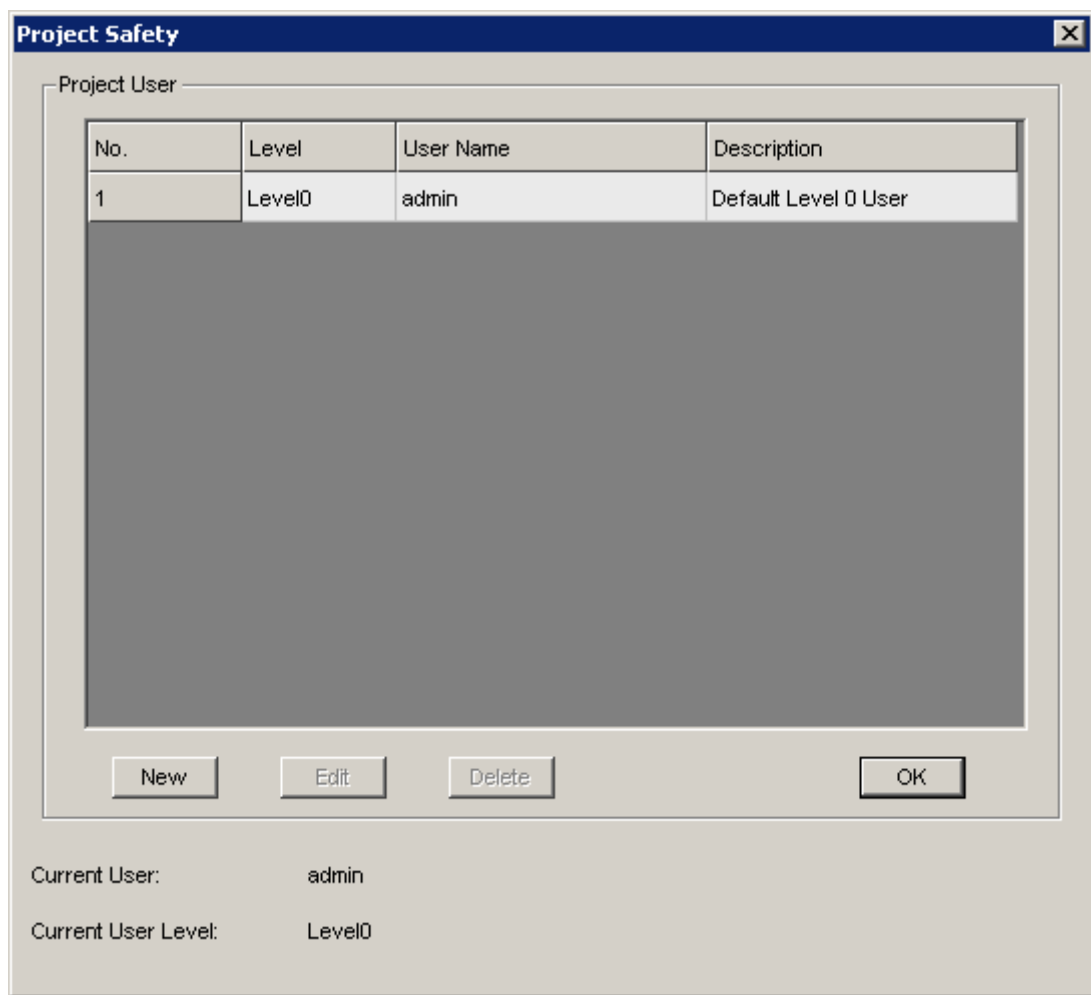
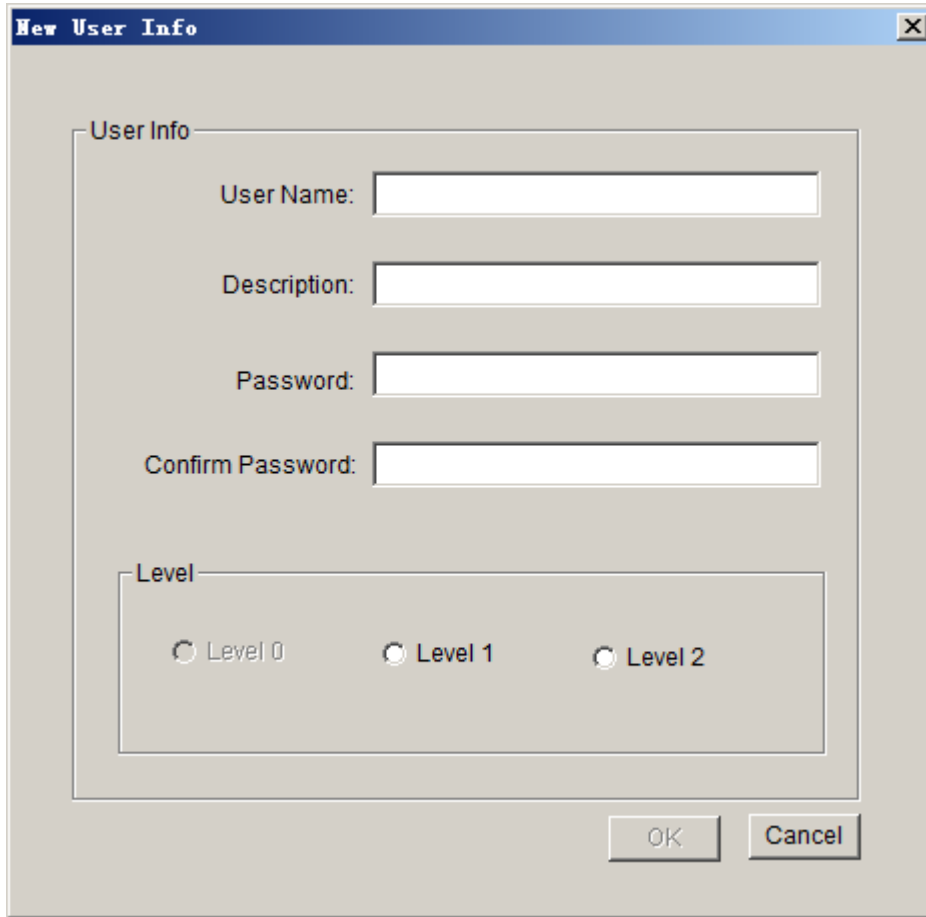


Figure 40 Project safety

The newly created project defaults to level 0 users.

6.4.1 New

This command is used to create a new user; the new user level can be set to Level1 or Level2.



The dialog box titled "New User Info" contains two main sections: "User Info" and "Level".

User Info section:

- User Name: [Text input field]
- Description: [Text input field]
- Password: [Text input field]
- Confirm Password: [Text input field]

Level section:

- Level 0: ☐
- Level 1: ☐
- Level 2: ☐

At the bottom right, there are "OK" and "Cancel" buttons.

Figure 41 New user

User Name: The name can only consist of alphabetic, numeric and underline (" _"), the first character must be an alphabetic character or underline, and the length of user name should not exceed 32 bytes.

Description: It describes the user information briefly and can be any characters; the length can't exceed 256 bytes at most.

Password: It can contain any character, the length is 6~12 characters.

Confirm Password: The characters inputted here must completely keep consistent with the characters inputted in the field of Password above.

Level: Set the user authority level. Safety FA-AutoThink provides three levels of authority: Level0, Level1 and Level2, in descending order. The authority level of new user must be lower than that of the current user. More detail about authority level and operate restriction, see the Table 5 for the specific operation authority of each level.

Table 5 User authority list

Authority	Level0	Level1	Level2	Description
Create project	√	×	×	
Modify the user name and password	√	×	×	A high-level user can edit the user name, description, and password of a lower-level user. Same-level users can only modify their own

Authority	Level0	Level1	Level2	Description
				user name and description, can not modify the password and user information of other same-level users
Create, modify or delete level1 user	√	×	×	
Create, modify or delete level2 user	√	√	×	
Offline Configuration	√	√	×	
Modify project and save project	√	√	×	
Download	√	√	×	
Run, Stop and Pause task	√	√	×	
Online monitoring	√	√	√	
Simulation	√	√	√	
Debug variable	√	×	×	
Upload project file	√	√	√	
Download project file	√	√	×	
Variable debug readback	√	√	×	
Clear controller	√	√	×	
Controller calibration time	√	√	×	
Reset, Cold reset and Warm reset	√	√	×	
Upload SOE File	√	√	√	
NTP Settings	√	√	×	



- When the Level0 user forgets the password, it cannot perform the highest authority operation on the project. Please contact the vendor for assistance.

6.4.2 Edit

You can change the existing user information, such as password, permissions level; but the level 0 users cannot modify their own permission level.

6.4.3 Delete

Delete an existed user. You can't delete a user whose authority is above or equal the authority of Current User.

6.4.4 Action timing

In a project, it needs to verify project safety in the following situations:

- Login

When you open an existed project, it needs to verify project safety;

- Switch a user to login

When you select [File] - [Switch User] command to switch a user to log into a project, it needs to verify project safety.

- Set controller password

Through [Tool] - [Assistant tool] -[Controller Operation]-[controller lock]to set controller password for online operation, before operation, you need to enter the controller lock password. Details see Chapter [11.1.3 Controller lock](#).

6.5 Library safety

You need to enter a password to open user library and set password through [Library Safety].



- Menu bar: Click [Project] - [Library Safety].

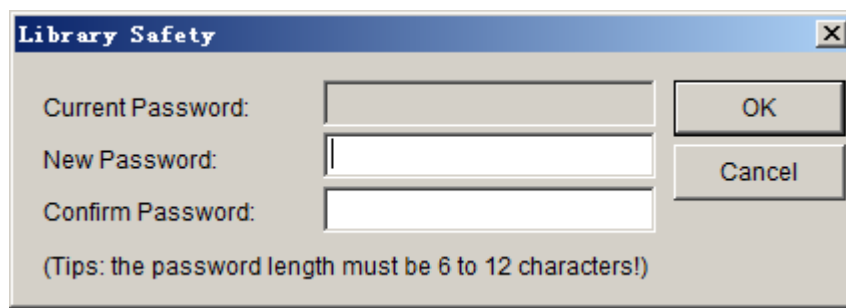


Figure 42 Set library password

The password can be set to any character with a length of 6~12 characters.

6.6 Project setting



- Menu bar: click [Project] - [Options].

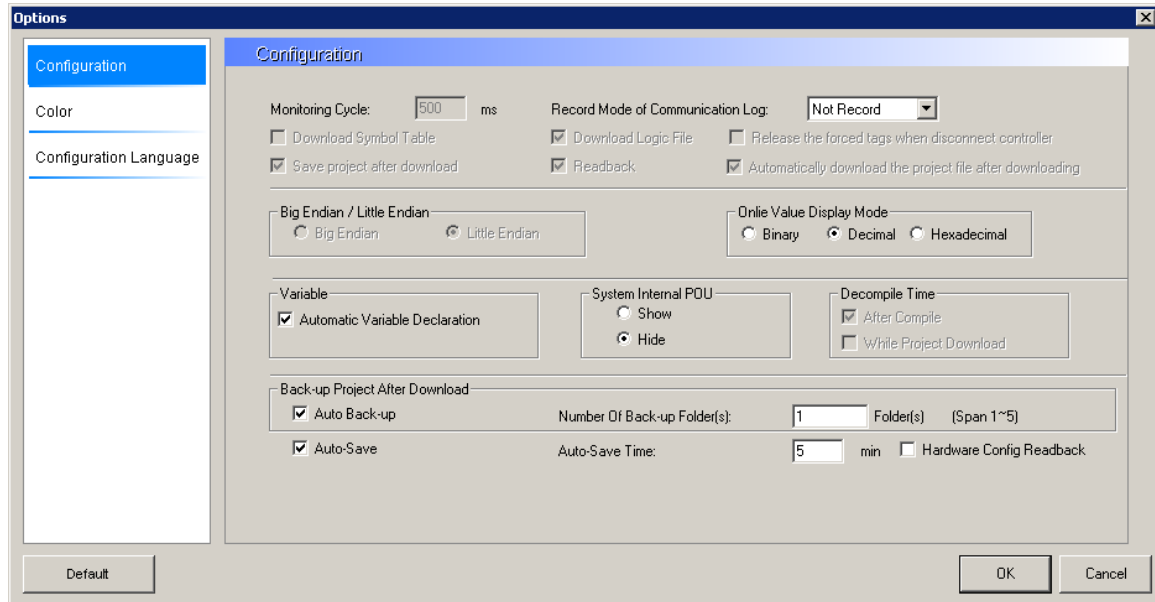


Figure 43 Options

6.6.1 Configuration

- **Monitoring Cycle:** Safety FA-AutoThink software from the controller to read the data interval, the default is 500ms.
- **Record Mode of Communication Log:** Set whether to record into communication log and the type of records. There are three modes: Not Recorded, Compact Mode and Full Mode.
- **Download Symbol Table:** Download symbol table generated by project compiling to specific control station. Default status is selected.
- **Download Logic File:** Download executable logic generated by project compiling to specific control station. Default status is selected.
- **Release the forced tags when disconnect controller:** set whether the force function is related to online connection.
- **Save project after download:** Set whether the project is automatically saved after download.
- **Readback:** Set whether the data can be read back to the project in debugging.
- **Automatically download the project file after downloading:** Set whether the project file is automatically downloaded after the successfully download.
- **Big Endian/Little Endian:** It represents data storage. Little Endian indicates that the high-bit of data is stored in high address of memory. Big Endian indicates that the high-bit of data is stored in low address of memory. The default is Little Endian.
- **Online Value Display Mode:** Set the online data format in online or simulation. Users can choose Binary, Decimal or Hexadecimal, the default is Decimal.
- **Automatic Variable Declaration:** Set whether pop up an automatic variable declaration dialog to define the variable.
- **System Internal POU:** Used to set whether HSGetQ POU is displayed in TASK1. The program associated with DP slave diagnosis is configured in HSGetQ POU, and does not support editing.

- **Decompile Time:** set when to carry out the decompile operation, default to execute after compile.
- **Back-up project after download:** Set whether the current project is backed up automatically after download.

When system backup project automatically, all project documents in current project directory are backed up. Backup project is saved to BackUp_PRJ folder in project path with name :project name_ backup time (year_month_day hour-minute-second).

The storage of the path of the maximum number of backup depends on the **number of back-up folder** of preset; when the backup project number more than set to the maximum value, the system will automatically delete the project file backup. The abnormal case occurs during backup, the system will prompt you for backup.



- When you restore a backup, please close the project configuration window and replace the project source file with the backup file.

- **Auto-Save**

The project will be saved automatically at intervals of the set time.

When the project is closed abnormally and opened again, the user can choose to open the project automatically saved by the system or manually saved by the user.

- **Hardware Config Readback**

Check Hardware Config Readback, the software read-back user parameters for you to confirm after compiling passed, and you need to confirm read-back information when downloading.

6.6.2 Color

In this category, it displays color parameter settings related to program, the object as shown in Figure 44.

In this dialog box, current default colors will be shown to distinguish objects in program definition. Users can modify it by drop-down menu to make configuration content easier to distinguish, or use **Default** to set the configuration to default.

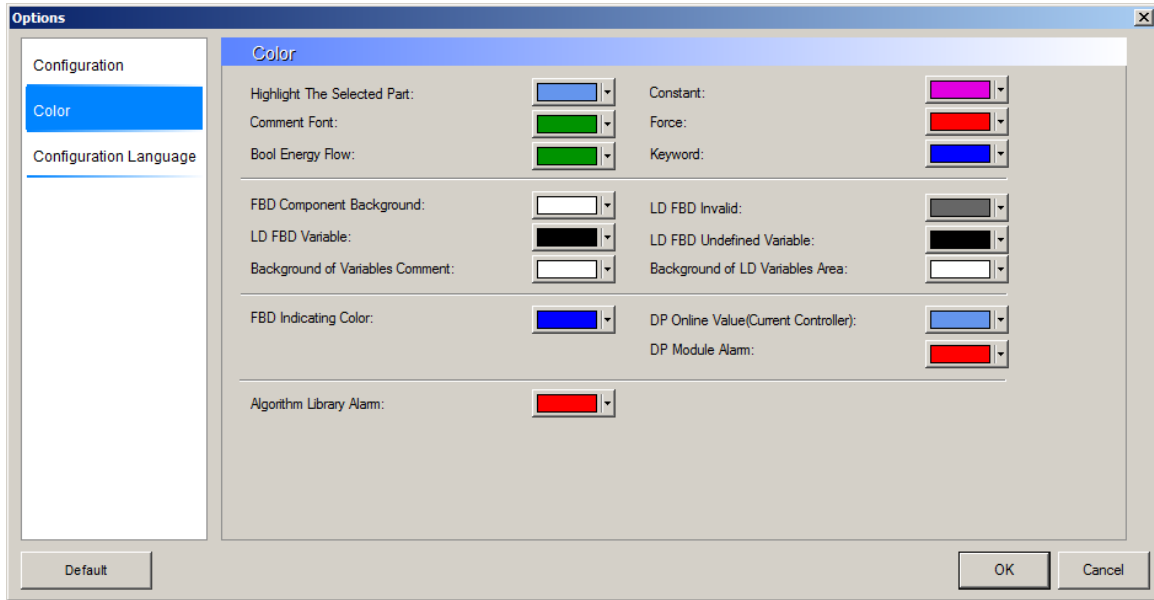


Figure 44 Color

- Bool Energy Flow: This is used to set the color of the pin line as the BOOL type pin output TRUE. The default is green, and you can change to other colors.

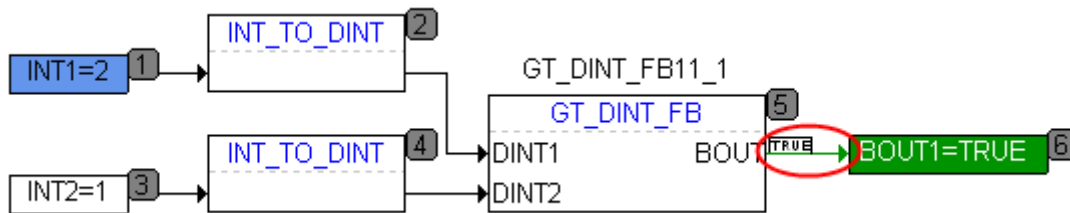


Figure 45 BOOL energy flow



- Bool Energy Flow configuration may affect the output pin color of BOOL type (except XOR and NE).

6.6.3 Configuration language

In this category, you can set show features of all kinds of IEC configuration languages.

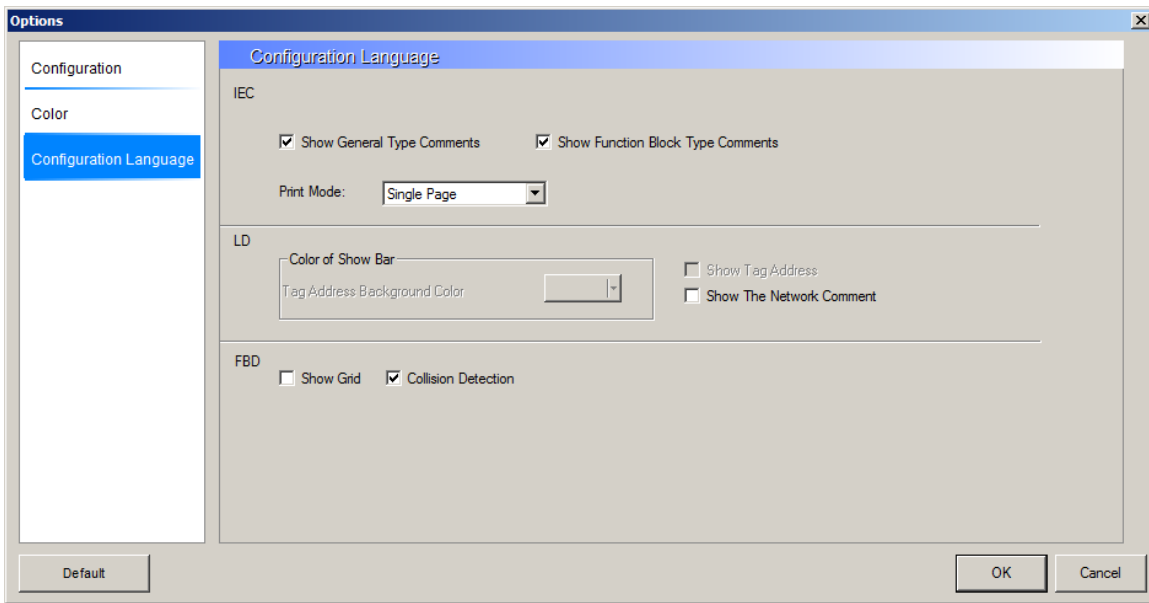


Figure 46 Configuration language

6.6.3.1 IEC

- Show General Type Comments: Used to display the variable description of common type variable. When the variable description is not input, the default is CMT.
- Show Function Block Type Comments: Used to display the variable description of function block type variable. When the variable description is not input, the default is CMT.
- Print Mode: Set the POU print mode, the default is Single page.
 - ☐ Multi Page: This way is in accordance with the printer set in the paper specifications print POU content. POU in FBD language is divided into several in accordance with the paper size in printer; the pages are separated with black dot lines.

This print, system will automatically add a page number for each page, the number arrangement according to from left to right, and from top to bottom order:

Each page number in LD according to its line number and column number, as 1-1, 1-2, 2-1.

No page number in FBD.

- ☐ Single Page: it is used to make the whole content of a POU on one paper and print. If the paper type/size in POU and the printer is different, then system will automatically adjust the content show percentage, to ensure the content is printed on one paper.

6.6.3.2 LD

Display related parameters setting of LD language.

Color of show Bar: Set the background color of address name in user program configured by LD language.

Show tag address: It is default that set to display address after adding a contact or coil.

Show the network comment: Set whether to display and edit the node note or hide it.

0001	comment:

Figure 47 Network comment

6.6.3.3 FBD

Display related parameter settings of FBD language.

Collision Detection: Detect whether each component is allowed to overlap when programming user program by FBD language. when collision detection is not checked , previous components will be partially or completely covered by the new ones.

Show Grid: Set whether to display grid in editing area. The default status is no-checked.

6.6.4 Default

Use this button to set all the parameters to default value.

6.7 Save project

6.7.1 Save



- Menu bar: click [File] - [Save];
- Toolbar: ;
- Shortcut key: **Ctrl+S**

All modifications of the current project are saved.

6.7.2 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].

6.8 Close project



- Menu bar: click [File] - [Close];
- Toolbar: .

Close the current project, but software program (Safety FA-AutoThink.exe) does not quit.

6.9 Recent projects list

- 
- Menu bar: Click [File]-[Recent Projects List].

The list of recently opened projects is displayed, including project path and project name. Using this order can rapidly switch between projects.

6.10 Upload project file

- 
- Menu bar: click [Online] - [Upload Project File].

Use this command to upload the project file from controller to local computer.

The [Upload Project File] and [Download Project File] are a pair of reciprocal operations to copy project file between local computer and controller.

6.11 Download project file

- 
- Menu bar: click [Online] - [Download Project File].

With this command, you can download project file to controller.

6.12 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 48, with detailed description of functions as follows:

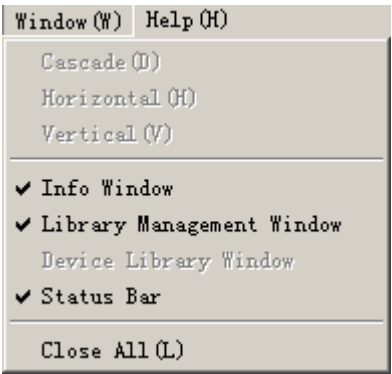


Figure 48 Window Menu

The operation orders and their functions in window menu are as shown in Table 6.

Table 6 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.

■ Common Windows

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

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

6.13 Help

[Help] menu provides [Content] and [Index] commands to open help window.

Click [About(A)...] to display the information window of the software, including software name, identification, version, copyright information.

Chapter 7 Hardware configuration

Hardware system is established under the [Hardware Configuration], meanwhile, module parameters and communication parameters are configured.

7.1 Controller module

LK220S CPU module is added under the [Hardware Configure] by default, and ETHERNET is added under the LK220S by default.

You can view the CPU device information, rename, and add the device through the relevant command in the right click menu.

7.2 Configure DP protocol

7.2.1 Add DP master device

You can add LK249S safety master station communication module via [Add Device] in right-click menu of LK220S.

Prompt the Add dialog as shown in Figure 49. Select LK249S to add.

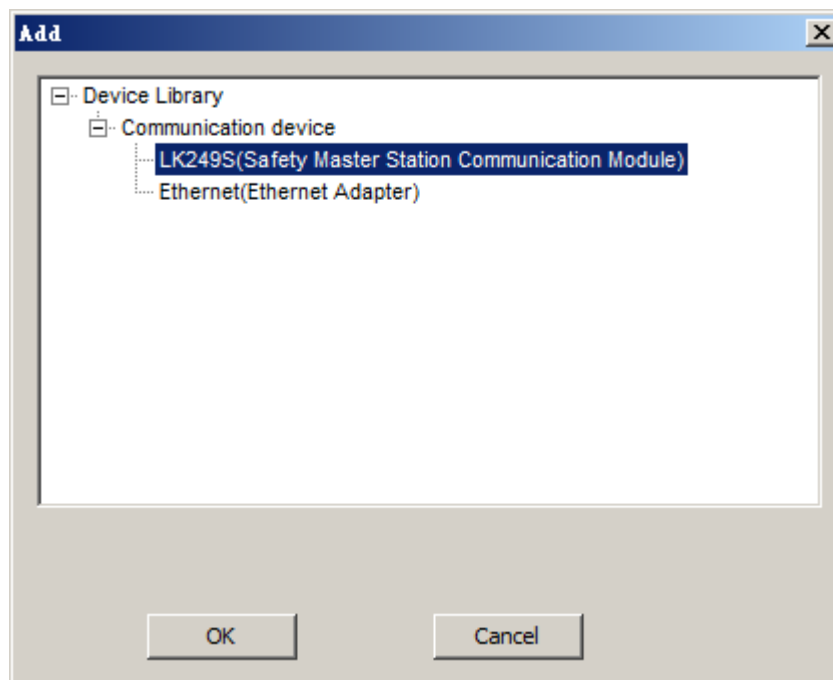


Figure 49 Add LK249S



- LK220S CPU module currently supports up to one safety DP master station communication module and one Ethernet adapter.

You can view device information of module, rename, add protocol and delete module through the right click menu on LK249S.

7.2.2 Add communication protocol

LK249S as communication master station of PROFI-safe/Profibus-DP, you need to add master station communication protocol. Select [Add protocol] to add.

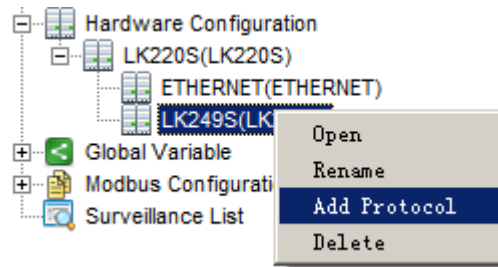


Figure 50 Right click menu on LK249S

The Add dialog is shown in Figure 51.

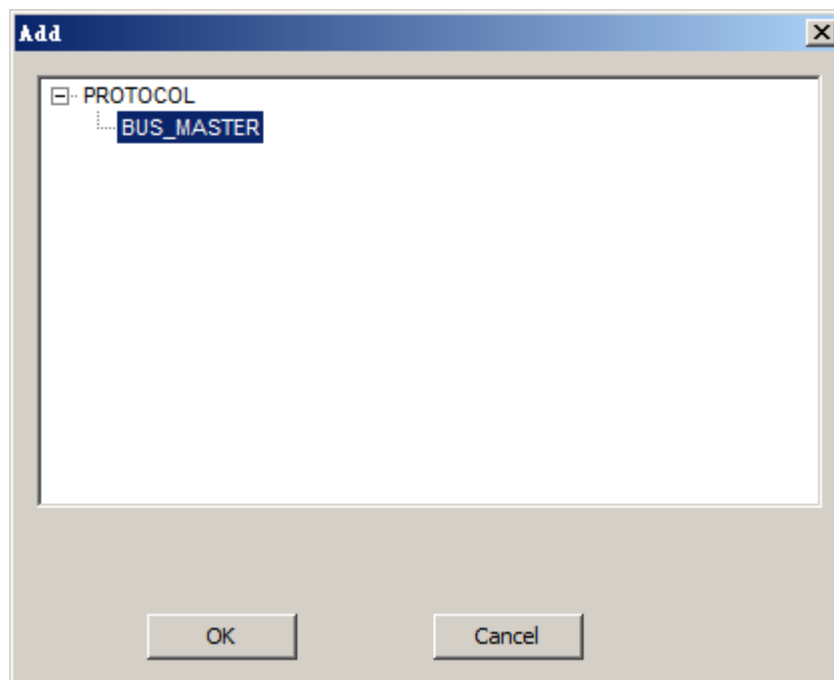


Figure 51 Add BUS_MASTER protocol



- Only support one BUS_MASTER protocol under LK249S module.

You can view device information, rename, add devices and delete devices through right-click menu on BUS_MASTER node.

Double-click BUS_MASTER to open the device information window.

BUS_MASTER(BUS_MASTER)	
Device Information	
Project	Content
Protocol name	BUS_MASTER
Port	COM
Baud rate	500.00
TSL(0~65535)	200
Min TSDR(0~65535)	11
Max TSDR(0~65535)	100
TQUI failure/Repeater switching time(0~255)	0
TSET(0~255)	1
TTR	3416
Gap	10
Retry limit(0~255)	1
Slave interval(0~65535)	50
Poll timeout	500
Data control time	600
DP polling cycle	0
Protocol	Profibus-DP/PROFIsafe
Address range of slave station	2~125
Position of master/slave station	Profibus-DP master station/PROFIsafe master station

Figure 52 BUS_MASTER device information window

Parameter information of communication master station is displayed in the device information window, Port and Baud rate can be set. Port is set as COM1. Choose different baud rate, other bus parameters switch to best value automatically.

7.2.3 Add PROFIsafe/Profibus-DP slave station

You can add PROFIsafe safety slave devices and Profibus-DP non-safety slave devices to safety DP master station device.

7.2.3.1 Import device description file

The LK249S support the module of other companies as the DP slave station. Before configuration, you need to import the required device description file, and the corresponding devices shall be generated in the device library for the user configuration.

Device description file corresponding to module is as system file, which should not be modified or deleted.

It is must ensure that format and content exported is correct when new device description file is imported.



- Menu bar: Click [Project]-[Import Device Description File].

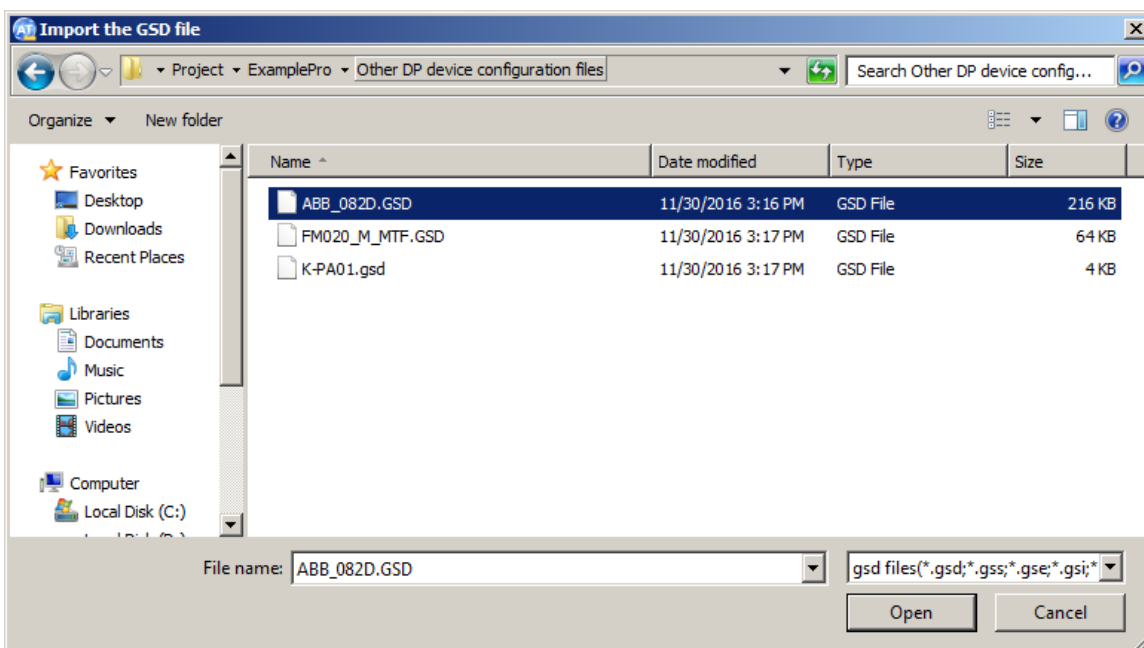


Figure 53 The Dialog of adding configuration file

Choose GSD file and click **Open** to import the device.

7.2.3.2 Add PROFIsafe/Profibus-DP device

Select [Add Device] command to add the slave station.

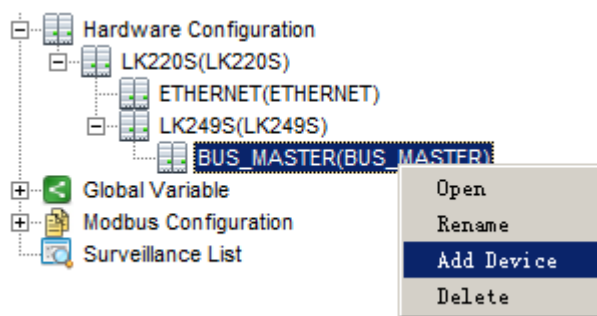


Figure 54 Add devices command

The Add dialog is shown in Figure 55. PROFIBUS-DP as a non-safety protocol and PROFISAFE as a safety protocol, slave modules are selected according to whether the system is a safety system. The default slave station address is displayed in the address box and can be modified.

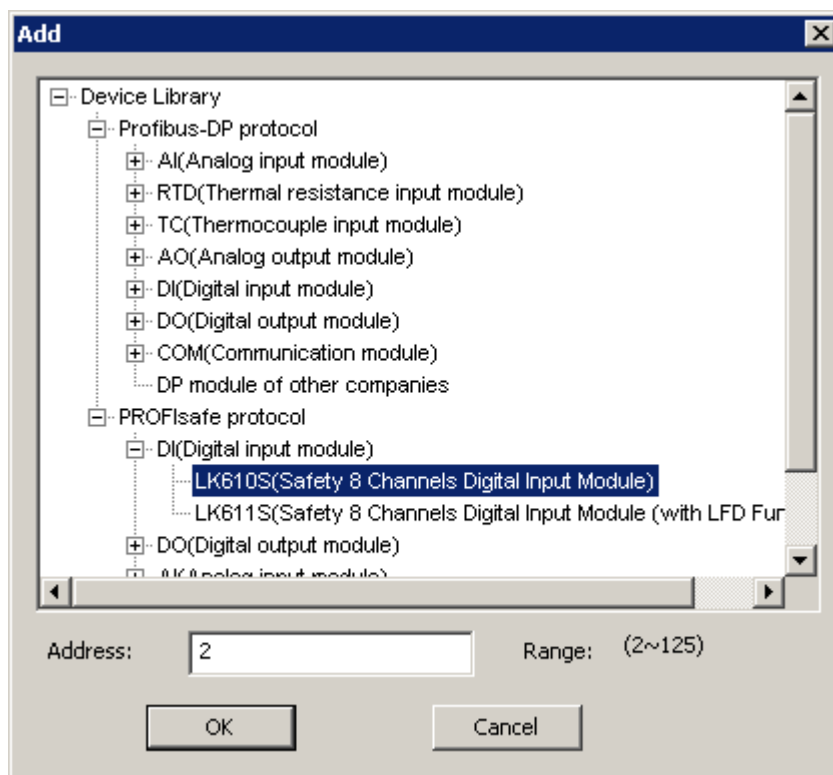


Figure 55 Add slave devices

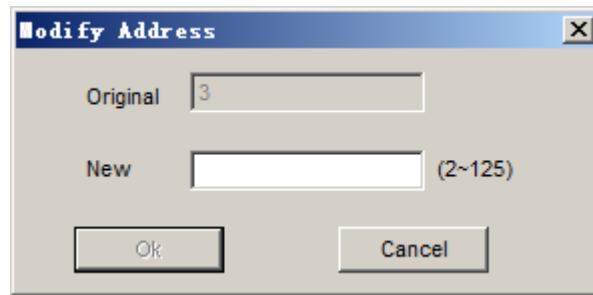
Name of slave module is composed: slave name (slave address: module name). Added serial number is displayed behind the slave name when the same module is added repeatedly. You can modify slave name through [Rename] in right-click menu.

7.2.4 Modify slave address

Double-click blue area in Device Address to pop up dialog of Modify Address.

LK610S(3:LK610S)	
Device Information	
Project	Content
Module Type	LK610S
GSD File Name	LK610S.gsd
Description	Safety 8 Channels Digital Input Module
Device Address	3(Double-click Configuration)
Redundancy	NO
Input Starting Address	IB16
Output Starting Address	Not configured
Device Property	Double-click Configuration
Number of Channels	8DI
Maximum power consumption	50mA@24VDC
Operating ambient temperature	0~60 degrees centigrade

(a)



(b)

Figure 56 *Modify address of slave station*

Enter the new address, range from 2 to 125. When the address you entered is illegal or reduplicative, The button **OK** is not available.

7.2.5 Configure slave station parameters

Double-click blue area in Device Property (as shown in Figure 56(a)) to open the Module Properties dialog, as shown in Figure 57. The dialog contains [Input / Output selection] tab and [User Parameter].

