Smarter. Greener. Together.

Industrial Automation Headquarters
Delta Electronics, Inc.
Taoyuan Technology Center
No.18, Xinglong Rd., Taoyuan City,
Taoyuan County 33068, Taiwan
TEL: 886-3-362-6301 / FAX: 886-3-371-6301

Asia
Delta Electronics (Jiangsu) Ltd.
Wujiang Plant 3
1688 Jiangxing East Road,
Wujiang Economic Development Zone
Wujiang City, Jiang Su Province,
People's Republic of China (Post code: 215200)
TEL: 86-512-6340-3008 / FAX: 86-769-6340-7290

Delta Greentech (China) Co., Ltd.
238 Min-Xia Road, Pudong District,
ShangHai, P.R.C.
Post code : 201209
TEL: 86-21-58635678 / FAX: 86-21-58630003

Delta Electronics (Japan), Inc.
Tokyo Office
2-1-14 Minato-ku Shibadaimon,
Tokyo 105-0012, Japan
TEL: 81-3-5733-1111 / FAX: 81-3-5733-1211

Delta Electronics (Korea), Inc.
1511, Byucksan Digital Valley 6-cha, Gasan-dong,
Geumcheon-gu, Seoul, Korea, 153-704
TEL: 82-2-515-5303 / FAX: 82-2-515-5302

Delta Electronics Int'l (S) Pte Ltd
4 Kaki Bukit Ave 1, #05-05, Singapore 417939
TEL: 65-6747-5155 / FAX: 65-6744-9228

Delta Electronics (India) Pvt. Ltd.
Plot No 43 Sector 35, HSIIDC
Gurgaon, PIN 122001, Haryana, India
TEL : 91-124-4874900 / FAX : 91-124-4874945

Americas
Delta Products Corporation (USA)
Raleigh Office
P.O. Box 12173,5101 Davis Drive,
Research Triangle Park, NC 27709, U.S.A.
TEL: 1-919-767-3800 / FAX: 1-919-767-8080

Delta Greentech (Brasil) S.A
Sao Paulo Office
Rua Itapeva, 26 - 3° andar Edificio Itapeva One-Bela Vista
01332-000-São Paulo-SP-Brazil
TEL: +55 11 3568-3855 / FAX: +55 11 3568-3865

Europe
Deltronics (The Netherlands) B.V.
Eindhoven Office
De Witbogt 15, 5652 AG Eindhoven, The Netherlands
TEL: 31-40-2592850 / FAX: 31-40-2592851

DVP-4949620-01

*We reserve the right to change the information in this catalogue without prior notice.

2013-01-18

ISPSoft User Manual

ISPSoft User Manual

www.delta.com.tw/ia

Smarter. Greener. Together.

Preface

ISPSoft is Delta's new generation software development tool for programmable logic controllers. IEC 61131-3, which supports five programming languages and a large number of applied instructions, is adopted. Besides, ISPSoft manages projects by means of integrating tasks. The efficient and convenient development environment that ISPSoft provides enables users to apply PLCs to more complex control systems as well as to small control systems.

About IEC 61131-3

In early days, PLC developers established their own programming languages. These different programming languages were burdens on users. In view of this, IEC 61131-3, which was published by the International Electrotechnical Commission (IEC), appeared. IEC 61131-3 integrated the characteristics of the programming languages. It considered users' backgrounds and users' habits, and defined programming language standards that manufacturers and users could follow. The program codes which meet the IEC61131-3 standard are more readable and compatible. The description of IEC 61131-3 is as follows.

- IEC 61131-3 integrated the programming languages established by PLC developers, and defined five programming languages for programmable control systems: instruction lists (IL), structured texts (ST), ladder diagrams (LD), sequential function charts (SFC), and function block diagrams (FBD). Besides, continuous function charts (CFC), derived function block diagrams, are also a common PLC programming language.
- The concept related to symbols is adopted. Users can replace a device with a symbol. A program is more readable, and the time of assigning devices is saved,
- The architecture related to program organization units (POUs) is adopted. A traditional program is divided into several program organization units which can be developed independently. The architecture of a program can be more modular and can be maintained more easily be means of calling functions and function blocks.
- Program organization units are managed and organized through the concept related to tasks. The development of programs is upgraded to the management of projects. The large-scale development of programs can be managed more easily.

About the manual

The manual aims to introduce the functions of the software. The related concepts are also introduced. The slight variations among different models which are not obstacles to the operation of the software are mentioned briefly. Users have to refer to the related operation manuals or programming manuals for more information.

The contents of the manual are oriented towards the introduction of the operation of the software. Users can easily find topics related to the operation of the software in the table of contents, such as setting communication parameters, inserting a contact in a ladder diagram, monitoring devices online, and enabling a simulator. If the topic of a chapter is related to a certain concept, the concept will be introduced in the front part of the chapter. Users who are equipped with basic knowledge, or who want to know the operation of the software can skip this part, and directly refer to the related sections. Besides, if parts of the contents of a chapter are related to the contents of other chapters, the sections to which users have to refer will be noted in the chapter. The brief introduction of the chapters in the manual is as follows.

- Chapter 1 and chapter 2 introduce the installation of the software, the environment, and the basic setting.
- Chapter 3 is about configuring and setting a PLC system.
- Chapter 4 provides a simple example, and leads users to create a traditional ladder diagram in ISPSoft in a short time without using the IEC61131-3 architecture.
- Chapter 5~chapter 7 introduce the concepts related to IEC61131-3, and the ways in which the concepts are applied to ISPSoft.
- Chapter 8~chapter 12 introduce ladder diagrams (LD), function block diagrams (FBD), instruction lists (IL), structured texts (ST), and sequential function charts (SFC).
- Chapter 13 introduces the auxiliary tools provided by ISPSoft.
- Chapter 14 is about monitoring a program, testing a program, and debugging a program through ISPSoft.

- Chapter 15 introduces the password setting, and the data protection mechanism in ISPSoft.
- Chapter 16 is about configuring a network through ISPSoft, and creating a data exchange mechanism.
- Chapter 17 is about backing up data and restoring data through ISPSoft.

Graphic representations

The graphic representations used in the manual are listed in the following table.

Graphic representation	Description
	Clicking the left mouse button
	Clicking the right mouse button
	Double-clicking the left mouse button
	Pressing and holding the left mouse button, and then moving the mouse without releasing the button.
	Typing with a keyboard
	Operating sequence (The graphic representation is used when an operating sequence is mentioned. For example, and .)
	Number used with a picture
	A point for attention (The item mentioned may be related to damage to equipment, property, or a human body.)

Trademark declaration

The products and trademarks which do not belong to Delta Electronics, Inc. belong to the companies which produce and declare them.

Chapter 1 Introduction of the Software

Table of Contents

1.1 Introduction of ISPSoft and System Requirements 1-2
1.1.1 Characteristics 1-2
1.1.2 System Requirements 1-2
1.1.3 Installing ISPSoft 1-3
1.1.4 Uninstall ISPSoft 1-5
1.2 Introduction of COMMGR 1-6
1.2.1 Working Mode of COMMGR 1-6
1.2.2 Installing COMMGR 1-7
1.2.3 Uninstalling COMMGR 1-9

1.1 Introduction of ISPSoft and System Requirements

ISPSoft is a software development tool for Delta's new generation programmable logic controllers. IEC 61131-3, which supports five programming languages and a large number of applied instructions, is adopted. In addition to basic programming functions, ISPSoft also contains many auxiliary tools. The multilingual environment and the friendly user interface provide users with a convenient and efficient development environment.

1.1.1 Characteristics

- It supports the international standard IEC 61131-3 and a large number of applied instructions.
- It supports five programming languages. They are ladder diagrams (LD), sequential function charts (SFC), function block diagrams (FBD), instruction lists (IL), and structured texts (ST). Users can use more one programming language in one project.
- It supports traditional Chinese, simplified Chinese, and English.
- The **Find** and **Replace** functions can be applied to a present window, or a whole project.
- It provides a user-defined operating environment.
- The project management adopts an interface which uses a hierarchical tree structure.
- Users can develop several models in a group of projects.
- It provides many convenient functions such as making comments, creating bookmarks, activating/inactivating networks, managing devices and symbols, simulation, and etc.
- It supports several types of connections such as monitoring programs online, editing programs online, monitoring devices, debugging programs, operating/setting a PLC, and etc.
- Users can import and export projects by means of the **Import** and **Export** functions.
- A file (*.dvp) created with WPLSoft can be opened directly, and can be converted into an ISPSoft format (*.isp).
- It provides several password setting and data protection mechanism.
- It supports COMMGR, a new generation communication manager.
- It supports new generation Delta AH500 series modules.
- There three built-in configurations.
  - HWCONFIG: It is used to configure hardware for a system, and manage parameters.
  - NWCONFIG: It is used to configure a network for a PLC system, and manage data exchange.
  - CARD Utility: Users can backup and restore a system through a management wizard and a memory card.

1.1.2 System Requirements

Before using ISPSoft, users have to make sure that an operating system meets the requirements below.

Item	System requirement
Operating system	Windows 2000/NT/Me/XP/Vista/7
CPU	Pentium 1.5 G or above
Memory	256 MB or above (A memory having a capacity of 512 MB or above is recommended.)
Hard disk drive	Capacity : 500 MB or above
CD-ROM drive	For installing ISPSoft It is optionally required.
Monitor	Resolution: 800×600 or above (Setting which is suggested: 1024x768/96 dpi)
Keyboard/Mouse	A general keyboard/mouse, or devices compatible with Windows
Printer	A printer with a driver for Windows (It is used to print a project, and is optionally required.)

Item	System requirement	
RS-232 port	For connecting to a PLC	One of them is used, but a PLC which is connected must have a corresponding port. (*1)
USB port	For connecting to a PLC	
Ethernet port	For connecting to a PLC	
Communication software	COMMGR, a communication manager, must be installed on a computer. (*2)	
Models which are supported	AH500 series PLCs/DVP series PLCs (exclusive of DVP-PM series PLCs)/VFD-C2000 series AC motor drives/VFD-C200 series AC motor drives/VFD-CP2000 AC motor drives /VFD-E AC motor drives (*3)	

***1. ISPSoft supports several ways in which a computer is connected to a PLC. Users have to make sure of the port and the mode supported by a PLC before a computer is connected to the PLC.**

***2. Please refer to section 1.2 for more information about COMMGR.**

***3. In addition to ISPSoft, users must use PMSoft version 2.05 or above to develop AH10PM-5A and AH20MC-5A.**

***4. The functions and specifications mentioned above are only applicable to ISPSoft version 2.00 or above. The older versions are not equipped with the complete functions.**

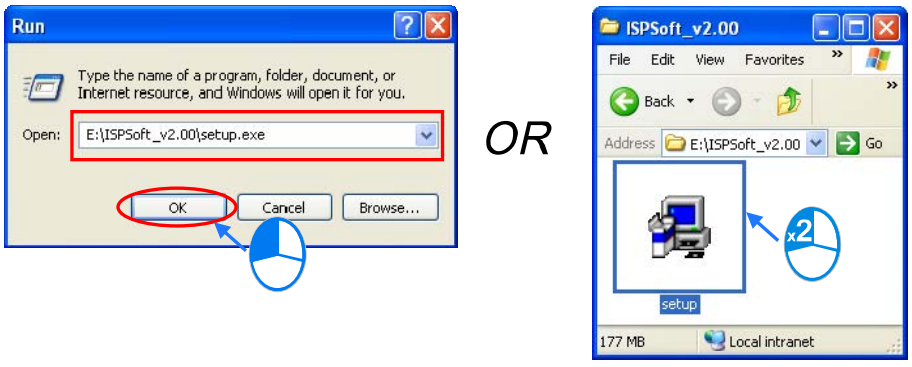
1.1.3 Installing ISPSoft

If an older version of ISPSoft has been installed on a computer, users have to uninstall it before install ISPSoft. (Pleases refer to section 1.1.4 for more information about uninstalling ISPSoft.)

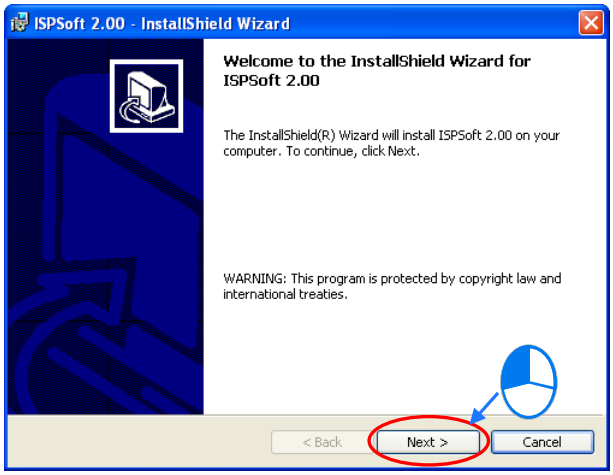
(1) Start a Windows 2000/NT/Me/XP/Vista/7 operating system. Users have to log in the system as a system administrator before they install ISPSoft.

(2) Put an ISPSoft CD in the CD-ROM drive, or download the installation program from http://www.delta.com.tw/ch/index.asp. (Before the installation program downloaded from the website is installed, it has to be decompressed.)

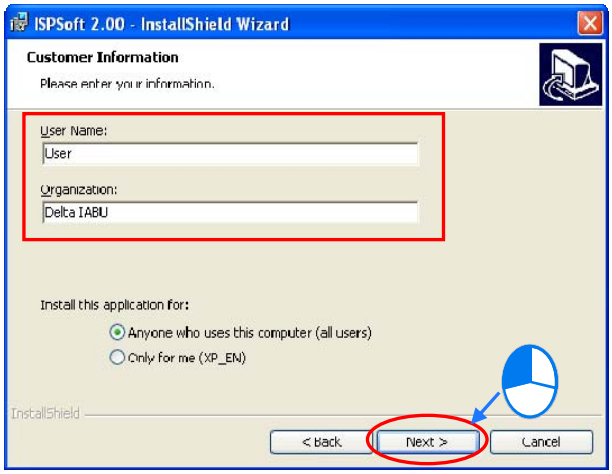
(3) Click **Start**, and then click **Run…** to open the **Run** window. Specify a path which denotes a file called setup.exe in the **Open** box, and then click **OK**. Users can also double-click the **setup** icon to execute the installation program.