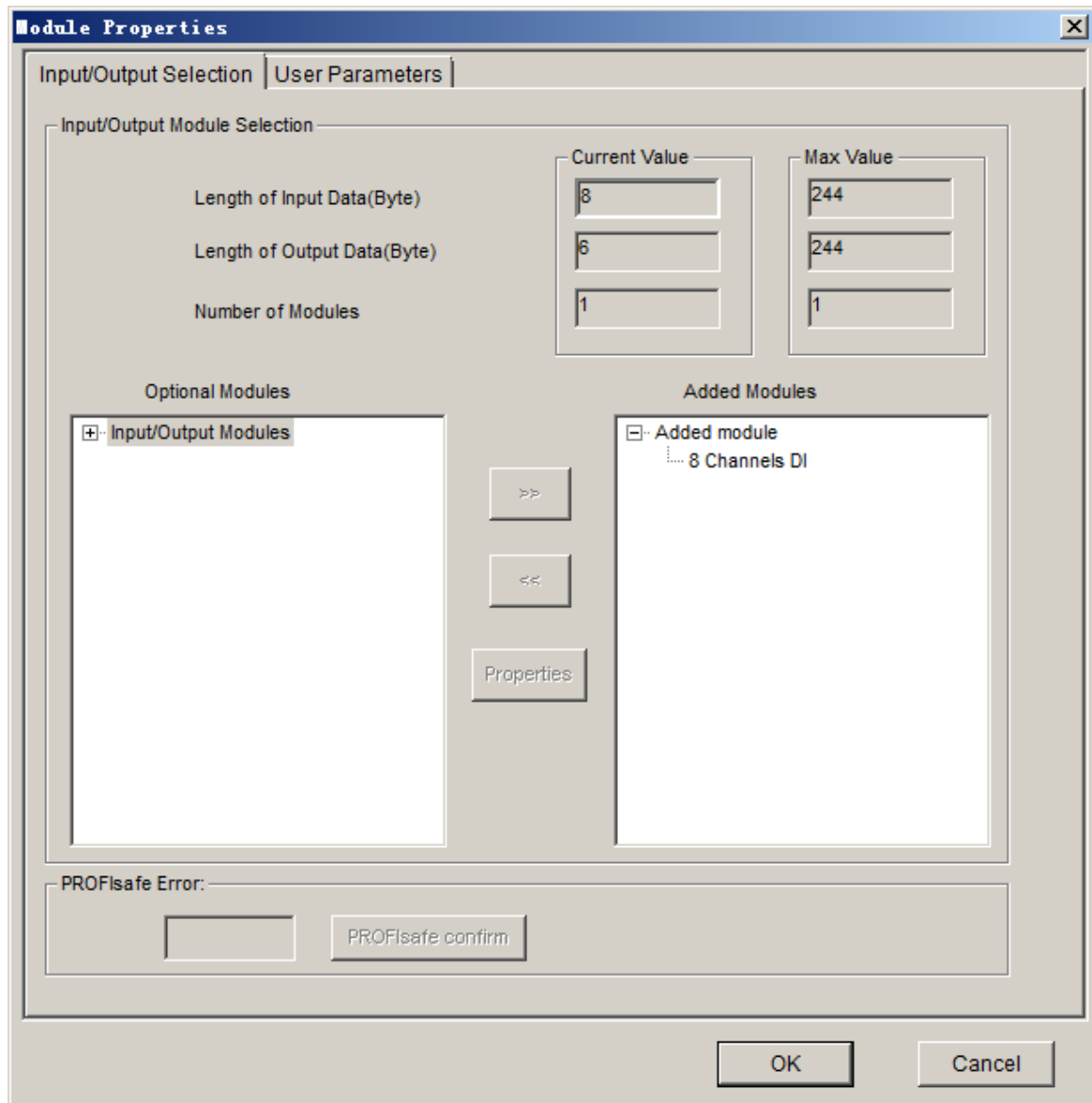
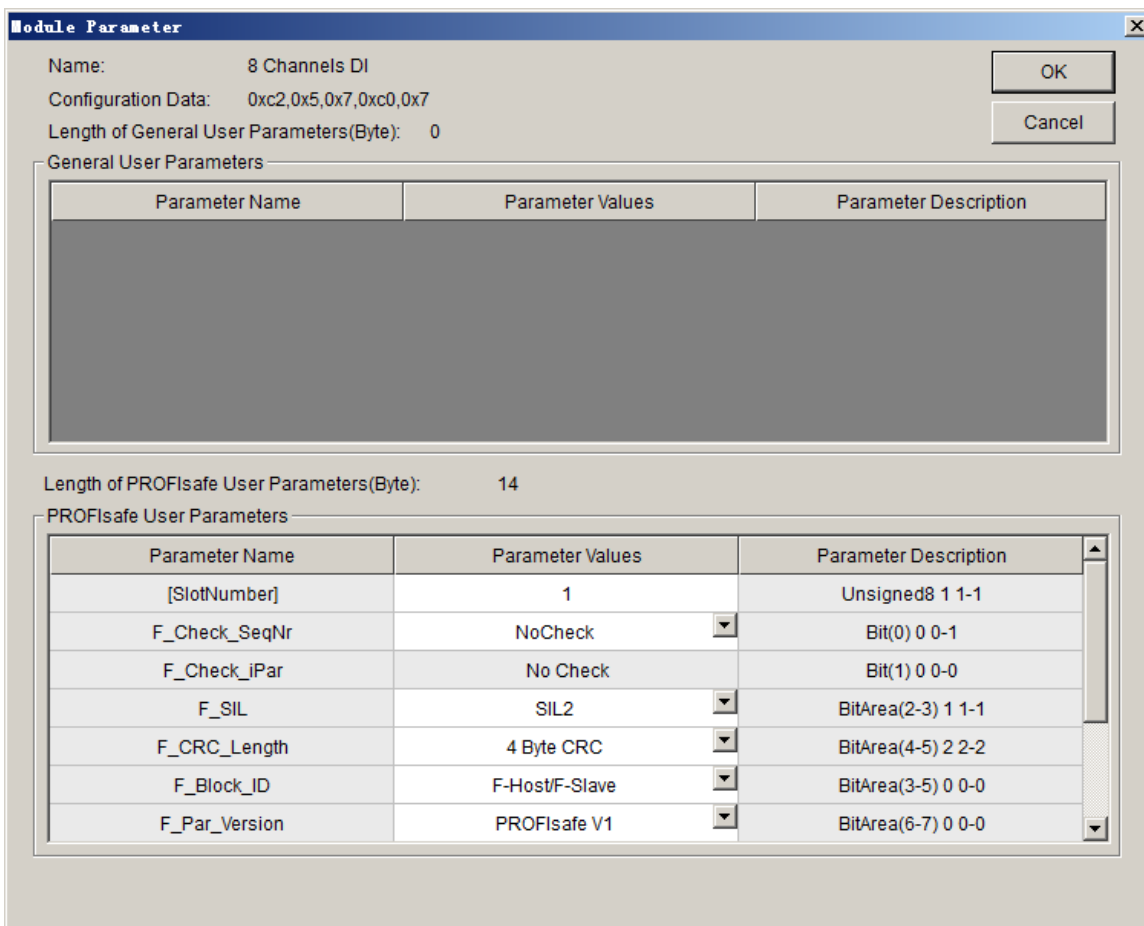


Figure 57 Input / Output selection tab

7.2.5.1 Module Parameter

You can view and configure module parameters through the **Properties** button.



Module Parameter

Name: 8 Channels DI

Configuration Data: 0xc2,0x5,0x7,0xc0,0x7

Length of General User Parameters(Byte): 0

OK

Cancel

General User Parameters

Parameter Name	Parameter Values	Parameter Description

Length of PROFIsafe User Parameters(Byte): 14

PROFIsafe User Parameters

Parameter Name	Parameter Values	Parameter Description
[SlotNumber]	1	Unsigned8 1 1-1
F_Check_SeqNr	NoCheck	Bit(0) 0 0-1
F_Check_iPar	No Check	Bit(1) 0 0-0
F_SIL	SIL2	BitArea(2-3) 1 1-1
F_CRC_Length	4 Byte CRC	BitArea(4-5) 2 2-2
F_Block_ID	F-Host/F-Slave	BitArea(3-5) 0 0-0
F_Par_Version	PROFIsafe V1	BitArea(6-7) 0 0-0

Figure 58 Module parameters

See Parameters of Chapter 7 IO Module in *LKS Safety Control System Product Manual* for details.

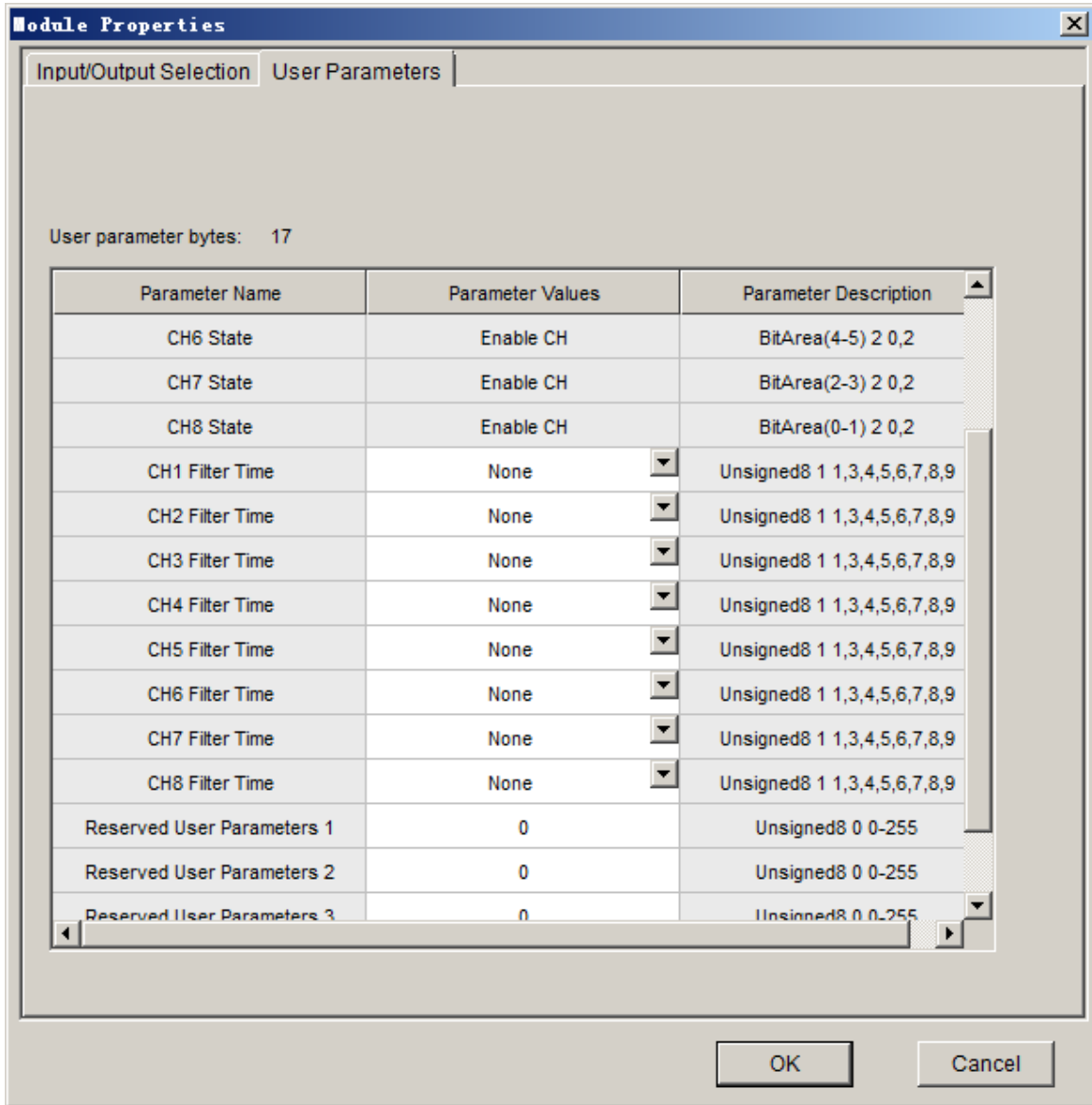
7.2.5.2 PROFIsafe fault status

Safety module LK610S, LK611S, LK710S, LK411S and LK630S can confirm PROFIsafe faults. See Chapter 7 in *LKS Safety Control System Product Manual* for details.

7.2.5.3 User parameters

User parameters are configured in follows window.

The user parameters that can be configured for each module are different. Here, LK610S module is taken as an example to illustrate. For user parameter configuration of other modules, please refer to the introduction of each module in the Device Library.



Module Properties

Input/Output Selection | User Parameters

User parameter bytes: 17

Parameter Name	Parameter Values	Parameter Description
CH6 State	Enable CH	BitArea(4-5) 2 0,2
CH7 State	Enable CH	BitArea(2-3) 2 0,2
CH8 State	Enable CH	BitArea(0-1) 2 0,2
CH1 Filter Time	None	Unsigned8 1 1,3,4,5,6,7,8,9
CH2 Filter Time	None	Unsigned8 1 1,3,4,5,6,7,8,9
CH3 Filter Time	None	Unsigned8 1 1,3,4,5,6,7,8,9
CH4 Filter Time	None	Unsigned8 1 1,3,4,5,6,7,8,9
CH5 Filter Time	None	Unsigned8 1 1,3,4,5,6,7,8,9
CH6 Filter Time	None	Unsigned8 1 1,3,4,5,6,7,8,9
CH7 Filter Time	None	Unsigned8 1 1,3,4,5,6,7,8,9
CH8 Filter Time	None	Unsigned8 1 1,3,4,5,6,7,8,9
Reserved User Parameters 1	0	Unsigned8 0 0-255
Reserved User Parameters 2	0	Unsigned8 0 0-255
Reserved User Parameters 3	0	Unsigned8 0 0-255

OK Cancel

Figure 59 User parameters

- CHn State: Channel enable status. The channel is enabled by checking Channel Enable.

8 Channels DI							
Channel Number	Channel Name	Channel Types	Channel Address	Channel Description	Channel Enable	Mode of Safety Value	Safety Value
1	DPIO_2_1_4_1	BOOL	%IX24.0		☒	Hold On	
2	DPIO_2_1_4_2	BOOL	%IX24.1		☒	Hold On	
3	DPIO_2_1_4_3	BOOL	%IX24.2		☒	Hold On	
4	DPIO_2_1_4_4	BOOL	%IX24.3		☒	Hold On	
5	DPIO_2_1_4_5	BOOL	%IX24.4		☒	Hold On	
6	DPIO_2_1_4_6	BOOL	%IX24.5		☒	Hold On	
7	DPIO_2_1_4_7	BOOL	%IX24.6		☒	Hold On	
8	DPIO_2_1_4_8	BOOL	%IX24.7		☒	Hold On	

Figure 60 Channel enable

- CHn Filter Time: Channel filter time. The filter time values of each module are different. Please refer to optional range of the actual configured module.
- Reserved User Parameters n: Reserved parameters.

7.3 Configure Modbus TCP protocol

7.3.1 Configure Modbus TCP master protocol

When the controller is as a Modbus TCP master station, you need to complete the following configuration.

7.3.1.1 Add the master station protocol

In the [ETHERNET] node's right-click menu, select [Add Protocol] command, dialog is shown in Figure 61. Select MODBUSTCP_MASTER protocol to add.

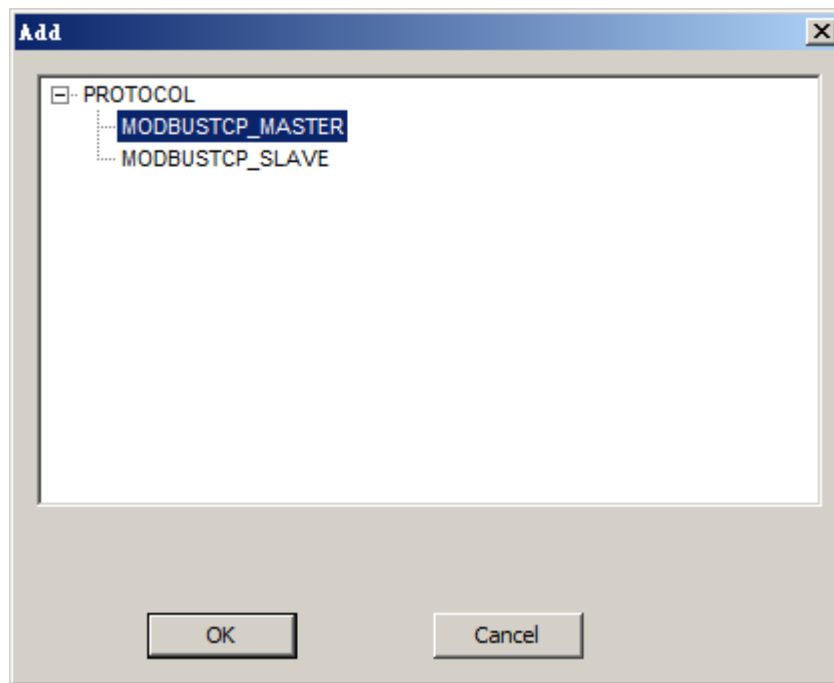


Figure 61 Add Modbus TCP Master Protocol



- Modbus TCP master is a non-safety protocol and is used only in safety project.

7.3.1.2 Configure Modbus TCP master station

Double click MODBUSTCP_MASTER node to open master station configuration window.

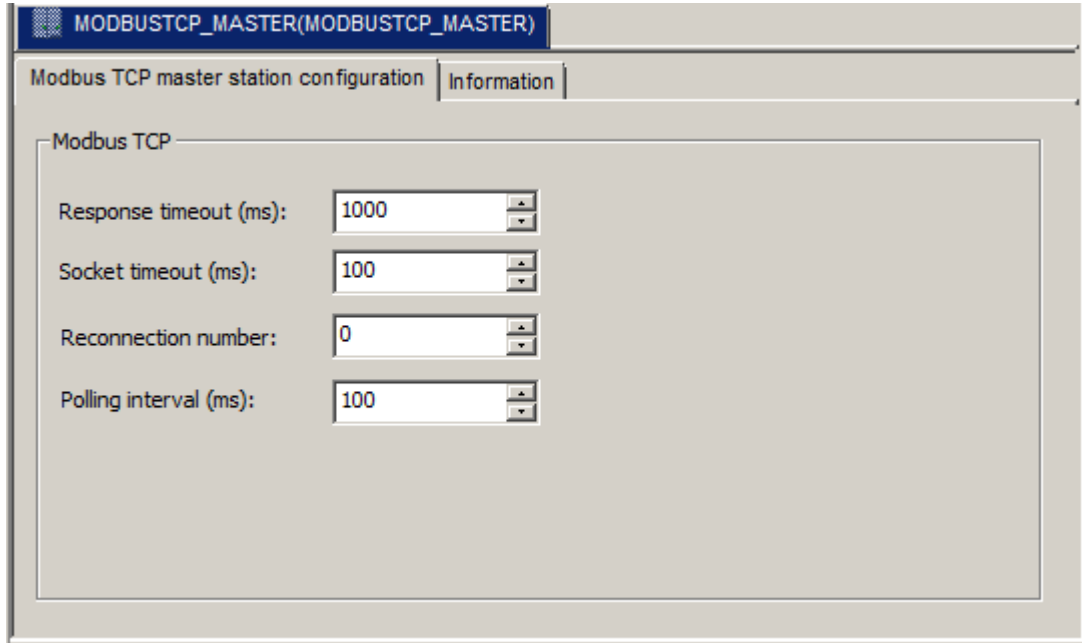


Figure 62 Configuration of Modbus TCP master station

Table 7 Communication configuration parameters of master station

Parameters	Value	Default Value	Description
Response timeout (ms)	10~2,147,483,000	1000	The allowed delay response time after master station sends a request frame
Socket timeout (ms)	10~2,147,483,000	100	TCP/IP connection Socket timeout
Reconnection number	0~10	0	The number of times that the master station re-sends the request after an abnormal response
Polling interval(ms)	100~2,147,483,000	100	The time interval from the moment when the Modbus TCP master station receives the response frame from the slave station to the moment when it sends the next request frame. If the response made by the slave station is timed out for the last frame, then the master station can ignore the time interval and send the request frame directly

To ensure the validity of the polling interval, you should note the following points in configuration:

- It is suggested that you need to use less instructions to read data from slave station, in other words, each read instruction to read more data.

- If slave station has been configured, please ensure that communication link is normal between the master and slave, and slave stations run fine. If the slave station does not exist, to delete the configuration of slave station.
- Theoretical calculation formula of polling interval in worst case: $80\text{ms} \times \text{the number of instruction configured in slave station}$ (Note: if all slaves run fine, the worst value is not reached).

7.3.1.3 Configure Modbus TCP slave station

As master station, controller can configure one or more slave stations for data communication. Select [Add Device] in the right click menu of the MODBUSTCP_MASTER node, and select MODBUSSLAVE_TCP in dialog to add.

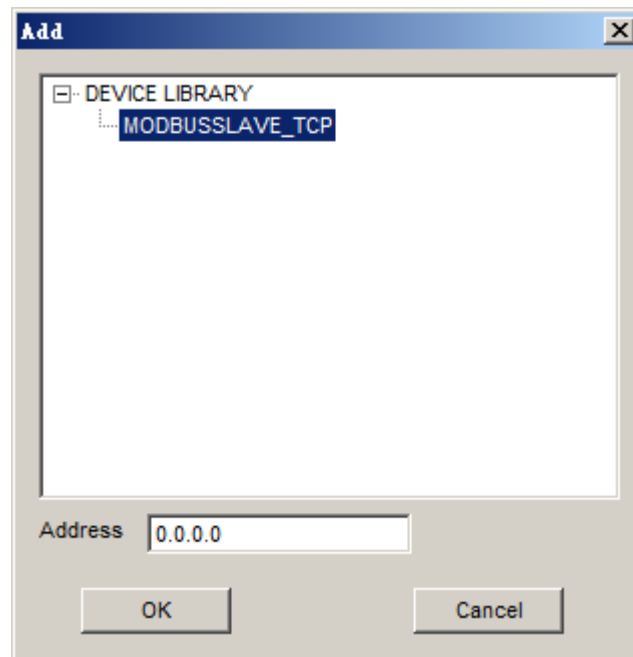


Figure 63 Add Modbus TCP slave station

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

- Modbus TCP slave station configuration
Double click MODBUSSLAVE_TCP node to open slave station configuration window.

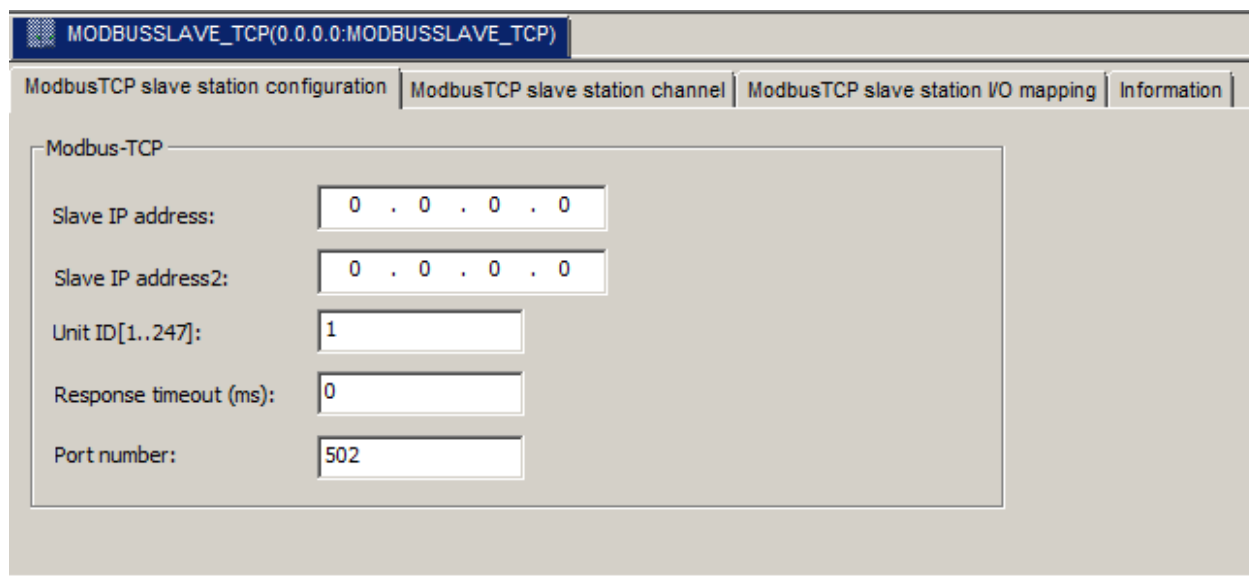


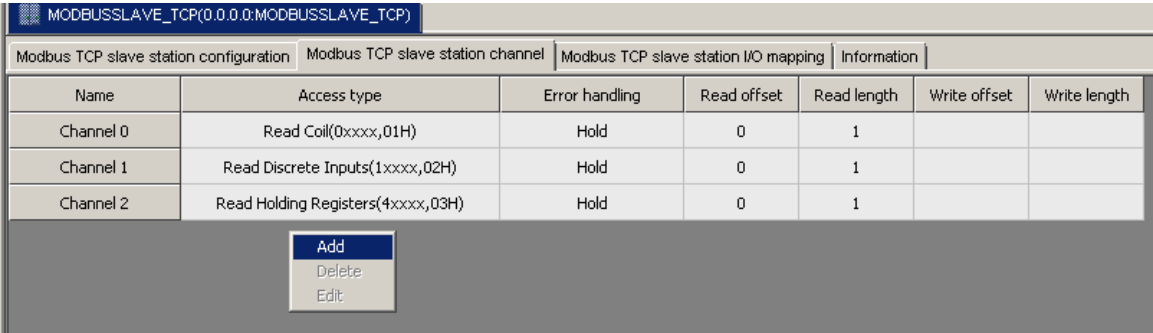
Figure 64 Modbus TCP slave station configuration window

Table 8 Parameters of slave station

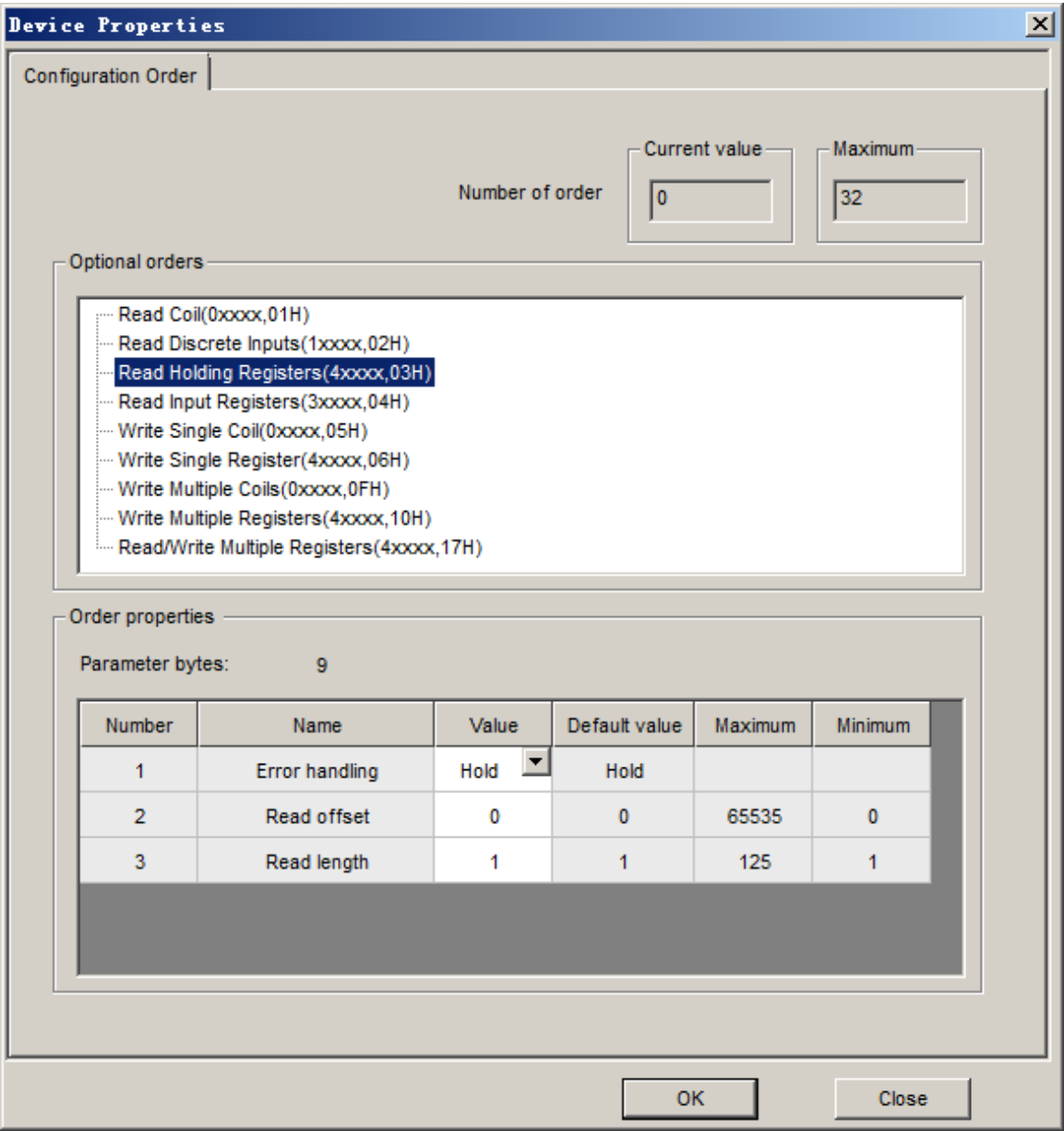
Parameters	Value	Default Value	Description
Slave IP address	Set according to the actual IP address of slave station	0.0.0.0	IP address of the slave station requested by master station The slave station can configure two IP addresses, one for connection and the other for standby. It should be noted that dual IP configuration is only supported in version V1.2.1 and above. At the same time, it needs to match with firmware version LK220S-A03 and above, so the function can be used normally.
Unit ID	1~247	1	Modbus TCP Protocol Unit ID
Response timeout (ms)	0~2,147,483,000	0	Default case: the default timeout time of slave is zero. In this case, the response timeout time of the master is adopted. The Response timeout of a slave station can be configured separately. If the parameter is greater than zero, The timeout time is based on the current slave station configuration
Port number	1~65,535	502	The Modbus TCP protocol port

■ Configure slave station orders

In the [Modbus TCP slave station channel] tab, you can add orders for the slave station through the [Add] command in the right-click menu, as shown in Figure 65.



(a)



(b)

Figure 65 Add slave station orders

Choose order in the **Optional orders** box, order parameters will be displayed and you can double-click the parameter value item to set.

You can add up to 32 orders.

Table 9 Order parameters

Parameters	Value	Default Value	Description
Error handling	Hold, Clear	Hold	Hold: The current data is held after abnormal response Clear: The current data is cleared after abnormal response
Read /Write offset	0~65,535	0	Corresponding offset value of start address on the slave
Read /Write length	1~2000	1	The length value corresponds to the number of corresponding channels of the slave station

■ Slave station I/O mapping

After configuring the orders, the corresponding I/O channels are mapped in the [Modbus TCP slave station I/O mapping] tab.

- ☐ Modbus start address of each instruction = 00001 + read/write offset address.
- ☐ Channel name: The initial value is TCPIO_ device number _ protocol number _ four-bit slave station IP address _ channel number. Channel name supports edit and you can access Modbus variables via channel name.
- ☐ Channel addresses: Channel addresses are automatically allocated by the system and cannot be modified.

MODBUS_SLAVE_TCP(0.0.0.0:MODBUS_SLAVE_TCP)					
Modbus TCP slave station configuration		Modbus TCP slave station channel		Modbus TCP slave station I/O mapping	
Channel Number	Modbus Address	Channel Name	Channel Types	Channel Address	Channel Description
Channel 0					
1	000001	TCPIO_1_1_0_0_0_0_1	BOOL	%IX0.0	Read Coil(0xxxx,01H)
2	000002	TCPIO_1_1_0_0_0_0_2_1	BOOL	%IX0.1	Read Coil(0xxxx,01H)
Channel 1					
3	100001	TCPIO_1_1_0_0_0_0_3_1	BOOL	%IX4.0	Read Discrete Inputs(1xxxx,02H)
4	100002	TCPIO_1_1_0_0_0_0_4	BOOL	%IX4.1	Read Discrete Inputs(1xxxx,02H)
5	100003	TCPIO_1_1_0_0_0_0_5	BOOL	%IX4.2	Read Discrete Inputs(1xxxx,02H)
Channel 2					
6	400001	TCPIO_1_1_0_0_0_0_6	WORD	%IW8	Read Holding Registers(4xxxx,03H)
7	400002	TCPIO_1_1_0_0_0_0_7	WORD	%IW10	Read Holding Registers(4xxxx,03H)

Figure 66 Add slave station orders

7.3.1.4 Modbus instruction diagnosis

When the Modbus master station instruction is configured, after compilation, the diagnosis variables of instructions are generated in the variable group ModbusOrderDiagVar, as shown in Figure 5.5-68. The parameter of variable group does not support editing. The user can find specific instructions according to the diagnostic information which is displayed in **Variable Description** column.

ModbusOrderDiagVar					
No.	Name	Description	Data Type	Initial Value	Area
0001	SYS_OrderDiag_0	No.1 communication device, No.1 protocol, Address is 0...	WORD	0	S Area
0002	SYS_OrderDiag_2	No.1 communication device, No.1 protocol, Address is 0...	WORD	0	S Area
0003	SYS_OrderDiag_4	No.1 communication device, No.1 protocol, Address is 0...	WORD	0	S Area

Figure 67 Modbus Variable Diagnostic Message

In online status, you can through code value of instruction state to diagnose the instruction state.

Table 10 Code Value of Master Instruction State for Modbus TCP

Code value of instruction state	meaning
0	No error
2	Time out
4	Function code is error
16	Unite ID not match(the address of slave station is error)
32	The TCP connection fails
33	The request Message sent fails
34	The confirmation message received fails
64	Data sent is not match with data received
128+1	The function code which is not supported by slave station
128+2	Data address overflow
128+3	Data range overflow
128+4	Slave device has fault
128+6	Slave device is busy
128+15	Other fault of slave station

7.3.2 Configure Modbus TCP slave protocol

LK220S only as a Modbus TCP slave station in safe project.

7.3.2.1 Add slave station protocol

Right click the ETHERNET node to select [Add Protocol] command, and protocol dialog is shown in Figure 68. Select protocol ModbusTCP_SLAVE to add.

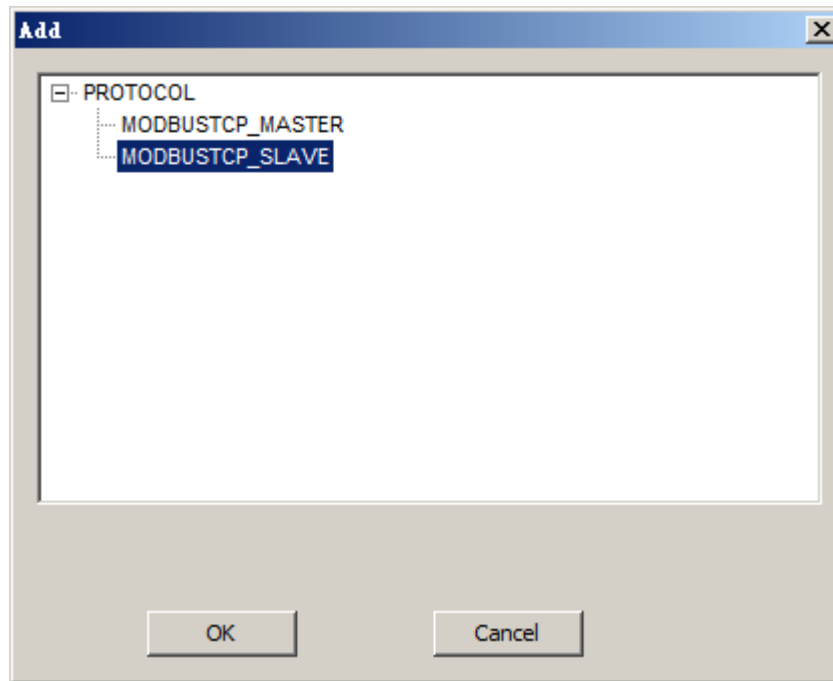


Figure 68 Add slave station protocol

7.3.2.2 Configure slave station parameters

Double click the protocol node to open the slave parameter configuration window.

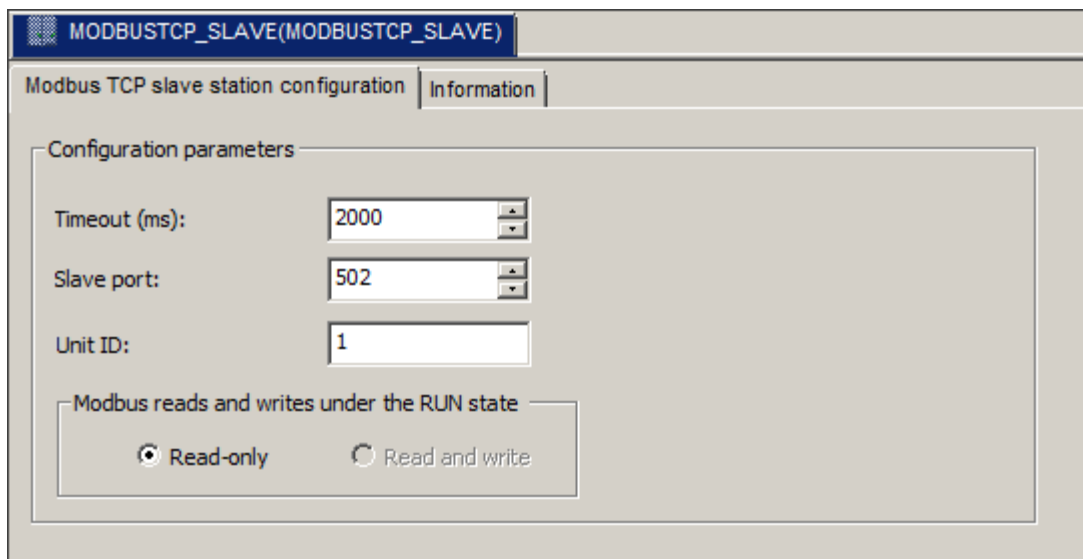


Figure 69 Slave station configuration

Table 11 Configuration parameters

Parameters	Value	Default Value	Description
Timeout (ms)	500~2,147,483,000	2000	The time interval in which the slave station has not received the data sent by the master station. If the set time is exceeded, the slave station

Parameters	Value	Default Value	Description
			disconnects the communication link to the master station.
Slave station port	1~65,535	502	Port number of Modbus TCP protocol
Unit ID	1~247	1	Unit ID of Modbus TCP protocol

- Set Modbus read and write properties in RUN mode
 - ☐ Read-only: As a default settings in safety system.
 - ☐ Read and write: As non-safety settings, master station can read from slave station and write to slave station.

7.3.2.3 Modbus variable configuration

All variables for Modbus communication need to be defined in [Modbus Configuration]. There are four kinds of variable groups: Coil, Digital Input Signal, Input Registers and Holding Registers, as shown in Figure 70, the variable types in each of the above variable group are different.

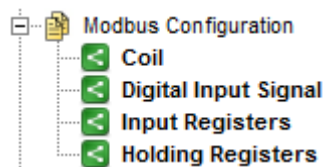


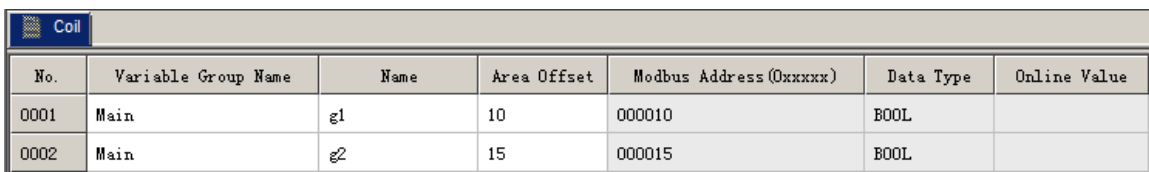
Figure 70 Modbus variable groups

Add variables into the corresponding groups under [Modbus Configuration], the limitations of address and scope are shown in Table 12.

Table 12 Modbus variables definition

Storage Area	Data Type	Access Permission	Area Offset	Modbus Address
Coil	BOOL	Read/Write	1~65535	000001~065535
Digital Input Signal	BOOL	Read-only	1~65535	100001~165535
Input Registers	WORD\DINT\DWORD\REAL	Read-only	1~65535	300001~365535
Holding Registers	WORD\DINT\DWORD\REAL	Read/Write	1~65535	400001~465535

Double-click sub-node below [Modbus Configuration] to open variable group window.



No.	Variable Group Name	Name	Area Offset	Modbus Address (0xxxxx)	Data Type	Online Value
0001	Main	g1	10	000010	BOOL	
0002	Main	g2	15	000015	BOOL	

Figure 71 Modbus variable group window

Modbus variables are configured in four ways:

(1) Manually add

You can add variables through [Add] command in the right-click menu of variable group window.

(2) Add by command

Variables that are defined in PRG type POU, global variable group and channel are add to the corresponding Modbus variable group by [Send To] command in right click menu.

(3) Copy Variables to Modbus variable group

Copy variables from PRG or global variable group to Modbus variable group.

Variable properties in the Modbus variable group are as follows:

- Variable Group Name: Variable group name sent is shown below.
POU local variables: POU(PRG) name where the variable is located.
Variable in global variable group: Variable group name under the [Global Variable] node.
Channel variables: Shown Physical.
- Name:
The variables sent show original variable name.
BOOL/WORD pins of function block: Function block name.pin name.
- Area Offset: Define area offset in variable group according to the slave address and register type set by master station. Master station access Modbus data of slave station through Modbus address and read/write data.
Setting instructions:
 - ☐ Offset address length of DWORD, DINT, REAL type variables is 2 words, set the offset address to $n + 2$, n is an address that has been occupied. For example, REAL variables $g1$ and $g2$, set offset address of $g1$ to 1, then address of $g2$ is set to 3.
 - ☐ The offset address of DWORD, DINT, REAL type variables cannot be set to 65535.
- Modbus Address: Modbus address is composed of register type and area offset.
The register is divided into the following four types:
 - ☐ 0: Coil
 - ☐ 1: Digital input signal
 - ☐ 3: Input register
 - ☐ 4: Holding register

(4) Import or export Modbus variables

When you first import the Modbus variable, you need to export the template of Modbus variable from the software.

- Export



- Menu bar: click [Project] - [Export Modbus Variable].



Figure 72 Template excel

- Import

You fill in excel with Modbus variables, and import excel to project.

Variable Group Name, Name, Area Index are required to fill. Software will first empty original Modbus variables and import new variables.



- Menu bar: click [Project] - [Import Modbus Variable].

Import results will be displayed in the [Information window].

7.4 System diagnosis

Online, you can view master-slave controller status and system diagnosis information through the variable group SysDiagVar under the [Global Variable].

SysDiagVar					
No.	Name	Description	Data Type	Initial Value	Area
0001	sys_LocalMSSState	Local master/slave state	WORD	0	S Area
0002	sys_LocalRSSState	Local single/hot standby state	WORD	0	S Area
0003	sys_LocalABState	Local A/B state	WORD	0	S Area
0004	sys_LocalRedLink	Local RED module operation network state	WORD	0	S Area
0005	sys_LocalTaskState	Local task running state	WORD	0	S Area
0006	sys_LocalKeyState	Local key switch state	WORD	0	S Area
0007	sys_RemoteMSSState	Remote master/slave state	WORD	0	S Area
0008	sys_RemoteRSSState	Remote single/hot standby state	WORD	0	S Area
0009	sys_RemoteABState	Remote A/B state	WORD	0	S Area
0010	sys_RemoteRedLink	Remote RED module operation network state	WORD	0	S Area
0011	sys_RemoteTaskState	Remote task running state	WORD	0	S Area
0012	sys_RemoteKeyState	Remote key switch state	WORD	0	S Area
0013	sys_TaskFirstRun	First run flag of the task after the download	WORD	0	S Area
0014	sys_ModuleState	Local module exist status	WORD	0	S Area
0015	sys_TaskCycleAbnormal	Task cycle does not match the current project	WORD	0	S Area
0016	sys_CPUModuleFatalErr1	CPU module fatal error 1	WORD	0	S Area
0017	sys_CPUModuleFatalErr2	CPU module fatal error 2	WORD	0	S Area
0018	sys_CPUModuleMinorErr1	CPU module general error 1	WORD	0	S Area
0019	sys_CPUModuleMinorErr2	CPU module general error 2	WORD	0	S Area
0020	sys_REDModuleFatalErr1	RED module fatal error 1	WORD	0	S Area
0021	sys_REDModuleFatalErr2	RED module fatal error 2	WORD	0	S Area
0022	sys_REDModuleMinorErr1	RED module general error 1	WORD	0	S Area
0023	sys_REDModuleMinorErr2	RED module general error 2	WORD	0	S Area
0024	sys_DPModuleFatalErr1	DP module fatal error 1	WORD	0	S Area
0025	sys_DPModuleFatalErr2	DP module fatal error 2	WORD	0	S Area
0026	sys_DPModuleMinorErr1	DP module general error 1	WORD	0	S Area
0027	sys_DPModuleMinorErr2	DP module general error 2	WORD	0	S Area
0028	sys_ReservedDiag	System reserved diagnostic information	ARRAY[0..31] OF WORD		S Area

Figure 73 SysDiagVar diagnosis variable group

Table 13 Diagnosis information

Variable	Diagnosis value	Description
sys_LocalMSSState	Local master/slave state	0: Unknown status 1: Initial status 2: Hardware ready status 3: Hot ready status 4: Single ready status

Variable	Diagnosis value	Description
		5: Fault status 6: Error status 7: Engineering redundancy 8: Engineering verification 9: Slave 10: Master 11: Unqualified host 12: Wait to enter the fault status 13: Wait to enter the wrong status
sys_LocalRSSState	Local single/hot standby state	0: Unknown status 1: Single 2: Dual
sys_LocalABState	Local A/B state	0: Unknown status 1: A 2: B
sys_LocalRedLink	Local RED module operation network state	0: Unknown status 1: First optical fiber 2: Second optical fiber
sys_LocalTaskState	Local task running state	The status of each task is represented by 2 bits 0: Unknown status 1: Running 2: Stopped 3: Hang up
sys_LocalKeyState	Local key switch state	0: UNKNOWN 1: RUN 2: REMOTE 3: PRG
sys_RemoteMSSState	Remote master/slave state	Same value with local
sys_RemoteRSSState	Remote single/hot standby state	Same value with local
sys_RemoteABState	Remote A/B state	Same value with local
sys_RemoteRedLink	Remote RED module operation network state	Same value with local
sys_RemoteTaskState	Remote task running state	Same value with local
sys_RemoteKeyState	Remote key switch state	Same value with local
sys_TaskFirstRun	First run flag of the task after the download	The status of each task is represented by 1 bit 0: Non first run 1: First run
sys_ModuleState	Local module exist status	(Online state of communication module) bit0~bit3: Module ID in slot2 bit4~bit7: Module ID in slot3 bit8~bit11: Module ID in slot4, reserved bit12~bit15: Module ID in slot5, reserved Module ID=0: No module is in slot Module ID=7: LK240S is in online Module ID=8: LK249S is in online
sys_TaskCycleAbnormal	Task cycle does not match the current project	0: The task cycle setting match with the current project 1: The task cycle setting does not match the current configuration project

Variable	Diagnosis value	Description
sys_CPUModuleFatalErr1	CPU module fatal error1	Bit2=1: Double Ethernet fault Bit7=1: Redundancy timeout fault Bit8=1: Redundancy state failure (for 240S is offline under redundant configuration)
sys_CPUModuleFatalErr2	CPU module fatal error2	Reserved
sys_CPUModuleMinorErr1	CPU module general error1	Bit0=1: History run state fault Bit6=1: Anomal power supply of RTC Bit7=1: Task cycle timeout fault
sys_CPUModuleMinorErr2	CPU module general error2	Reserved
sys_REDModuleFatalErr1	RED module fatal error1	Bit0=1: Self checking fault
sys_REDModuleFatalErr2	RED module fatal error2	Reserved
sys_REDModuleMinorErr1	RED module general error1	Bit0=1: First optical fiber fault Bit1=1: Second optical fiber fault Bit2=1: Double optical fiber fault bit3=1: AB switch conflict Bit5=1: Optical fiber communication fault
sys_REDModuleMinorErr2	RED module general error2	Reserved
sys_DPModuleFatalErr1	DP module fatal error1	Bit0=1: Master controller conflict fault Bit4=1: DP1, DP2 network fault
sys_DPModuleFatalErr2	DP module fatal error2	Reserved
sys_DPModuleMinorErr1	DP module general error1	Bit3=1: DP communication fault Bit4=1: DP1 network fault Bit5=1: DP2 network fault
sys_DPModuleMinorErr2	DP module general error2	Reserved
sys_ReservedDiag	System reserved diagnostic information	Reserved

7.5 Hardware list

Table 14 The correspondence between hardware version and software version

Model	Module name	Software version
LK130-B	4 slot local backplane module	V1.0.0
LK117-B	11 slot extended backplane module	
LK118-B	5 slot extended backplane module	
LK220S-A01	Safety main control module	
LK240S-A01	Safety redundant communication module	
LK249S-A01	Safety DP master station communication module	
LK232S-A01	Safety Profibus-DP bus repeater module	
LK610S-A01	Safety 8 channels digital iInput module	
LK710S-A01	Safety 8 channels digital output module	

Model	Module name	Software version
LKA104-B	PROFIBUS-DP Bus connector	
LK411S-A01	Safety 8 channels current type analog input module	V1.0.1
LK220S-A02	Safety main control module	
LK249S-A02	Safety DP master station communication module	
LK610S-A02	Safety 8 channels digital input module	
LK710S-A02	Safety 8 channels digital output module	
LK630S-A01	Safety 8 channels digital input (with SOE function) module	V1.2.1
LK220S-B01	Safety main control module	V1.2.2
LK249S-A03	Safety DP master station communication module	
LK610S-A04	Safety 8 channels digital input module	
LK710S-A05	Safety 8 channels digital output module	
LK630S-A02	Safety 8 channels digital input (with SOE function) module	
LK611S-A01	Safety 8 channels digital input module (with LFD function)	V1.3.1

Chapter 8 Program organization unit

The POU (Program Organization Unit) is important, only basic programming unit in Safety FA-AutoThink software. POU's can be divided into function, function block and program.

Program is the main logic to perform the Safety FA-AutoThink software, is a collection in order to complete a task and the preparation of the statement sequence or a set of instructions. It is divided into two types: the main program and the subroutine.

General definition of general global variables in the program, mapping hardware addresses global variables. Implement the practical logic by calling the program.

Function block: Function block is the basic algorithm unit of object oriented property, and the function block has internal state.

One or more values are generated when the function block is executed. You can create a function block multiple named instances; each instance should have a related example name and contains the output and internal variables of a data structure, and with examples relating to the input variables values or variable input reference. Internal variables necessary output variables and the data structure of all values should be from a functional block one execution of the keep to the next execution; therefore, input variables of the same function blocks do not always produce the same output value. Such as flow accumulation.

Function is a basic algorithm unit that has at least one input variable, no internal state, and a return value.

There is no instance name in the use of Function, whose output value is only related to the input value of the unified cycle and its own logic. For example ADD.

This chapter mainly introduces the type of POU, how to create the POU and the call relationship between POU's.

8.1 Add POU



- Project Management: right-click [User Program], click [Add POU].

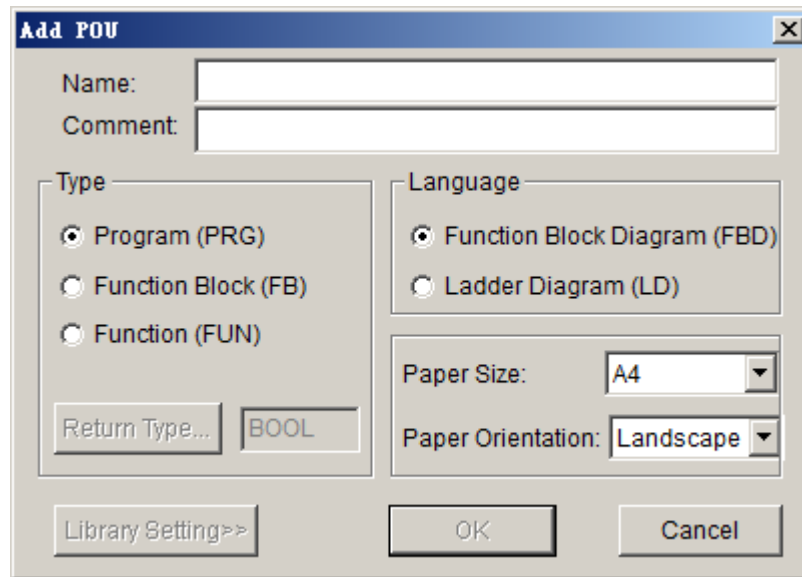


Figure 74 Add POU

The window is structurally classified into three parts from top to bottom: name, type and language.

8.1.1 Common

Introduce the common parts of three types in this dialog.

8.1.1.1 Name

Name: POU name, required.

POU name only can include letter, digit, underline, and cannot be started from AT_ , SYS_ , and the length cannot exceed 32 characters, the exceeding part cannot input.

POU name cannot be the same as the variable name, the variable group name, data type (customized or system defaulted), key words, library name or function block name.

POU name cannot be the same as the reserved device name(CON、 PRN、 AUX、 NUL、 COM0~9、 LPT0~9) in Windows system.

If the name meets the above rules, the **OK** is available, and vice versa.

Comment: POU's note, which no limit of special illegal character; the maximum number of limited character is 512.

8.1.1.2 Type

POU type: PRG, FB and FUN.

8.1.1.3 Language

The programming languages are FBD (function block diagram) and LD (ladder diagram).

8.1.1.4 Paper size

It is used for determining the editable paper size. The default is A4; in addition there are optional types A0, A3, A4, B5 and Ax (its size at 4 times the size of A0).

After selecting, the system automatically adjusts to the standard size.

Paper Orientation: Select the orientation of the drawings, the default is Landscape orientation.

8.1.2 Difference

Add function, and you need to set the data type of return value via return type.

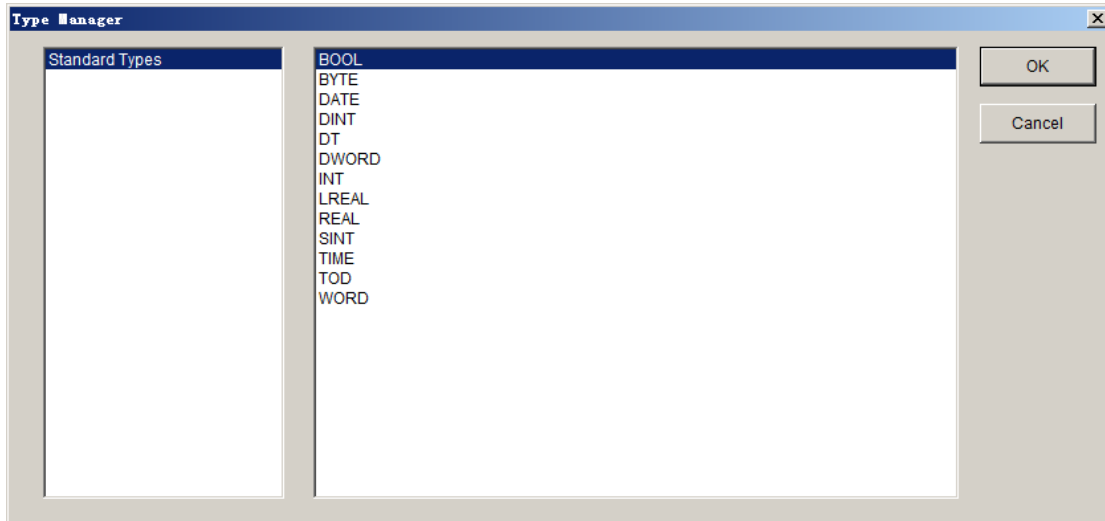


Figure 75 Type manager

8.2 Edit POU



- Project Management: Right-click POU under [program] node, click [Edit].
- Project Management: Double-click POU under [Program] node.

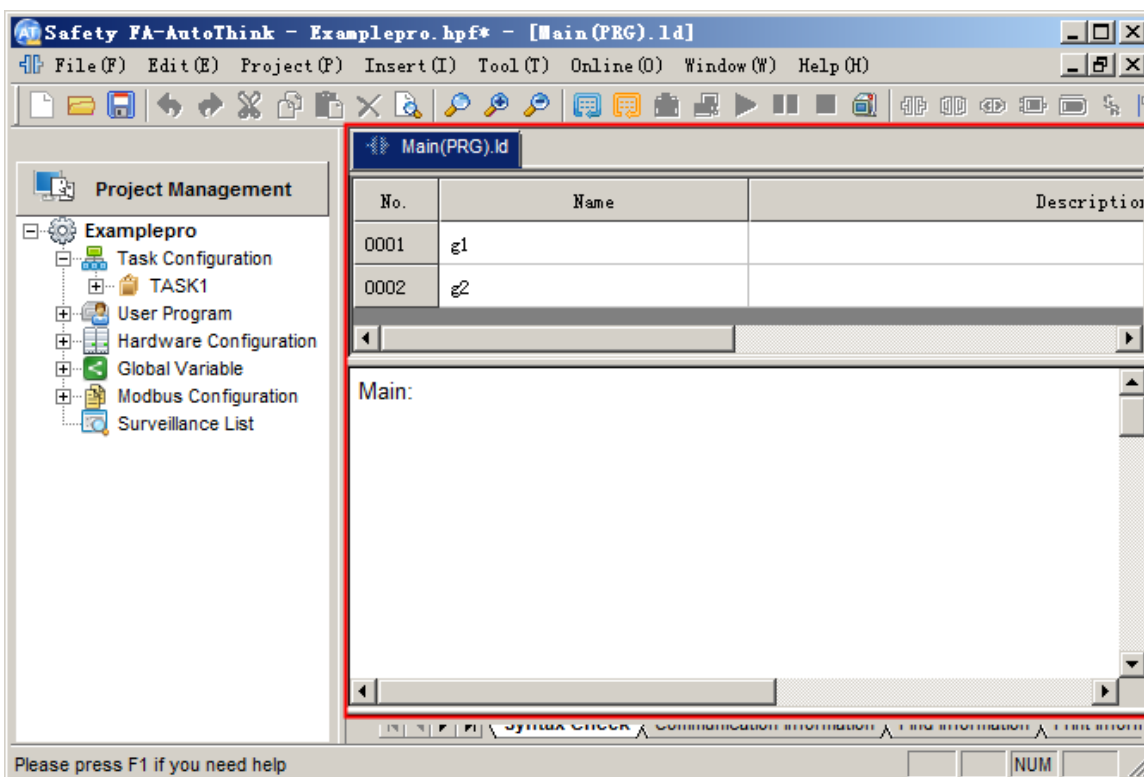


Figure 76 POU edit window

Each POU consists of a declaration part and a body, the former is a table editor, the latter is written in one of the IEC programming languages which include FBD or LD.

The screen divider between the declaration part and the implementation, you can move up or down the screen divider with the mouse pointer.

8.2.1 Variable declaration

Safety FA-AutoThink software offers a variety of basic variable types, including BOOL, WORD, BYTE, INT, SINT, TIME, DATE, ARRAY, DINT, DT, DWORD, LREAL, REAL, TOD and so on, these data types can meet the demand of industrial control industry.

In POU, Variables can be divided into 5 types according to the function: VAR_POU, VAR_INPUT, VAR_OUTPUT, VAR_IN_OUT, VAR_GLOBAL.

Variable types and functions supported by POU are shown in table.

Table 15 Variable types and functions supported by POU

Variable Type	Means	Read-Write		Application		
		External	Internal	Program	Function Block	Function
VAR_POU	Local Variable, it is used only in POU internal and cannot be accessed externally	-	R&W	Supported	Supported	Supported
VAR_INPUT	Input Variable, it can be called as	R&W	R	Unsupported	Supported	Supported

Variable Type	Means	Read-Write		Application		
	external parameter for POU					
VAR_OUTPUT	Output Variable, it is usually as POU's return value.	R	R&W	Unsupported	Supported	Unsupporte d
VAR_IN_OUT	Input / Output Variable, it has both input variable and output variable properties	R&W	R&W	Unsupported	Supported	Unsupporte d
VAR_GLOBAL	Global Variable	R&W	R&W	Supported	Supported	Unsupporte d

8.2.2 Body

Safety FA-AutoThink software supports two kinds of powerful programming language, which LD and FBD are graphical programming interface, the project is extremely easy to use, has clear logical thinking.

8.3 POU property

With the help of the POU Property window on the one hand you can learn POU primary information (such as the use environment and role) and on the other hand you can easy change the part of the options value.

Here are ways to open the properties window of current POU.



- Project Management: Right-click the sub node under [User Program], click [Property].

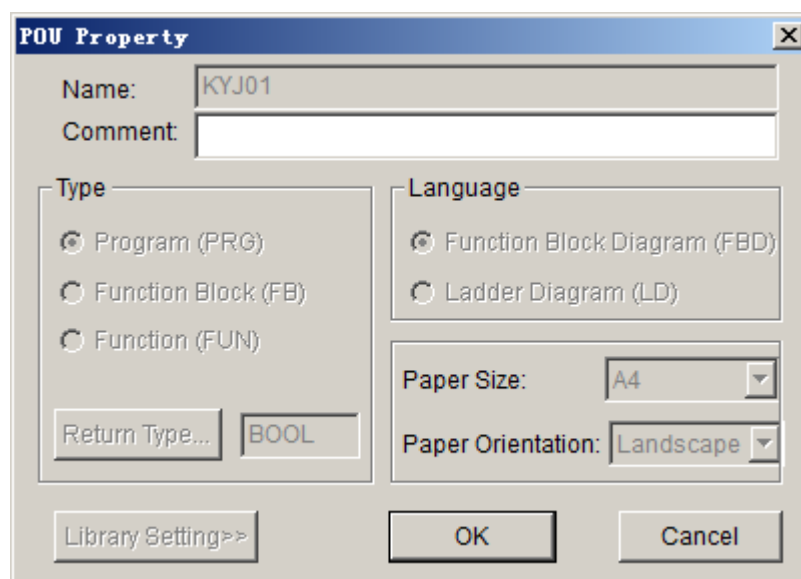


Figure 77 POU property dialog

This dialog is the same as the content structure of the **Add POU** dialog, there is a little difference among some items specific to the operating authorization. For introduction of related parameter, please see [8.1 Add POU](#).

8.4 Rename POU



- Project Management: Right-click the sub node under [User Program], click [Rename].

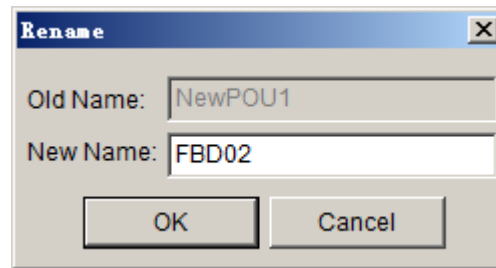


Figure 78 Rename dialog

Enter new POU name and click **OK** button to finish the modifying. See [8.1 Add POU](#) for naming rules.

8.5 Copy and paste

The paste result is **NewPOU***, "*" is presented by natural number (exclude 0); when the copied POU name is existed with duplicated POU, "*" value is accumulated and increased progressively. The operation description is shown as follows:

Right-click [Copy POU] on the selected POU node to copy, as shown in Figure 79.

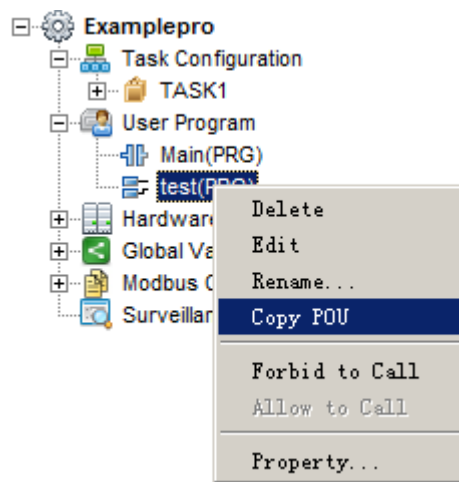


Figure 79 Copy POU

Then right-click **Paste POU** on **User Program** node.

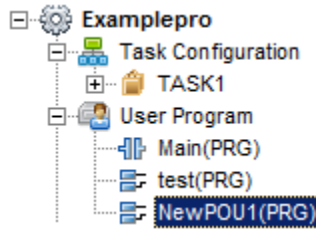


Figure 80 Paste

8.6 Delete POU



- Menu bar: click[Edit] - [Delete(D)];
- Toolbar: ;
- Shortcut Key: **Delete**;
- Project Management: right-click the POU node under [User Program] node, click [Delete].

POU cannot be recovered once it is deleted, please be careful.

8.7 Zoom POU



- Tool bar: 

Online and offline, the display proportion of the contents in the POU edit window can be adjusted.

The zoom proportion that you set will affect the current all POU display, and the proportion does not automatically restore to 100% until the project is closed.

The display ratio of the hardware configuration window will change with the proportion of POU display.

POU Scale will affect the display ratio of hardware configuration window; the hardware configuration window will be in accordance with the proportion of the final operation of the zoom display.

8.8 Print POU

8.8.1 Print preview

It is used to preview the result of the current content in the work area.

You can only preview the POU.



- Menu bar: click [File] - [Print Preview].

Preview result will differ depending on print mode.

The print mode of POU consists of Multi-page and Single-page. The default mode is Single-page; you can set via [Project] - [Options] - [Configuration Language].

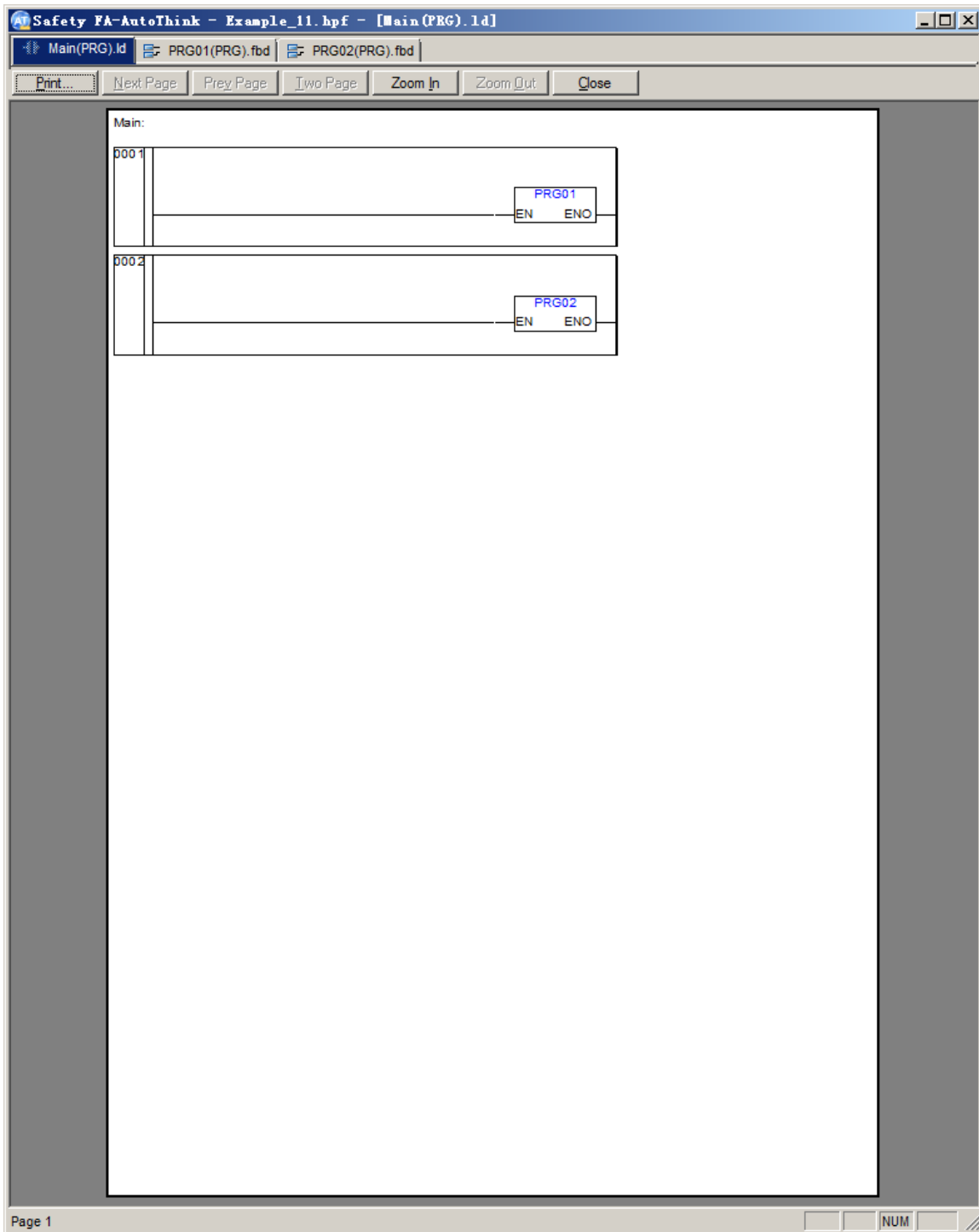


Figure 81 the preview result

Print: to set the print conditions.

PreV Page and Next Page: they are used for turning page of preview result. These buttons are only offered for Multi-page Print mode; otherwise they appear grayed out.

Two Page or One Page: it used to show the preview result on one/two columns. When the result has only one page, this button is displayed in gray.

Zoom in and Zoom out: used to adjust the display scale.

Close: it is used to exit this function.

8.8.2 Print



- Menu bar: click [File] - [Print].
- Shortcut Key: **Ctrl+P**.

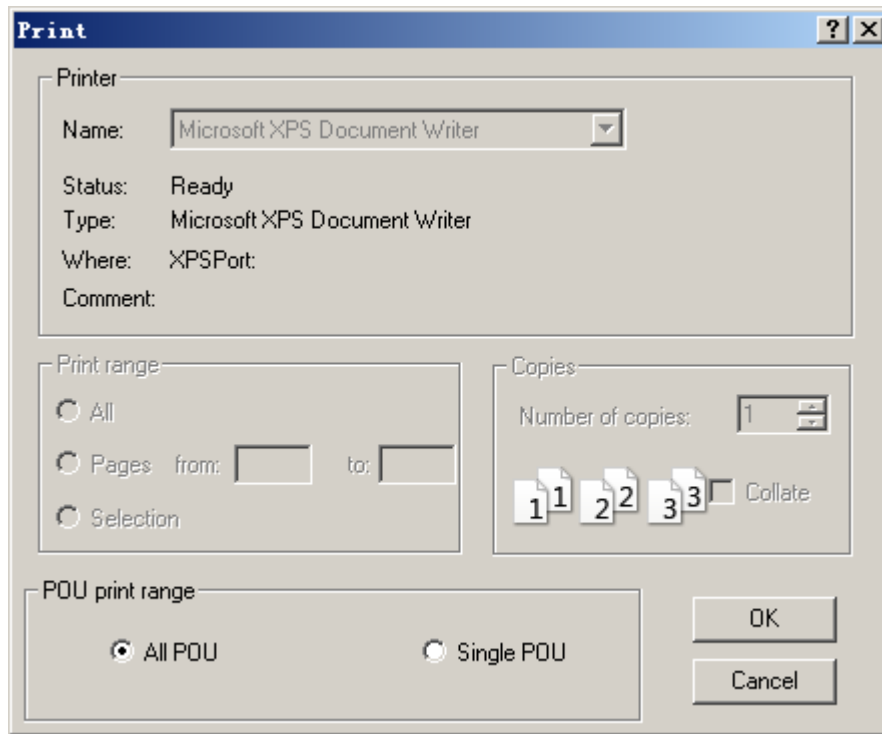


Figure 82 Print dialog

8.9 Find



- Menu bar: click [Edit] - [Find(F)];
- Toolbar: ;
- Shortcut Key: **Ctrl+F**.

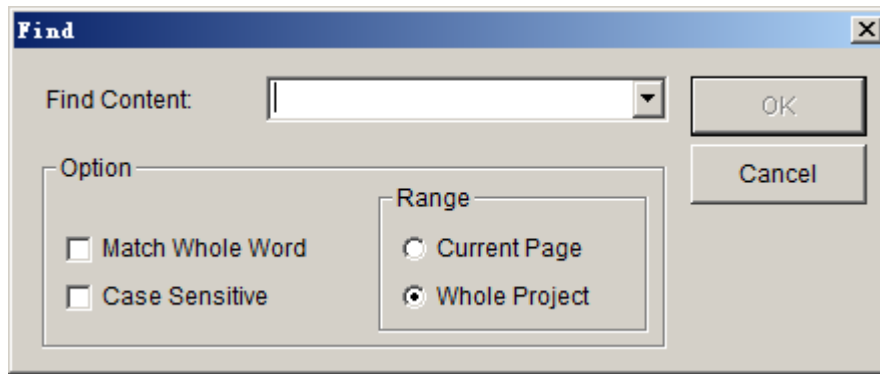


Figure 83 Find dialog

Search for a certain text in current editor window by this command.

Open the **Find** dialog to input the character string to find in Find field. Users can set searching options, including whether match whole word only, whether case sensitive, search the whole project or current page.

System launches the operation of search starting from current page after clicking **OK** button. Search results are displayed in the output window [Search Information] in the bar.

8.10 Replace



- Menu bar: click[Edit] - [Replace];
- Shortcut Key: **Ctrl+H**.

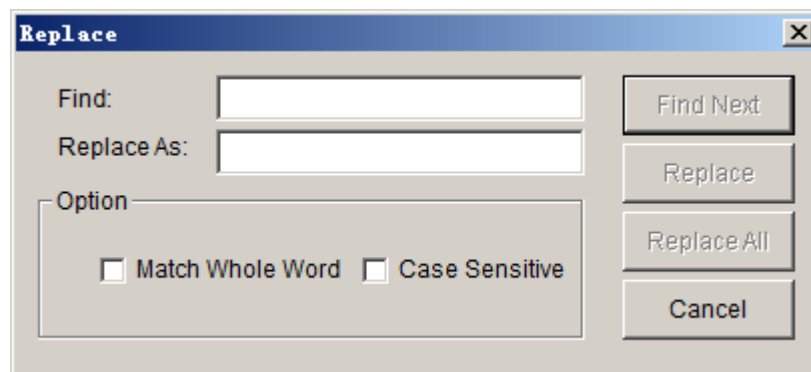


Figure 84 Replace dialog

Using this command, you can quickly replace one with another, such as the variable name in the algorithm.

The using method of this dialog is the same as the **Find** dialog, see [8.9 Find](#).

8.11 Call POU

Each POU does not operation until it was called, the POU calling rules as shown in Figure 85.

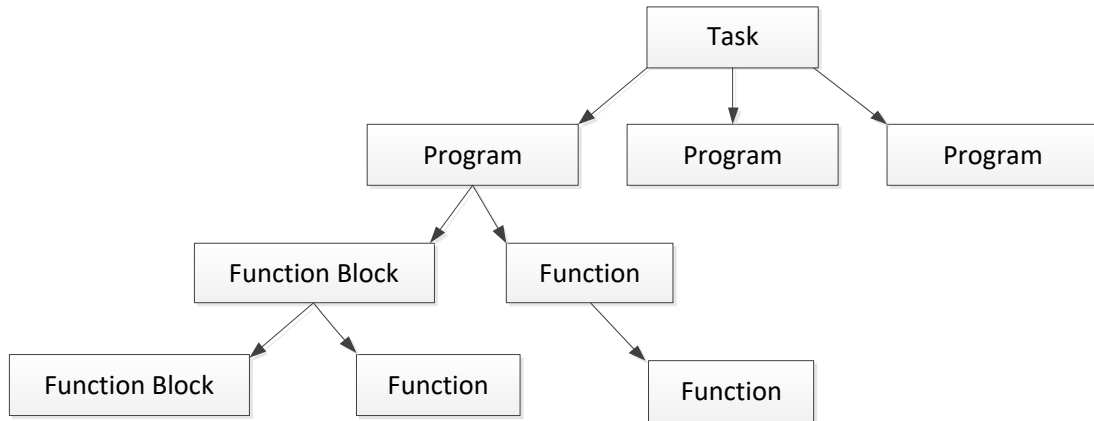


Figure 85 POU calling relationship graph

You can use [View the Calling Tree] understand the relationships between task and other POUs.

8.11.1 Program calls Function Block

8.11.1.1 Add block

In a LD or FBD POU, select block element at the insertion point. After adding the end, you need to make an instance declaration on the block element (in the default text ??? input instance name).

8.11.1.2 Call declared Function Block (FB)

In a project, a program refers to a functional block that has been declared in another program in a form of **program name.instance name**.

8.11.2 Program calls Function

The method of function call is the same as the method of user program calling the function block. Just the function does not need to define the instance name.



- When a user program calls a function, you must assign the return value of the function to the other variable; otherwise, the compilation error.

8.11.3 Call between program

Here the program focuses on the user program.

Program call contains Task call and program call.

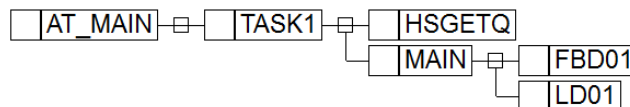


Figure 86 Call relationship between task and program

The user program must be called by the task to be executed. See Chapter [6.2.1 Program calling](#).

How to achieve the user program calls between each other:

Select node, and drag the program from the [User Program] node to the call location. Take the Main calls LD01 program for example, it is as shown in Figure 87.

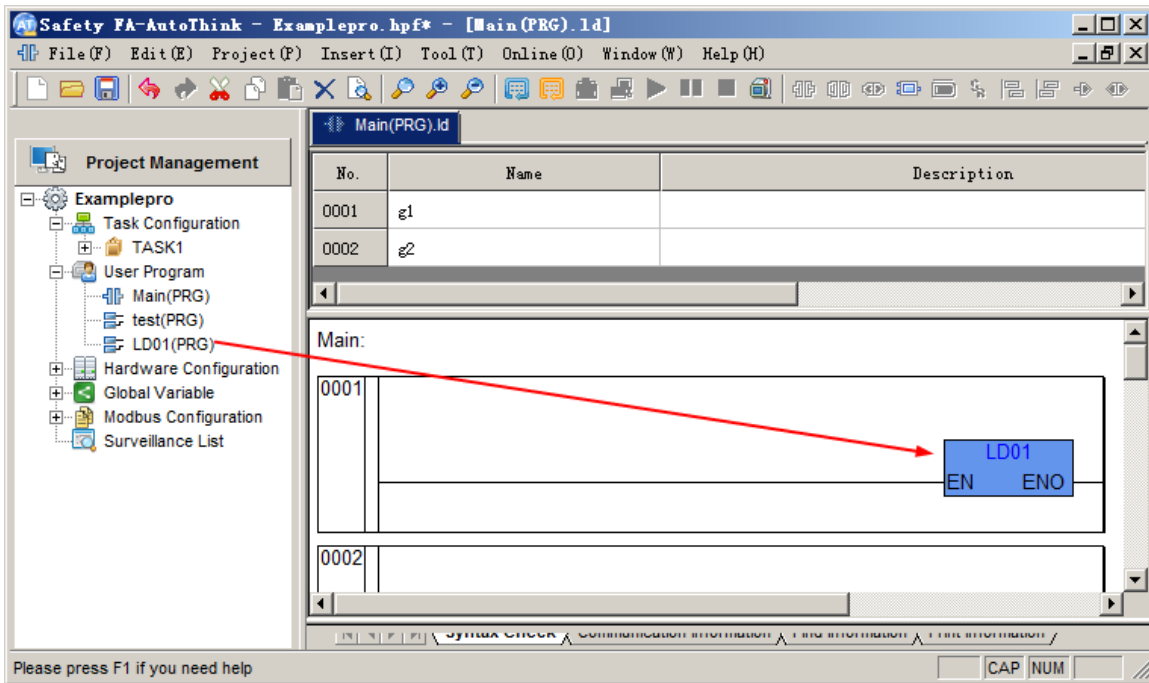


Figure 87 Calling by drag

8.11.4 Call between Function Blocks

The method of calling between function blocks is the same as [8.11.1 Program calls Function Block](#), but display results are different.