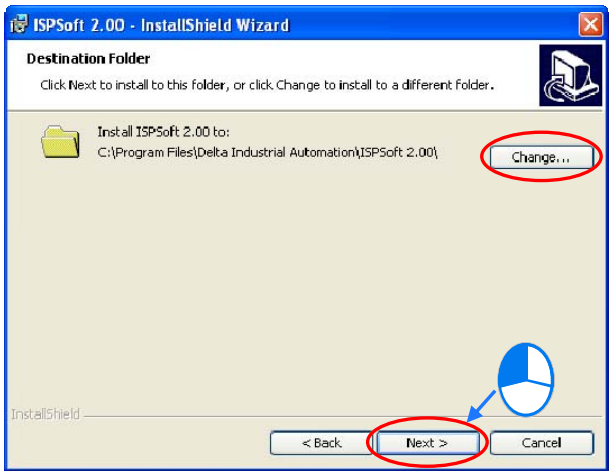
(4) After the **ISPSoft 2.00 – InstallShield Wizard** window appears, click **Next**.



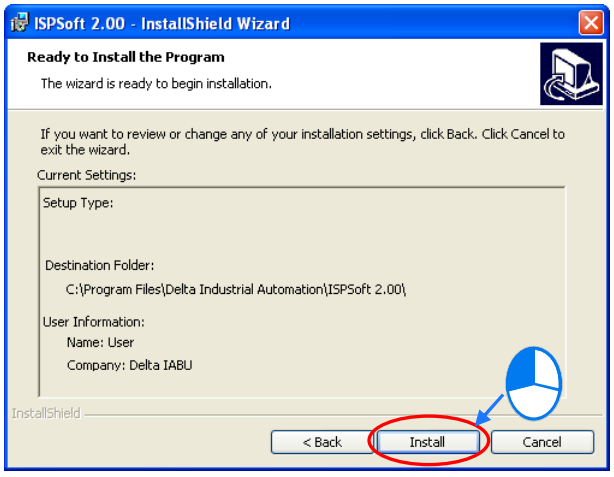
(5) Type related information in the **User Name** box and **Organization** box, and then click **Next**.



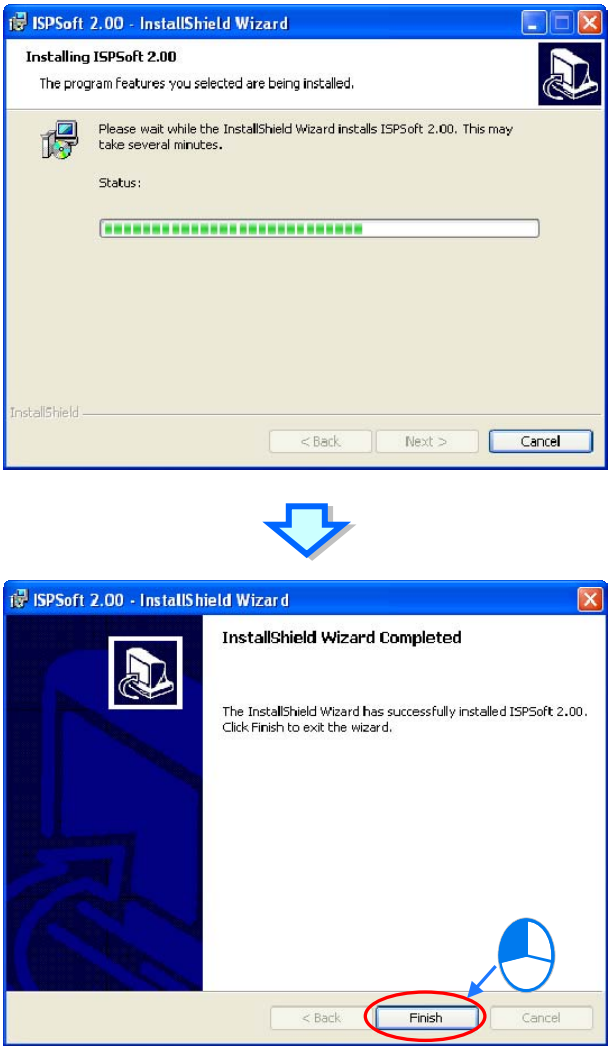
(6) Leave the default path unchanged, or click **Change…** to change the path. Click **Next** to proceed to the next step.



(7) Check the installation information, and then click **Install**.



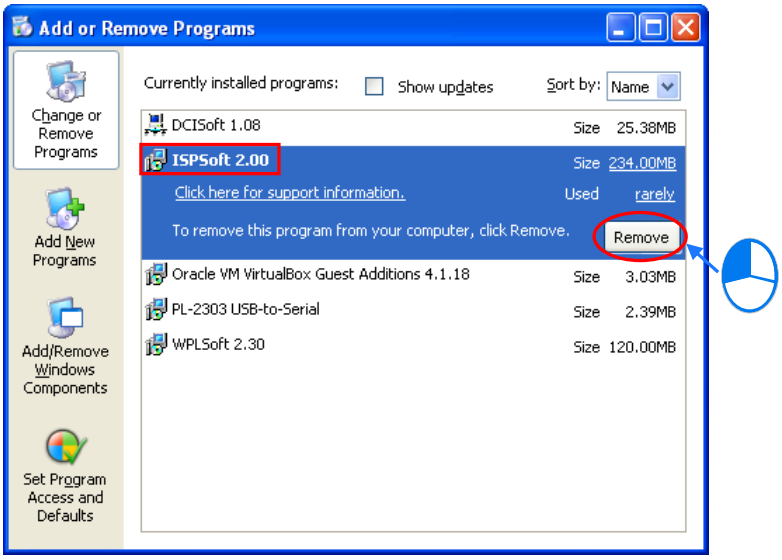
(8) After ISPSoft is installed, shortcuts to the program are created on the desktop and the **Start** menu. Click **Finish** to complete the installation.



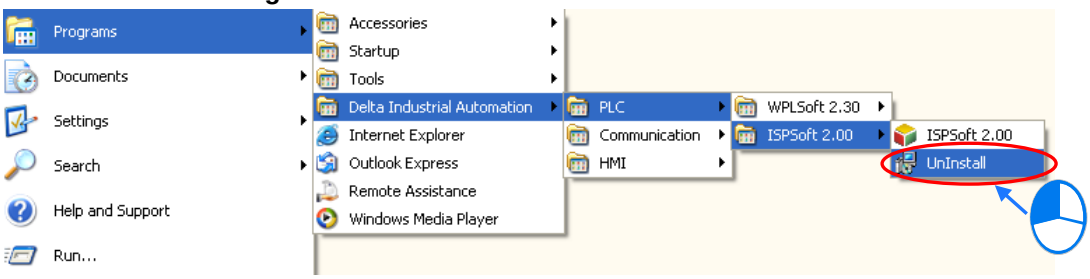
1.1.4 Uninstall ISPSoft

(1) There are two methods of uninstalling ISPSoft.

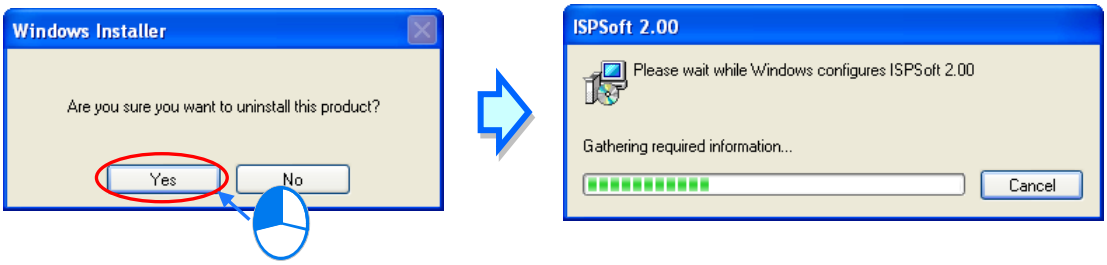
- Method 1: Open the **Control Panel** window, and click **Add or Remove Programs**. In the **Currently installed programs** box, click **ISPSoft x.xx**, and then click **Remove**.



- Method 2: **Start>Programs>Delta Industrial Automation>PLC>ISPSoft x.xx>Uninstall**



(2) After users click **Yes**, ISPSoft will be removed.

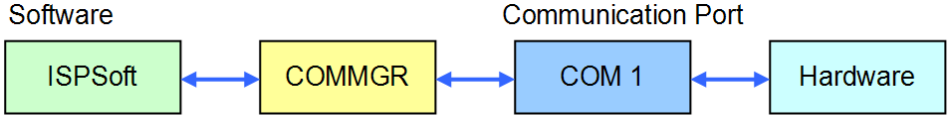


1.2 Introduction of COMMGR

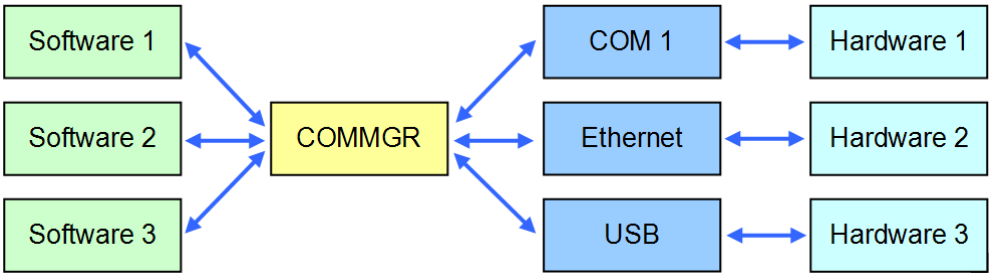
COMMGR is a new generation communication management tool developed by Delta Electronics, Inc. in 2011. It functions as a communication bridge between Delta software and hardware. Communication becomes more convenient and more efficient through the management of COMMGR.

1.2.1 Working Mode of COMMGR

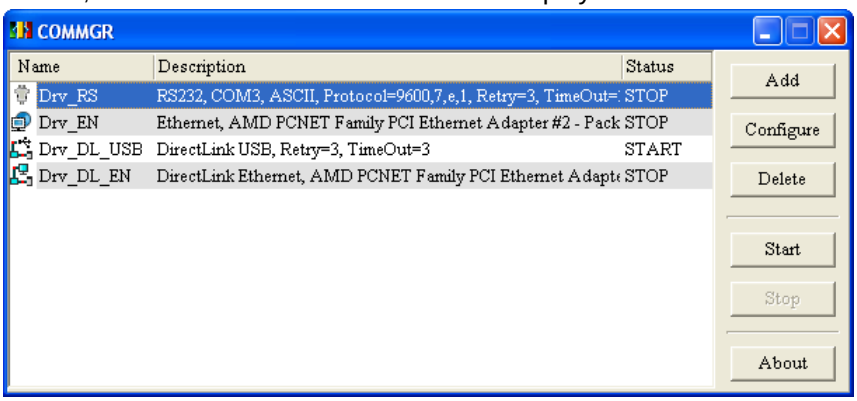
Users can create communication parameters which must be set on the management list in COMMGR in advance. The communication parameters which have been created in advance are called a driver. Users can start or stop a driver in COMMGR. If a driver is started, a connection will be created automatically. After the users specify a driver in ISPSoft, a communication will be carried out.



In addition to ISPSoft, other software communicating with hardware through COMMGR can operate simultaneously. COMMGR manages all communication commands, and makes software connect to hardware.



The **COMMGR** window and the management list are shown below. Drivers which are named by users are displayed in the **Name** column, parameters related to the drivers are displayed in the **Description** column, and the statuses of the drivers are displayed in the **Status** column.



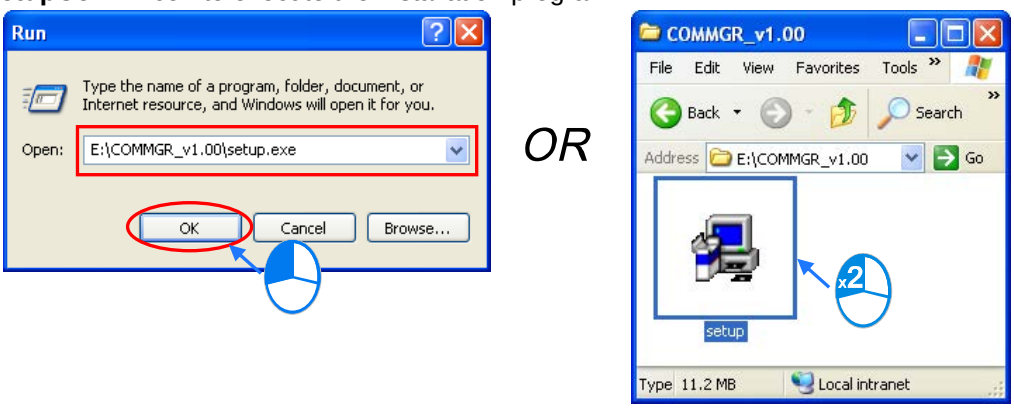
1.2.2 Installing COMMGR

COMMGR is software independent of ISPSoft. It must be installed and uninstalled separately. If an older version of COMMGR has been installed on a computer, users have to uninstall it before install COMMGR. (Pleases refer to section 1.2.3 for more information about uninstalling COMMGR).

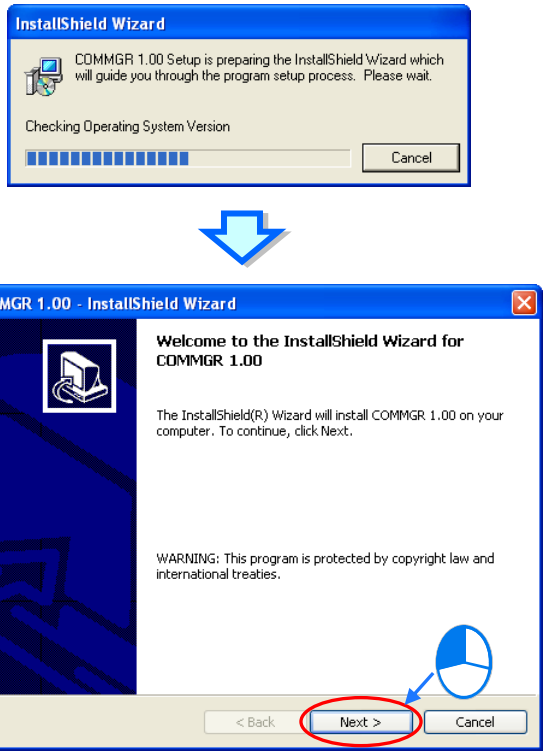
(1) Start a Windows 2000/NT/Me/XP/Vista/7 operating system. Users have to log in the system as a system administrator before they install COMMGR.