When you drag the FB type block to the POU(FB), the software will automatically assigns the instance name (PRG name + FB type name + _ serial number), and displays the variable declaration box, it is as shown in Figure 88.

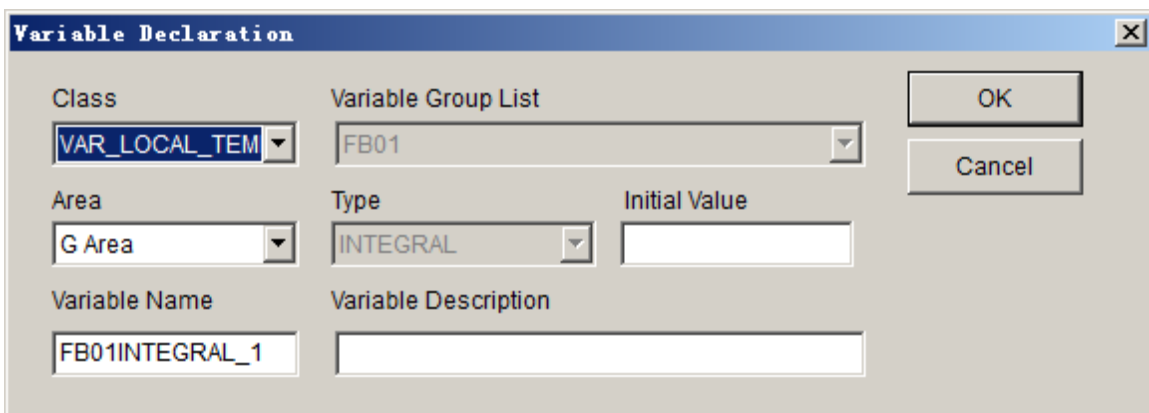


Figure 88 Variable declaration

Class: Set the variable property, including global variable and local variable, please see [9.1 Variable type](#) for each type meaning.

The use of other items the same as the same name items in [9.3 Variable declaration](#).

8.11.5 View the calling tree

Using a tree diagram describes the current project of program calls to each other, you can expand or close. Only after the project compiles successfully, you can view the calling tree.



- Menu bar: click [Project] – [Viewing the calling Tree].

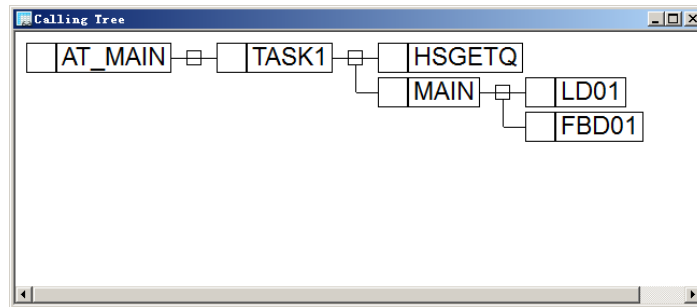


Figure 89 Calling tree

8.12 Folder

8.12.1 Add

You can be classified according to the configuration content of the POU, such as the characteristics of the equipment easy to Management.



- Project Management: Right-click [User Program], click [Add Folder];

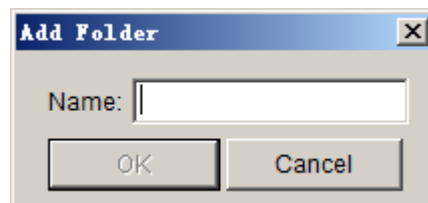


Figure 90 Folder name

In the edit box, enter the name of the folder, the name of the folder:

- 1 cannot contain symbols (in English) \, /, :, *, ?, "", <, >, |;
- 2 cannot be named as user program;
- 3 can be the same as the name of the POU;
- 4 name length maximum of 32 characters.

Click OK button to close this dialog box. Folder icon is displayed as .

When you enter an invalid name, **OK** is not operational.

8.12.2 Delete



- Project Management: Right-click [User Program], click [Delete].

The folder under [User Program] node to be deleted will pop up when you are prompted to confirm the delete operation; Note after you have deleted and unrecoverable, please attention during operation.

8.12.3 Rename

Rename the folder.



- Project Management: Right-click [User Program], click [Rename].

Chapter 9 Variable

The operation object of controlling and computing is data. The changeable data is named as variable stored in RAM of CPU module. In order to conveniently use and mark these variables in the process of data computing, each of them has a name, variable name. Only define the variables in advance, these variable names and types can be used in control algorithm configuration. This process is known as variable declaration.

Every variable need to declare as using, this section focuses on the basic variable declaration formats, as well as the specific methods are declared in different types of variables.

9.1 Variable type

9.1.1 Classify according to structure

9.1.1.1 Simple variable

Simple variable refers to a single change; it is given a clear value, a simple variable only represents one meaning. As BOOL, REAL type variables.

9.1.1.2 Function Block Instance

Function block instance contains a group of specific variables depending on the function block type of the case. So the function block instance closes to the concept of data structure.

9.1.2 Classify according to the value range of variable

Global Variable (VAR_GLOBAL): it is a wide use variable. A global variable can be used in more than one place (like the graph, algorithm) at the same time in a project. A same name is defined several times is not allowed in a project.

Local Variable: relative to the global variable, it has a limited use range which is the variable belongs to the function block (or function), and only can be used while you create function block (or function). A same name is defined several times is allowed in a project.

The value range of local variable in use can be sorted: input variable, output variable, Input/output variable and local variable.

(1)Input Variable: it is entered from the external, and only readable for the function block (or function) it belongs to.

(2)Output Variable: it is produced to the external, readable and writable for the function block it belongs to and readable for the other.

(3)Input/output Variable: it is a produce of combined the input variable with the output variable, and readable and writable for a function block.

(4)Local Variable: it is a temporary variable you need to achieve a function block (or function); it doesn't connect with the external, and is only valid for the function block it belongs to. It is known as parameter (VAR_TEMP) in function block information which is opened by view command.

9.2 Variable naming rules

Variable names only consist of letters, underscore, or digit, but if the variables start with digit, it should be combined with letter or underline to use; variable name cannot be all digits, if all for digit system will default to constant type.

Variable name can identify underline. For example, AB_CD and ABC_D are different ones.

Variable name is not case sensitive. For example, VAR1, Var1 and var1 represent the same variable

Variable name cannot be invalid and not contain blank space. For example, AB CD is wrong.

Variable name does not contain special characters, such as – and +, etc. For example, AB-CD and AD+CD are wrong.

Variable name cannot same as type name (include custom type), POU name, enumeration name, task name or type conversion function name.

Variable name cannot begin with sys_ and AT_.

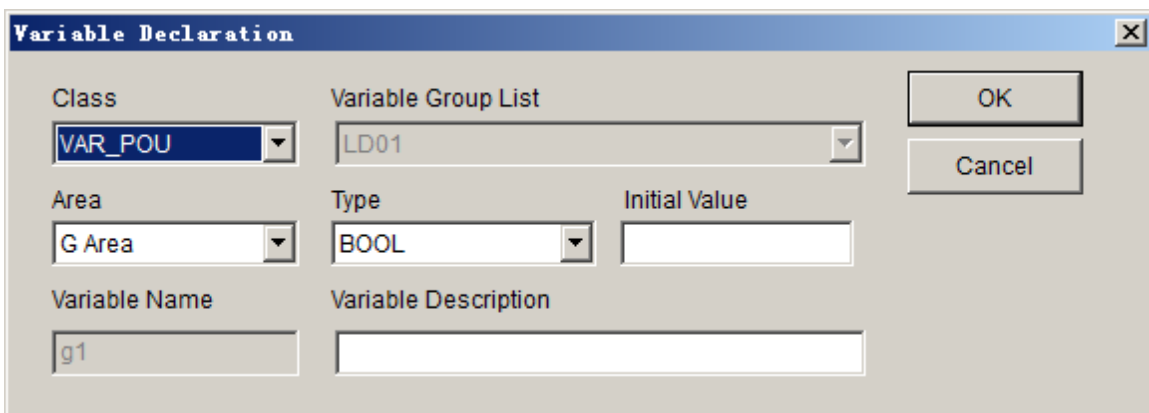
Variable name cannot be the same as keyword.

Variable name in the project guarantees uniqueness.

9.3 Variable declaration

The variables are declared automatically when you check **Automatic Variable Declaration** in [projects] - [option] – [configuration] dialog.

When you create variables in the POU, enter a valid variable name, press **Enter** key, it will pop up **Variable Declaration** dialog. It is as shown in Figure 91.



The dialog box titled "Variable Declaration" contains the following fields and controls:

- Class:** A dropdown menu with "VAR_POU" selected.
- Variable Group List:** A dropdown menu with "LD01" selected.
- Area:** A dropdown menu with "G Area" selected.
- Type:** A dropdown menu with "BOOL" selected.
- Initial Value:** An empty text input field.
- Variable Name:** A text input field containing "g1".
- Variable Description:** An empty text input field.
- Buttons:** "OK" and "Cancel" buttons are located on the right side.

Figure 91 Variable declaration

Class: defines a variable availability scope. Please see [9.1.2 Classify according to the value range of variable](#).

You can add variable classes related to POU's type.

A POU can add the maximum number of variables, please see Table 16.

Table 16 The maximum number of variables supported by one POU

Scope	POU(PRG)	POU(FB)	POU(FUN)
-------	----------	---------	----------

Scope	POU(PRG)	POU(FB)	POU(FUN)
VAR_GLOBAL	unlimited	/	/
Input_Variable	/	64	28
Output_Variable	/	64	/
Input/Output_Variable	/	32	/
Local_Variable	/	128	128

Area: Please refer to Chapter 6.3 Set project safety property.

Variable name: The name of the new variable.

Variable Group List: POU (PRG), it is used to show the POU name

Type: The data type of this variable is set and defaulted as BOOL; you can choose by the drop-down menu.

Initial Value: Set the initial value for the variable; when you first download this variable, this value will be downloaded to the controller and involved in the operation.

Variable Description: Description information of variables which cannot exceed 64 characters at most.

9.4 Standard data type

In declaring variables, in order to ensure the correctness of the type selected, you can see the Table 17 to understand the basic type first.

Table 17 Data type

No.	Key words	Data Type	Value Range	Storage bit number	Note
1	INT	Integer	-32,768~32,767	16 bit	
2	SINT	Short integer	-128~127	8 bit	
3	DINT	Double integer	-2,147,483,648~2,147,483,647	32 bit	
4	WORD	Word type	0~65,535	16 bit	This type variable can also be used for bitwise operations
5	DWORD	Double word type	0~4,294,967,295	32 bit	This type variable can also be used for bitwise operations
6	BYTE	Byte type	0~255	8 bit	This type variable can also be used for bitwise operations
7	BOOL	Boolean type	FALSE(0) TRUE(1)	8 bit	
8	REAL	Real/single precision floating-point type	-3.402823466E+38~-1.175494351E-38 1.175494351E-38~3.402823466E+38	32 bit	
9	LREAL	Double Real/single precision floating-point type	-1.7976931348623158e+308~-2.2250738585072014e-308 2.2250738585072014e-308~1.7976931348623158e+308	64 bit	

No.	Key words	Data Type	Value Range	Storage bit number	Note
10	TIME	Time	It is always composed of an initial t and T (time or TIME) and a digit symbol. Therewith, it is the exact time statement including day (displayed as d), hour (displayed as h), minute (displayed as m), second (displayed as s), and millisecond (displayed as ms). Please attention, the time item must be aligned according to the time order (d is in front of h, h is in front of m, m is in front of s, and s is in front of ms), but needs not include all time parts. The maximum value: 49 days, 17 hours, 2 minutes, 47 seconds, and 295 ms (4294967295ms)		The correct form is shown as: T#14ms T#100S12ms (it is permitted that the highest component can exceed the limit) T#12h23m50s Error form is shown as: T#5m68s (the case that the limit in the lower component is exceeded) T15 m (# lost) T#14s12d (mistake of time order)
11	DATE	Date	1970-01-01 to 2106-02-07		A date constant is started from d, D, DATE or date, and then followed with #. Then the date is input according to the format of year/month/day. e.g. DATE#1996-05-06 d# 1972-03-29
12	TOD	Time of day	00:00:00~23:59:59.999		It is used for storing time of one day, started from od#, TOD#, TIME_OF_DAY#, or time_of_day #, and followed by the time appeared in the format of time/minute/second. Second can be input as the real digit, or input by the fraction form of second. e.g. tod#00:00:00
13	DT	Date and Time	1970-01-01-00:00:00 to 2106-02-07-06:28:15		The combination of date and time: it is started from dt#, DT#, DATE_AND_TIMEor date_and_time. Date and time are connected by a hyphen. e.g. dt# 1972-03-29- 00: 00: 00

9.5 Complex data type

Besides standard data type, there is complex data type which is composed of standard data type, that is Array and Function block.

9.5.1 Array type

Array (ARRAY) is convenient to quote for sequentially combining data of same type to an organic whole.

Users can define one-dimensional array according standard data type to sequentially combine data of same type. Each element in array has a fixed number or subscript.

You can initialize all elements in array or not.

For example: assign initial value to a one-dimensional ARR1 and the subscripts are from 0 to 6 with the variable of REAL. As shown in Figure 92.

No.	Name	Description	Data Type	Initial Value	Area
0001	ARR1	array1	ARRA...		G Area

No.	Name	Description	Data Type	Initial Value	Area
0001	ARR1 [0]		REAL	0	G Area
0002	ARR1 [1]		REAL	0	G Area
0003	ARR1 [2]		REAL	0	G Area
0004	ARR1 [3]		REAL	0	G Area
0005	ARR1 [4]		REAL	0	G Area
0006	ARR1 [5]		REAL	0	G Area
0007	ARR1 [6]		REAL	0	G Area

Figure 92 Assign initial value to array component

In LD Editor, to assign a value to each of the array elements, you can use MOVE to achieve this purpose, as shown in Figure 93.

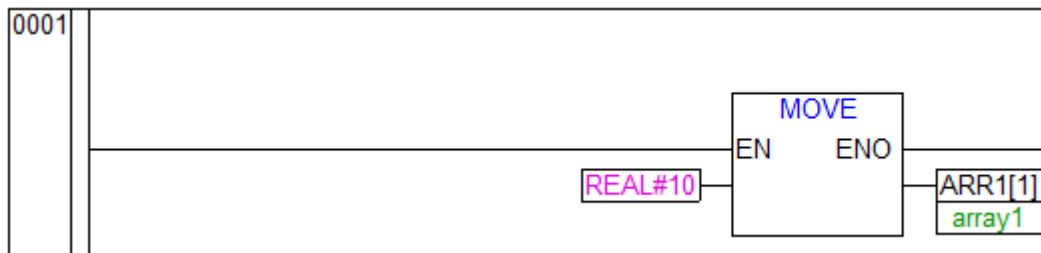


Figure 93 Assign value to array component



- In program area, use a constant in the format as "data type+"#"+numerical value".

Define Array

Users can define one-dimensional array type variable in dialog variable declaration. Select item ARRAY in data type drop-down list by clicking drop-down arrow in Variable Type unit of variable lines after adding variable. The popup dialog of Array definition is shown in Figure 94.

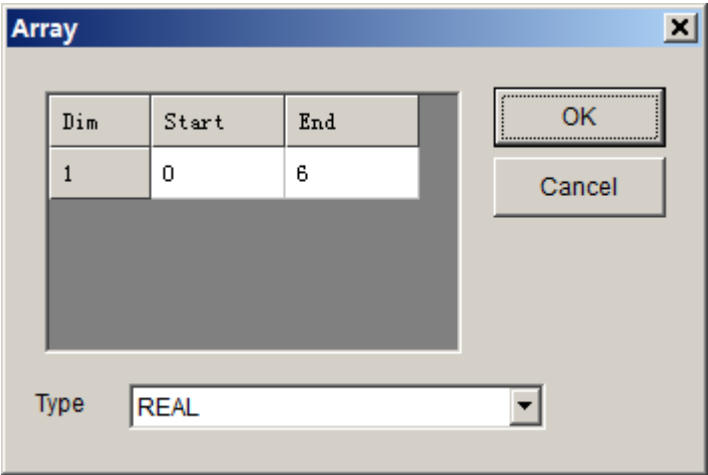


Figure 94 Define the array

Define the array dimension through Dim list and only one dimensional array is supported. Start expresses the start subscripts and Column End is end subscripts. Users can select the element type, including all standard types in drop-down menu of Type. Then click **OK** button to finish the definition.



- The limitation of array subscript is from 0 to 2147483646. The default subscript starts with 0. Users can change the start bit of start subscript according to different needs.
- The maximum number of all elements in array can't exceed 8000.

Select variable tag line in variable list and double-click the Serial Number or right-click command Details to pop up the window of defining each sub-item of this variable, as shown in Figure 95. The window includes Name, Description, Data Type, Initial Value, Area. You can modify the detail information of each sub-item.

No.	Name	Description	Data Type	Initial Value	Area
0001	ARR1	array1	ARRA...		G Area

LD01. ARR1					
No.	Name	Description	Data Type	Initial Value	Area
0001	ARR1 [0]		REAL	100	G Area
0002	ARR1 [1]		REAL	0	G Area
0003	ARR1 [2]		REAL	2	G Area
0004	ARR1 [3]		REAL	3	G Area
0005	ARR1 [4]		REAL	60	G Area
0006	ARR1 [5]		REAL	70	G Area

Figure 95 Each sub-item of array element

■ Quote Array Element

Array element can be quoted in program, such as quoting one-dimensional array element in FBD language. For example, quote ArrayVar[2] in a program.

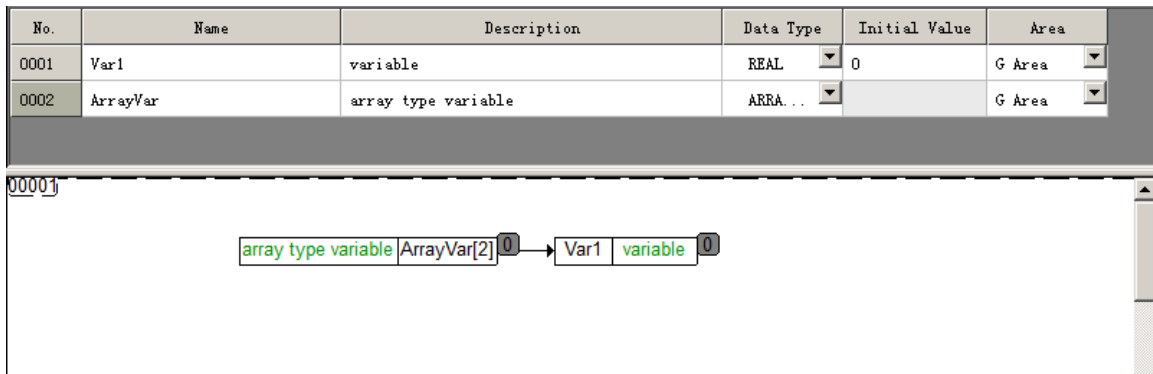


Figure 96 Quote array element

9.5.2 Function block type

Function Block is a group of data elements as organic as array, the whole unit is used to achieve some specific functionality. Each elements of a Function Block can be a basic data type and vary from each other.

You can use the system-defined Function Block or the self-defined Function Block, as shown in Figure 97. The system-defined Function Block can be referred in Library Managing Window.

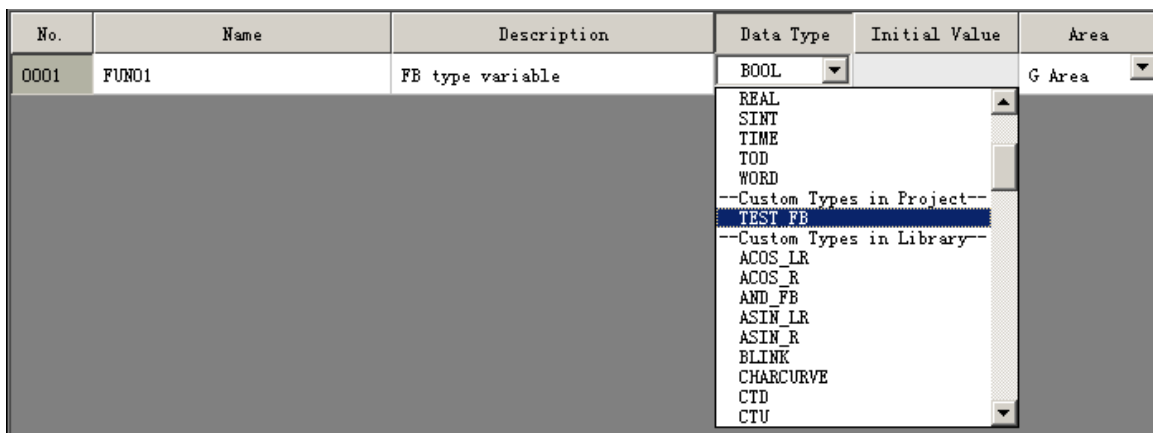


Figure 97 Custom type in project

For example: define a variable tag with ACOS_LR as the Data Type, the instance FUN01 is as shown in Figure 98

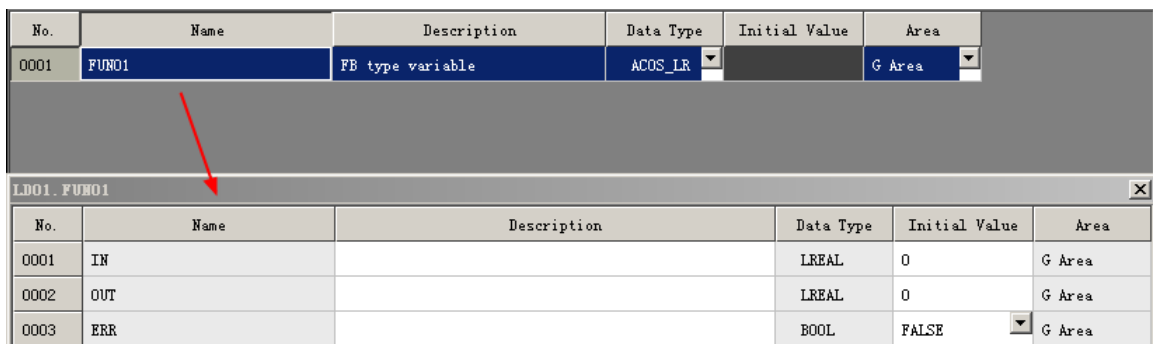


Figure 98 Elements of function block

- Quote Element of Function Block

You can quote an element of Function Block. For example, quote ERR of ACOS_LR in FBD language, as shown in Figure 99.

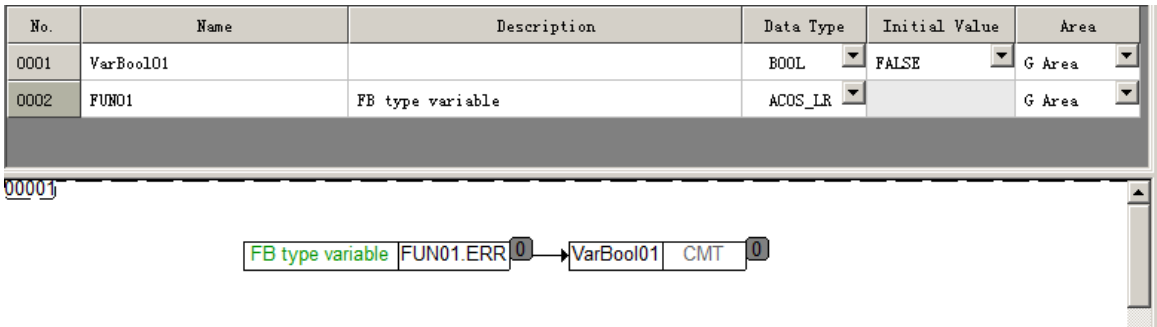


Figure 99 Quote a sub-item of a function block



- You can only modify the Initial value of the sub-items of Array or Function Block, but not the Initial value of the variable of Array or Function Block.

9.6 Surveillance list

The monitoring list is used to classify and manage the important variables, which is easy to control and track the numerical changes in the monitoring mode. Multiple monitoring lists can be created by the [Surveillance list] node under the project management tree.



- Project Management: right click [Surveillance list], click [Add List].

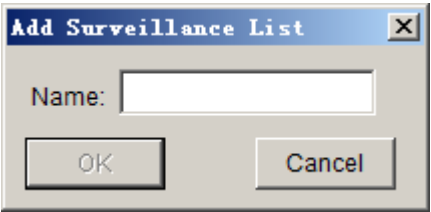


Figure 100 Add surveillance list

In the edit box, enter the list name, which can be used with any character, with no more than 32 characters in length.

The created surveillance list will be displayed in the form of a child node for [Surveillance list] node. It is as shown in Figure 101.

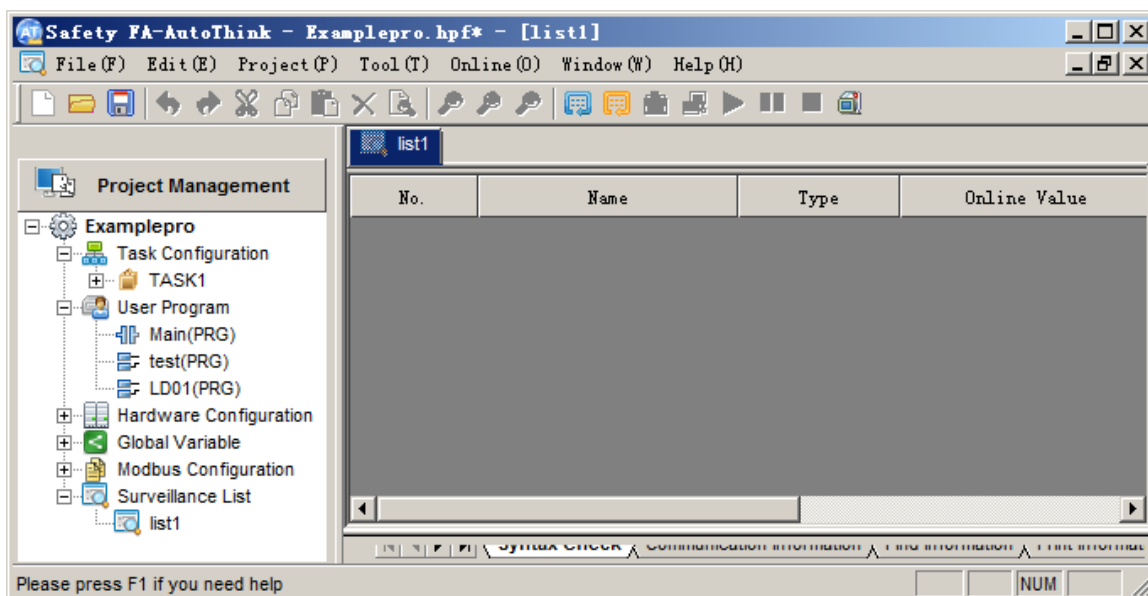


Figure 101 Surveillance list

The context menu in the edit area can help you to add or delete variables as in Figure 102.

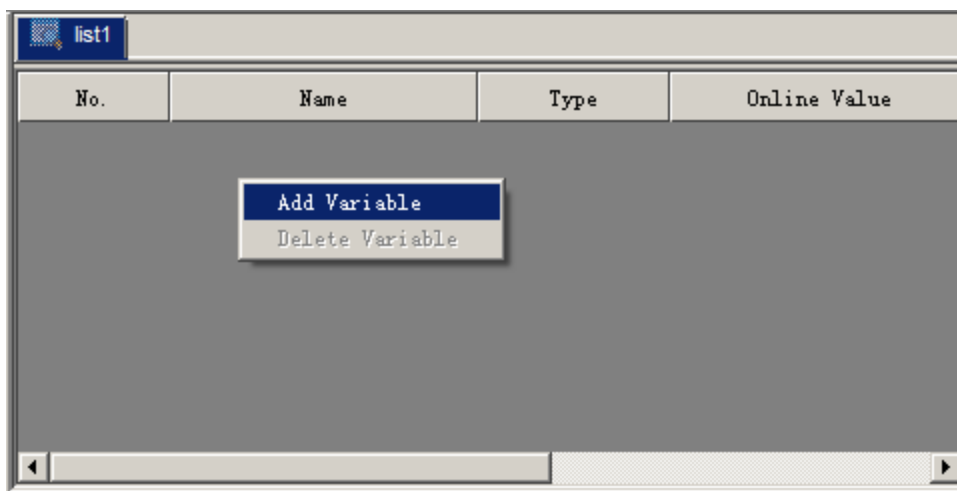


Figure 102 The context menu of work area

[Add] commands each time can add a variable, except the variable name is required of variable line; the other items are not required, because when the software in monitor mode, the software will automatically obtain the variables related to content and display. But the variable name you fill in must be declared successfully.

You can add global variables to the monitor list, and add local variables from user program; the latter written format is **program name.variable name**, such as **TEST1.VAR1**.

9.7 General operation of variable

9.7.1 Add

Select [Add] command to add a record of variable, please refer to [9.3 Variable declaration](#) for the operation.

The newest record is always the end record of variable list.

9.7.2 Insert

Select a variable or multi-variables, and then select [Insert] command; the new variable will be inserted before the selected variable which has the minimum No. You can insert only one variable at a time.

9.7.3 Copy and paste

Cut, if the variable name with new variables in the table of variables no duplicate names, paste after the variable name and other information remain unchanged; if duplicate with new variables in the table of variables, paste after the variable name automatic system in accordance with the naming rules renaming and other information remains unchanged.

Paste, Paste above and Paste below: the difference is that after pasting three different locations.

Paste: the variable will be added to the end of the variable list after pasting them.

Paste above: Paste objects inserted above the selected row.

Paste below: Paste objects inserted below the selected row.

9.8 Customized variable group

You can do the following operations to [Global Variable] node in project tree.

9.8.1 Add

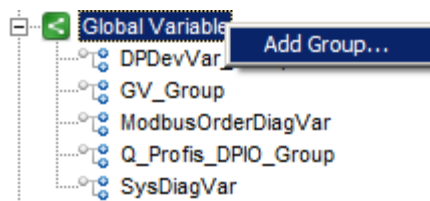


Figure 103 Customized variable group

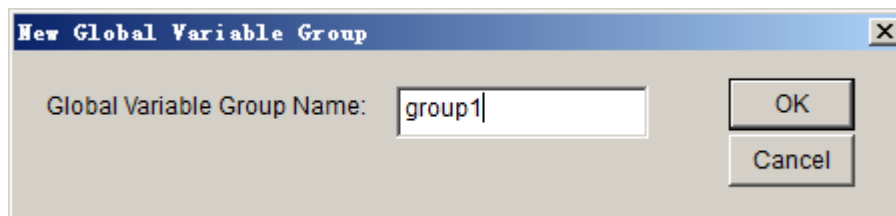


Figure 104 Set group name

Input a global variable group name, and click **OK** to finish the operation.

Double-click the new added global variable group, you can edit the variables in its variable editor window, such as Insert, Add, Delete, Copy, Cut and so on, see [9.7 General operation of variable](#) for details.

9.8.2 Rename

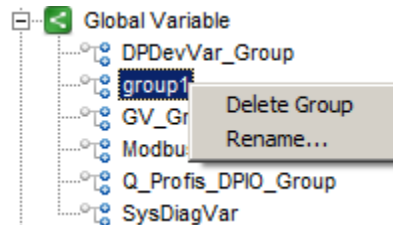


Figure 105 Rename

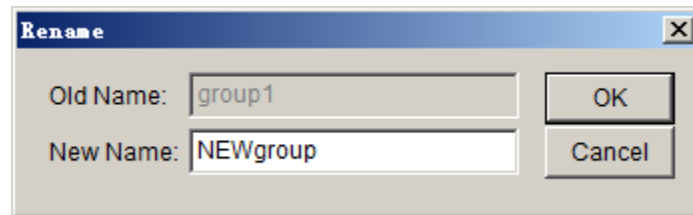


Figure 106 Rename dialog

Input a new group name in the field of New Name, and click **OK** to confirm the changing name operation.

9.8.3 Delete

You can delete the non - System - defined variable groups.



- The group cannot be recovered as deleting.
- The naming rules of the global variable group is as the same as the [9.2 Variable naming rules](#), it will not be described here anymore.

9.9 Export and Import Variables

9.9.1 Export Variables

1. Overview

If you want to batch edit global variables or import global variables from other projects, first, you need to export all global variables from the project.

The global variables which are defined in global variable group and program (PRG) can be exported in the form of .xls and the variables are classified by tab in table.

2. Requirement

The project has been opened.

3. Step

To export the global variables, follow the following steps:

- Step 1.** . Click the "Project" menu.
- Step 2.** Select the "Export Variable" command.
"Save as" dialog box will pop up.
- Step 3.** Enter a file name in save path.
Default name is "project name.xls".
- Step 4.** Click the **Save** button.

9.9.2 Import Variables

1. Overview

You can import the variable table that has be modified after exporting or the variable table exported by other projects into the current project. The global variables which are defined in global variable group and program (PRG) can be imported in the form of .xls.

- Import format requirements
 - ☐ Variable table name is not required, suffix is .xls.
 - ☐ The first row format of the global variable group table must be :variable group name + (COMMON).
 - ☐ The first row format of the PRG variable table must be: program name + (PRG).
 - ☐ There cannot be empty rows between variables row.
 - ☐ Each property of variable cannot be empty.
 - ☐ See [9.2 Variable naming rules](#) for variable name requirements.
- Import rules
 - ☐ Support importing new variable groups and does not support importing new program variables.
 - ☐ If you do not define a data area for variable, the G area is imported as a default data area.
 - ☐ Variables that have been defined in the project can't be imported.
 - ☐ When a variable group in a project is imported again, the variables with the same name in the group are overwritten.

2. Requirement

A variable table.

3. Step

To import the global variables, follow the following steps:

- Step 1.** Click the "Project" menu.
- Step 2.** Select the "Import Variable" command.
"Open" dialog box will pop up.
- Step 3.** Select a variable table.
- Step 4.** Click the **Open** button.

The variable table is imported with result in [Syntax Check] tab.

9.9.3 Export hardware Variables

1. Overview

You can export the channel variables of I / O modules, and modify the channel name and channel description in batch.

Editing rules:

- In the table, only the Channel Name and Channel Comment are allowed to modify.
- See [9.2 Variable naming rules](#) for channel naming.
- The channel name cannot be duplicate.

2. Requirement

The I/O modules have been added.

3. Step

To export the global variables, follow the following steps:

Step 1. Click the "Project" menu.

Step 2. Select the "Export Hardware Point" command.

Prompt dialog box will pop up.

Step 3. Click the **OK** button.

"Save as" dialog box will pop up.

Step 4. Enter a file name in save path.

Default name is "project name Channel.xls".

Step 5. Click the **Save** button.

9.9.4 Import Hardware Variables

1. Overview

After the exported channel variables are edited, you can import to the project.

2. Requirement

A variable table

3. Step

To import the channel variables, follow the following steps:

Step 1. Click the "Project" menu.

Step 2. Select the "Import Hardware Point" command.

Prompt dialog box will pop up.

Step 3. Click the **OK** button.

"Open" dialog box will pop up.

Step 4. Select a channel variable table.

Step 5. Click the **Open** button.

The variable table is imported with result in [Syntax Check] tab.

9.10 Variable Group Management

9.10.1 Create a File

1. Overview

Global variable groups can be classified and managed by folders.

You can create multiple folders in the Global Variable node, or create folders under the folder.

2. Requirement

The project has been opened.

3. Step

To create a new folder, follow the following steps:

Step 1. Right-click the "Global Variable" node.

Step 2. Select the "Add Folder" command in the right-click menu.

"Add Folder" dialog box will pop up.

Step 3. Enter a folder name.

- ☐ Cannot contain symbols (English characters) \, /, :, *, ?, "", <>, |
- ☐ Can't be named "Global Variable"
- ☐ Can't have the same name as the variable group name
- ☐ Name length up to 32 characters

Step 4. Click the **OK** button.

The **OK** button is not operational when the name is not valid.

9.10.2 Delete

1. Overview

When you delete a folder, all folders and variable groups under the folder will be deleted. Please confirm whether to delete folders before deleting.

2. Requirement

A folder has been created.

3. Step

To delete a folder, follow the following steps:

Step 1. Right-click the folder.

Step 2. Select the "Delete" command in the right-click menu.

Confirmation dialog box will pop up.

Step 3. Click the **OK** button.

9.10.3 Rename

1. Requirement

A folder has been created.

2. Step

To rename a folder, follow the following steps:

Step 1. Right-click the folder.

Step 2. Select the "Rename" command in the right-click menu.

"Rename" dialog box will pop up.

Step 3. Enter a new name.

For naming rules see [9.10.1 Create a File](#).

Step 4. Click the **OK** button.

The **OK** button is not operational when the name is not valid.

9.11 Access of Variable

Access of variable in the process of computing includes read and assign values to the variable.

- Access a simple variable by entering variable name directly.
- Access an item of function block case by the format: "function block name.item name".
For example, R_TRIG01.Q represents that read set value item of R_TRIG01 and the premise is the type of R_TRIG01 has declared to R_TRIG.
- Access a local variable (quote it in other program) by the format: "program name. variable name".

For example, VAR1 (local variable defined in POU1) is quoted in POU2 by the format of POU1.VAR1.

- Quick access of variable

Select a variable name in programming area and press **Shift+F2** key. A Variable Declaration dialog box related to this variable will pop up and the variable property will quickly be displayed in this window.

9.12 Delete unused global variables

This command is used to collect and remove global variable in the project, which have been created but not involved in the configuration of the algorithm or Modbus communication.

When the project is successfully compiled, the command is available.



- Menu bar: Click [Project] – [Delete unused global variables].

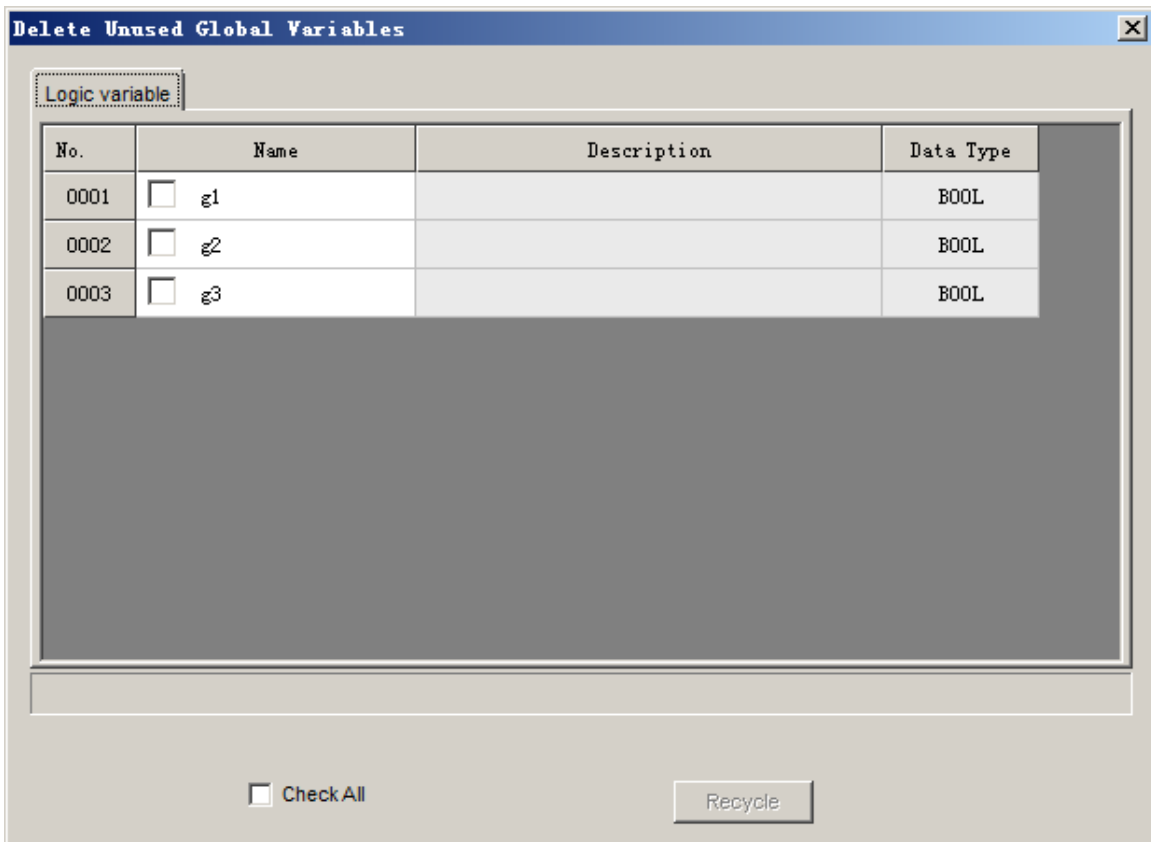


Figure 107 Delete unused global variables

The window displays variables defined in global variable group and POU of PRG type but not used.

Manually check the variables to be deleted, or rapidly select all by click the box of **Check All**, click the **Recycle** button to delete variables.



- Once the logical variable is deleted, it cannot recover.

9.13 Show cross reference table

After successful compiling, you can check the cross table about the relationship on variables and programs of the project.

Select [Project] – [Show Cross Reference Table] to display the Cross Table window, as shown in Figure 108. It shows the cross information in current project, the information contains these items: variable name, POU Name, and Position. Double-click any one of the information to locate to the cross position in POU.

Cross Reference Table			
No.	Name	POU Name	Position
0001	fbd16.var2	fbd16 (PRG). fbd	00001
0002	fbd16.var1	fbd16 (PRG). fbd	00001
0003	fbd15.var1	fbd15 (PRG). fbd	00001
0004	fbd15.var2	fbd15 (PRG). fbd	00001
0005	fbd1.var1	fbd1 (PRG). fbd	00001
0006	ld20.var1	ld20 (PRG). ld	0001
0007	ld3.div_real_1	ld3 (PRG). ld	0001
0008	ld3.var2	ld3 (PRG). ld	0001
0009	ld19.acos_lr_1	ld19 (PRG). ld	0001
0010	ld17.var1	ld17 (PRG). ld	0001
0011	ld1.var1	ld1 (PRG). ld	0001
0012	fbd11.asin_r_1	fbd11 (PRG). fbd	00001
0013	ld11.var1	ld11 (PRG). ld	0001
0014	ld3.var1	ld3 (PRG). ld	0001
0015	ld4.div_lreal_1	ld4 (PRG). ld	0001
0016	fbd15.log_lr_1	fbd15 (PRG). fbd	00001
0017	fbd7.var1	fbd7 (PRG). fbd	00001
0018	ld18.asin_lr_1	ld18 (PRG). ld	0001

Figure 108 Cross reference table

9.14 Variable debug readback



- Menu bar: click [Online] - [variable debug readback].

Use this command to assign online value of variable in M area to its initial value when exit the monitoring mode.

9.15 Input Assistant

Using the Input Assistant, you can quickly select the variable or function block names to replace for a variable or block component.



- Menu bar [Click Edit] - [Input Assistant];
- Shortcut key: **F2**.

■ Variable

In the program edit area or watch list, select a variable name input box, press **F2**, select the target variable you want to replace in the help management dialog box, and click Confirm Change.

■ Block element

In the Program Edit area, select the block component, press **F2**, select the target block element to be replaced in the Help Management dialog box, and click Confirm Change.

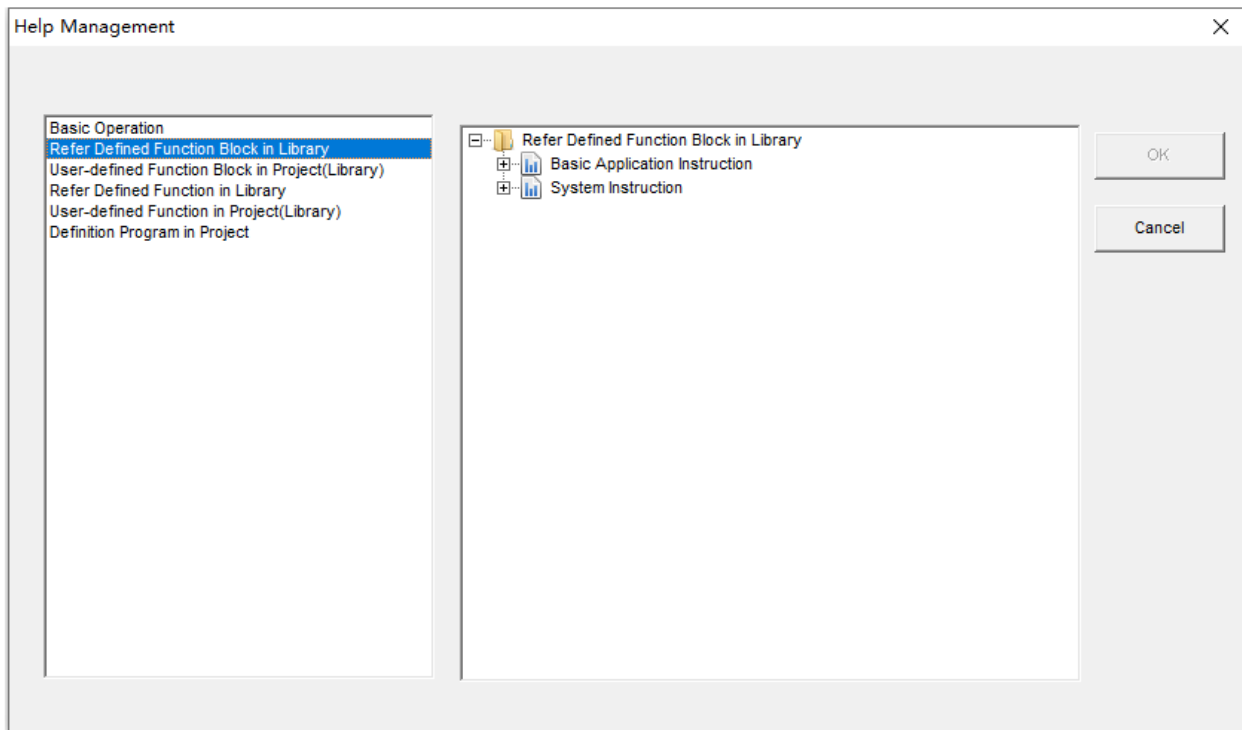


Figure 109 Enter Assistant Help Management



- Note: If you select a variable for a POU in a PRG local variable, the text is returned as POU name.local variable name.

9.16 Variable Property

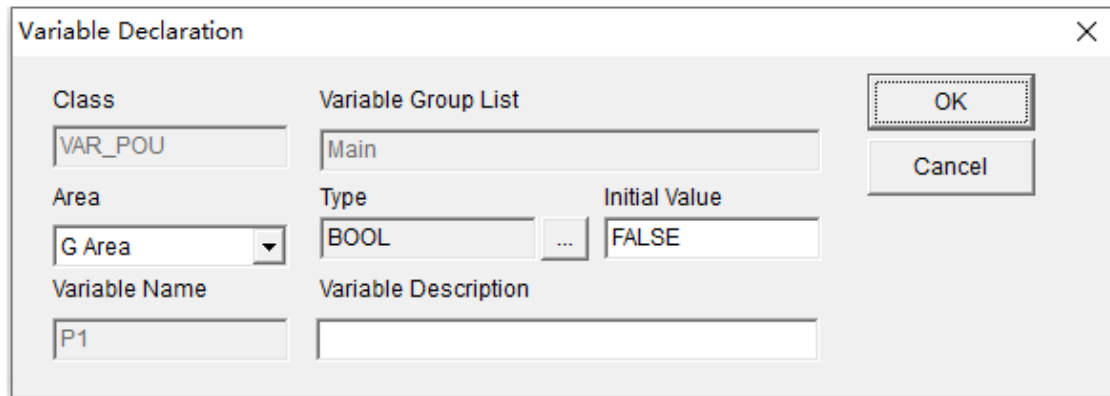
The properties window for the currently selected variable can be quickly displayed by the Variable Properties menu item.

In the Program Editor, select the variable name and view the variable properties in the following way.



- Program area: Right-click element or variable name, click [Variable Properties];
- Menu bar: Click [Edit] - [Variable Properties];
- Shortcut key: **Shift + F2**.

The Variable Declaration dialog box pops up for this variable, as shown in the figure. You can modify Area, Type, Initial Values, and Variable Descriptions.



The image shows a 'Variable Declaration' dialog box with a close button (X) in the top right corner. It contains several input fields and buttons:

- Class:** A text box containing 'VAR_POU'.
- Variable Group List:** A text box containing 'Main'.
- Area:** A dropdown menu showing 'G Area'.
- Type:** A text box containing 'BOOL', followed by an ellipsis button '...'.
- Initial Value:** A text box containing 'FALSE'.
- Variable Name:** A text box containing 'P1'.
- Variable Description:** An empty text box.
- Buttons:** 'OK' and 'Cancel' buttons are located on the right side of the dialog.

Figure 110 Variable Declaration

When the POU is set to a function, the Area option cannot be modified in the Variable Declaration dialog box.

Chapter 10 Programming language

POU can be written in 2 languages: FBD and LD, their characteristics as follows:

FBD: Function Block Diagram, is a kind of programming language, which is similar to the structure of digital logic circuit, and uses Boolean algebra to represent the logic controls, some complex functions are expressed by the instruction frame. Programmers have a basic knowledge of digital circuit suitable for the use of this language.

LD: Ladder Diagram, is evolved from the contactor, relay in the electrical control system, and corresponding to the operation schematic diagram, for the relay control electrical personnel are familiar with, especially suitable for digital logic control and is most widely used PLC programming languages, but not for writing large control program.

10.1 FBD editor

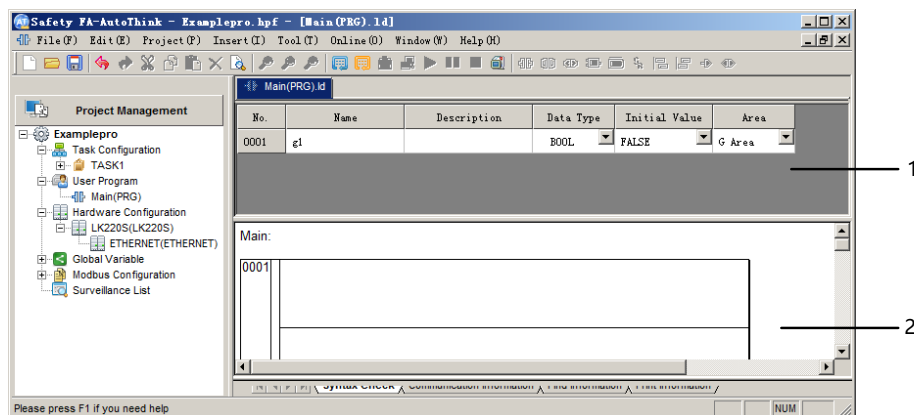


Figure 111 POU Editing in FBD Language

- 1: local variable area
- 2: programming area

The local variable area is located on the upper part of the FBD editor, which is used to display detailed information about the local variables that are declared by the current POU. The classification of local variables varies according to the type of POU.

The function blocks and signal flow lines are programmed in programming area.

The programming area provides a grid in order to facilitate positioning of the blocks, while observing minimum distances. The user can place the corners of blocks and signal flow lines only on the grid lines. Variables and constants can be placed anywhere in the POU; they are displayed and edited in rectangles. The grid can be switched on and off.

When the POU documentation print mode is Multi-page Print, the programming area of a FBD's POU is laid out several pages. The entire graphic region is covered by a raster. The divisions between the individual pages are shown as dotted lines on the screen. When the POU is printed out page-for-page, each page exactly reflects a page in the edit area.

Care should be taken to avoid positioning any objects on the dotted lines as they would then be split up over two pages when the documentation is printed.

The element can place anywhere since the graphic region hasn't the snap grid facility. The continuous process elements contain block, input, output, and return jump and so on. Block elements are linked with one another by linking the output of one element with the input of another. Here the connections formed by signal flow lines are represented. The signal flow lines are edited by dragging with the left mouse button depressed simultaneously. The course of the signal line is automatically determined by the system.

10.1.1 Cursor Position

In FBD programming editor window, click each component or pin to select the object which display blue. Double-click the text area on component to input variable name or description text or other content, which displays blue.

10.1.1.1 On Block, Input, Output, Jump, Label, Return and Comment

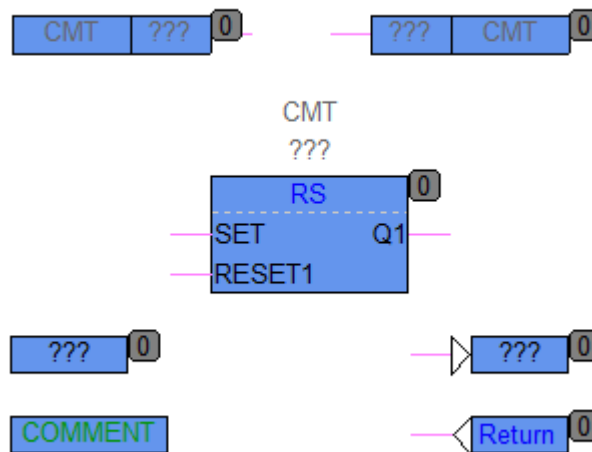


Figure 112 Select Elements

10.1.1.2 On the text of Block, Input, Output, Jump, Label, Return and Comment

Left double-click on the text area, you can select.

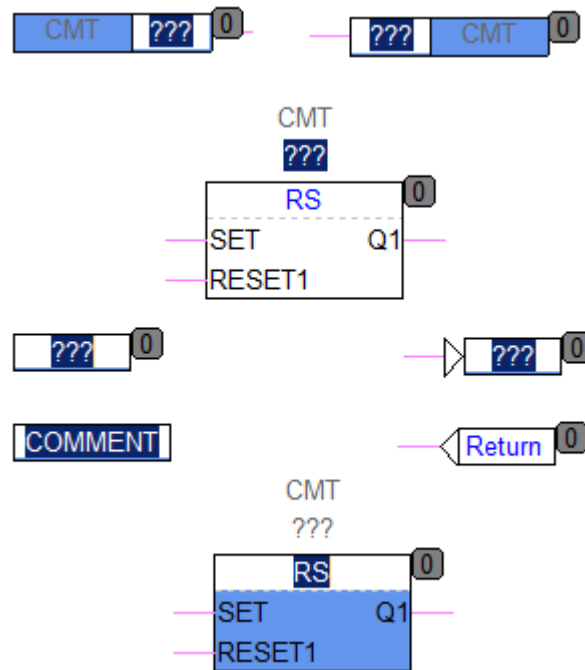


Figure 113 the text of select elements

10.1.1.3 On the input pin of Block, Input, Output, Jump, Label, Return and Comment

You can click the left button to select input pins, but only select one at a time.

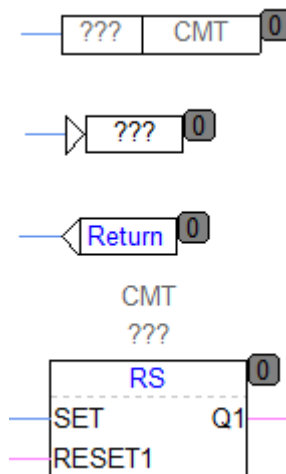


Figure 114 the Input Pin of Select Elements

10.1.1.4 On the output pin of Block and Output

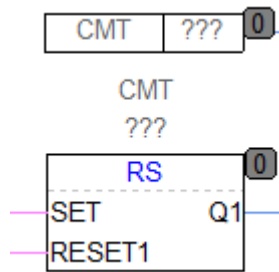


Figure 115 the Output Pin of Select Elements

The color of element in the selected status can be modified through menu [Project] – [options] – [Color].

If the multi-selection is to be conducted, it can be applied by selecting the required elements while pressing the **Ctrl** key; or it can be applied by right clicking and hold on it and dragging the mouse in the editor to draw the rectangle area to select some elements of them.

Use the combination key of **Ctrl+A** to select all elements in the current editor.

10.1.2 Network line



- Menu bar: click [Insert] – [network line];
- Shortcut key: Ctrl+W;
- POU: right click the blank area, click [network line].

In programming environment of FBD language, you can set execution sequence of program using network line to divide programming area into difference section.



- When POU of FBD type, system will add a network line with number of 00001 at top of editor window automatically, and this network line can't be dragged.
- If you move the component position in the network line will not trigger the project compile; if the move across the network line, then it will trigger the project to compile.

10.1.3 Input



- Menu bar: click [insert] – [Input Component];
- Tool bar: ;
- Shortcut Key: **Ctrl+I**;
- POU: right click the blank area, click [Input Component].

This command is used for inserting one input, the occurred text“???” can be selected and replaced with variable or constant.

The newly inserted input element can move along with the mouse, left click after it is moved to the proper place and the insertion is successful. It is as shown in Figure 116.

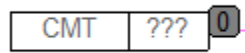


Figure 116 Input

When adding an input element in the programming area, the default input element displays a variable name and a comment; you can make the content of the notes hidden after setting in [Configuration language](#).

10.1.4 Output



- Menu bar: click [insert] – [Output];
- Tool bar: ;
- Shortcut Key: **Ctrl+U**;
- POU: right click the blank area, click [Output].

This command is used for inserting one output, and the text??? Can be selected and replaced with variable, the value related to the input of the output will be distributed to this variable.

You executed this command to insert the output element. The newly inserted output element can move along with the mouse, click after it is moved to the proper place, it is as shown in Figure 117.

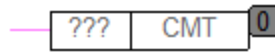


Figure 117 Output

The output element is the same with the input element, after inputting the name of variable in FB type; the system will automatically add the default item name to it.

10.1.5 Block



- Menu bar: click [Insert] – [Block];
- Toolbar: ;
- Shortcut key: **Ctrl+B**;
- POU: right click the blank area, click [Block].

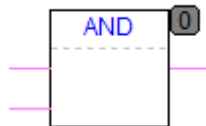


Figure 118 Block

Double click on the AND can be changed to valid function block, function and program name.

Select the desired function block name from [library manager], such as RS, selected from the library manager drag to the target position, when the mouse is  state, release the left key, block element is automatically added to the programming area.

If block is added to POU (FUN/FB) by dragging, see Chapter [8.11.3 Call between program](#).

10.1.6 Jump and Label

Under FBD language circumstance, jump is used in conjunction with the label. How to add jump and label will be introduced in the following part.

10.1.6.1 Jump



- Menu bar: click [Insert] – [Jump];
- Toolbar: ;
- Shortcut key: **Ctrl+J**;
- POU: right click the blank area, click [Jump].

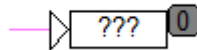


Figure 119 Jump

You can insert Jump Element by this command. The new inserted jump Element can be moved by mouse, click the appropriate position to insert it. Input jump mark name same as the name of label mark definition in text area ????. The jump conditions only for BIT or BOOL type and other types which can be forcedly converted to BOOL type.

10.1.6.2 Label



- Menu bar: click [Insert] – [Label];
- Toolbar: ;
- Shortcut key: **Ctrl+L**;
- POU: right click the blank area, click [Label].

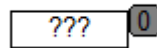


Figure 120 Label

The new inserted label Element can be moved by mouse, click the appropriate position to insert it. Input mark name of label in text area ???.

When the jump condition is **true**, system will jump the program segment before jump label and execute the program line at the position of jump label, it is as shown in Figure 120. When the VAR1 is **true** in program segment 1, jump the program segment 2 and execute program segment 3 where label0 is.

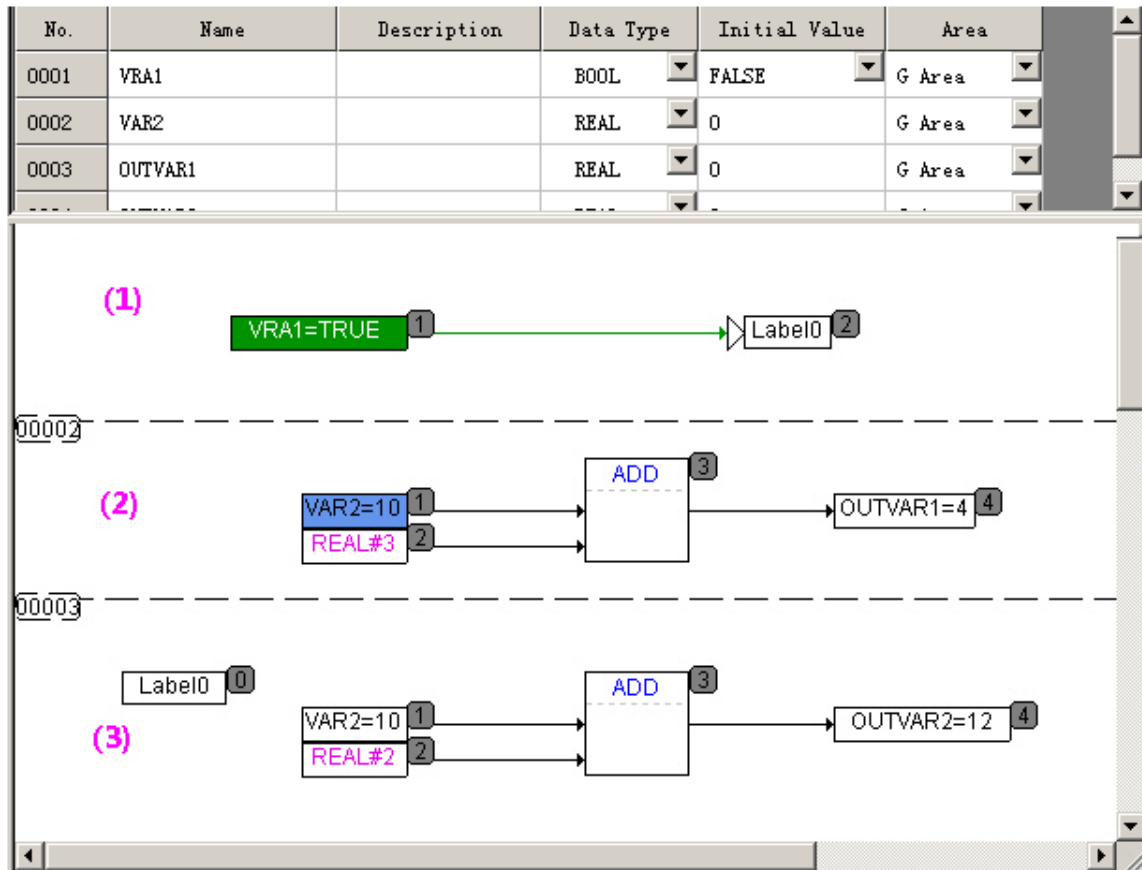


Figure 121 Example for jump and label

10.1.7 Return



- Menu bar: click [Insert] – [Return];
- Toolbar: ;
- Shortcut key: **Ctrl+R**;
- POU: right click the blank area, click [Return].

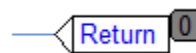


Figure 122 Return

The Return shall provide early exit from a function, function block or program.

Users can insert Return Element by this command. The new inserted return Element can be moved by mouse, click the appropriate position to insert it.

Return element can only be connected to BIT or BOOL type variables and other types which can be forcedly converted to BOOL type.

10.1.8 Comment



- Menu bar: click [Insert] – [Comment];
- Toolbar: ;
- Shortcut key: **Ctrl+K**;
- POU: right click the blank area, click [Comment].

COMMENT

Figure 123 Comment

You add comments to the program, which can enhance the readability of the control logic.

Comment component does not take part in execution of program, and the affection of it is insert text information shows description in program interface of FBD.

You can insert component with the default text comment by this command. Double-click this text area to modify the note. The new inserted Return element can be moved by mouse, click the appropriate position to insert it.

10.1.9 Set /Reset



- Menu bar: click [Insert] – [Set/Reset];
- Toolbar: ;
- Shortcut key: **Ctrl+T**;
- POU: Right click Output, click [Set/Reset].

This command can be used to select the input pin of output element.

S is set symbol, R is reset.

Select an output element, use this command, set the Set or reset function. It is as shown in Figure 124.

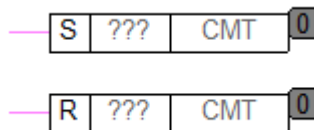


Figure 124 Set and Reset

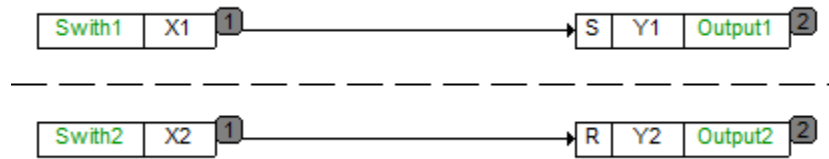


Figure 125 Example

When X1=True, Y1=True; Y1 keep this value, even if the X1=False;

When X2=True, Y2=False; Y2 keep this value, even if the X2=False;

If you repeat active this command, which can make the output state will switch between the set, reset and normal.

10.1.10 Negate



- Menu bar: click [Insert] – [Negate];
- Toolbar: ;
- Shortcut key: **Ctrl+Q**;
- POU: Right click pin of element, click [Negate].

This command is used to invert the commands of input, output, jump and return, and one circle on the link is used for expressing the negation.

When the block, input, jump or return is selected, their input will be negated.

The output or input of block will be negated when they are selected.

The former negation can be cancelled through the secondary negation.

Select the pin of one element and use the negate function to invert the value of the pin.

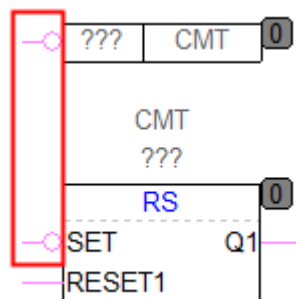


Figure 126 Negate



- Negate function is available only for BOOL, BYTE, WORD, DWORD and BIT.

10.1.11 Component Align

You can use this feature to quickly typeset POU elements. The arrangement of elements: horizontal left/right and vertical alignment.

In POU, you circle the two or more components, and select relevant command from the context menu, it is as shown in Figure 127.

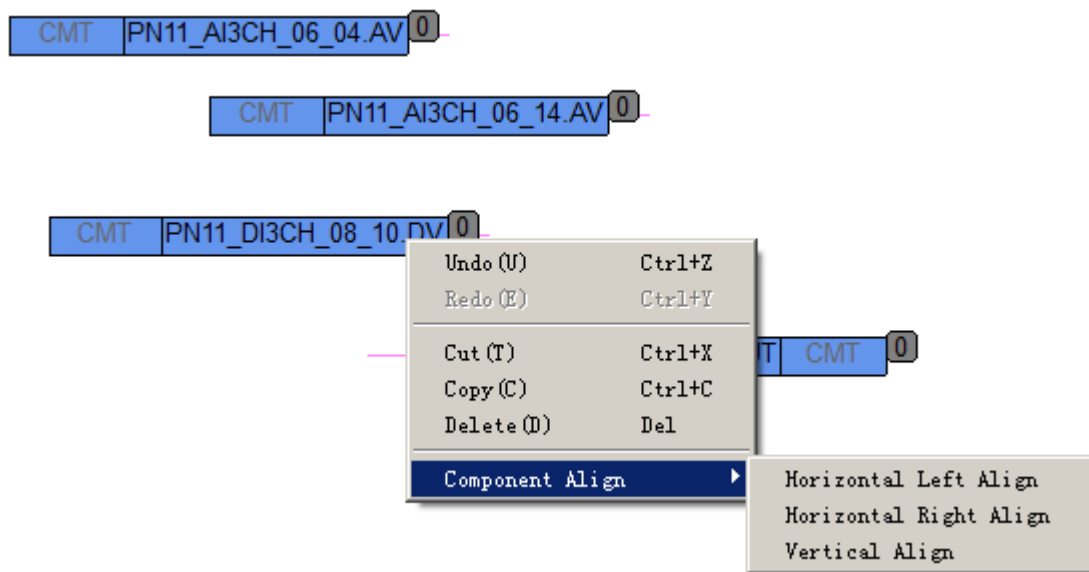


Figure 127 Alignment

Horizontal left/right align: the system will select the left/rightmost position element from the covered range as a benchmark, other elements aligned with the baseline in the horizontal direction. See Figure 128 and Figure 129.

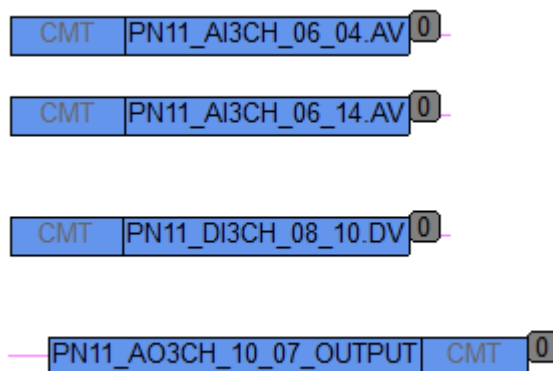


Figure 128 Horizontal left align

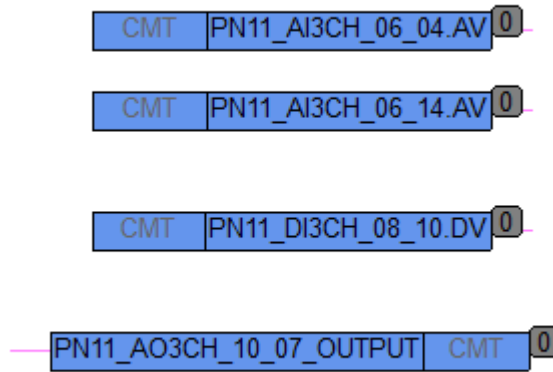


Figure 129 Horizontal right align



- You execute the component alignment commands, or save the alignment results, which will trigger the current project to compile.
- When some components are aligned, a component may be moved to boundary line (black line); this time, the component and the elements behind it will not move.

10.1.12 Execute sequence and number

The top left corner of other component will automatic numbering **0**, except Comment component. This number used for executing sequence of FBD component in online mode or simulation mode.

When add a new component, it will acquire an execution sequence automatically with default number of 0. When you move it, the number will retain the same.

The execution process will start from number 1 component.

In this software, FBD will generate calculation or execution sequence of different components in certain principle as compiling.

Firstly introduce the concept of End Component; it indicates in legal network, there is no right pin or component with null right pin. According to syntax rule of FBD (specified in [10.1 FBD editor](#)):

Output, Jump, Return component and PRG can be considered as end component only;

FB can be end component or not;

Input component, standard function and Function must not be end component.

Take instance as shown in Figure 130 as an example, based on above definition about end component, sequence principle of component in legal network of FBD is:

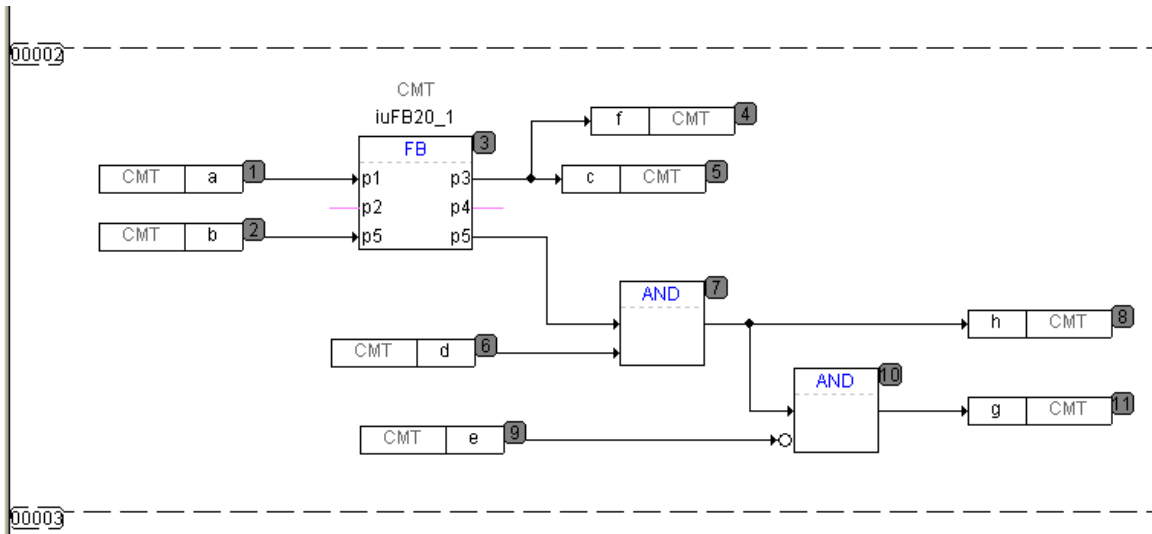


Figure 130 Example of execution order

- (1)Initially, execution numbers of all components are 0, after compiling and sorting, the first number is 1.
- (2)Execution sequence of different end components shall comply with geometry position: from up to down, from left to right. In above example, the end components are: f, c, h, g. Comply with geometry relation, the execution sequence of them is f, c, h, and g.
- (3)Handle each end component according to sequence in (2), according to principle (4) to sort start from end component.
- (4)If any one component want to acquire a sequence number, only when components of all left pins connected have acquired sequence number; or else, you must handle the component connect with left pins comply with pin sequence, the handle principle is (4) as well.

10.2 LD Editor

The Ladder Diagram is a graphically-oriented IEC61131-3 programming language.

The LD language originates from the field of electromagnetic relay systems and describes the flow of current through the individual networks of the program organization units (POU) of a programmable controller.

When the POU documentation print mode is Multi-page Print, the work area of a LD POU is laid out several pages. An implicit raster is applied to the entire work area.

The title bar in LD's POU and FBD's POU is no differences in position and use.

An LD network is delineated on the left and right by the so-called power rail. The state of the left rail shall be considered ON at all times.

1. An LD network comprises the following graphic elements:

- (1) Power flow lines,
- (2) Variables and constants,
- (3) Contacts,
- (4) Coils,

- (5) Conditional and unconditional jumps,
- (6) Functions and function blocks.

There can be several networks on different levels within an LD's POU; these will be processed passing from the top to the bottom unless any explicit jumps have been programmed in.

2. An LD's POU is processed in accordance with the following rules:

- (1) No network element may be calculated until the states of the inputs have been calculated;
- (2) The calculation of a network element is not concluded until the states of the outputs have been calculated.
- (3) The calculation of a network element is not concluded until all the outputs have been calculated, even if the network contains jumps-either forward or backward;
- (4) Networks are processed top-down.

The 4th rule, however, is also dependent on the power flow of the POU, as rules 1-3 must also be obeyed. By way of illustration, the following algorithm is used for determining the processing sequence:

- Step 1. Sort all network elements from top to bottom and, within that sort, from left to right;
- Step 2. Search for the first element in which all the inputs are calculated.
- Step 3. Calculate that element;
- Step 4. If there are other uncalculated elements, go to Step2.

In contrast with the FBD language, no explicit processing sequence can be specified for the blocks and outputs, but instead it emerges from the structure of the program. Feedback messages are also not permitted, as they contravene rule 1.

When analyzing the logical relationship in the ladder diagram, it could be thought that there is DC current flowing from left to right between two Power rail lines. It is as shown in Figure 131.

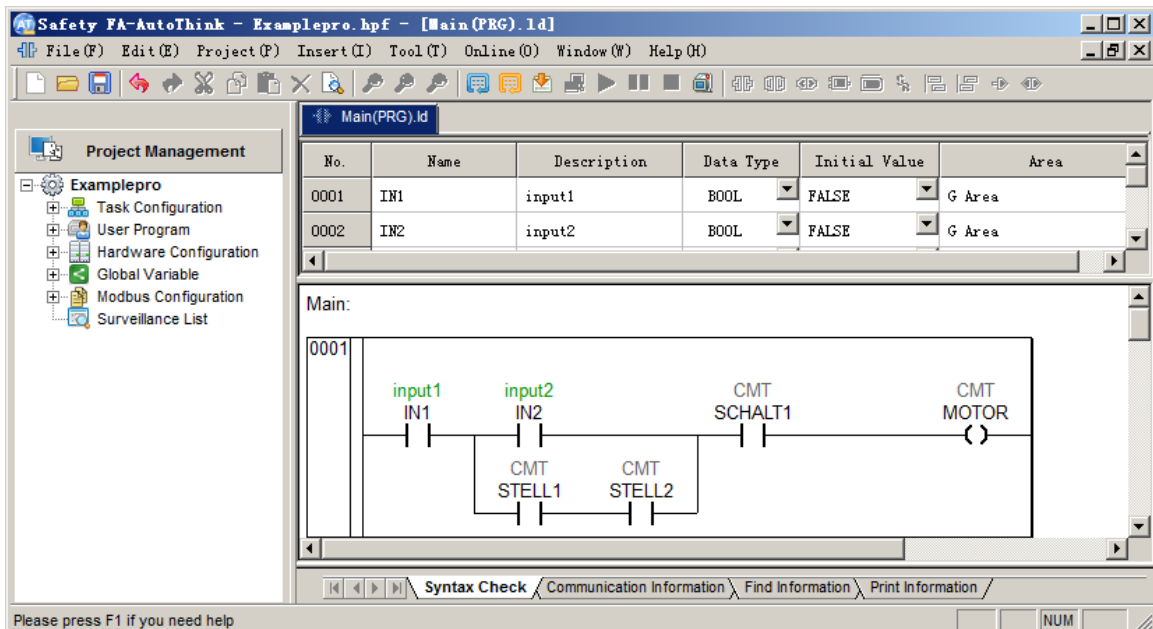


Figure 131 LD editor

10.2.1 Cursor position

The selected content displayed by marking the blue background, there are cursor position examples.