(2) Put a COMMGR CD in the CD-ROM drive, or download the installation program from http://www.delta.com.tw/ch/index.asp. (Before the installation program downloaded from the website is installed, it has to be decompressed.)

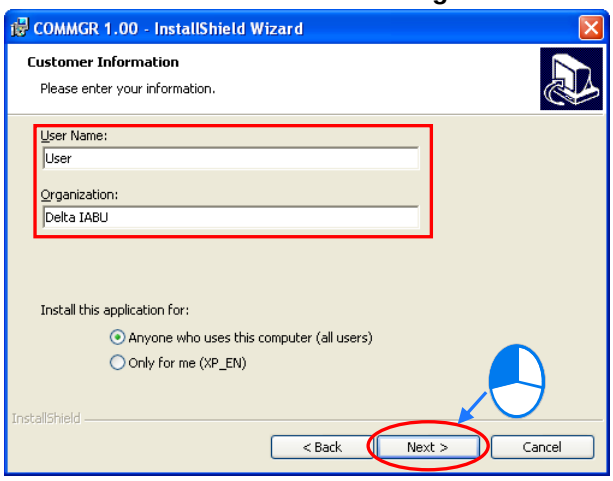
(3) Click **Start**, and then click **Run…** to open the **Run** window. Specify a path which denotes a file called setupComm.exe in the **Open** box, and then click **OK**. Users can also double-click the **setupComm** icon to execute the installation program.



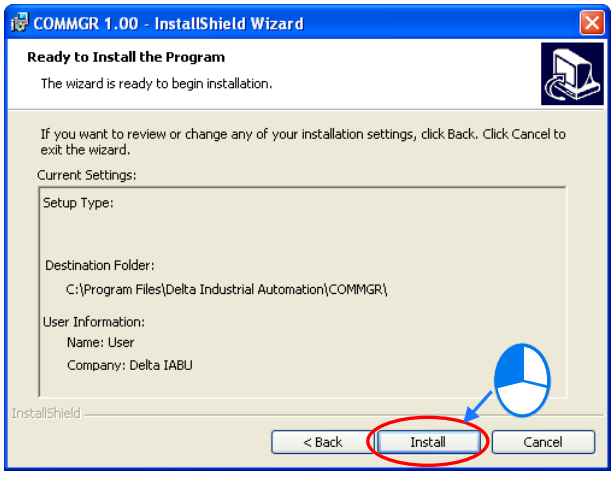
(4) After the **COMMGR 1.00 – InstallShield Wizard** window appears, click **Next**.



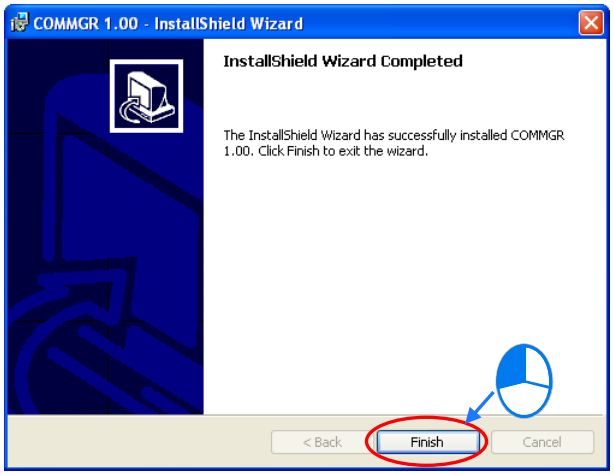
(5) Type related information in the **User Name** box and **Organization** box, and then click **Next**.



(6) Check the installation information, and then click **Install**.



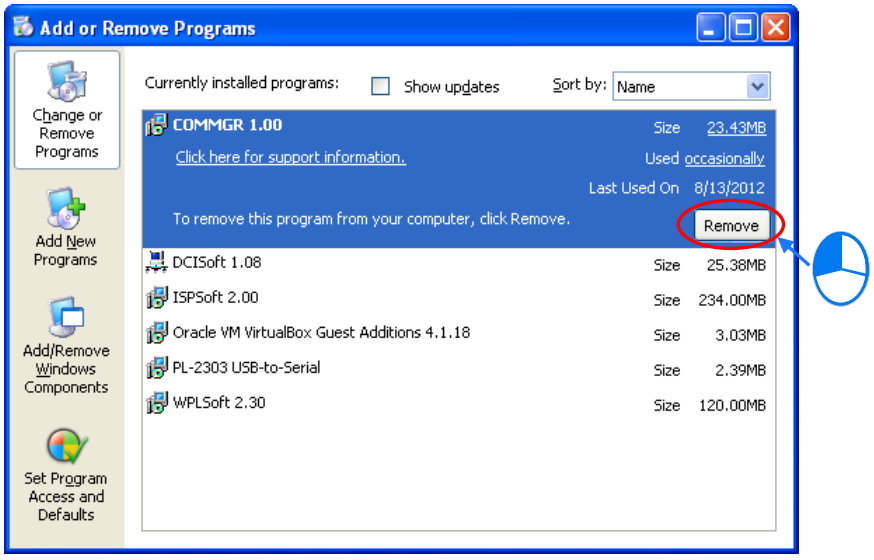
(7) After COMMGR is installed, a shortcut to the program is created on the **Start** menu. Click **Finish** to complete the installation.



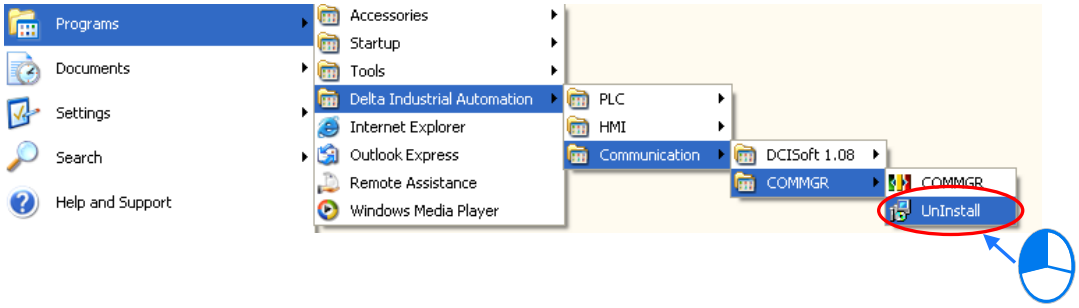
1.2.3 Uninstalling COMMGR

(1) There are two methods of uninstalling COMMGR.

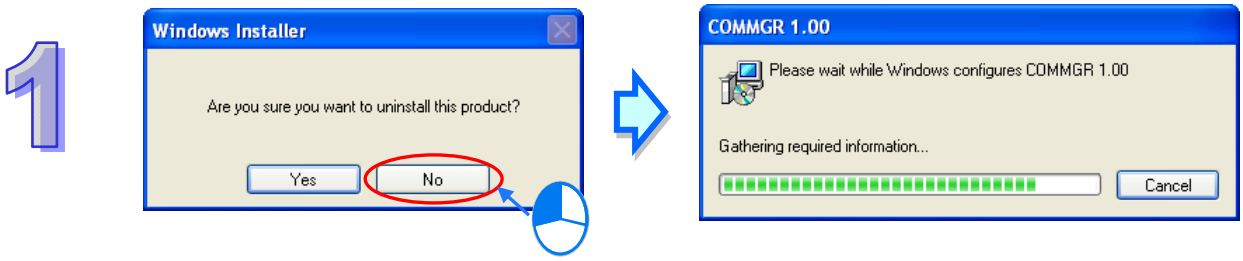
- Method 1: Open the **Control Panel** window, and click **Add or Remove Programs**. In the **Currently installed programs** box, click **COMMGR x.xx.xx**, and then click **Remove**.



- Method 2: **Start>Programs>Delta Industrial Automation>Communication>COMMGR>Uninstall**



(2) After users click **Yes**, COMMGR will be removed.


1
Windows Installer
Are you sure you want to uninstall this product?
Yes
No
COMMGR 1.00
Please wait while Windows configures COMMGR 1.00
Gathering required information...
Cancel

Chapter 2 Starting and Setting

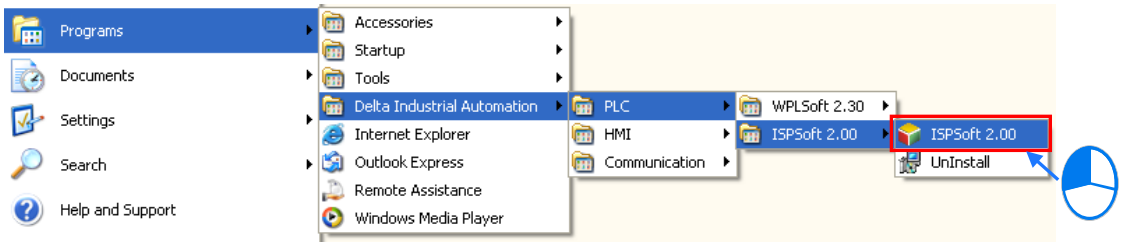
Table of Contents

2.1 Guide to Starting ISPSoft and Introduction of the Environment 2-2
2.1.1 First Step of Entering ISPSoft.....2-2
2.1.2 Window Title and Status Bar 2-5
2.1.3 Menu Bar.....2-6
2.1.4 Toolbars.....2-8
2.1.5 Project Management Area and Message Display Area 2-9
2.1.6 Working Area.....2-11
2.2 Project Framework.....2-13
2.2.1 Single Project and Group of Projects 2-13
2.2.2 Integration of PMSoft into ISPSoft.....2-14
2.2.3 Managing a Single Project 2-15
2.2.4 Managing a Group of Projects.....2-20
2.2.5 Managing a PMSoft Project through ISPSoft.....2-26
2.3 Basic Setting in ISPSoft.....2-30
2.3.1 System Setting and Environment Setting 2-30
2.3.2 Importing and Exporting Preferences.....2-35
2.4 Communication Setting.....2-36
2.4.1 Starting/Closing COMMGR 2-36
2.4.2 Managing Drivers 2-37
2.4.3 Creating a Connection—Creating a Driver 2-38
2.4.3.1 Setting Communication Parameters for RS232.....2-39
2.4.3.2 Setting Communication Parameters for USB (Virtual COM)..2-40
2.4.3.3 Setting Communication Parameters for DirectLink (USB) 2-41
2.4.3.4 Setting Communication Parameters for Ethernet 2-41
2.4.3.5 Setting Communication Parameters for DirectLink (Ethernet) 2-42
2.4.3.6 Setting Communication Parameters for a DVP Simulator.....2-43
2.4.3.7 Setting Communication Parameters for a AH Simulator 2-43
2.4.4 Creating a Connection—Starting/Stopping a Driver.....2-44
2.4.5 Creating a Connection—Configuring/Deleting a Driver 2-45
2.4.6 Creating a Connection Between ISPSoft and COMMGR 2-45
2.4.7 Connecting a PLC and a Communication Port.....2-47
2.4.8 Practical Connection Test.....2-50

2.1 Guide to Starting ISPSoft and Introduction of the Environment

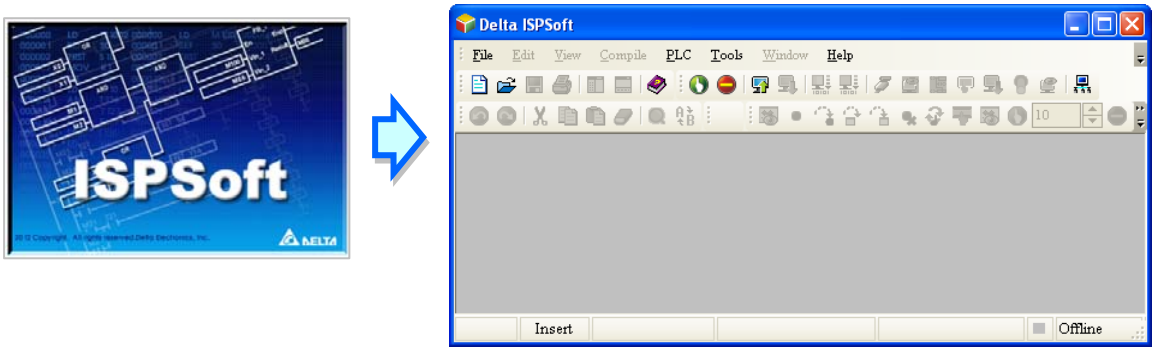
2.1.1 First Step of Entering ISPSoft

After ISPSoft is installed, shortcuts to the program are created on the desktop and the **Start** menu. Users can click the shortcut on the **Start** menu or double-click the shortcut on the desktop to start ISPSfot. Besides, several ISPSoft windows are allowed. Users can start ISPSoft again in the same way.

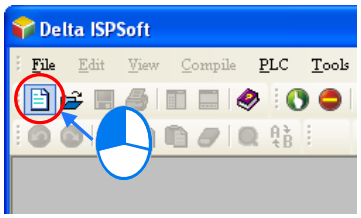


OR

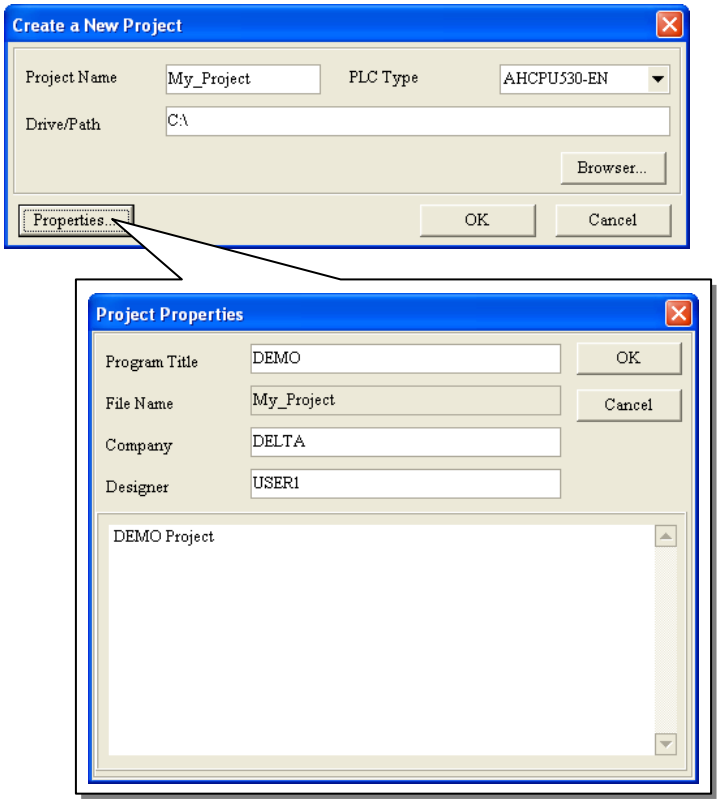
After the welcome screen disappears, the **Delta ISPSoft** window appears. There are basic functions available to users.



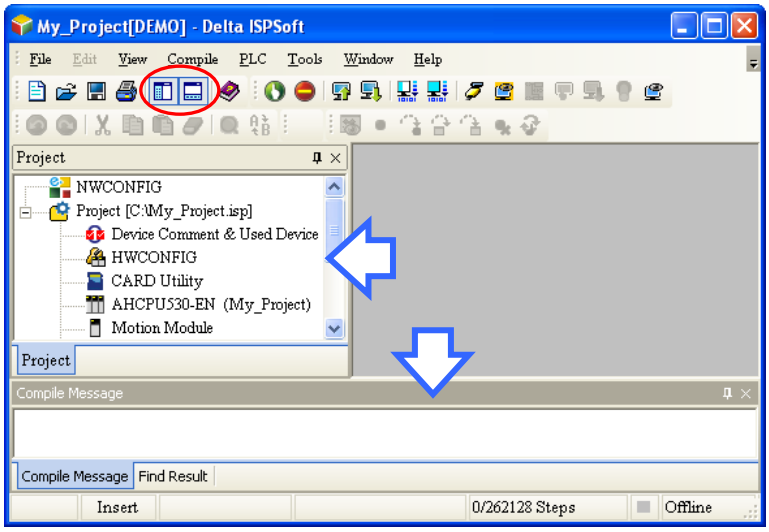
After on the toolbar is clicked, a new project is created.



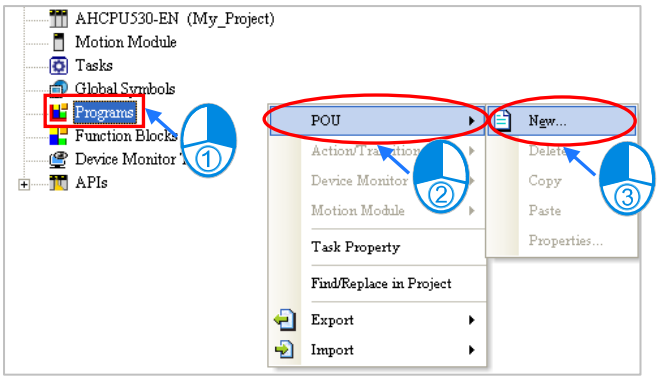
In the **Create a New Project** window, type a project name in the **Project Name** box and a path in the **Drive/Path** box, and select a PLC in the **PLC Type** drop-down list box. After users click **Properties…**, they can give a description of this project. Finally, click **OK**.



After the project is created successfully, a project management area will appear at the left side of the main screen. The relation between the items listed in the project management area is represented by a hierarchical tree structure. If the project management area does not appear, users can click **Workspace** on the **View** menu, or click on the toolbar. Besides, a message display area will appear if users click **Output Window** on the **View** menu or click on the toolbar.

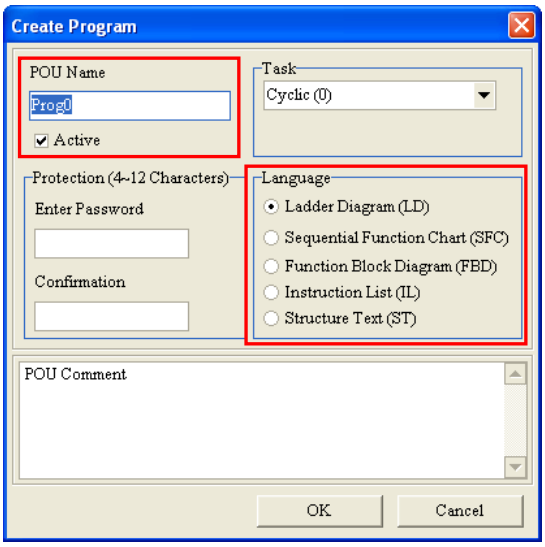


After the project is created, the main working area is blank. If users want to write a program, they can right-click **Programs** in the project management area, point to **POU** on the context menu, and click **New**….

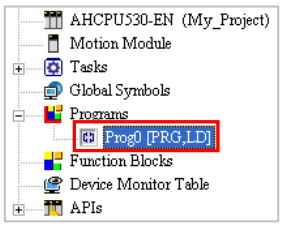


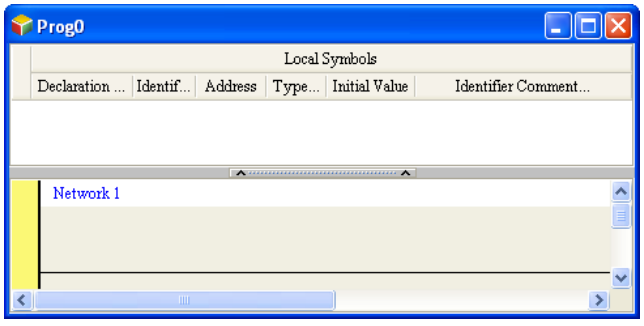
2

In the **Create Program** window, users can type a program name in the **POU Name** box, and select a programming language in the **Language** section. The other default values are retained, and will be introduced in section 5.4.

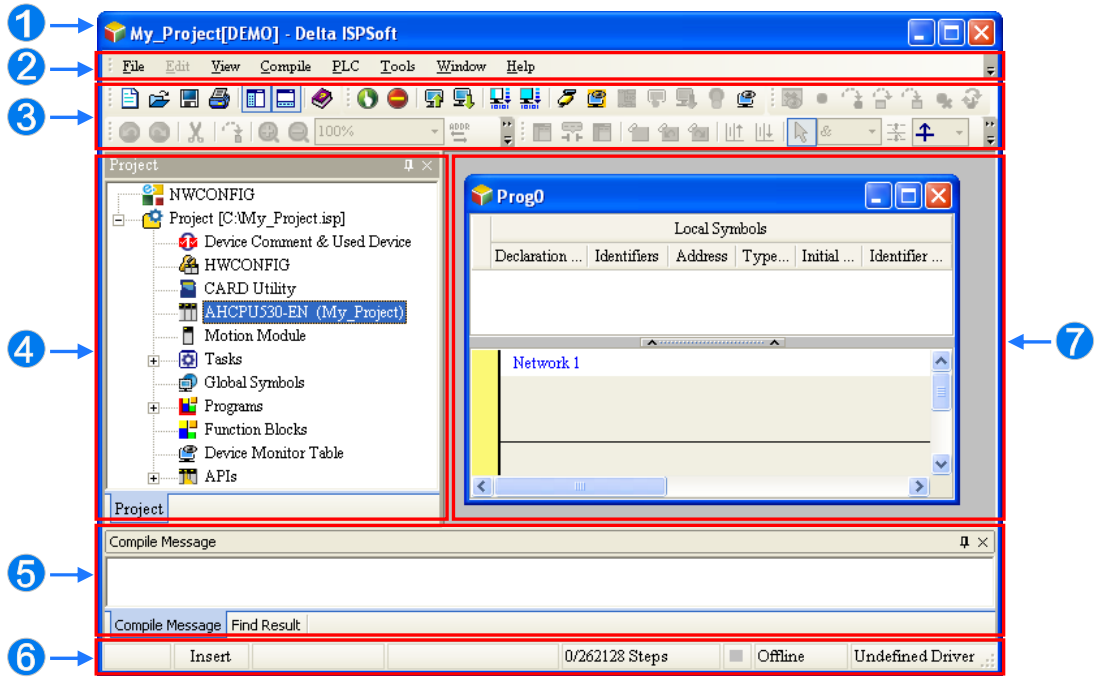


After a program organization unit is created successfully, an item will appear under **Programs** in the project management area, and a program editing window will appear in the main working area.





The main screen of ISPSoft is shown below.



❶ Window title: It displays a project name and a program title.

❷ Menu bar: There are eight menus.

❸ Toolbar: There are five toolbars.

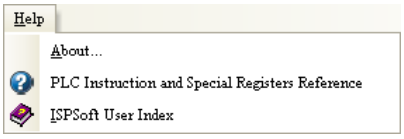
❹ Project management area: It adopts an interface which uses a hierarchical tree structure.

❺ Message display area: The messages related to the compiling of a program and search for data in a project are displayed in this area.

❻ Status bar: Information about the editing of a program and connection is displayed.

❼ Working area: A program editing area, a symbol table, a device table, and etc. are displayed in this area.

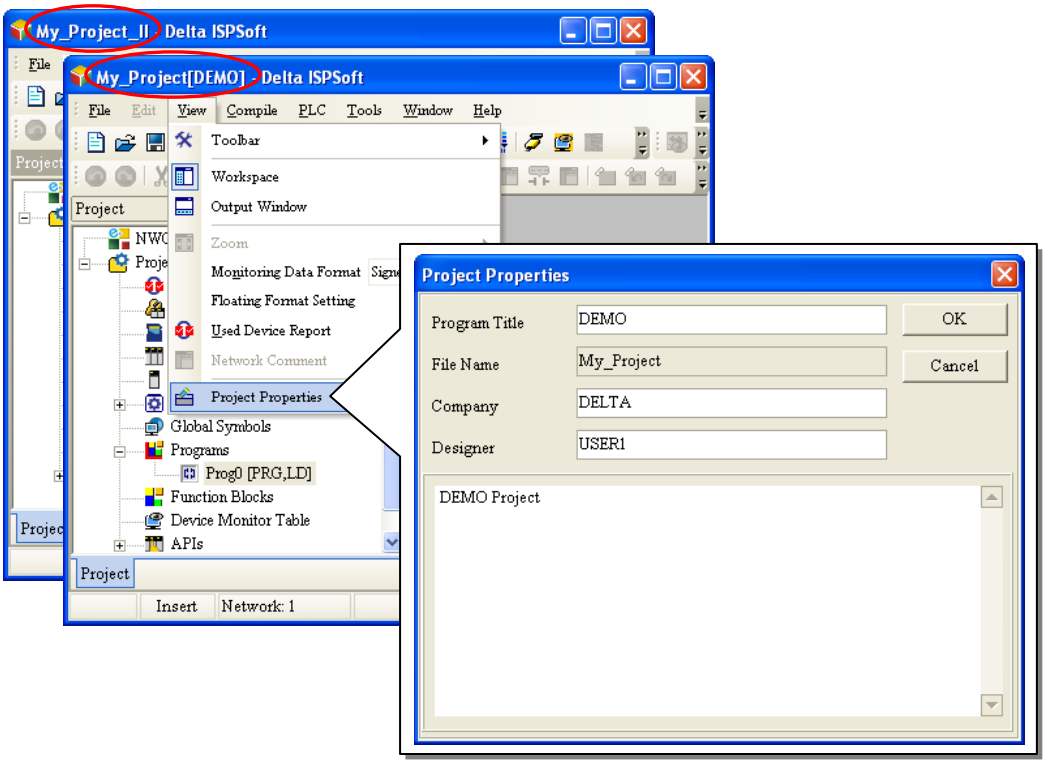
When using ISPSoft, users can click an item on the **Help** menu to get help.



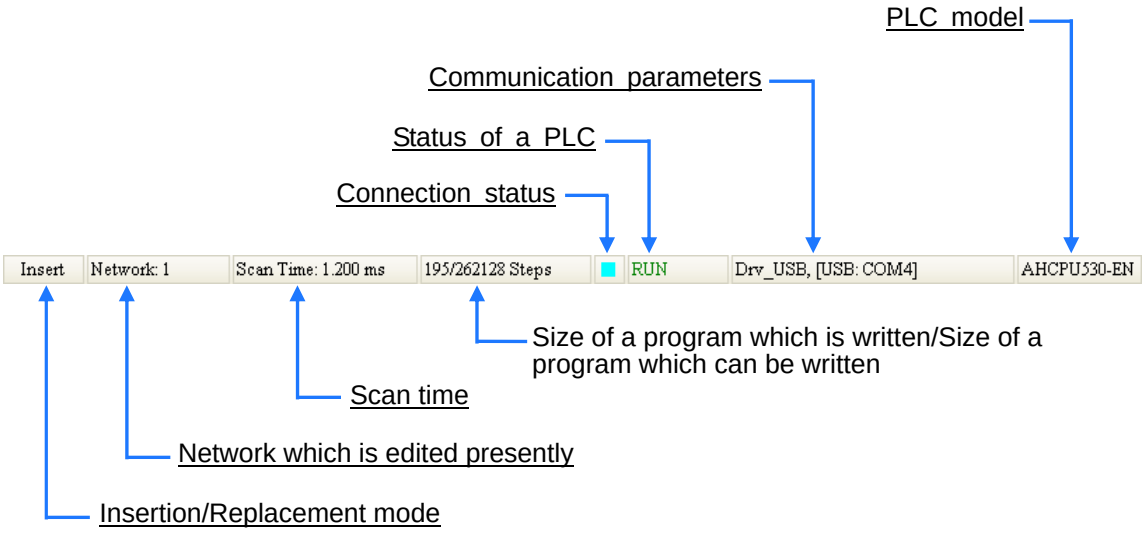
Item	Description
About…	Users can get the information about the version of the software, and the date when the software is released.
ISPSoft User Index	Users can get the usage of the software. (Users can also click on the toolbar to get the usage of the software.)
PLC Instructions and Special Registers Reference	Users can get the information about the instructions and the registers. If users click an instruction in the program editing area and press F1 on the keyboard, they can get the information bout the instruction.