10.2.1.1 Network

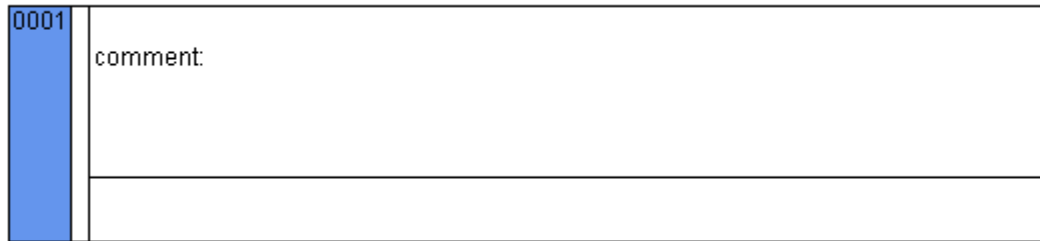


Figure 132 Network

10.2.1.2 Contact

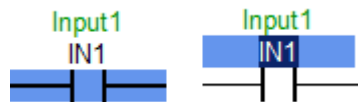


Figure 133 Contact and text

10.2.1.3 Coil



Figure 134 Coil and text

10.2.1.4 Block

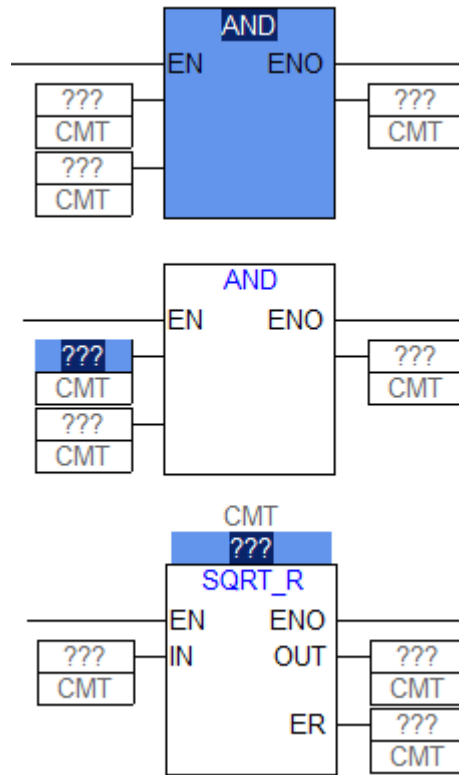


Figure 135 Block and text

10.2.1.5 Return

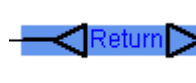


Figure 136 Return

10.2.1.6 Jump

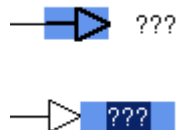


Figure 137 Jump and text

10.2.1.7 Comment and label

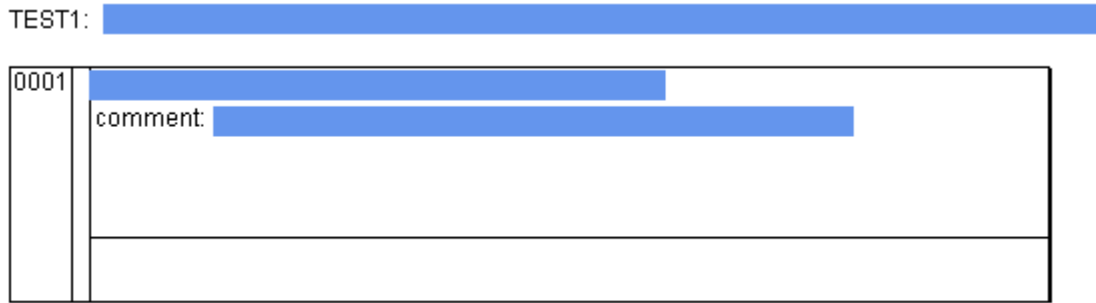


Figure 138 Comment and Label

10.2.2 LD elements

10.2.2.1 Network

Network is an important concept in Safety FA-AutoThink as well as a basic unit of LD program; each LD language POU is formed by network. The network number default is started from 001; every POU can add 999 networks at most.

10.2.2.2 Contact

The contact can be used in parallel, the contact represents the logic input condition, such as switch, button, and internal condition; when one parallel branch transmits the **ON** state, and the parallel branch can transmit **ON** state. Or the contact is linked in series; at this moment, when the contact transmits the **ON** state, the last contact can transmit the **ON** state, these are in accordance with the parallel/series circuits.

├─┤: Normally open contact: if the variable value is **ON**, the state of the left side link will be transmitted to the right side link; otherwise, the right side link is **OFF**.

10.2.2.3 Coil

There are some so-called coils at the right side of the ladder diagram, the coil is generally used for indicating the output result of the logic calculation and controlling the external indicator lamp, the contactor and the internal output condition, etc.; these coils are expressed by () and connected by a horizontal line. The coil transmits the connection state from left to right and copies the state to the BOOL variable, which can describe the state of the entrance line is ON (TRUE of corresponding Boolean variable) or OFF: state (FALSE of corresponding Boolean variable).

-()-: Coil, A coil copies the state of the link on its left to the link on its right without modification, and stores an appropriate function of the state or transition of the left link into the associated Boolean variable.

10.2.2.4 Block

The function block and program can be added in the network diagram; they must be possessed of input and output of Boolean type value and can be applied to the left side of the ladder diagram as the contact.

The diagram illustrates that the basic block is presented by multiple inputs and multiple outputs. The input always indicates the left side of the block structure, and the output always indicates the right side

of the block structure. The name of the block (function block type) is displayed in the middle of the block structure. The example name is displayed at the upper part of the block structure.

The block element includes function block , function and program.

10.2.2.5 Illustration

The ladder diagram includes a series of nodes, the left and right sides are respectively provided with one vertical current line (energy line) to define its range. The middle part is a circuit diagram consisting of the contact point, coil, and connection line. It is as shown in Figure 139.

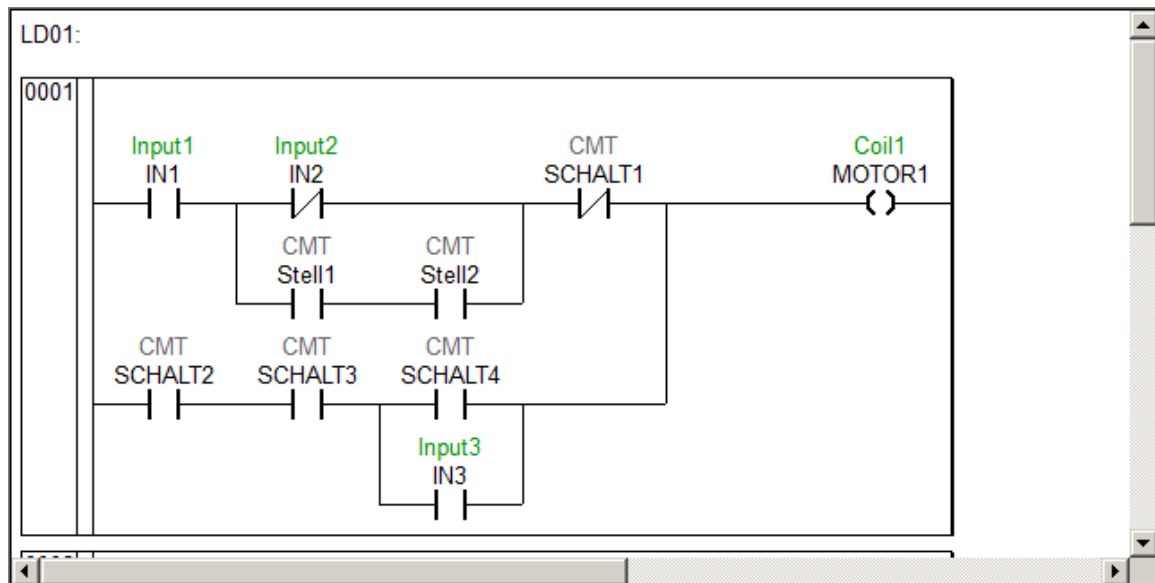


Figure 139 LD

10.2.3 Network

10.2.3.1 Add network



- Menu bar: click [Insert] – [network (Before)]/[Network (After)];
- Shortcut key:  /  ;
- POU: Right click blank space, click [Network (Before)] / [Network (After)].

Then, a new network will appear in front of or behind the selected network in LD editor.



- Each section from POU in LD language, you can add up to 32 elements in the horizontal direction, for a maximum of 16 elements in the vertical direction.

10.2.3.2 Delete

Select the desired deleted Networks directly using the **Delete** Key to remove.

10.2.3.3 Copy and paste network

When the cursor is located in the network, users can copy and paste the selected network in a same or different POU.

Step 1. Copy



- Menu bar: click [Edit] – [Copy];
- POU: Right click blank area, click [Copy].

Select signal network: Click the desired nodes are copied, select the [Copy] command to copy.

Select multiple networks: when you need to select multiple successive networks, select the first one; press **Shift** button and click the last network to select all network between the two. Pay attention that continuously press **Shift** button and right-click Copy command on the last network to copy all selected networks.

Step 2. Paste



- Menu bar: click [Edit] – [Paste];
- POU: Right click blank area, click [Paste].

Paste the network of the cut sheet to the front or back of the current selection.

10.2.4 Series contact



- Menu bar: click [Insert] - [Series contact];
- Toolbar: ;
- Shortcut key: **Alt+I+F**;
- POU: Right click Network / Contact; click [Contact (Before)]/[Contact (After)].

Insert a series contact in the current cursor position.

You can use the [Series contact] command in the following two locations:

If the cursor locates at a network, the new contact will be inserted in the head of this line.

If the cursor locates at contact S1, new contact will be successively inserted behind the current contact S1. If you use Contact (Before), new contact will be inserted before the current contact S1, as shown in Figure 140.

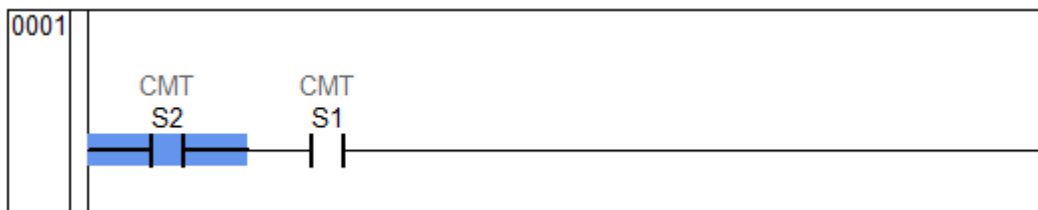


Figure 140 Contact



- If the number of contact is more than 1 in a network, you can add only one coil in this network.

10.2.5 Parallel contact



- Menu bar: click [Insert] - [Parallel Contact];
- Toolbar: ;
- Shortcut key: **Alt+I+P**.

By adding a contact and the existing contacts to establish a parallel circuit relationship.

If you need to create a parallel relationship for a single contact object, just check the contacts and execute the [Parallel Contact] command.

If you need to create a parallel connection for multiple series contact objects, use the **Shift** key to select the associated series contacts, and then execute the [parallel Contact] command. If you want to achieve parallel effect in the Figure 141, select IN1 and IN2 after adding parallel contact.

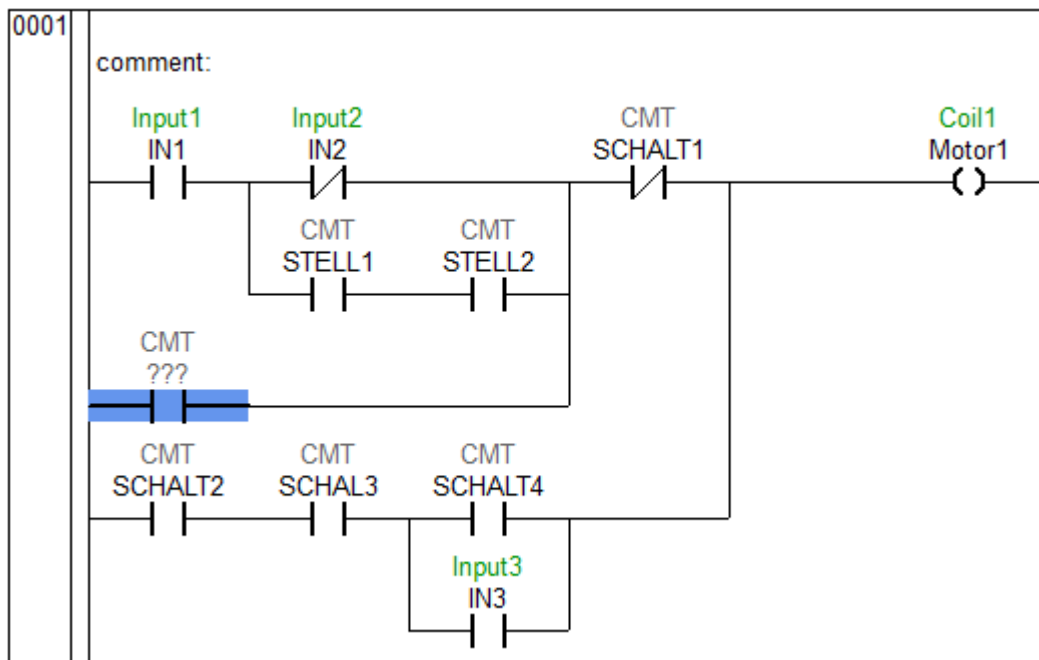


Figure 141 Multiple objects simultaneously add parallel contacts

10.2.6 Coil



- Menu bar: click [Insert] - [Coil];
- Toolbar: ;
- Shortcut key: **Alt+I+L**;
- POU: Right clicks Network, click [Coil].

Add an output coil on the network where the cursor is.

If the cursor locates at a network and there is contact but no coil in the program line, the new coil will be inserted in the end of this line.

If the cursor locates at a coil, new contact will be successively inserted under primary coil, as the selected coil as shown in Figure 142.



Figure 142 the position of the coil can be added



- When a network has only one contact, the coil in this network can establish a parallel relationship.
- A network does not support the coexistence of coil with block element, jump element, or return element.

10.2.7 Block



- Menu bar: click [Insert] - [Block];
- Toolbar: ;
- Shortcut key: **Alt+I+B**;
- POU: Right clicks Network, click [Block].

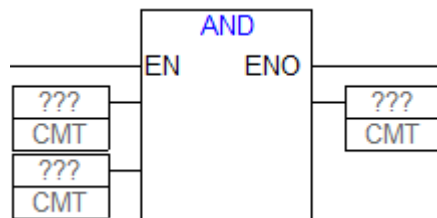


Figure 143 Block

Double click **AND**, you can type the name of the valid function block or function or program.

Multiple blocks can only be built in parallel relation, which is not supported in series.



- A network does not support the coexistence of block element with coil, jump element, or return element.

10.2.8 Set/Reset



- Menu bar: click [Insert] - [Set/Reset];
- Toolbar: ;
- Shortcut key: **Ctrl+T**;
- POU: Right clicks coil, click [Set/Reset].

A setting coil shall be gotten if this command is executed in coil.

The coil is defined to setting or resetting mode.

 : Setting coil. The status of left link will be copied into the left link through setting coil. If the left link is true, the relevant BOOL variable will be set to be true; otherwise, relevant BOOL variable remain unchanged. BOOL variable could be reset by resetting coil.

 : Resetting coil. The status of left link will be copied into the right link through resetting coil. If the left link is true, the relevant BOOL variable will be set to be true; otherwise, relevant BOOL variable remain unchanged. BOOL variable could be set by setting coil.

If you repeat active this command, which can make the output state will switch between the set, reset and normal.

10.2.9 Negate



- Menu bar: click [Insert] - [Negate];
- Toolbar: ;
- Shortcut key: **Ctrl+G**;
- POU: Right click pins of contact/coil, click [Negate].

This command is used for negating contact and coil.

For the coil or contact, a slant will appear in the parenthesis of coil or between straight lines of connection, like  or .

: Negating coil, the status of left link will be copied into right link, and the reverse status of left link shall be copied into relevant BOOL variable through negating coil. If the left link is OFF, the right link is also OFF but relevant BOOL variable is ON.

: Normally closed contact. If its variable value is OFF, the status of left link shall transfer to right link; otherwise, the right link is OFF.

Double negatives cancel negative.

10.2.10 Comment

In order to improve the readability, it is need to add note to program, network and variable or input skip mark to indicate the skip direction of program line.

The specific categories are as follows:

10.2.10.1 Program comment

Input execution statement of current POU to finish the program notes of POU, such as Explain Current POU as shown in Figure 144.

LD01: A brief description of the current POU

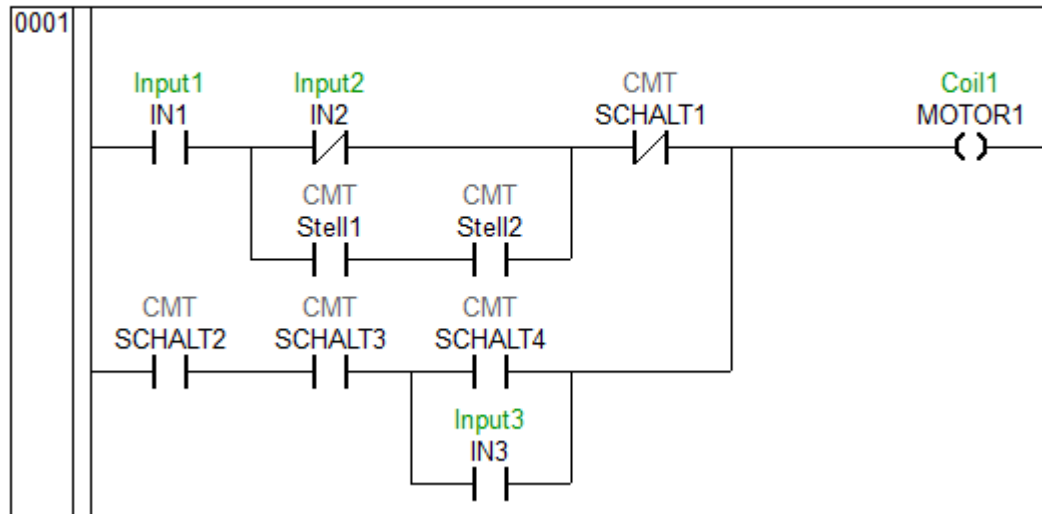


Figure 144 Program comment

10.2.10.2 Network node notes

It is the description of current network.

Select item **Show the Comments of Network Nodes** by the operation of [Project] – [Options] – [Configuration] - [LD] firstly.

Whose default note of this line is **comment** in program area, and then click the note area of blue rectangle box and input notes to it. Click other position of edit window to confirm the completion of the input, it is as shown in Figure 145.

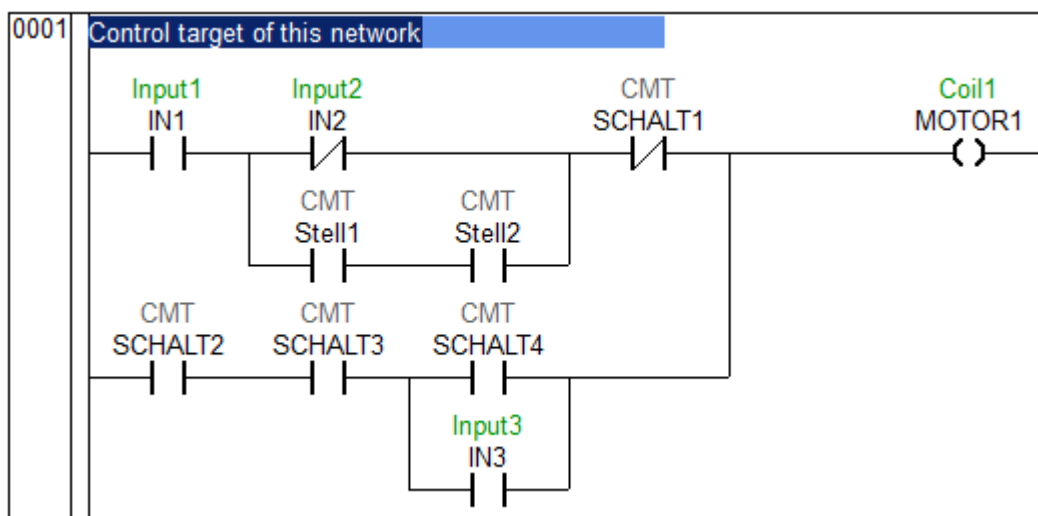


Figure 145 Network comment

10.2.10.3 Variable comment

System allows you to note the variable name and only variable descriptions of instance name of contact, coil and function block will be displayed in LD programming window. When begin to define the variable, you can note or describe it directly in dialog Variable Declaration or modify it in variable groups corresponding to Variable Definition, it is as shown in Figure 146.

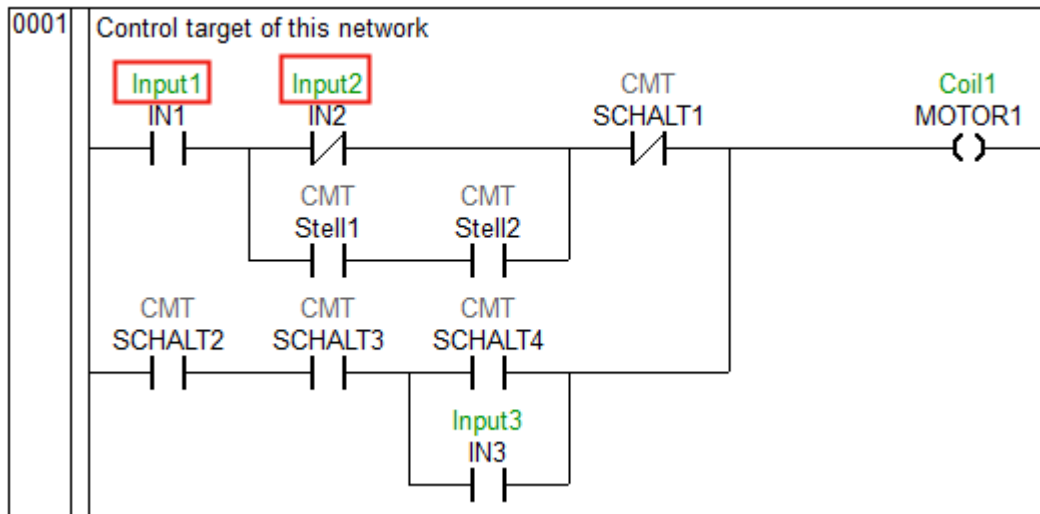


Figure 146 variable comment

If you expect to show as Figure 146, please check the items **Show General Type Comments** and **Show Function Block Type Comments** in the setting window of [Project] - [Options] - [Configuration Language] - [IEC].

10.2.10.4 Jump label

Each network has a mark number locating at the first line and the default value is null. Click the position of first line of this network and a bright blue rectangle input box will be displayed to input jump label only. When the conditions are satisfied, it will jump to related network with specified mark number. it is as shown in Figure 147. Pay attention that the label name should be remain the same as the name behind Jump Element.

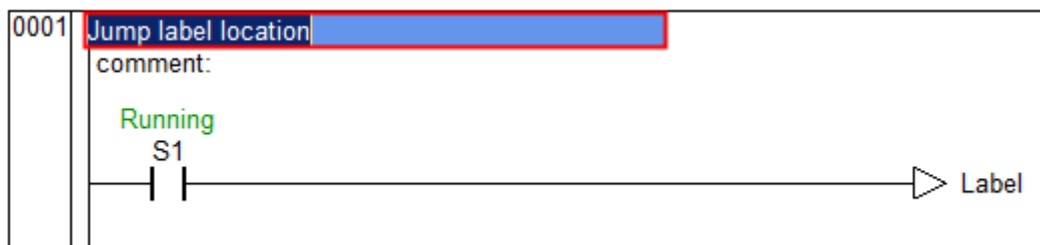


Figure 147 Jump label

If a comment to show branches, please use the key combination **Ctrl+ Enter**.

10.2.11 Jump and return

The commands of jump or return are all to change the execution sequence. Under normal circumstances, controller executes the program according to the network sequence of main program.

10.2.11.1 Jump



- Menu bar: click [Insert] - [Jump];
- Shortcut key: **Alt+ I+ J**;
- POE: Right click network, click [Jump].

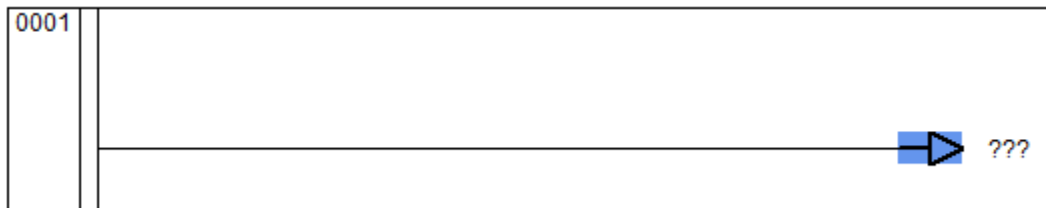


Figure 148 Jump

After adding the jump, it need to input the mark number (default value is ???). The function of mark number is to identify the destination of jump. Figure 149 shows to jump to network 0003 and not to execute the program of network 0002 when the jump conditions are satisfied.

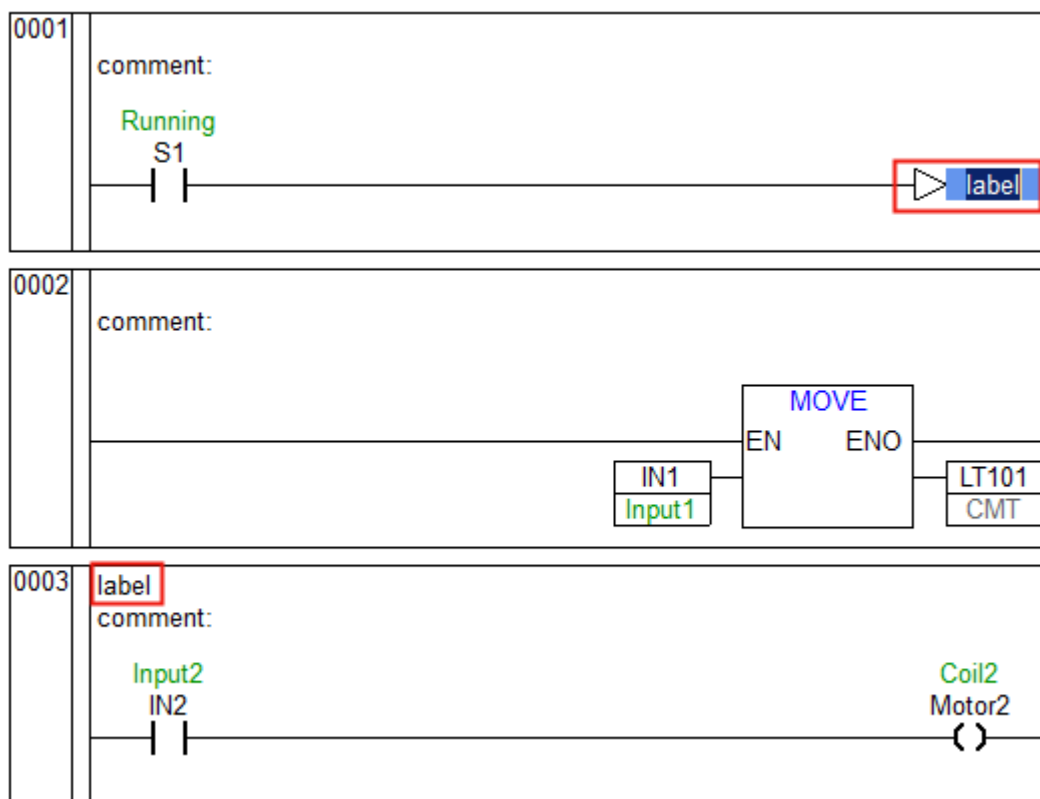


Figure 149 Jump and label



- When a block or coil is added in a section, the jump component is not allowed.

10.2.11.2 Return



- Menu bar: click [Insert] - [Return];
- Shortcut key: **Alt+I+U**;
- POU: Right clicks Network, click [Return].

Users can use return to make the called POU not execute and return to POU who calls it at upper layer when the conditions are satisfied.

When cursor locates on network, insert this element, as shown in Figure 150.

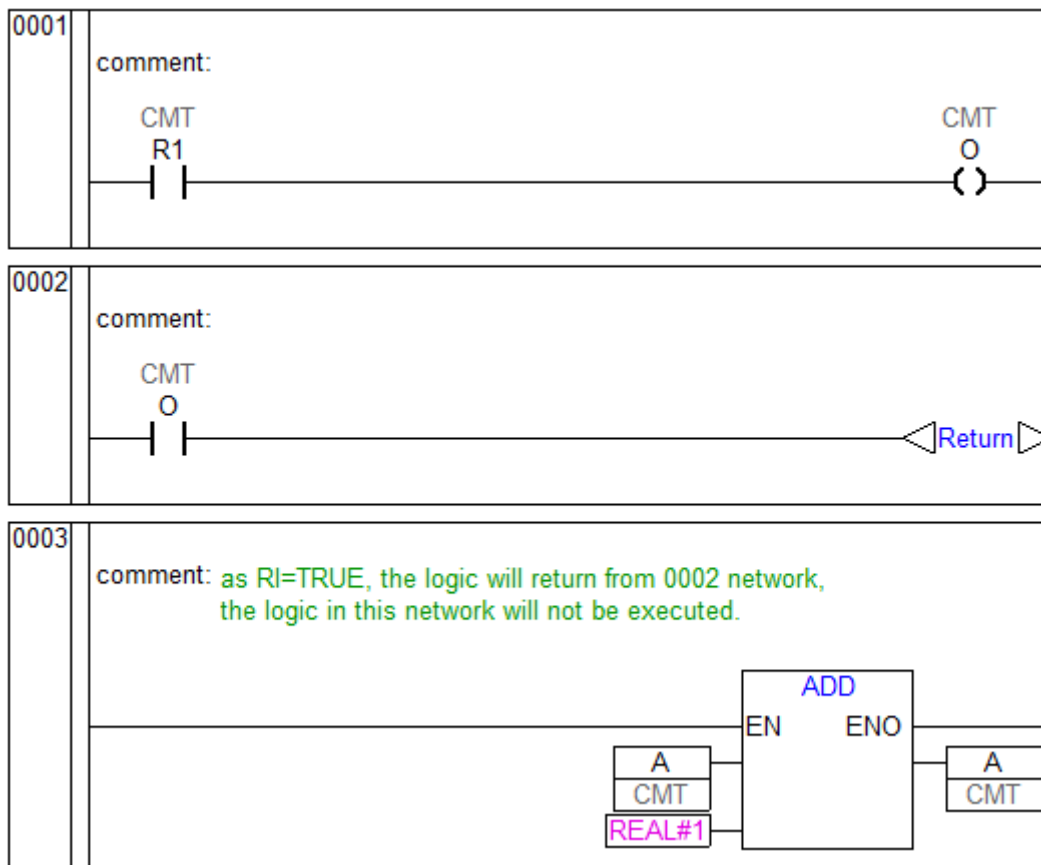


Figure 150 the applied example of return



- If a block or coil has been added to a Network, then the return element is not allowed.

10.2.12 Move the element

Element in the POU of LD can be moved to a different position through dragging.

Select a desired element (contact, coil, function block), press the mouse and drag it to leave the current position, all positions where the POU can move to indicated by a blue rectangle  , move the element to one (display ) of these positions and release the mouse key, and insert the element to a new position. It is as shown in Figure 151.

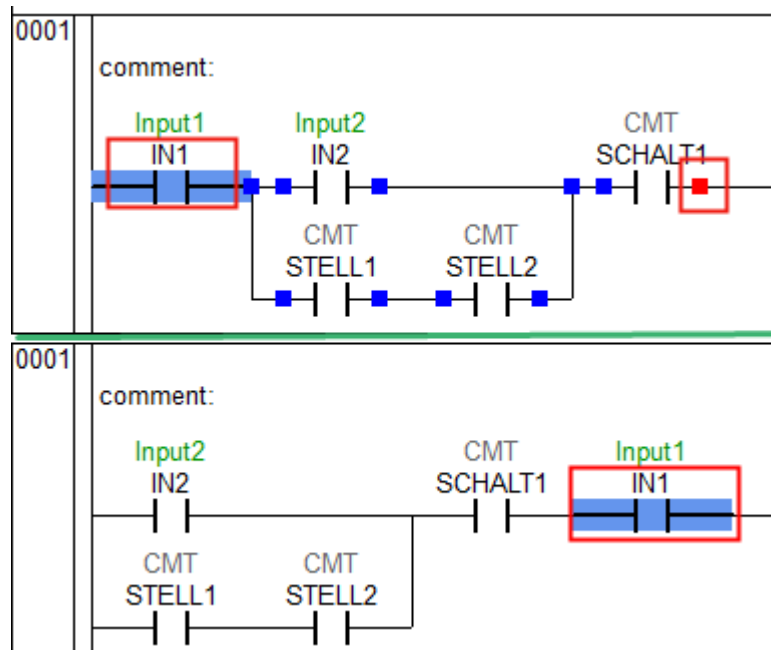


Figure 151 Move the element

Chapter 11 Controller operation

11.1 Controller operation

Controller operation includes [Controller information], [Controller lock], [Network configuration], [Project upgrading], [Firmware upgrading], [Log reading] and [IP scan].

11.1.1 Connect the controller

Before operating each tab, you must set the controller IP address and connect to the controller successfully. Otherwise, you will be prompted to set the controller IP when operating on each tab.

The setup interface of Controller IP is shown in Figure 152.

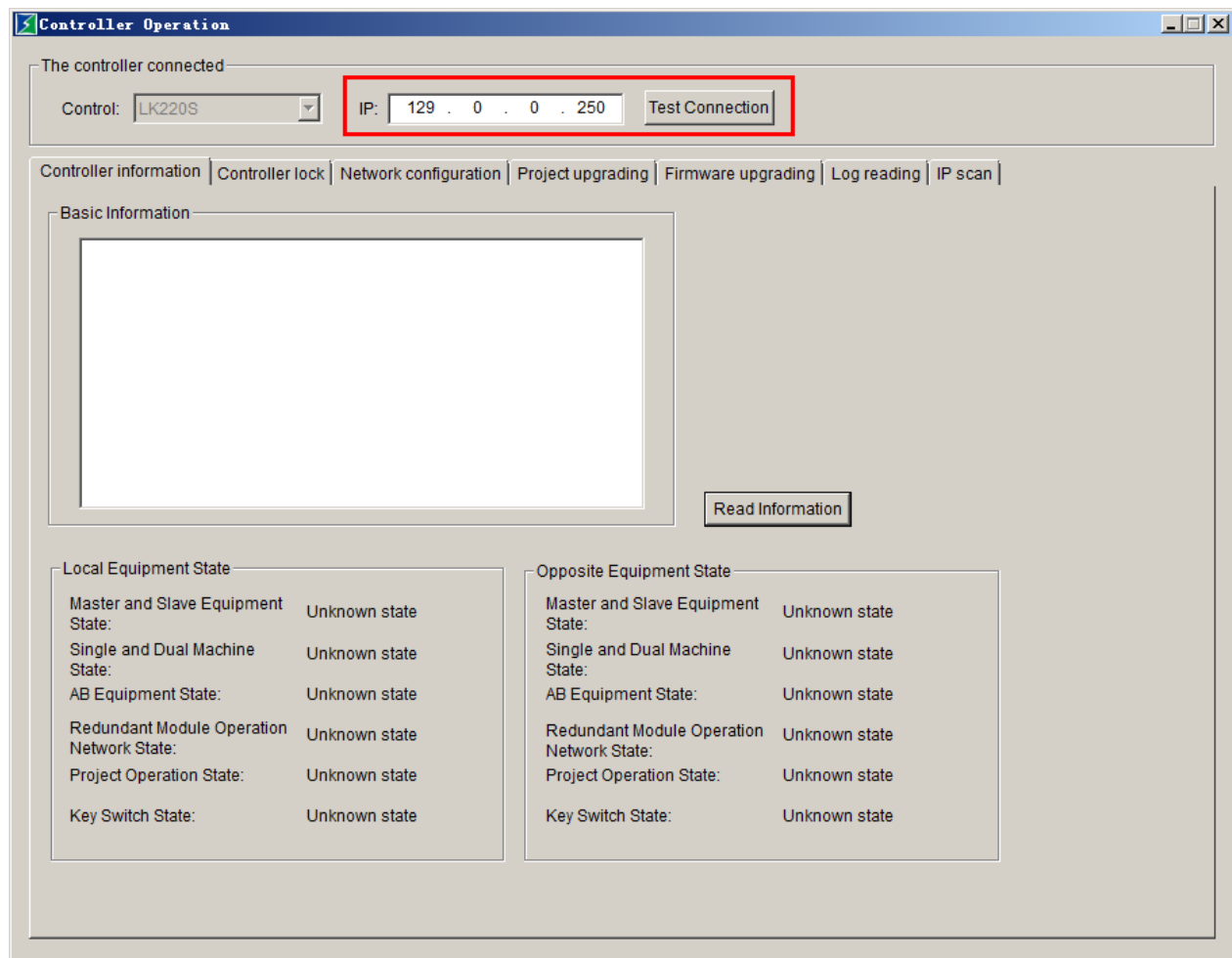


Figure 152 Controller IP tab

Input the IP address of the current controller in the IP address edit box, click the Test Connection button to pop up the dialog of the controller is connected successfully and click the **OK**, the lock state of current controller is shown. You can operate each tab after connecting controller.

11.1.2 Controller information

It can read the version information of controller in the [Controller information] tab, and it can obtain the master-slave status of the current controller.

In the [Controller information] tab, click Read Information button to obtain the version information of the current controller, which is displayed in the white window. A prompt box of Read version information successfully! pops up at the same time, as shown in Figure 153.

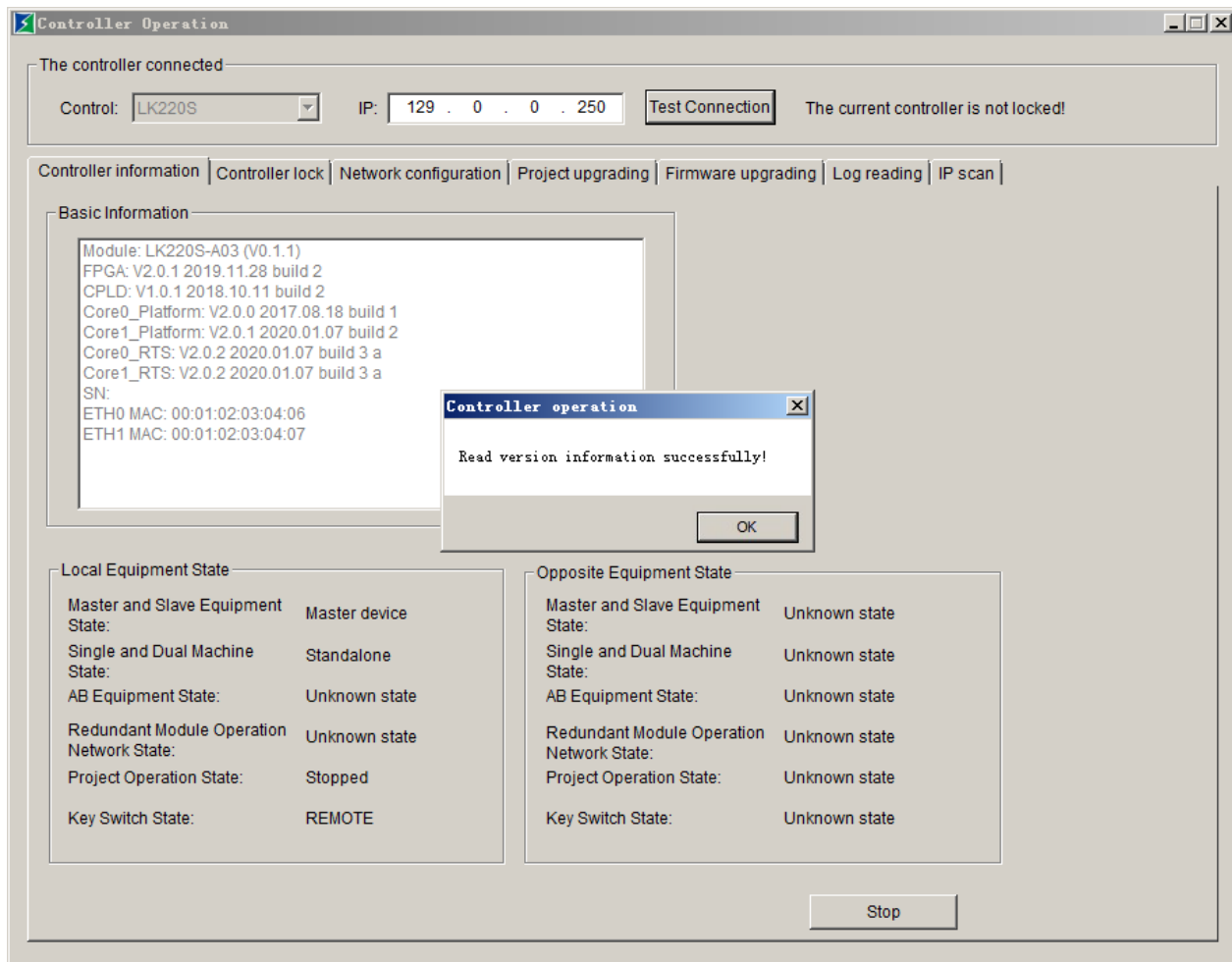


Figure 153 Read controller version information

It can display the redundancy status information of controllers in the [Local Equipment State] and [Opposite Equipment State] fields. The displayed status information refers to Table 18.