2.1.2 Window Title and Status Bar

After a project is created successfully, the window title displays its project name and its program title. The format is **project name [program title]**. If users do not set a program title, the window title will only display a project name. Besides, if users want to view the information about a program title, they can click **Project Properties** on the **View** menu.

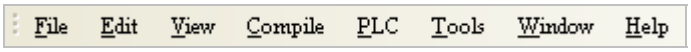


The status bar at the bottom of a window displays all kinds of working states, including an editing mode (insertion/replacement), a network which is edited presently, a scan time, the size of a program which is written/the size of a program which can be written, a connection status, the status of a PLC, communication parameters, and a PLC model.

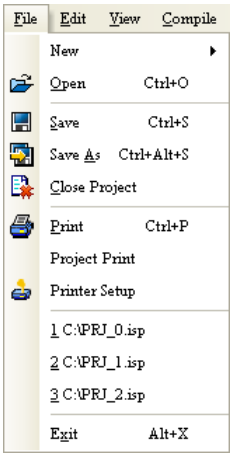


2.1.3 Menu Bar

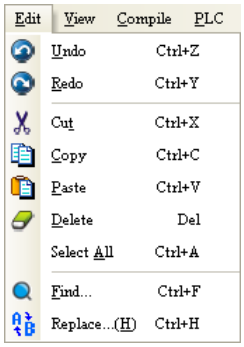
There are eight menus in the menu bar. The items on the menus vary with editing work carried out, and a model selected. A brief introduction of the menu bar is presented here, and a more detailed introduction the menu bar is presented in the following chapters.



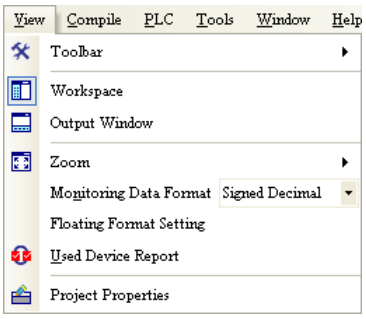
- **File**: It provides the function of accessing projects.



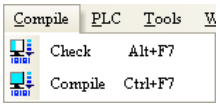
- **Edit**: It provides the function of editing a project.



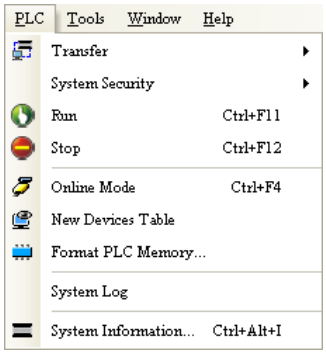
- **View**: It provides the function of viewing information about a project, and the function of setting a work environment.



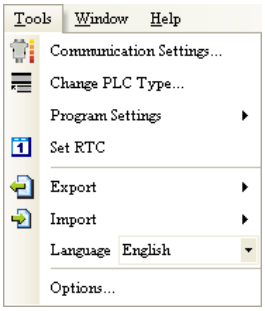
- **Compile**: It provides the function of checking syntax, and the function of compiling a program into an execution code.



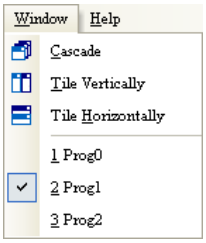
- **PLC**: It provides the function of operating or setting a PLC through ISPSoft.



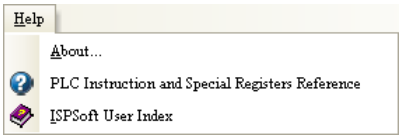
- **Tools**: It provides some convenient functions.



- **Window**: It provides the function of managing windows in the working area.

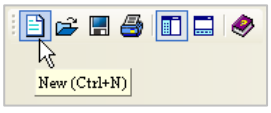


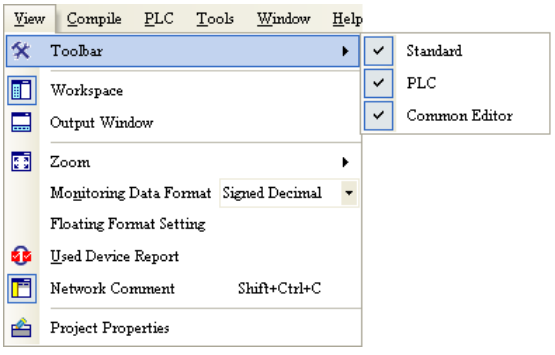
- **Help**: It provides help to users.



2.1.4 Toolbars

There are five toolbars. The icons on the toolbars also vary with editing work carried out, and a model selected. Besides, if a mouse cursor is moved to an icon and stays there for a while, the function represented by the icon and the shortcut to the function will be displayed. Users can select a toolbar which they want ISPSoft to display or hide by clicking **Toolbar** on the **View** menu.





- **File toolbar**: It provides the functions related to the processing a project.

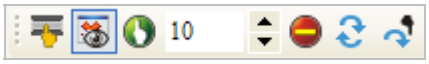
- **Editing toolbar**: It provides the functions related to editing work.

- **PLC toolbar**: It provides the functions related to operating a PLC.

- **Debugging toolbar**: It provides the functions related to debugging a program. The icons on the toolbar vary with a model selected.

AH500 series:

DVP series:



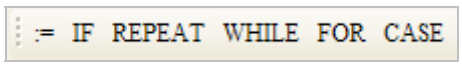
- **Programming toolbar**: It provides the functions related to editing a program. The icons ton the toolbar vary with a programming language selected.

LD:

FBD:

IL:

ST:



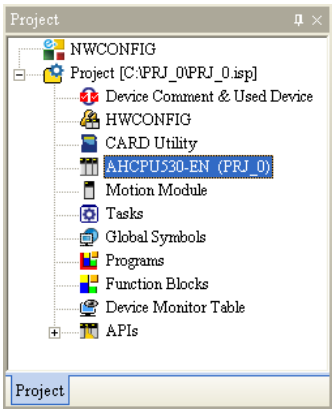
SFC:

2.1.5 Project Management Area and Message Display Area

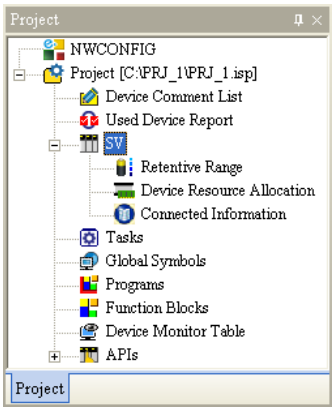
The contents of the project management area are items related to project development. The project management area adopts an interface which uses a hierarchical tree structure. The contents of the project management area include a hardware configurator, information about devices and applied instructions, symbols created by users, programs, tables of devices monitored, and etc. The items listed in the project management area vary with a model selected.

The project management area appears at the left side of the main screen. If users want to close the area, they can click in the upper right corner of the area. They can click on the toolbar, or click **Workspace** on the **View** menu to open or close the project management area. Besides, after users right-click an item in the project management area, a context menu will appear. The available choices on the context menu are actions related to the item.

● AH500 Series

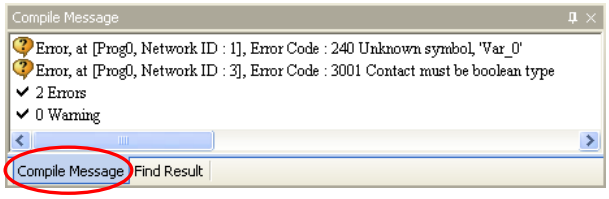


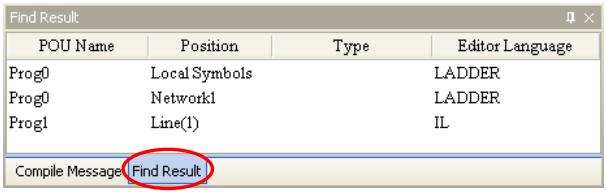
● DVP Series



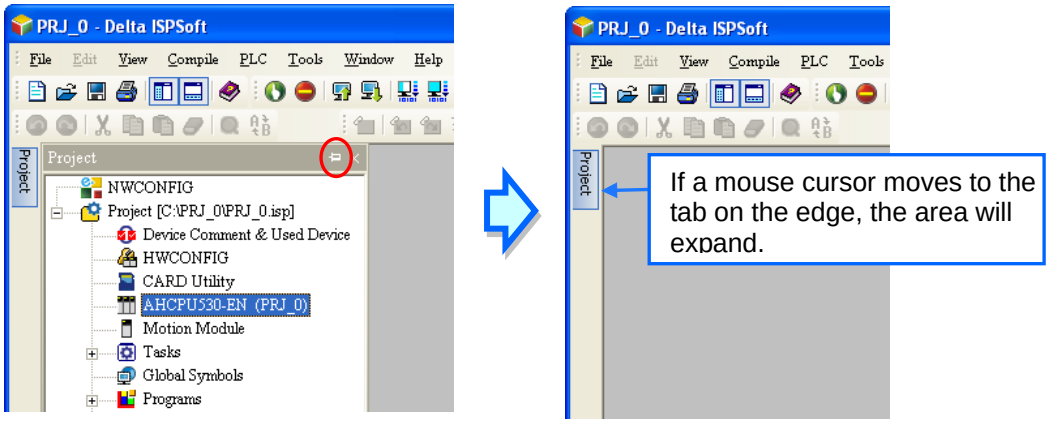
The message display area is composed of the **Compile Message** page and the **Find Result** page. Users can switch between the pages by clicking the tabs under the message display area. The **Compile Message** page shows a result or message related to the compiling of a program or the checking of syntax, and the **Find Result** page shows a result related to the search for data in a project.

The message display area appears at the bottom of the main screen. Users can click on the toolbar, or click **Output Window** on the **View** menu to open or close the message display area. If users want to close the area, they can click in the upper right corner of the area.



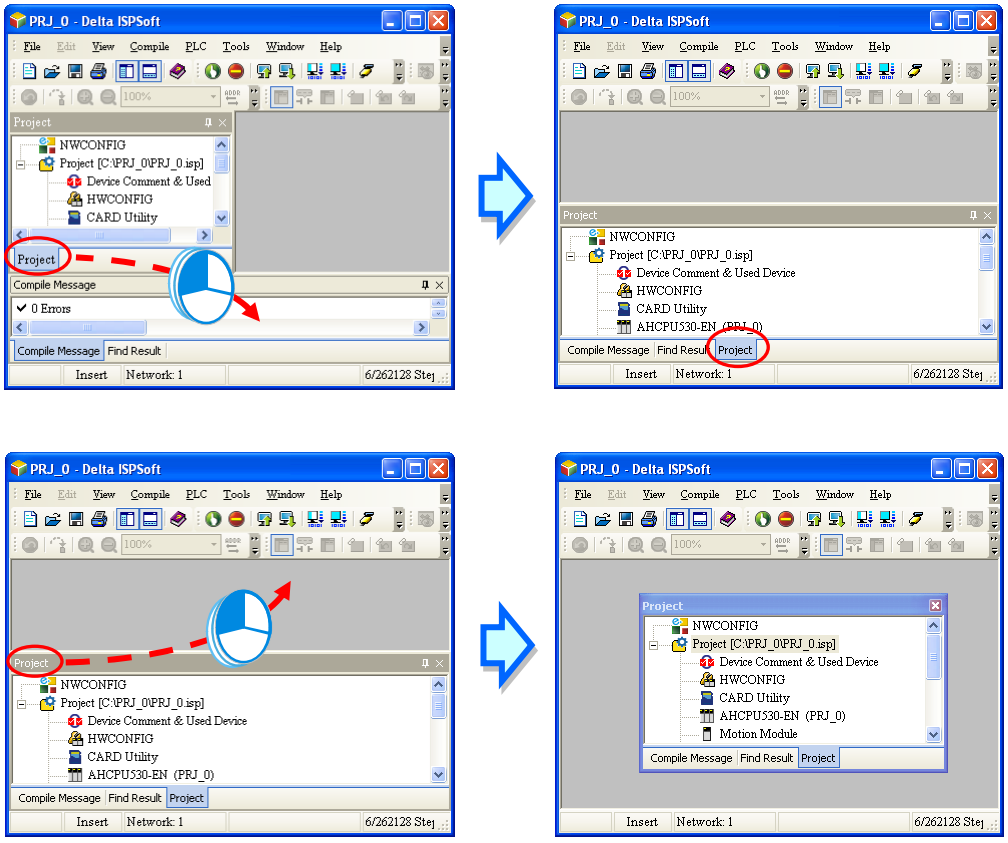


Users can change the way in which the project management area or the message display area is displayed by clicking in the upper right corner of the area. If the icon is , the area will be hidden automatically. After a mouse cursor leaves the area for a while, the area will contract. If the mouse cursor moves to the tab on the edge, the area will expand.



Besides, if the area is not automatically hidden, users can drag the tab with a left mouse button to any position, or to a position combined with the other area. If users want to drag the whole area,

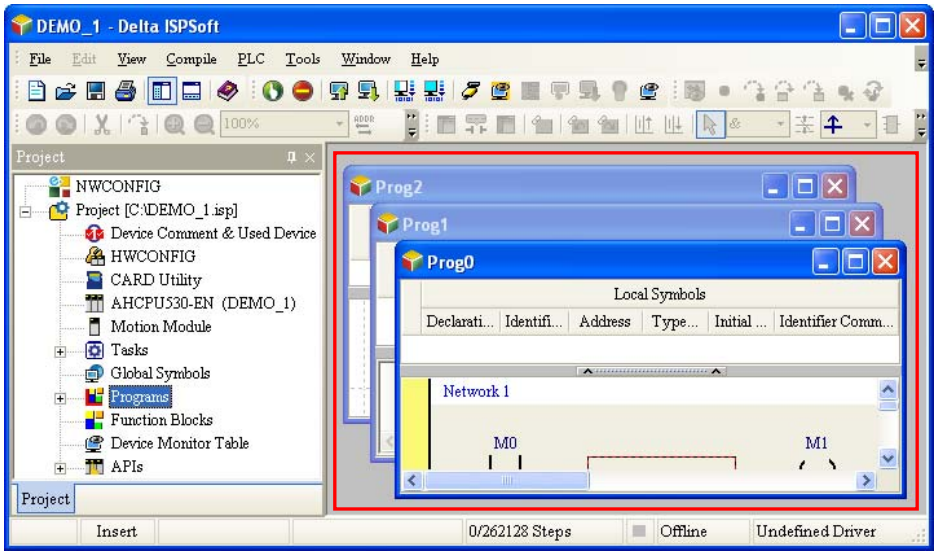
they have to drag the title bar.



2.1.6 Working Area

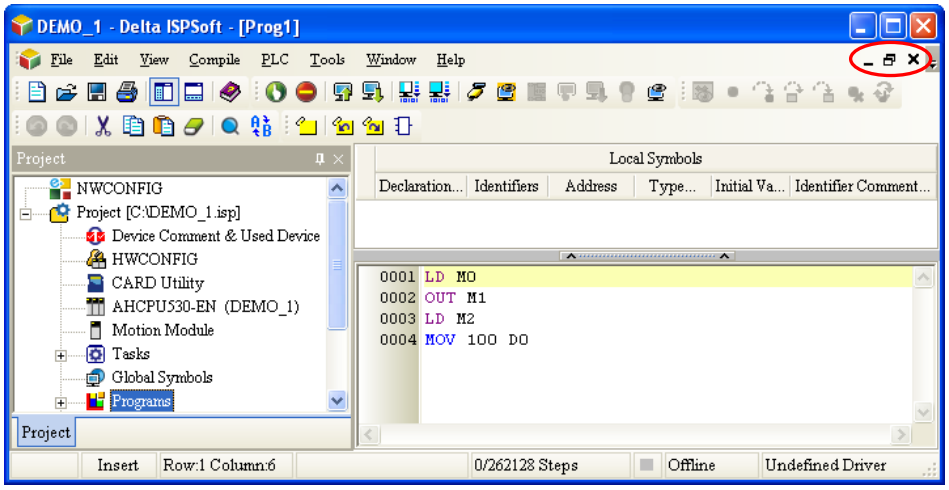
All windows are displayed in this area. If a window is maximized or minimized, it is maximized or minimized in this area. When a window is maximized, the status buttons for the window appear at the right side of the menus bar.

- **Display of windows**

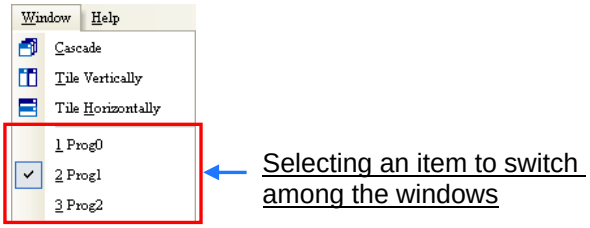


2

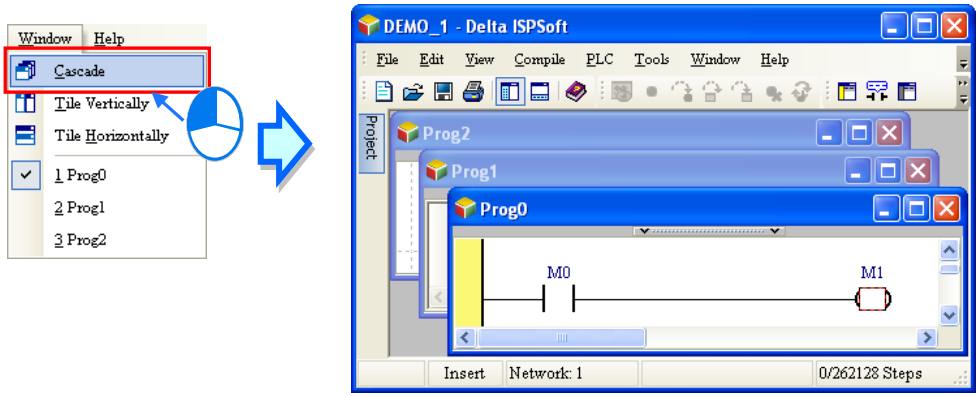
- **A window is maximized.**



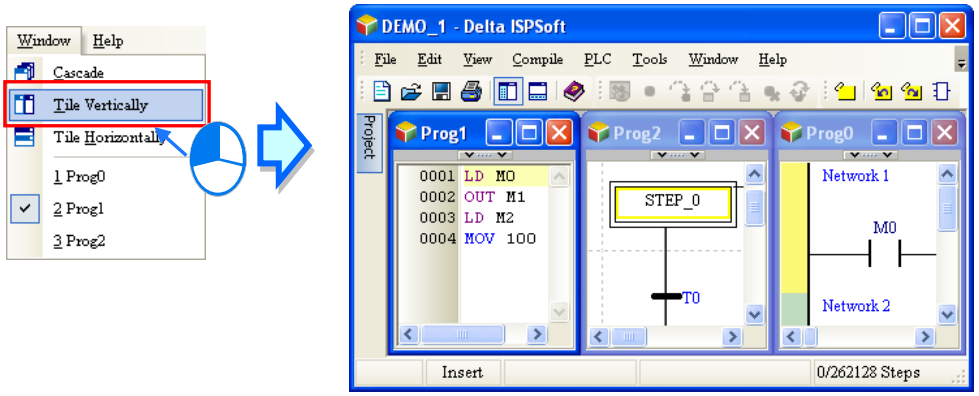
Users can manage windows and switch among windows through the choices on the **Window** menu.



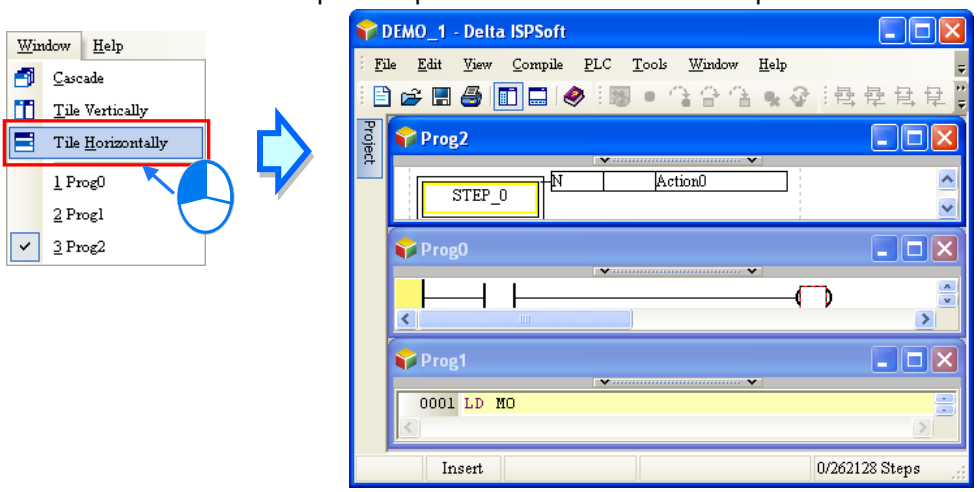
- **Cascade**: After the item is clicked, all windows will stack on top of each other, and the present window will in the front of the screen.



- **Tile Vertically:** Choosing **Tile Vertically** will cause windows to tile horizontally but take on a vertical shape. The present window will be at the leftmost side.



- **Tile Horizontally:** Choosing **Tile Horizontally** will cause windows to tile vertically but take on a horizontal shape. The present window will be the topmost window.



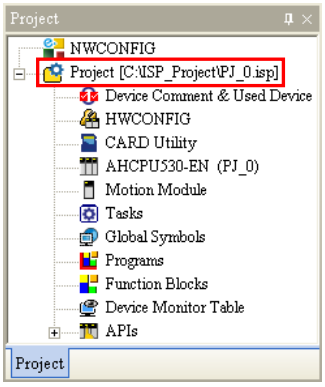
2.2 Project Framework

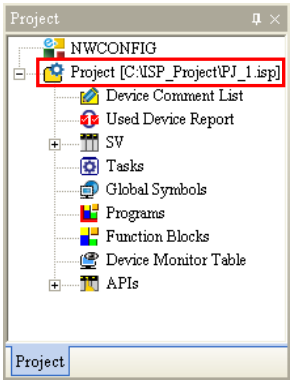
2.2.1 Single Project and Group of Projects

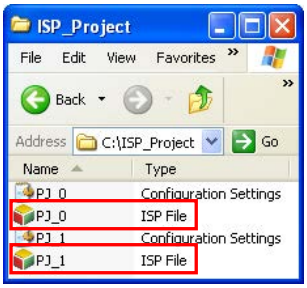
There are two types of project frameworks. They are described below.

- **A single project**

  A single project is a most basic type of project. Only one model is developed in a single project. In other word, a single project is applied to a single model. The filename extension for a project is .isp.



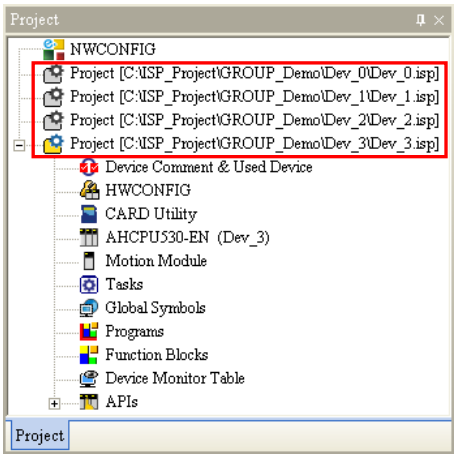


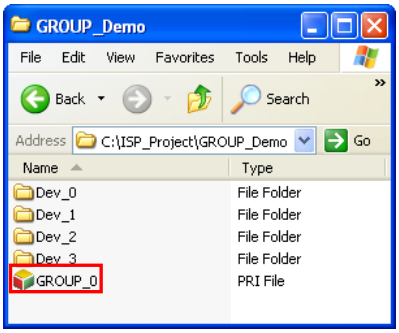


- **A group of projects**

  If several devices are connected on a network, users can create a group of single projects for the devices without restriction. Besides, the number of projects which can be created is unlimited. Users can make projects according to practical application, and they can carry out network configuration for the devices in a group of projects through **NWCONFIG** rapidly and easily.

  The filename extension for a group of projects is .pri, and the filename extension for every project in the group is .isp. The isp files are in the folders whose names are the same as the project names. Besides, users can import a single project which was created previously to a group.



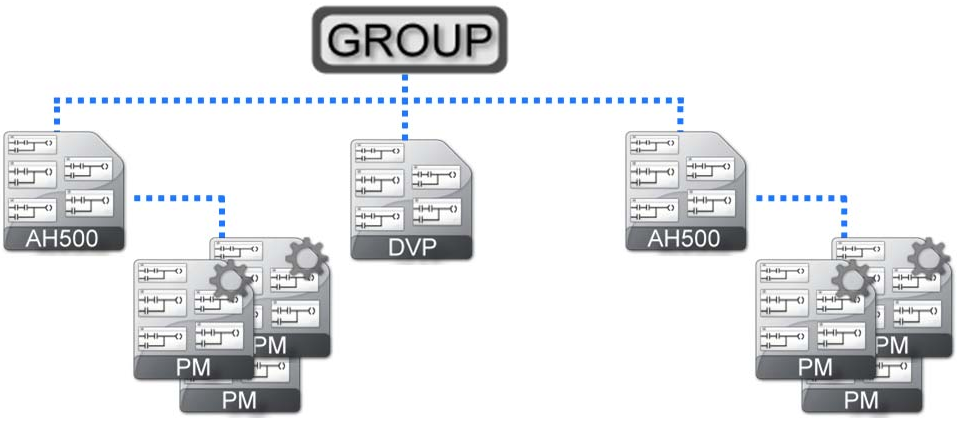


***. Please refer to chapter 16 for more information about NWCONFIG.**

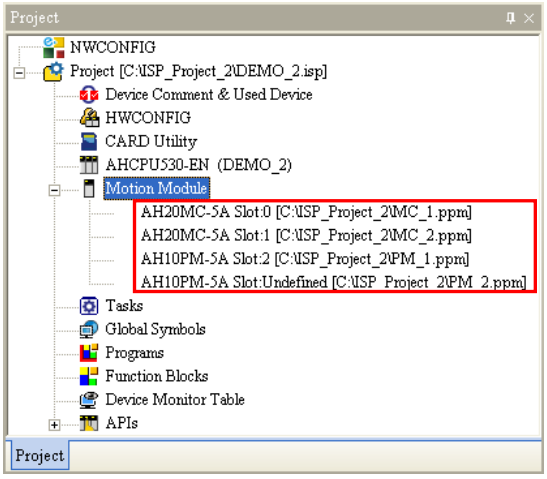
2.2.2 Integration of PMSoft into ISPSoft

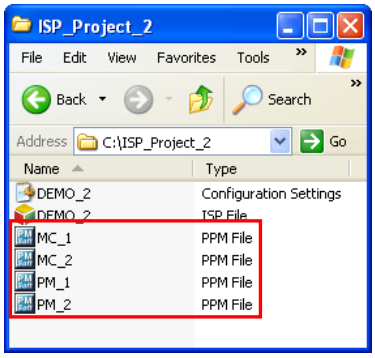
The AH500 series modules include two motion control modules. They are AH10PM-5A and AH20MC-5A. Users can write a program for a motion control module by means of PMSoft, software developed by Delta Electronics, Inc.. Please refer to PMSoft User Manual for more information about usage.

PMSoft is integrated into ISPSoft. Users can create a PMSoft project in ISPSoft, and configure hardware through **HWCONFIG**. Whether a project is a single project for an AH500 series module or a project for an AH500 series module created in a group, users can create a PMSoft in the project. The number of PMSoft projects which can be created is unlimited. However, whether a PMSoft project is executed or not depends on hardware configuration carried out through **HWCONFIG**. Please refer to section 2.2.5 for more information.



The filename extension for a PMSoft projects is .ppm. A PMSoft project and a single project in which the PMSoft project is created are saved in the same folder. Users can import a PMSoft project which was created previously to an ISPSoft project.