Table 18 Instructions to redundancy status information on controllers

Controller Status	Status Information	Description
Master and Slave Equipment State	Unknown state Initial state Hardware Ready state Dual device Ready state Standalone device Ready state Fault state	When the master-slave status of the local machine is Unqualified Master Machine, it can be cleared by clicking Clear.

Controller Status	Status Information	Description
	Error state Project redundancy Project verification Master device Slave device Unqualified master device	
Single and Dual-Machine State	Unknown state Standalone device Dual device	If [Local Status] is single-machine, then [Remote Status] is unknown
A/B Equipment State	Unknown state Device A Device B	Indicate whether the current machine is Machines A or B
Redundant Module Operation Network State	Unknown state First optical fiber Second optical fiber	The status value indicates whether the current working net is Nets A or B
Project Operation State	Unknown state Running Stopped Paused	Indicate the running status of a project
Key Switch Status	Unknown state RUN REMOTE PRG	Indicate the key switch status of the current Machines A or B

11.1.3 Controller lock

In [Controller lock] tab, implement locking and unlocking to current controller.

By locking the controller, it is forbidden to operate the controller when it is in offline state, including to update the controller, download the user file, download, upload user project, download user project, monitor, clear the controller, modify IP, project upgrading, firmware upgrading.

The Controller Lock window is shown in Figure 154.

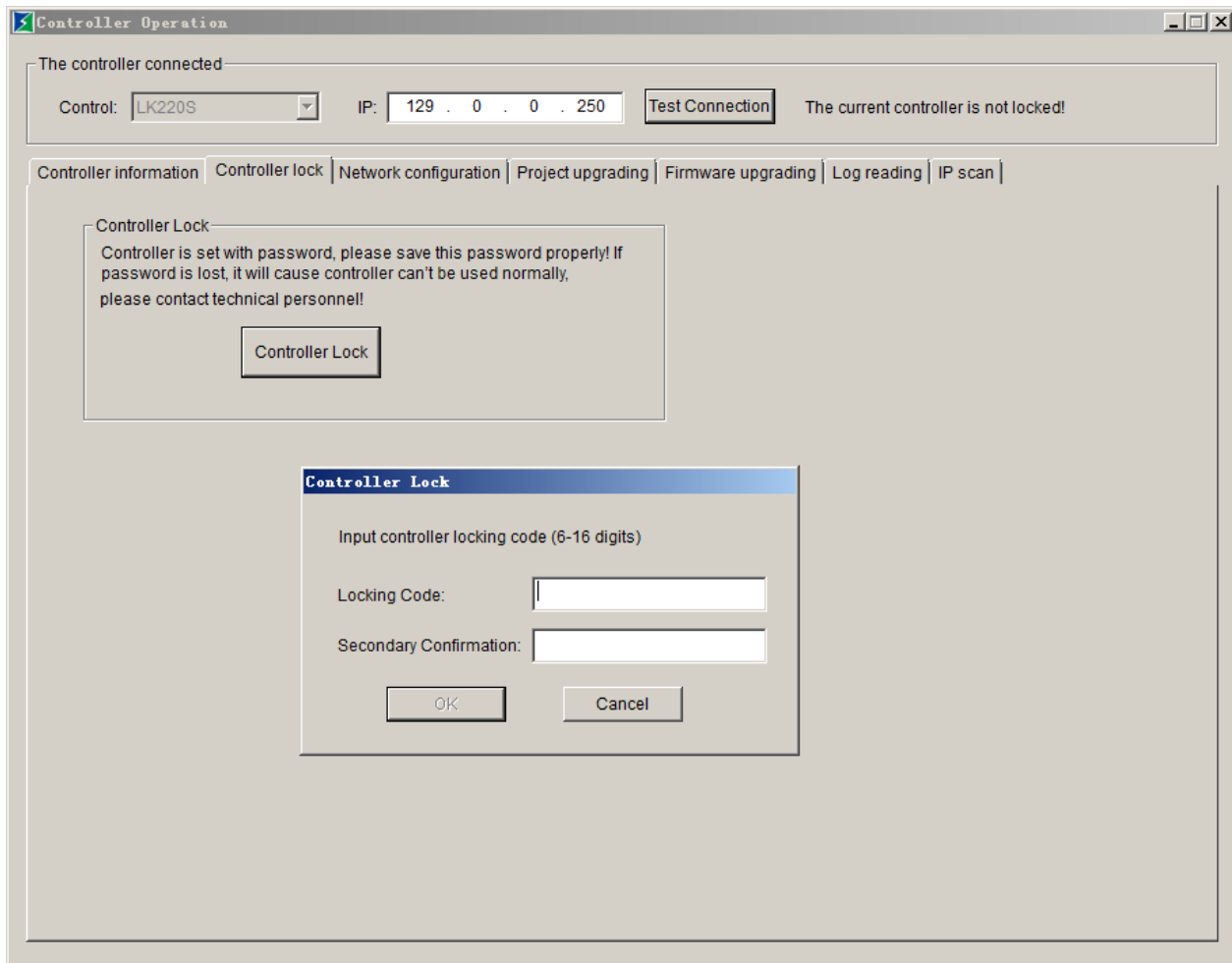


Figure 154 Controller lock

Click Controller Lock to pop up the password input dialog box of Controller Lock. After inputting the same 6-16-bit digital Locking Code twice, click OK to complete controller locking and the system prompts that the controller is locked successfully.

When operating the controller after it is locked, the operation cannot be done, with a message prompting the controller is locked currently, please unlock it and try again.

By click Controller Lock again after locking to unload the controller. The Unlock dialog box is shown in Figure 155. Input the previously set Locking Code, click **OK**. The controller is unlocked and the system prompts that the controller is unlocked successfully.

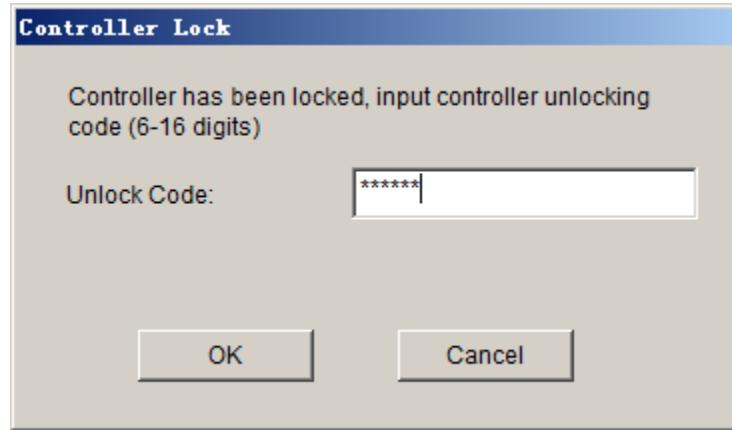


Figure 155 Unlock controller

11.1.4 Modify Controller IP

Read the controller IP, modify it, and then write IP to the controller. Before reading, please ensure that the connection between PC and controller is well.

Before modifying IP, the controller must be in single-frame mode, and the project has stopped.

Step1. Read IP

Click the IP Reading button to read the IP Addresses and Mask Code from the controller.

Step2. Modify IP

Check the Ethernet port, enter the new IP Address and Mask Code, click the IP Modification button. When the controller is redundant, set IP addresses of A and B controllers to the same value. When the redundant master-slave is established, the IP addresses of the master controller are set to setting value, and the IP addresses of the slave controller are set to setting value plus 1. For example, the IP addresses of two ports in A and B controllers are set to 128.0.0.250 and 129.0.0.250. When redundant master-slave is established, the IP addresses of the master controller are 128.0.0.250 and 129.0.0.250, and the IP addresses of the slave controller are 128.0.0.251 and 129.0.0.251. If master is switched to slave, the IP addresses of the new master controller are 128.0.0.250 and 129.0.0.250, and the IP addresses of the slave controller are 128.0.0.251 and 129.0.0.251.



- LKS system, when the controller is redundant, the fourth field in the IP address of the master controller can't be set to 254, otherwise, it can't communicate with the slave controller.
- LKS system, 128.0.0.254 and 129.0.0.254 are the reserved addresses of the controller. Do not use them.
- In the same network segment, IP addresses of multi-sets controllers cannot be continuous.

Step3. Restart controller

After modifying IP, you need to restart controller to take effect.

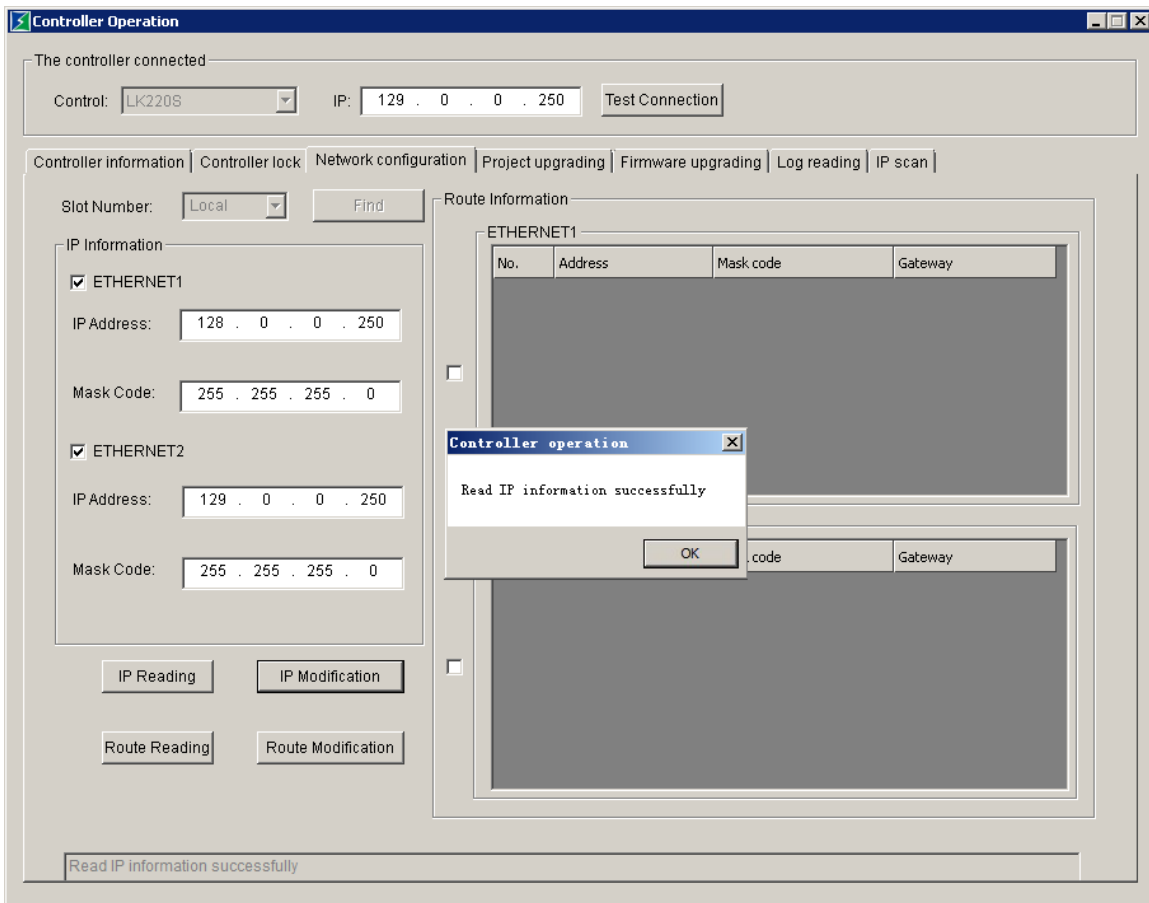


Figure 156 Network Configuration Dialog Box

11.1.5 Configure Route

Before configuring, please ensure that the connection between PC and controller is well.

11.1.5.1 Add Route

Configure route information for the controller network ports.

Before adding route, the controller must be in single-frame mode, and the project has stopped.

Step1. Check the Ethernet port you need to configure.

Step2. Right-click the route table area, select [Add] command, enter IP address, subnet mask, gateway in Add dialog, and click **OK**.

Step3. Click the **Route Modification** button.

Step4. Restart the controller and the route takes effect.

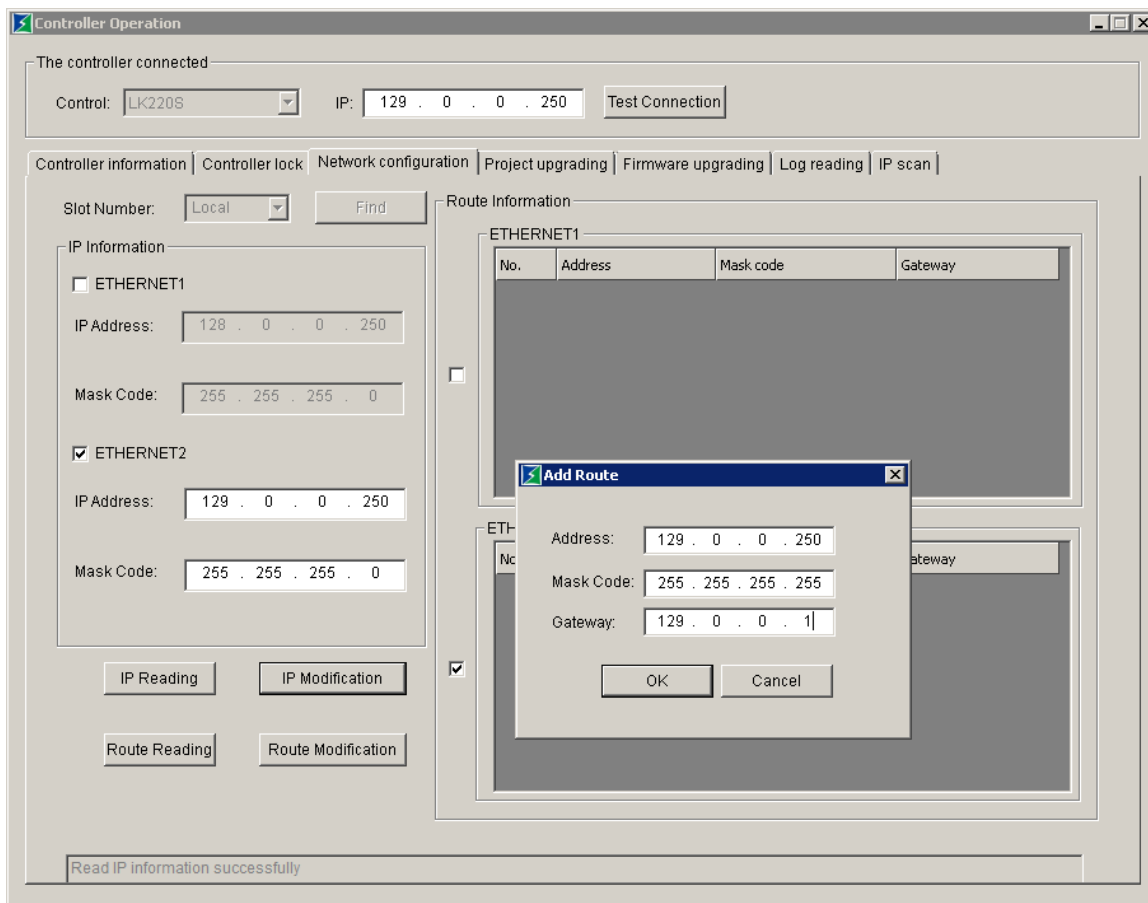


Figure 157 Add a Router

ETHERNET2			
No.	Address	Mask code	Gateway
0001	129.0.0.250	255.255.255.255	129.0.0.1
0002	128.0.0.250	255.255.255.255	128.0.0.1

Figure 158 Added Routing Message

11.1.5.2 Modify Route

Step1. Read route

Click the **Route Reading** button to read route information of the two Ethernet ports, including IP address, subnet mask, gateway.

Step2. Modify route

Check the Ethernet port, enter the new route information, and click the **Route Modification** button.

Step3. Restart controller

After modifying route, you need to restart controller to take effect.



- The route setting rules are the same as PC routing settings.

11.1.6 Project upgrading

11.1.6.1 Overview

If there is no source project for you to upgrade the project of controller, you can transfer the compiled project file to the controller through the auxiliary tool for updating.

During project is upgrading, the ".at" project compilation file and ".SDB" system file are transferred to the controller. Other files that need to be provided at the same time shall be determined according to the actual configuration of the project.

Project upgrading rules:

- During project is upgrading, the ".SDB" system file is required. Otherwise, project upgrading cannot be performed.
- If the SOE function is configured in the project, the ".soe" file is required during the upgrading, otherwise, the SOE function and NTP function will not take effect.
- If the Modbus function is configured in the project, the ".Mod" file is required during upgrading, otherwise, Modbus communication cannot work normally.

".at" and ".soe" files are generated by full compilation. ".SDB" and ".Mod" files are generated by downloading. When upgrading, please copy the corresponding files.

11.1.6.2 Requirements

- Prepare ".at" and ".SDB" files
- The auxiliary tool has been connected to the controller
- Controller cannot be in monitoring state (by software or auxiliary tool)
- Task is stopping

11.1.6.3 Steps

To upgrade your project, follow the following steps:

1. In the "Project upgrading" window, click the button .
2. Select the ".at" file and click the **Upgrade** button.

The upgrade confirmation prompt box will pop up.

3. Click **Yes**.

Start to upload the ".at" file to the controller. After the transfer, a confirmation prompt box will pop up.

4. Click **OK**.

11.1.7 Firmware upgrading

The tab realizes firmware upgrading and firmware file backup of controller.

The key switch is set to PRG or REM mode, and IEC is stopping when you upgrade firmware.

11.1.7.1 Controller firmware upgrade

1. Choose the firmware file

Click  in the [Upgrade] tab to select the .bin file of controller firmware, click **Upgrade** to pop up a prompt box of confirming upgrading, click **YES**, with the firmware file uploaded to the controller.

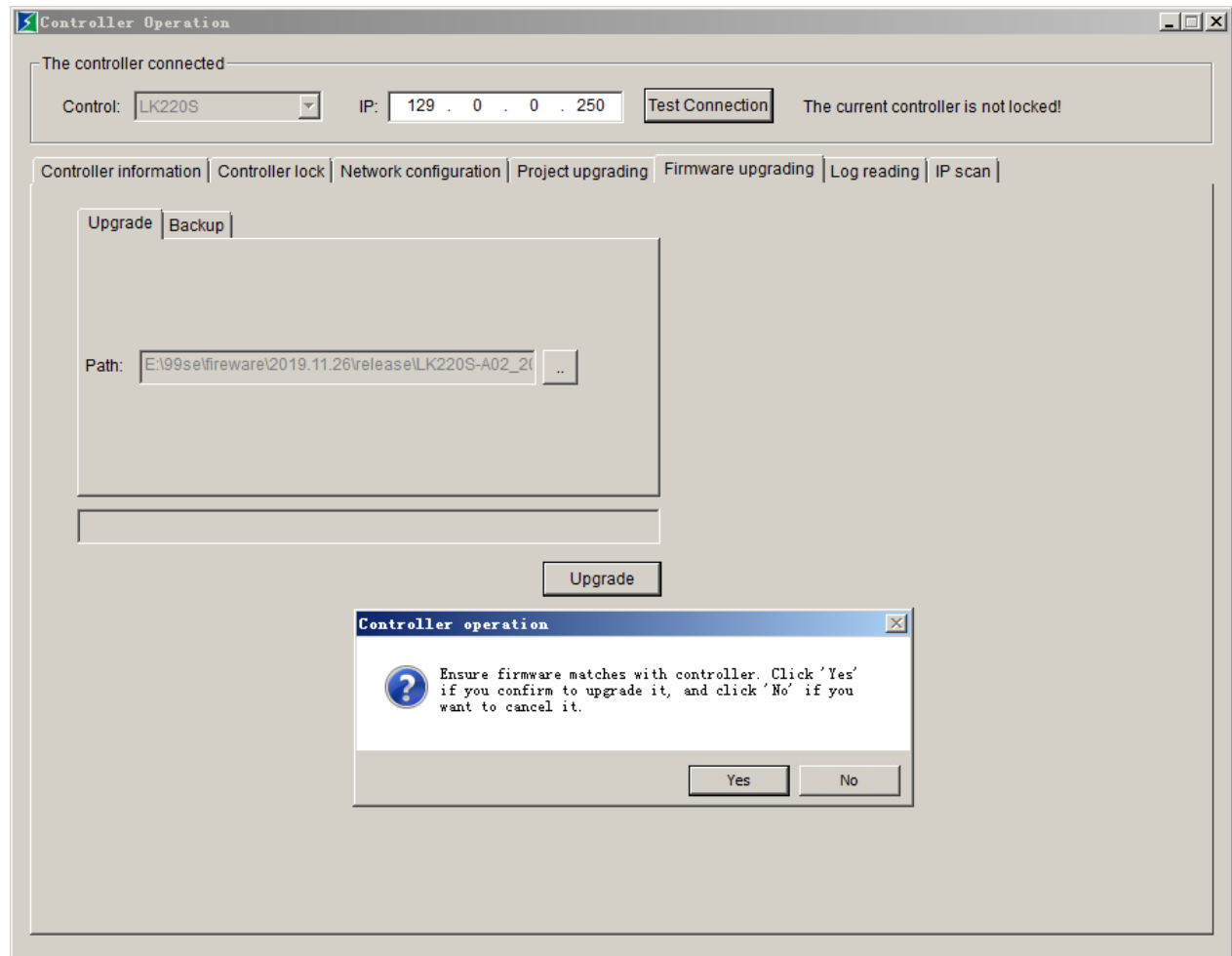


Figure 159 Firmware upgrade dialog

2. Start upgrading

The controller is updated after receiving the firmware file, with a prompt box as shown in Figure 160 popped up. Click OK to wait for the completion of controller update.

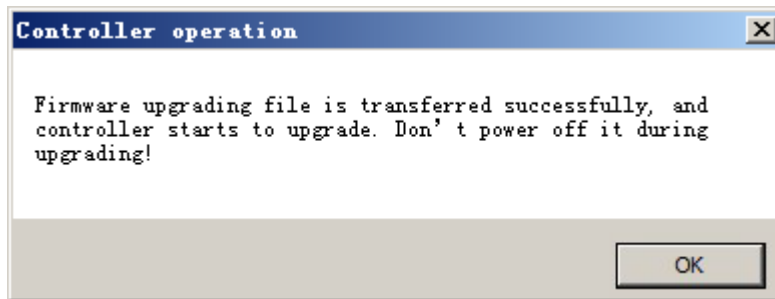


Figure 160 Prompt dialog of firmware file transmission

3. Confirm firmware version

After upgrading, you can check whether the new firmware version is correct by reading the controller information.

11.1.7.2 Backup firmware files

In the [Backup] tab, you can backup the firmware file in the controller to the local path, which can help the professionals analyze the problems.

Click  in the Path field, select the backup path, click **Backup** to complete the operation.

11.1.8 Log reading

Via the tab, you can read the log file in the controller, save it to the local path to help the professionals to analyze the problems.

Click  in the Path field to select the path, click Read to download the log files of the controllers to the local backup path, with a prompt box popped up when read successfully.

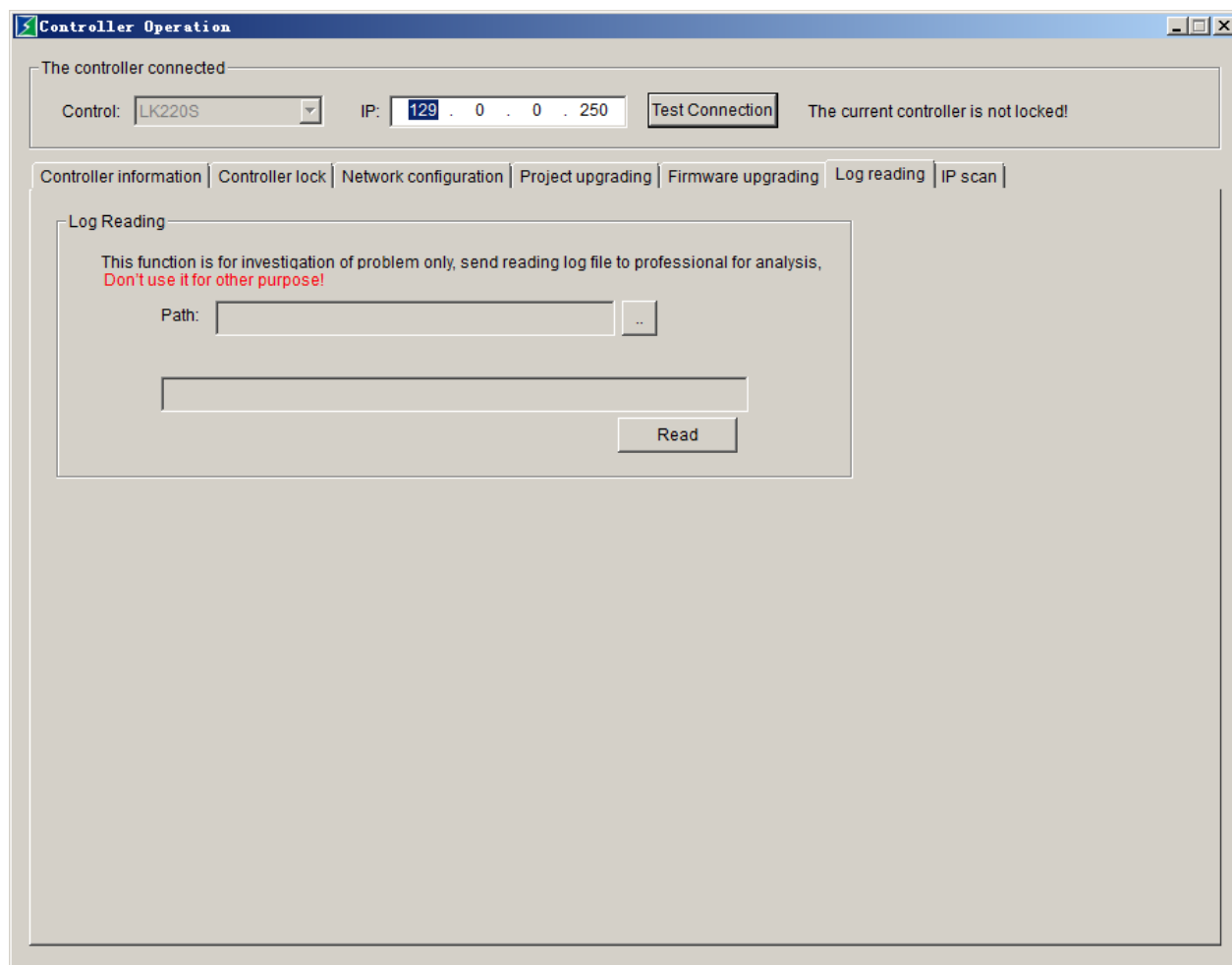


Figure 161 Read logs dialog

11.1.9 Scan IP

11.1.9.1 Scan IP

1. Concept

Through the IP scanning function of the Assistant Tool, you can find the IP addresses of all controllers in the LAN. It should be noted that IP scanning does not support routing. If the router is connected to the LAN, the devices connected by the router cannot be scanned.

When devices have been scanned with the same IP address, you can identify a device through MAC address. Then modify IP address and subnet mask to solve IP duplicate name.

If the PC has multiple network adapters, you need to check the network adapter model connected to network before scanning, and configure it in the Assistant Tool to ensure that the communication port is correct.

2. Requirements

- Network adapter model has been confirmed
- "Controller Operation" window has been open

3. Steps

To scan IP address, follow these steps:

Step 1. Click the "IP scan" tab.

Step 2. Select network adapter model in "Communication interface" drop-down box.

Step 3. Click the **Find CPU** button.

Start scanning, and the scanned controller and IP address will be displayed in the window.

After all controllers are found, prompt that the scan is complete.

Step 4. Click the **OK** button.

4. Next step

After scanning, you can select the IP address to detect the controller it belongs to, or set the IP address and station name.

5. Reference

[11.1.9.2 Modify IP](#)

[11.1.9.3 Test Controller](#)

[11.1.9.4 Rename](#)

11.1.9.2 Modify IP

1. Concept

You can select the scanned IP address to reset.

When the controller is redundant configuration, set IP addresses of A and B controllers to the same value. When the redundant master-slave is established, the IP addresses of the master controller are set to setting value, and the IP addresses of the slave controller are set to setting value plus 1.

2. Requirements

- "Controller Operation" window has been open
- IP address has been read

3. Steps

To set IP address, follow these steps:

Step 1. Select the IP address in the "IP scan" tab.

Step 2. Click the **Edit** button.

The IP address column becomes editable.

Step 3. Enter the IP address and subnet mask.

Step 4. Click the **Send** button.

A note dialog box will pop up here.

Step 5. Click the **Yes** button.

The controller IP address has been modified. A confirmation dialog box will pop up.

Step 6. Click the **OK** button.

11.1.9.3 Test Controller

1. Concept

You can identify the position of the controller by lighting the indicators. When lighting the indicators, the controller is identified by MAC address.

2. Requirements

- "Controller Operation" window has been open
- IP address has been read

3. Steps

To test controller, follow these steps:

Step 1. Select the IP address in the "IP scan" tab.

Step 2. Click the **Flash indicator** button.

At this time, the RUN ,ERR, FRC and BAT indicators of the controller are flashing at the same time, and stop flashing automatically after 30s.

You can also click the **Turn Off** button to stop flashing after confirming the controller.

11.1.9.4 Rename device

1. Concept

You can rename the scanned controller in the LAN.

2. Requirements

- "Controller Operation" window has been open
- The controller has been found

3. Steps

To modify controller name, follow these steps:

Step 1. Select the IP address in the "IP scan" tab.

Step 2. Click the **Edit** button.

The station name column becomes editable.

Step 3. Enter a new name.

- The name is character combination of a ~ z, A ~ Z, 0 ~ 9, "-", "." .
- The name cannot be longer than 20 characters.

Step 4. Click the **Send** button.

The name has been modified.

11.2 Reset



- Menu bar: Click [Online] - [Reset].

Reset is used to simulate the controller's status after the power-down and then power on, the controller should restore the variable value in the power-loss retentive memory to the value before power-down, and the variable value in other memory should be restored as the initial value.

11.3 Cold reset



- Menu bar: Click [Online] - [Cold Reset].

Cold reset is used to simulate the status after the project downloading, after the controller downloads the program, the data in each memory is restored to the initial value through cold reset function.

11.4 Warm reset



- Menu bar: Click [Online] - [Warm Reset].

Warm reset is used to simulate the status that the data in each memory is unchanged after restarting project. You can reload the user project through the warm reset and the data in each memory remains unchanged.

11.5 Clear controller

It shall be effective in the off-line mode, clear the project in PLC, re-initialize the PLC system.



- Menu bar: Click [Online] - [Clear Controller].

This function is used to clear files in the directory specified by the controller, and the popped up dialog box is as shown in Figure 162.

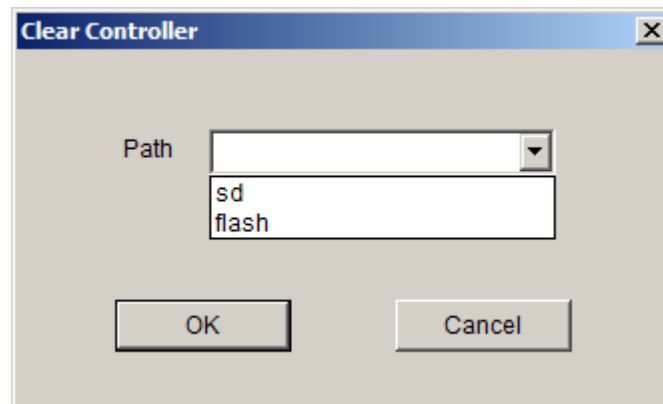


Figure 162 Clear controller

- Select the SD path, then all files in SD will be deleted, and can not be restored.
- Select the Flash path, then running program, user program and user files will be deleted, and can't be restored.

Choose the path, and click **OK**, confirmation information will be shown.



- It is required to reset or restart the controller after clearing the controller.

11.6 Controller calibration time



- Menu bar: Click [Online] - [Controller calibration time of the].

This feature is available in the online status. In the pop-up window, the current time of the controller is shown in the above, and below is the setting of new time of the controller. The new time can be entered manually, and then click the **Setting** button to take effect. Or directly click on the button of **Synchronous PC Clock**, set the controller time as the PC time. The interface of time calibration of the controller is as shown in Figure 163.

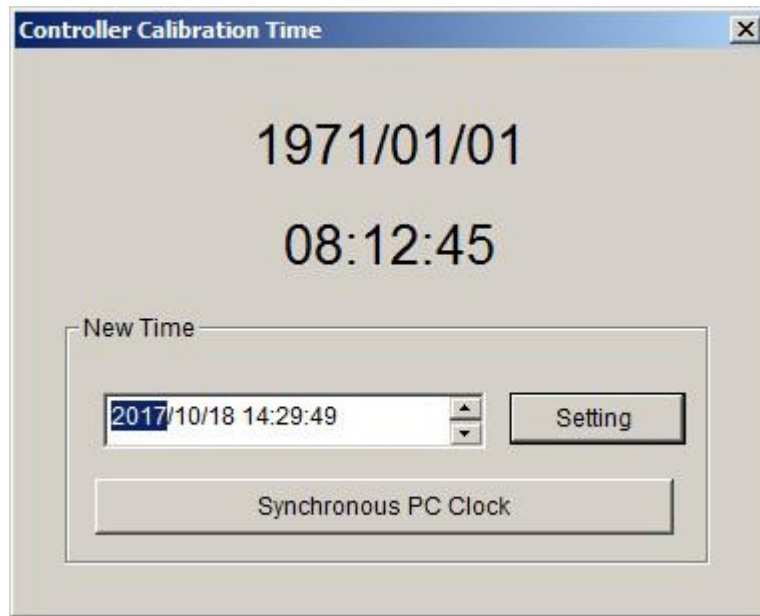


Figure 163 Time calibration of the controller

Time range supported:

- ☐ Year: 1971~2036
- ☐ Month: 1~12
- ☐ Date: 1~31
- ☐ Hour: 0~23
- ☐ Minute: 0~59
- ☐ Second: 0~59

Chapter 12 Compile

After the project configuration, you should compile it, whose purpose is to check whether there are some errors in configuration and to prove program correctness and perform ability. The programs configured will be converting into files suit for downloading; the compilation result will be displayed in information window.

Compilation includes two ways: full compilation and incremental compilation.

12.1 Incremental compilation

Only compile the modified and added logics. Following operations will cause the incremental compilation:

- Modify the POU (Add or delete POU, modify existing POU logic).
- Modify the variables (Add or delete variables, modify the variable definition).
- Modify the task properties (Add the POU to call, modify task properties).

Incremental compilation is performed in the following ways:



- Menu bar: click [Project] - [Compile];
- Toolbar: .
- Shortcut key: **F11**.

12.2 Full compilation

During the full compilation, the hardware configuration information shall be regenerated, all variables shall be reassigned, POU shall be re-translated, and generate new configuration logic information and hardware configuration information. Following operations will cause the full compilation:

- First compilation of new project.
- The .iec or .tmp files are incomplete or corrupted during compiling.
- Open the old project to cause upgrading of the TrgVersion file.
- The .iec or .tmp file is incomplete when the controller is downloaded.
- Execute [full compilation] command.
- The hardware configuration is changed, perform the [Compile].

Full compilation is performed in the following ways:



- Menu bar: click [Project] - [Full compile];
- Toolbar: .

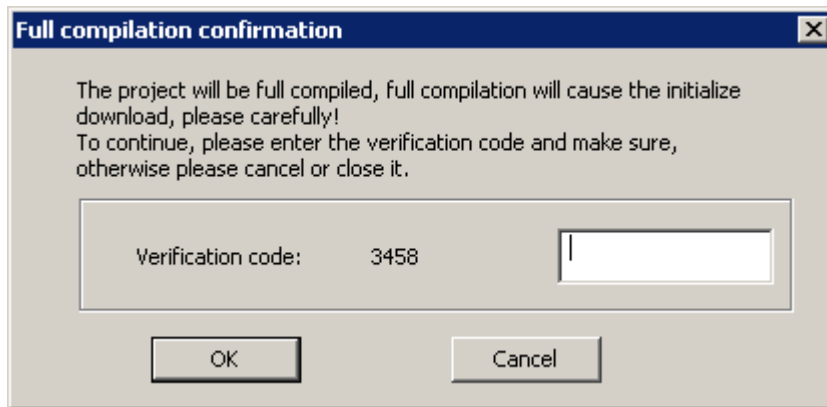


Figure 164 Confirmation for Full Compilation

12.3 Compilation result

The compilation result is displayed in the information window, the red font indicates the compilation error information, double-click to locate the error.

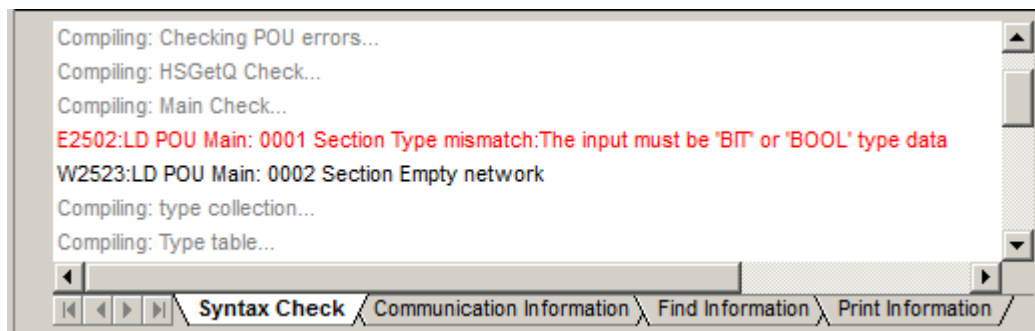


Figure 165 Compilation Results

12.4 Analysis for compiled information

1. Must be BIT or BOOL type data...

Reason: the project type of Negate, Set/Reset and specific elements (Jump Element) and some input in LD language must be BIT or BOOL.

Example: edit program like , in which w is DINT type, error as **Type mismatch:The input must be 'BIT' or 'BOOL' type data** will be displayed while compiling.

2. Undefined identifier ...

Represents a variable that is not defined, and needs to be declared.

3. * is not a member of *

Analysis: use some variables that not belong to the specific type

Example: edit a program as shown in Figure 166, the EXAMPLE block contains these members: a, g and w. But the input variable below use a variable e which isn't the member of EXAMPLE, so error shown as **Related data:'e' is not a member of type 'EXAMPLE'** will be displayed while compiling.

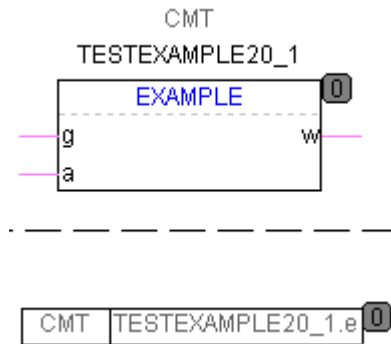


Figure 166 Example for hint

4. Only ... variable can be followed by ...

Analysis: specific symbol such as . and [can be used only after specific types of variables, pay attention please.

5. Jump location (or Label) is null.

Analysis: the jump location and label must be identified clearly.

Example: add a Label element 0 in program area, error shown as

Related data:Empty label will be displayed while compiling.

6. ... must not following with...

Analysis: some symbols, constants or variables can't be followed with other information or specific symbols.

Example: Take CMT 5TES 0 for example, error

Related data:Nothing is expected after constant '5' will be displayed while compiling.

7. ...out of range...

(Warning) Access an array element using variable subscript, could lead to array bounds and cause controller anomaly. It is strongly recommended to add array protection logic in the program!

Analysis: Pay attention to the scope of each data type and array subscript while programing. When programming, you can use a constant subscript or variable subscript to access the array element, and either kind of subscript may cause array bounds. If the array subscript is a constant and the constant value is not in bounds, Safety FA-AutoThink will give an ERROR message

Related data:'11' is out of the array index range when compiling. If the array index is a variable, when compiling Safety FA-AutoThink will give a WARNING message.

Access an array element using variable subscript, could lead to array bounds and cause controller anomaly. It is strongly recommended to add an array protection logic in the program!

Please note the range of array subscript during programing.

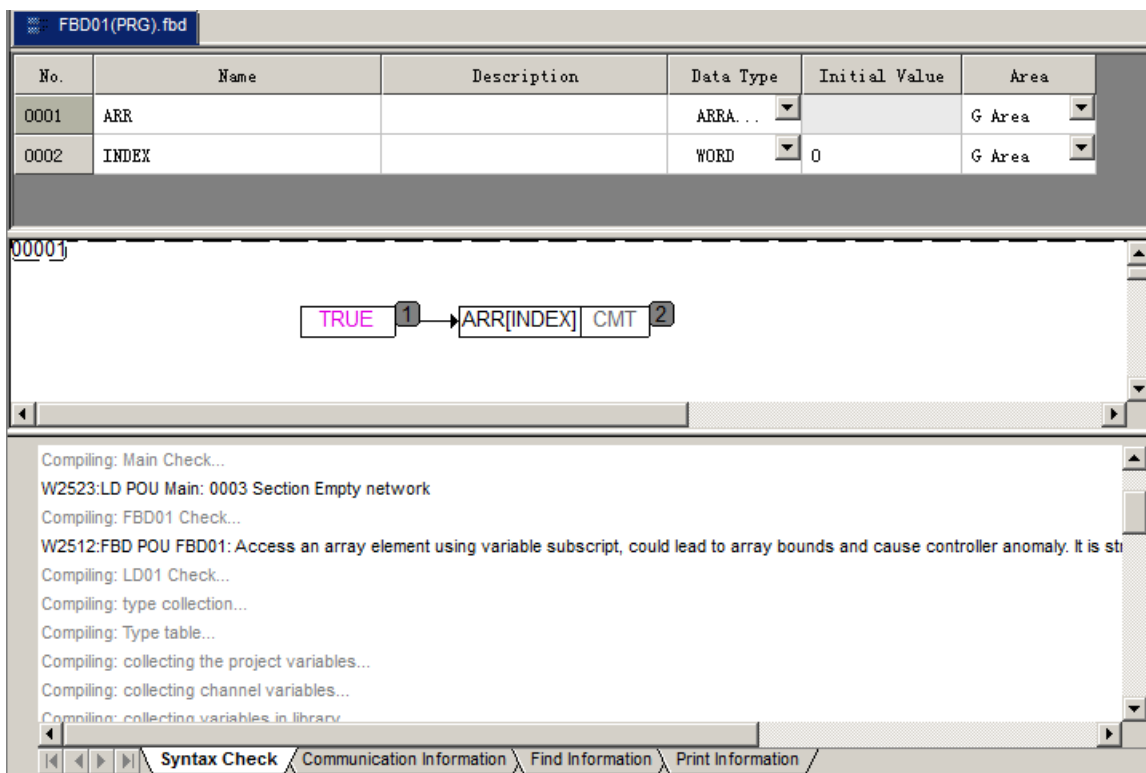


Figure 167 Example for warning

8. Lack of] to match [...] or only the unsigned integer data can be used as an array index

Analysis: [] should be used in pair, and array index must be INT or DINT.

Example: program like

CMT	g[1]	1
CMT	g[2.1]	0

 will cause these two errors

Related data: Lack of ']' to match '[' and Related data: Only the unsigned integer data can be used as an array index .

9. Cannot convert type to type

Analysis: variable types not match.

Example: in program

CMT	w	0	→	Fun01.IN	FB type variable	0
-----	---	---	---	----------	------------------	---

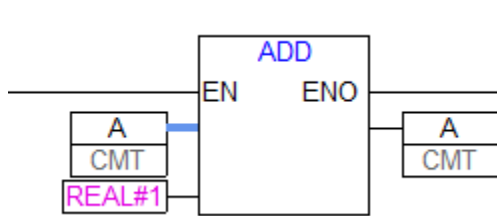
 , W is BOOL and

Fun01.IN is LREAL. While compiling, error shown as :Cannot convert 'BOOL' type to 'LREAL' type .

10. ...type cannot be participate in...operation

Analysis: meaningless symbol or constant will cause error. Some operations have special requirement for data type of input variables.

Example: edit a program shown as



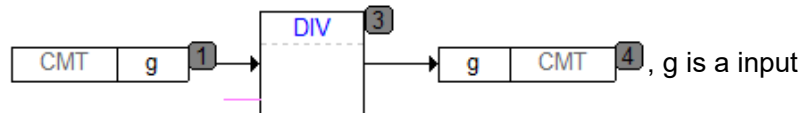
, in which A is BOOL type

data, error will be displayed as **Section Type mismatch: 'BOOL' type cannot be participate in 'ADD' operation**.

11. Read-only data ...cannot be assigned or lack of data

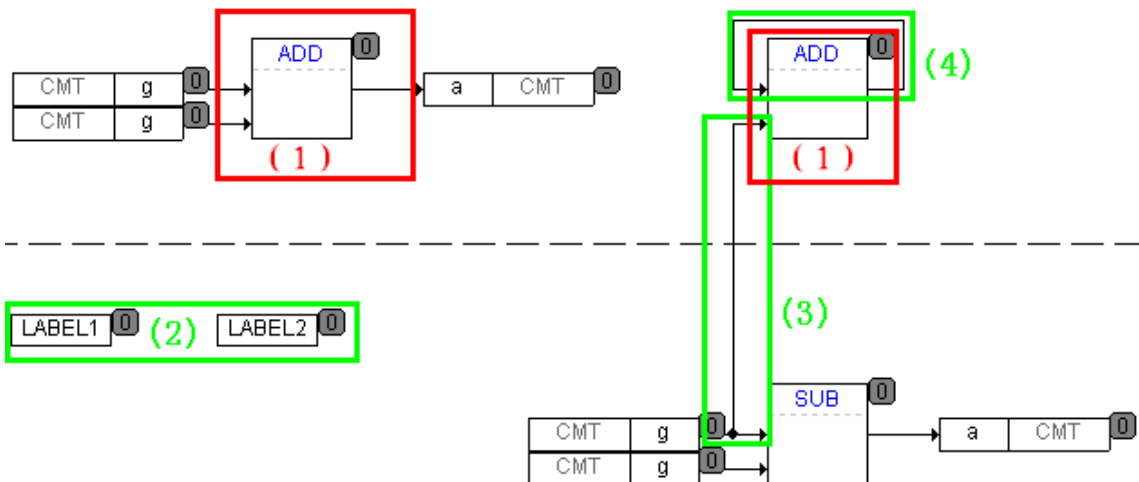
Analysis: Input variable is only-data, you can't write value to variable.

Example: in the function block



variable and other pin is null. While compiling, errors are shown as **Type mismatch:Lack of data** and **Read-only data 'g' cannot be assigned**.

12. The number of network logic tree cannot be more than one



13. The component is across network, block element does not support loop or null variable relates to...FB

Analysis: the FB does not relate to a variable.

12.5 Decompile

Decompile is used to ensure consistency of source project and actual one. After executing de-compilation, the system will show the comparison results of the original program and the decompiled program, and it will ask the user to check the correctness, completeness of the project, it is called as "Configuration Validation".

The de-compilation operation contains two steps:

1. Safety FA-AutoThink decompile

Safety FA-AutoThink decompiles program in the project to decompiled program and compares their consistency, then presents the result to users.

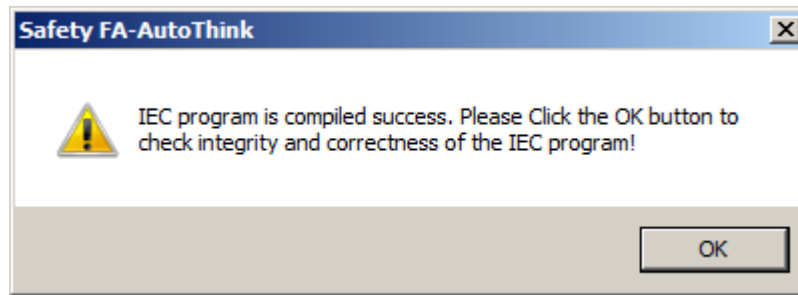


Figure 168 Check tip

When you got this box, please choose **OK** to the next step.

2. User conform

There are user programs sub-nodes which have been decompiled in Configuration logic validation program tree, double-click a user program sub-node to load its decompiled result and the original POU editor window.

In Figure 169, the original program shows at the left side in working area, while the corresponding decompiled program shows at the right side in working area. When you close the POU editor window, both of them will be closed.

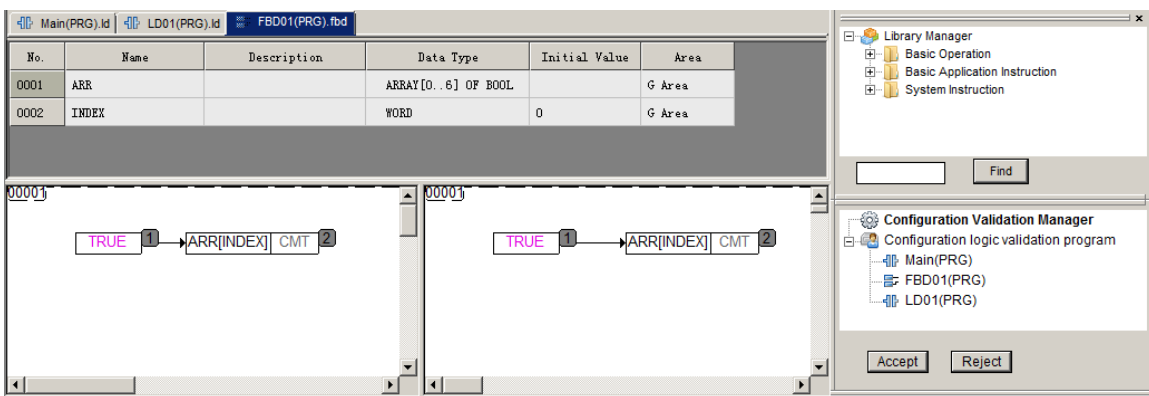


Figure 169 Comparison between the original program and the decompiled program

Accept: When the original program is consistent with the decompiled program, and you want to download, choose this button.

Reject: When the original program and decompiled program is inconsistent, or you want to abort the download, choose this button.

After compiling successfully, the user parameters information of module is read to confirm when downloading.



- There should be commissioning test by the user before project is put into use to detect errors caused by human or unsafe environment.

Chapter 13 Download

After compiling successfully, users can download the program to the controller. Before download, the connection between the engineer station and the controller should be set up.

13.1 Communication Settings

Click [Online] - [communication settings] command, it will display **Communication** Settings dialog box, as shown in Figure 170. Users can set IP address of current controller. Heartbeat time, reconnection time, port number use default value and cannot be modified. Click **OK** to complete communication settings.

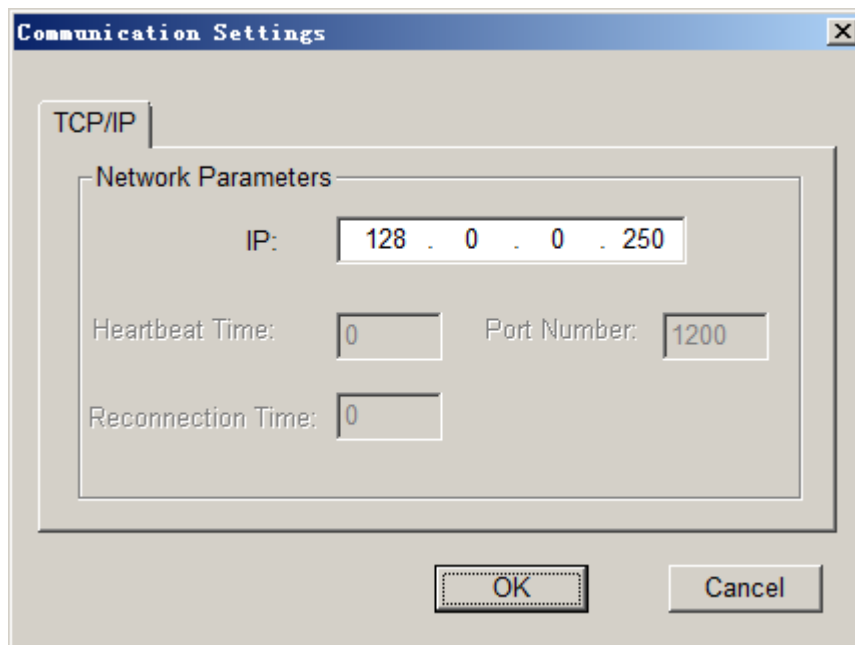


Figure 170 Communication parameters

13.2 Download

Download the project after compiling and communicate with controller successfully.

Please check the position of the dialing switch before downloading. If dialing switch is in the RUN state, it is forbidden to download.

Any of the following cases would lead to download.

- ☐ No project in the controller
- ☐ The project in the controller not consistent with that on the local machine
- ☐ Project compilation



- Menu bar: click [Online] - [Download];
- Toolbar: ;
- Shortcut key: **F5**.

After incremental compilation, click [Download], incremental download is performed, and after full compilation, click [Download], initialization download is performed.

Downloading, you need to verify that the read-back user parameters of module are correct, click "Yes", confirm to pass, and continue downloading. If you click "No", you need to check whether the user parameters are correctly configured and recompile without error.

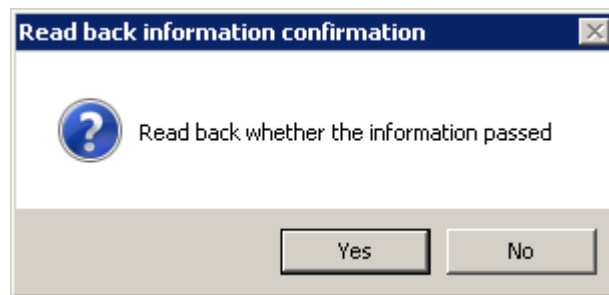


Figure 171 Confirm read-back user parameters

During initialization downloading, all safety IO is in safety-oriented side, when controller is running after downloading, all the IO outputs actual value. Please note the related information from the output window during the download.



- Before downloading, you need to ensure that no project file was opened, otherwise, backup file failed during downloading, at this time, users need to manually backup the project file.

Chapter 14 Debug

14.1 Simulation mode

Simulation Mode, which uses debug environment to simulate the logical correctness of user programs, provides a prerequisite of local online debugging and convenient modification and debugging of control logical. During the simulation, programs run in local simulation software.



- Menu bar: click [online] - [Simulation mode];
- Toolbar: .

After entering into simulation mode, then IEC program will auto-run, simulation interface displays just like the interface of monitoring in operation mode.

In the simulation mode, user can view the value of function block and member variable of array. Take function block for example: when double click the instance of function block, its value table will be displayed in which the content refreshed every 500ms according to the value in the local simulation software. As shown in Figure 172.

MAIN.MainHYSTERESIS_1					
No.	Name	Description	Data Type	Online Value	Area
0001	IN		INT	0	G Area
0002	High		INT	60	R Area
0003	Low		INT	30	R Area
0004	Out		BOOL	TRUE	R Area
0005	ERR		BOOL	FALSE	G Area

Figure 172 Variables' online value

14.2 Operation online

14.2.1 Write value

Write Values command is the operation which will modify the variable and channel online value. The following example shows how to write values.

After entering operation mode in a level0 user, the POU editor window is as shown in Figure 173. You can view the online value of simple variable directly in the window, while complex variable by double-clicking its variable name in variable area or its block in programming area, the online values of the elements of complex variable is as shown in Figure 174.

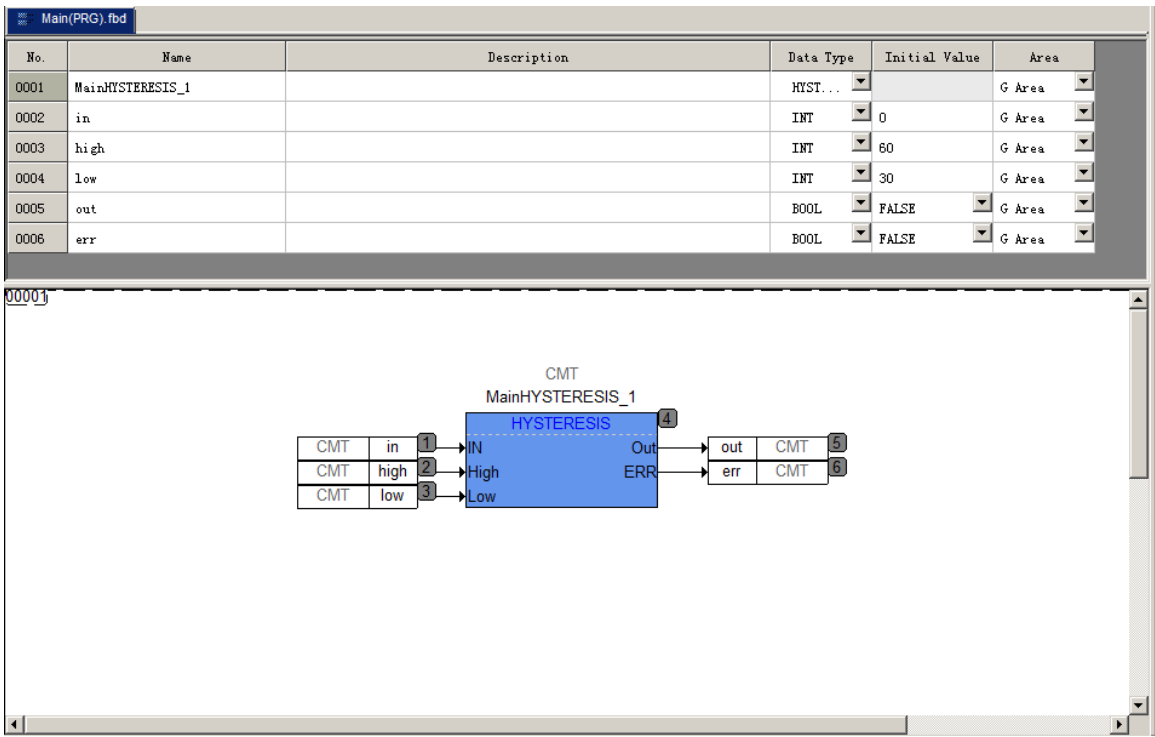


Figure 173 Operation mode

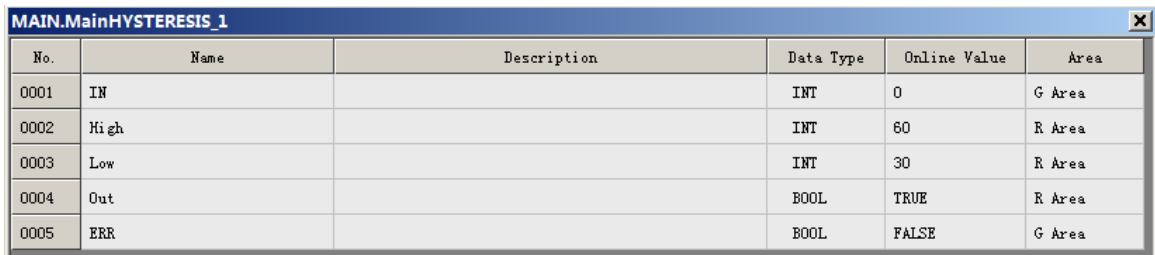


Figure 174 Function block variable's online value

14.2.1.1 Write value for single variable

Double-click the variable to be written, it will pop up Debug Variable dialog box, as shown in Figure 175. Current debugged variable name will be displayed in the field of Current Variable, while current value in the field of Current Variable Value. Input the target variable value in the field of Input Variable Value, and then click **Write** button to finish this operation. In this example, after click Write button, it will show *The online value of variable MAIN.mainHYSTERESIS_1.High is written as 100 successfully!* in Information Window, and the value will change to the new one.

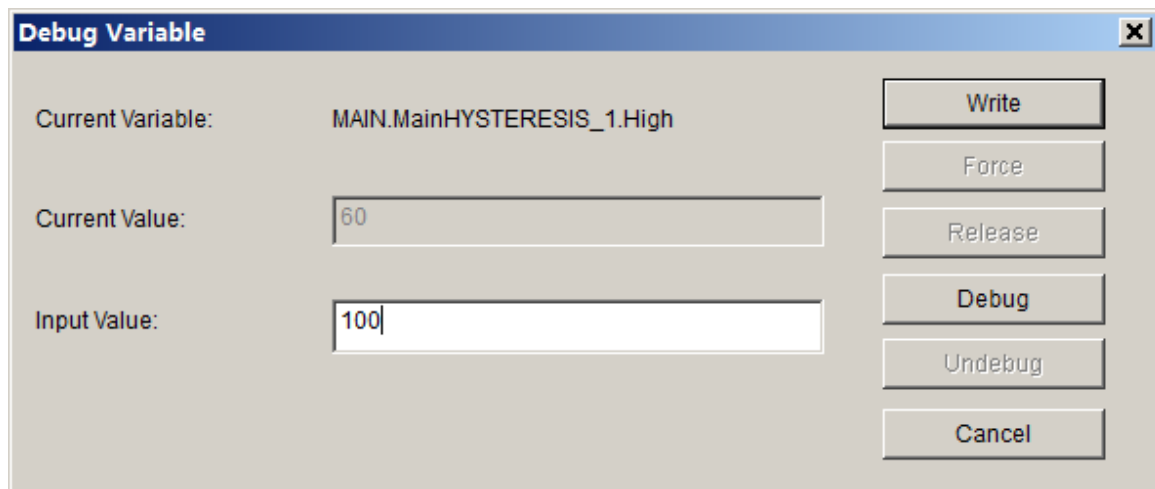


Figure 175 New value write

14.2.1.2 Write Value for multiple variables

Multiple variables written with a single variable write operation order and contents are basically the same, differs only in that, multiple variables written after the new value input, not directly select the [write] command, but choose to [Debug] command until all the variables of the new values are input into by write command makes to be debugged values come into effect at the same time.

Main.MainHYSTERESIS_1					
No.	Name	Description	Data Type	Online Value	Area
0001	IN		INT	0	G Area
0002	High		INT	60 < 100 >	R Area
0003	Low		INT	30 < 20 >	R Area
0004	Out		BOOL	TRUE	R Area
0005	ERR		BOOL	FALSE	G Area

Figure 176 Variable in the debug state

If you want to give up the debugging of a variable before selecting a write command, you can cancel its debug state by **Undebug** button.



- **Write** and **Debug** button will be disabled when users input illegal online value.
- The value changed by **Write** Values operation only works once, and will be changed according to the program running period.
- You write value to variable when controller is in RUN mode, information output window prompts that write value is not allowed in RUN status.

Chapter 15 Online monitor

After download, you can enter online monitor. The UI of online monitor is as follow:

Variable value change: after entering online monitor status, Safety FA-AutoThink will read the variable value from the controller periodically, and monitor the changing of value. The default monitoring period is 500ms, that is the value displayed in Safety FA-AutoThink is the value read from controller every 500ms. You can create surveillance list to monitor the values of some important variables centrally.

Hardware Configuration: you can monitor collected values in Hardware Configuration Window.

Program calculation: you can track the calculation status of programs in POU editor window.

15.1 Run



- Menu bar: click [Online] - [Run];
- Toolbar: .

In monitor status, select this command to enter the online mode. If the **Startup Type** of the task property is selected as **Auto**, the task can run automatically without clicking the Run button.

15.2 Stop



- Menu bar: click [Online] - [Stop].
- Toolbar: .

When the program operation is stopped, each variable will keep the operation value before the stop; when the program recovery operation, it will run the program from the beginning.

15.3 Pause

When the program operation is paused, each variable will keep the operation value before the pause; when the program recovery operation, it will run the program from the location where the program paused.



- Toolbar: .

15.4 Monitor



- Menu bar: click [Online] - [Monitor];
- Toolbar: .

Monitoring status of the entry and exit operation order exactly the same, you need to withdraw from the monitoring state; the command can be checked once.

15.4.1 Variable and program

Safety FA-AutoThink reads the variable value from the controller periodically, and monitors the changing of value. The variable value displayed in current opened POU editor window is changing, as shown in Figure 177; you can see the value changing in **Online Value** column of variable area and in variable area of components in programming area.

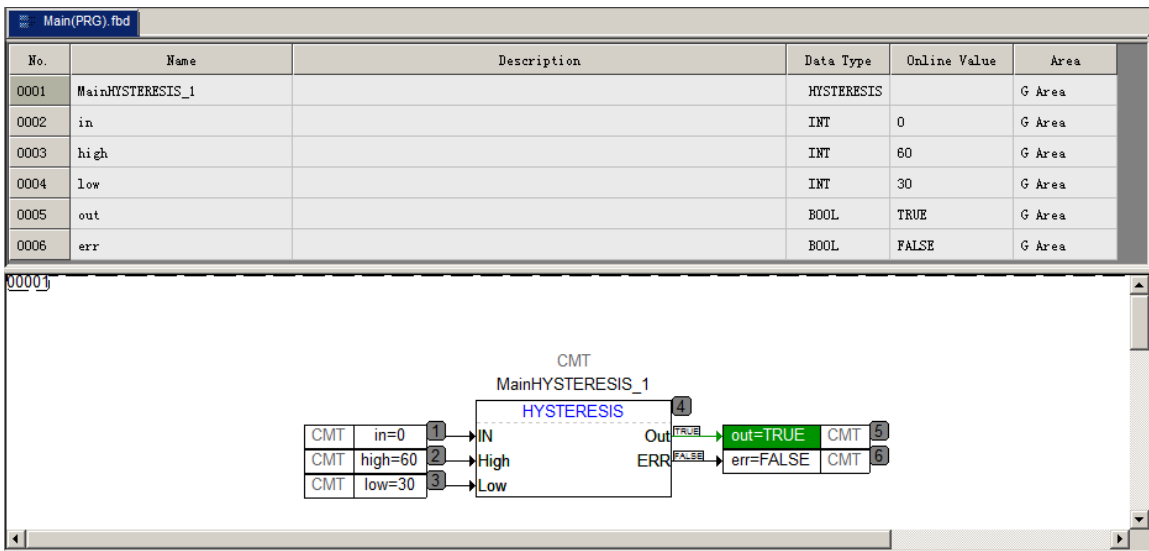


Figure 177 Monitoring function block instance

In monitor status, you can view changing of online values of the elements of Function Block or Array type. Here, take an instance of Function Block Type for example, double-click the instance of Function Block Type in programming area, it will pop up its elements table, as shown in Figure 178, the values displayed in Online Value column are refreshed every 500ms.

MAIN.MainHYSTERESIS_1					
No.	Name	Description	Data Type	Online Value	Area
0001	IN		INT	0	G Area
0002	High		INT	60	R Area
0003	Low		INT	30	R Area
0004	Out		BOOL	TRUE	R Area
0005	ERR		BOOL	FALSE	G Area

Figure 178 Online value of MainHYSTERESIS_1

If there are complex variables in the list, you can double - click the variable name to make the member variable information display.

15.4.2 Online Value and quality of channel

1~8 are online values of the channel, which are used to indicate real-time value of channel. 9~16 are quality bit of the channel, which are used to indicate that channel is good or not, and TRUE indicates channel is bad, and FALSE indicates channel is normal.

Channel Number	Channel Name	Channel Types	Online Value	Channel Address
1	DPIO_2_1_3_1	BOOL	FALSE	%IX16.0
2	DPIO_2_1_3_2	BOOL	FALSE	%IX16.1
3	DPIO_2_1_3_3	BOOL	FALSE	%IX16.2
4	DPIO_2_1_3_4	BOOL	FALSE	%IX16.3
5	DPIO_2_1_3_5	BOOL	FALSE	%IX16.4
6	DPIO_2_1_3_6	BOOL	FALSE	%IX16.5
7	DPIO_2_1_3_7	BOOL	FALSE	%IX16.6
8	DPIO_2_1_3_8	BOOL	FALSE	%IX16.7
9	Q_DPIO_2_1_3_1	BOOL	FALSE	%IX17.0
10	Q_DPIO_2_1_3_2	BOOL	FALSE	%IX17.1
11	Q_DPIO_2_1_3_3	BOOL	FALSE	%IX17.2
12	Q_DPIO_2_1_3_4	BOOL	FALSE	%IX17.3
13	Q_DPIO_2_1_3_5	BOOL	FALSE	%IX17.4
14	Q_DPIO_2_1_3_6	BOOL	FALSE	%IX17.5
15	Q_DPIO_2_1_3_7	BOOL	FALSE	%IX17.6
16	Q_DPIO_2_1_3_8	BOOL	FALSE	%IX17.7

Figure 179 Online value of channel parameters

15.5 View Operation Log

The operation log is used for recording all operations during debugging.



- Menu bar: click [Project] - [View Operation Log].

Online operation information, such as operation time, operation content, and recorded operation item, of the program can be recorded in the window. It is as shown in Figure 180.

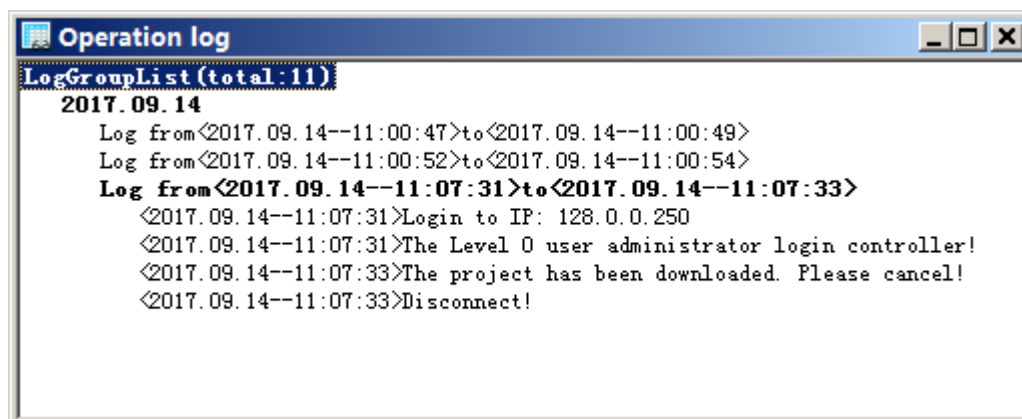


Figure 180 Operation log window

Each operation log file can store up to 5000 pieces of operation records. When the records are up to 5000 pieces, the system will create a new operation log file to store the new record.

The name of the operation log file obeys this naming rule: "Project Name + Date + a blank + Time + .log", for example, ExamplePro2017-9-14 11-18-30.log. The save path of operation log file is as the same as the project's.

15.6 Enable SOE

15.6.1 Overview

The controller supports SOE function. You can enable SOE to get the historical event information of variables. Currently, SOE variables include PRG variables, global variables, and LK630S channel variables.

15.6.2 Requirement

Variables have been created.

15.6.3 Step

To enable SOE function, follow the following steps:

Step 1. In the variable definition area or "User Parameters", enable SOE function.

Main(PRG).ld						
No.	Name	Description	Data Type	Initial Value	Area	SOE ENABLE
0001	g1		BOOL	FALSE	G Area	☒
0002	g2		BOOL	FALSE	G Area	☒
0003	g3		BOOL	FALSE	G Area	☒

Figure 181 Example_Enable SOE of PRG variables

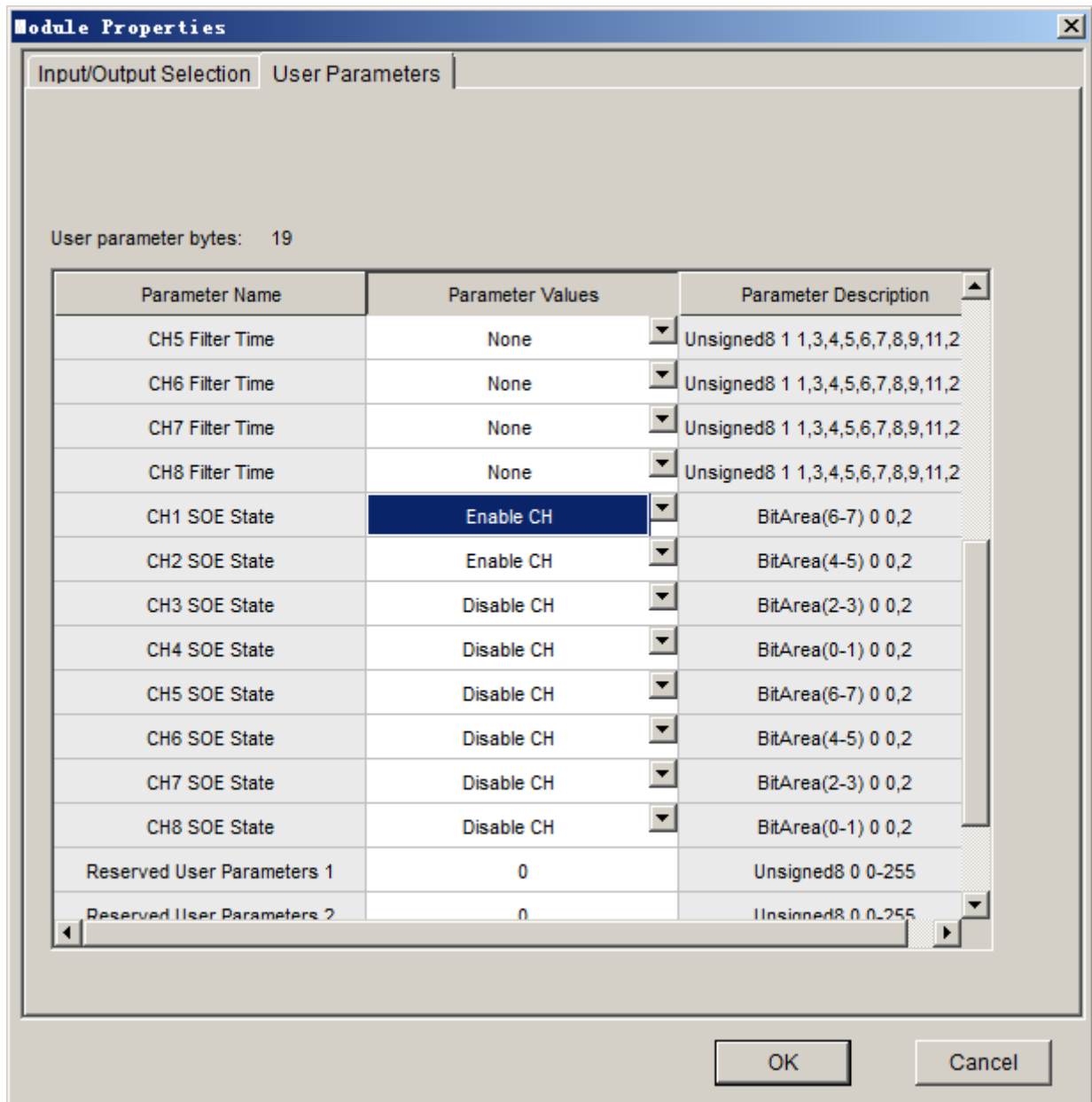


Figure 182 Example_Enable SOE of Channel

Step 2. Download project.

15.7 View SOE

15.7.1 Overview

The controller records real-time historical event information of BOOL variables that have enabled SOE function. Whenever a value changes from true to false, or from false to true, an event is recorded. Up to 6000 events can be saved.

When you need to view history information, you can read SOE from the controller. SOE information will be saved in the local path in ".xls" format.

See the follows table for detail event information.

Event information	Description
Transition time	The time of the events, accurate to milliseconds
Variable name	Variable name or channel name
Variable group name	POU name, module name , global variable group or library file name to which the variable belongs to
Description	Variable description is the same as variable definition
Transition type	Change state of variable value, from 0 to 1 or from 1 to 0

Non traceability history events caused by the project modification.

- After the variable is deleted, the previous history events cannot be traced.
- After the variable is renamed, the previous history events are displayed as the current latest variable name.

15.7.2 Requirement

- SOE function of variables has been enabled
- The project has been download
- Safety FA-AutoThink is offline
- Controller firmware version is LK220S-A03 and above

15.7.3 Step

To view SOE information, follow the following steps:

- Step 1.** Click Online menu.
- Step 2.** Select [Upload SOE File] command.
A confirmation dialog box will pop up.
- Step 3.** Click the **OK** button.
The "Save as" dialog box pops up.
- Step 4.** Enter the file name and select the save path.
- Step 5.** Click the **Save** button.

15.8 Set NTP

15.8.1 Overview

When the 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.

15.8.2 Requirements

Safety FA-AutoThink is offline

15.8.3 Steps

To set NTP calibration, follow the following steps:

Step 1. Click Online menu.

Step 2. Select [NTP settings] command.

The settings dialog box will pop up.

Step 3. Check the "NTP Enable" option.

Step 4. Enter the IP address of the NTP timing server.

Step 5. Please select a time value of hour, minute or second, and click the adjust button to increase or decrease the value.

It can be set from 00:01:00 to 23:59:59.

Step 6. After setting NTP parameters, please download project to the controller.

At this time, NTP timing function takes effect.



Beijing HollySys Intelligent Technologies Co., Ltd..

Di Sheng Middle Road, No.2

Economic-Technological Development Area

100176 Beijing, China

Tel: 010-5898 1588

Hotline: 4008111999

Fax: 010-5898 1558

<http://www.hollysys.com>