***1. To create a PMSoft project, PMSoft version 2.05 or above is used. A PMSoft project can not be created in an ISPSoft project for a DVP series PLC.**

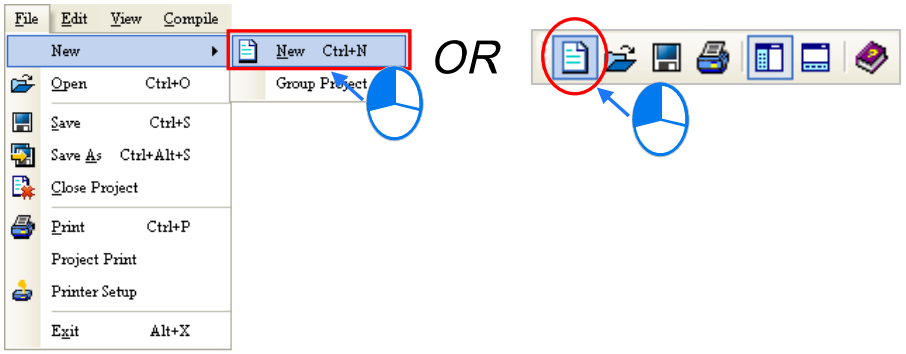
***2. A PMSoft project which is imported must be a file created with PMSoft version 2.05 or above, and a project for a DVP-PM series PLC is not supported.**

***3. Please refer to chapter 3 for more information about HWCONFIG.**

2.2.3 Managing a Single Project

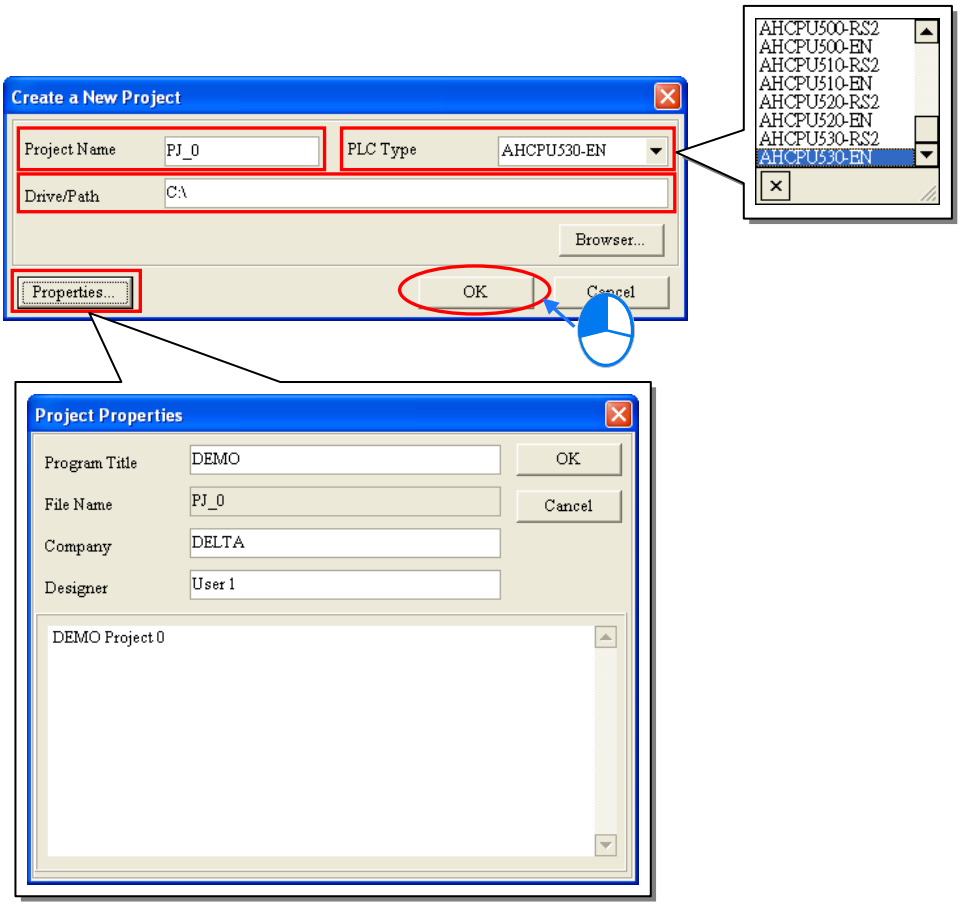
- **Creating a single project**

  User can click the **File** menu, point to **New**, and click **New** to create a new project. They can also create a new project by clicking on the toolbar.

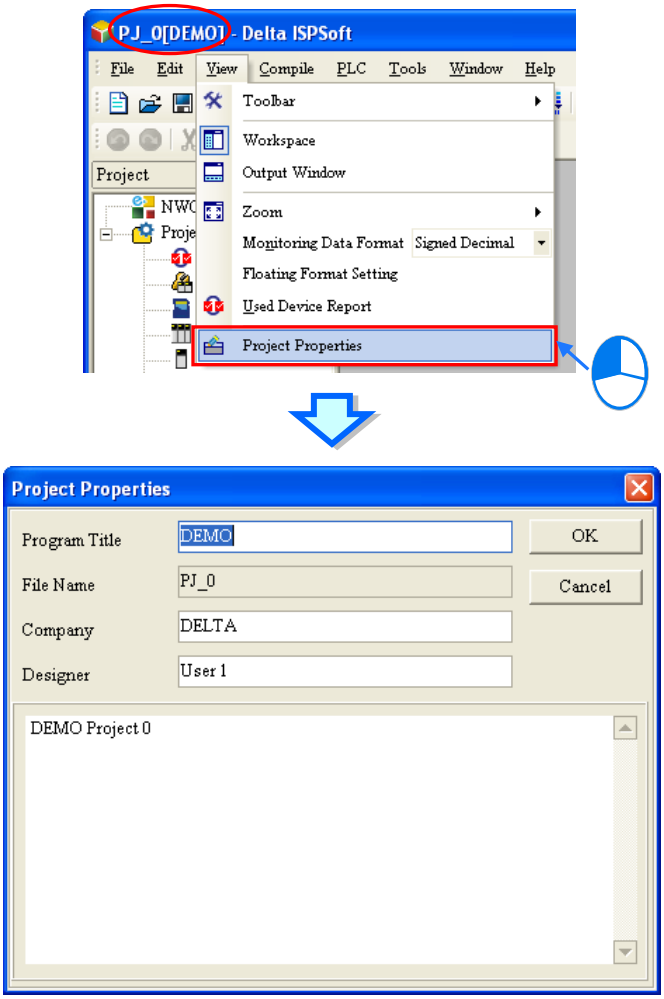


In the **Create a New Project** window, type a project name in the **Project Name** box and a path in the **Drive/Path** box, and select a PLC in the **PLC Type** drop-down list box. After users click **Properties…**, they can give a description of this project. Finally, click **OK**.

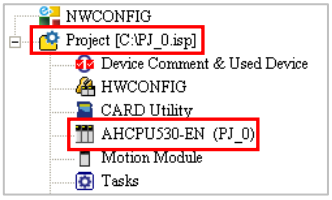
2



After a project is created successfully, the window title displays its project name and its program title. The format is **project name [program title]**. If users do not set a program title, the window title will only display a project name. Besides, if users want to view the information about a program title, they can click **Project Properties** on the **View** menu.



Users can see a path denoting a file and a model selected in the project management area. If a project created is a project for an AH500 series module, a name in parentheses will be attached to the model selected, and the name will be the project name. The name can be changed through **HWCONFIG**. Please refer to section 3.3.2.1 for more information.

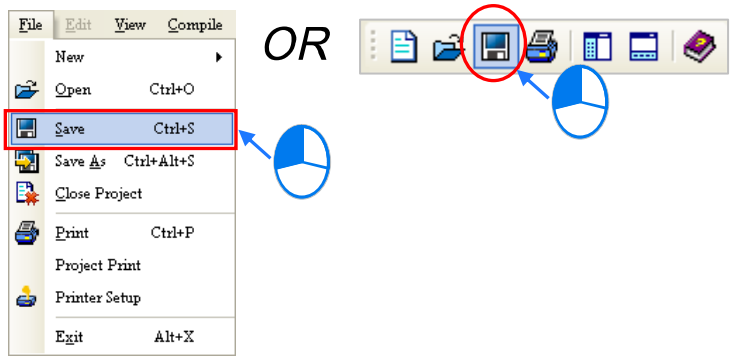


***. If a project created is a project for an AH500 series module, a name in parentheses will be attached to the model selected. If a project created is a project for a DVP series PLC, no name will be attached to the model selected. Users can identify a device by means of the name attached to it. When several devices are connected on a network, users can check whether a device connected to the computer is the device they expect by means of the name attached to the device.**

- **Saving a project**

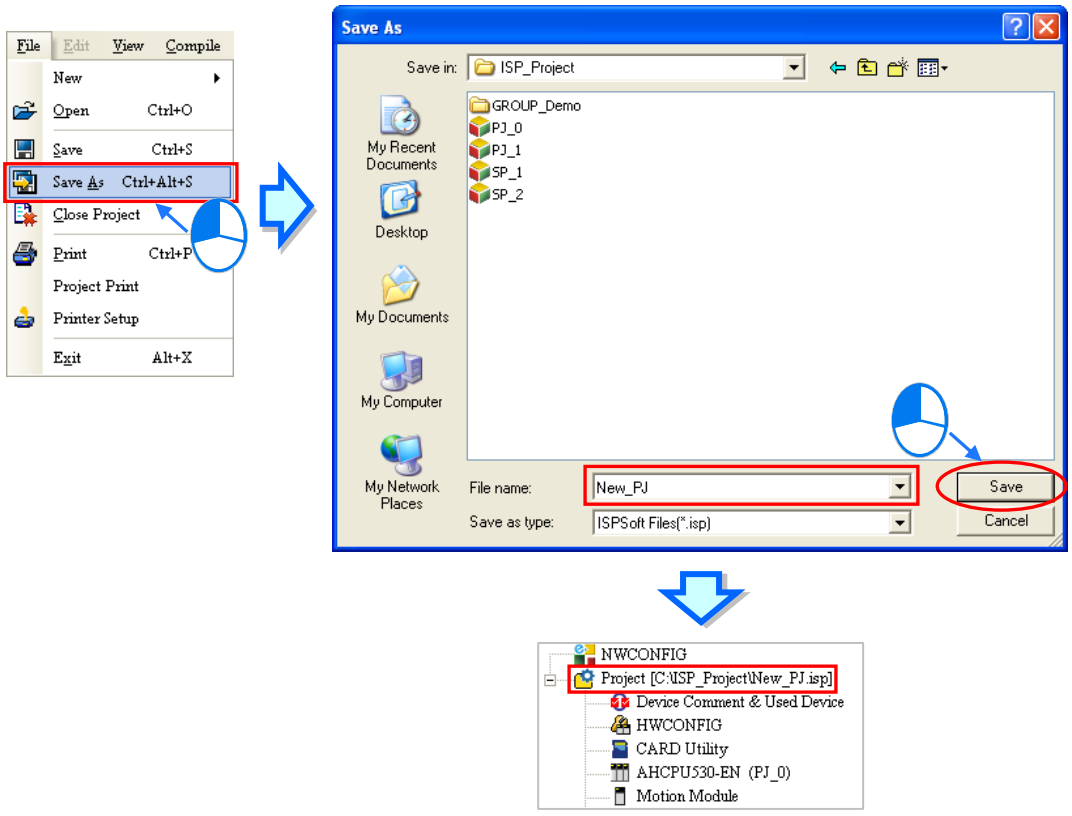
After a project is created, the file denoted by a path does not actually exist. The file will exist after users save the file. A file is saved in the following way.

Users can click the **File** menu, and click **Save** to save a file. They can also save a file by clicking on the toolbar.



2

If users want to save a file with a different name or want to specify a different path, they can click **Save As** on the **File** menu, specify a new path, type a file name in the **File name** box, and click **Save**.



***. After a file is saved with a different name, the name attached to the model selected will not change. The name attached to the model is the project name, and can be changed through HWCONFIG.**

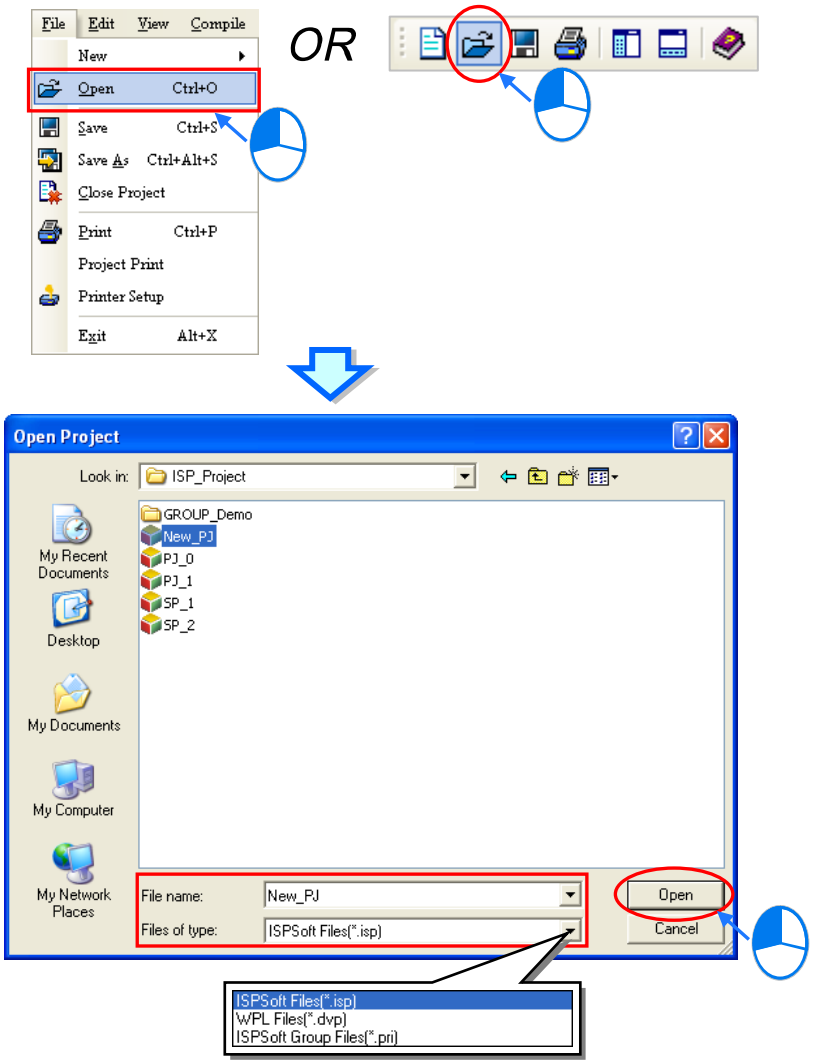
- **Opening an old project**

If users want to open a project which was created previously, they can click **Open** on the **File** menu, or click on the toolbar. After users click **Open** on the **File** menu, or click on the toolbar, they have to select a file format in the **Files of type** drop-down list box in the **Open Project** window, select a file they want to open, and click **Open**. The file formats which are supported are as follows.

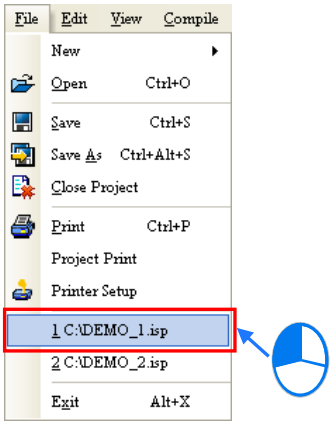
File type	Description
ISPSoft Files (*.isp)	ISPSoft files are single projects created with ISPSoft. If an old file format is opened, it is automatically converted to a new file format by ISPSoft when it is saved.
WPL Files (*.dvp)	WPL files are projects created with WPLSoft. If a WPL file is opened, it is automatically converted to an isp file by ISPSoft when it is saved.

File type	Description
ISPSoft Group Files (*.pri)	ISPSoft group files are projects in a group.

***. WPLSoft is also programming software released by Delta Electronics, Inc.. Please refer to WPLSoft User Manual for more information.**



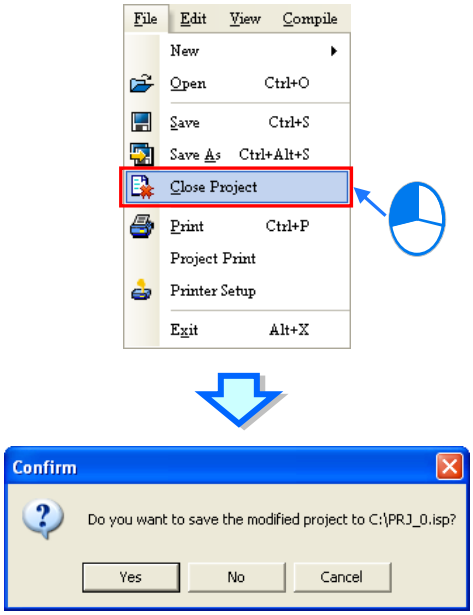
Besides, users can select a project recently edited on the **File** menu.



- **Closing a project**

  After the editing of a project is complete, users can click **Close Project** on the **File** menu to close

the project. If the project has not been saved, a window inquiring whether users want to save the project appears.



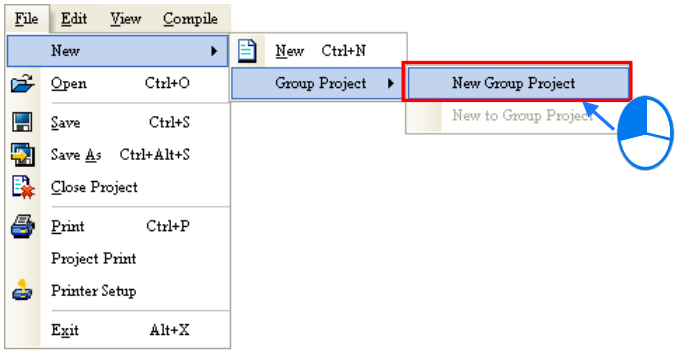
Additional remark

If users do not close ISPSoft before they close a project, the project will be opened automatically next time ISPSoft is started. (Users can decide whether a project will be automatically opened. Please refer to section 2.3.1 for more information.)

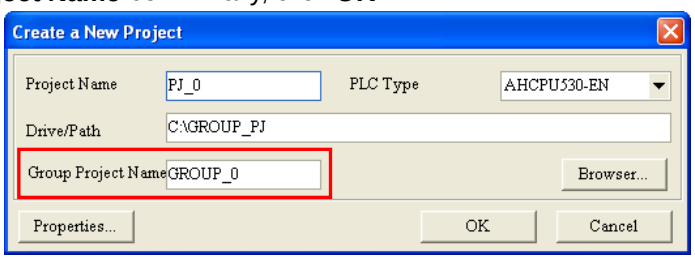
2.2.4 Managing a Group of Projects

- **Creating a group of projects**

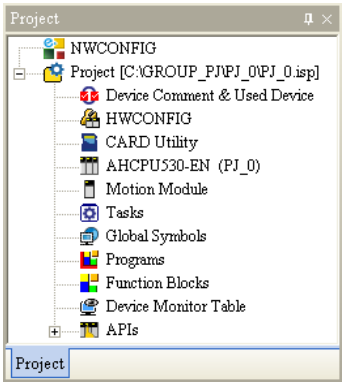
  Click the **File** menu, point to **New**, point to **Group Project**, and click **New Group Project**.



If users want to create a group of projects, they have to create the first single project in the group. In the **Create a New Project** window, type a project name in the **Project Name** box and a path in the **Drive/Path** box, and select a PLC in the **PLC Type** drop-down list box. After users click **Properties…**, they can give a description of this project. Besides, they have to type a group name in the **Group Project Name** box. Finally, click **OK**.

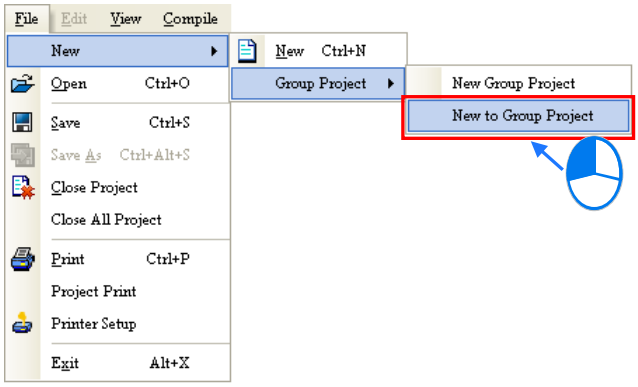


After the first single project is created, the information related to the first single project will be displayed in the project management area. Owing to the fact that there is only one project in the group, it is displayed in a way in which a single project is displayed, and it can be saved in a way in which a single project is saved. The only difference between the project in the group and a single project is that users can not click **Save As** on the **File** menu for the project in the group.

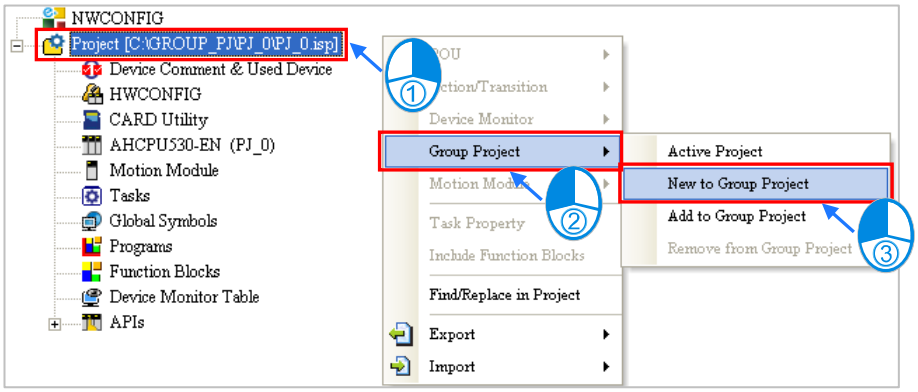


- **Adding a single project to a group**

  Users can click the **File** menu, point to **New**, point to **Group Project**, and click **New to Group Project** to add a new single project to the group. They can also right-click a project in the project management area, point to **Group Project** on the context menu, and click **New to Group Project** to add a new single project to the group.

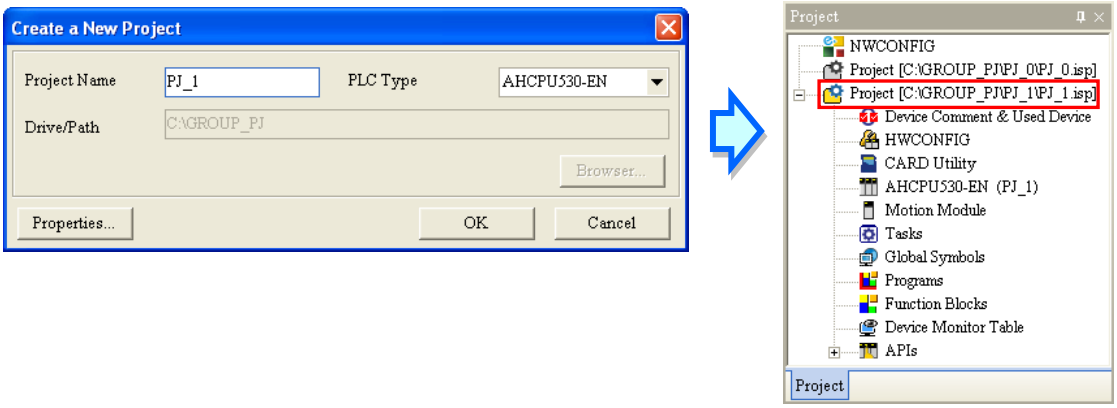


OR



Owing to the fact that one project is edited at a time, a project being edited will be closed if a new project is added to the group. If the project being edited has not been saved, a window inquiring whether users want to save the project appears.

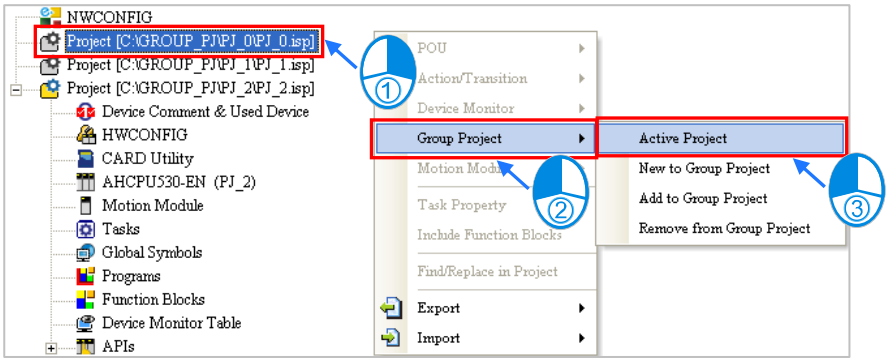
In the **Create a New Project** window, users can type a project name in the **Project Name** box, and select a PLC in the **PLC Type** drop-down list box. Besides, after users click **Properties…**, they can give a description of the project. However, users can not specify a path.



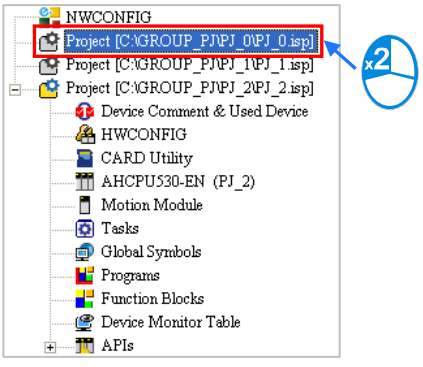
2

- **Activating a project**

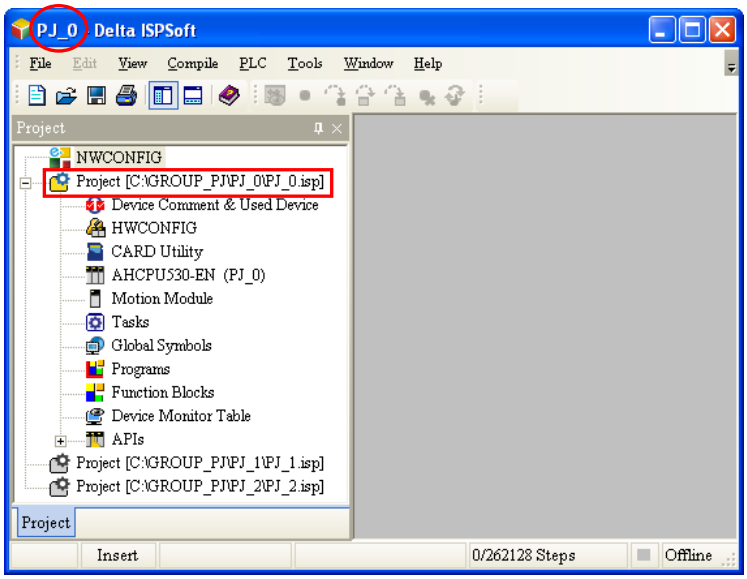
  One project is edited at a time. If users want to edit another project, they can double-click a project they want to edit, or they can right-click a project they want to edit, point to **Group Project** on the context menu, and click **Active Project**.



OR

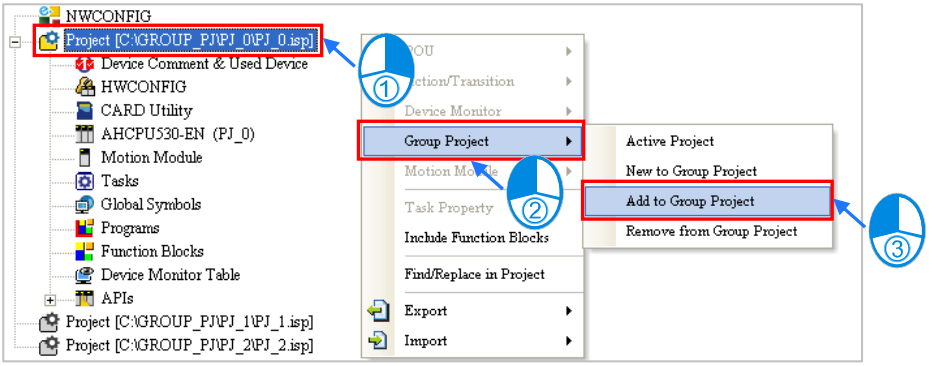


After a project is activated, the window title will be composed of the name of the project and the program title of the project. The projects which are not activated will become gray in the project management area.

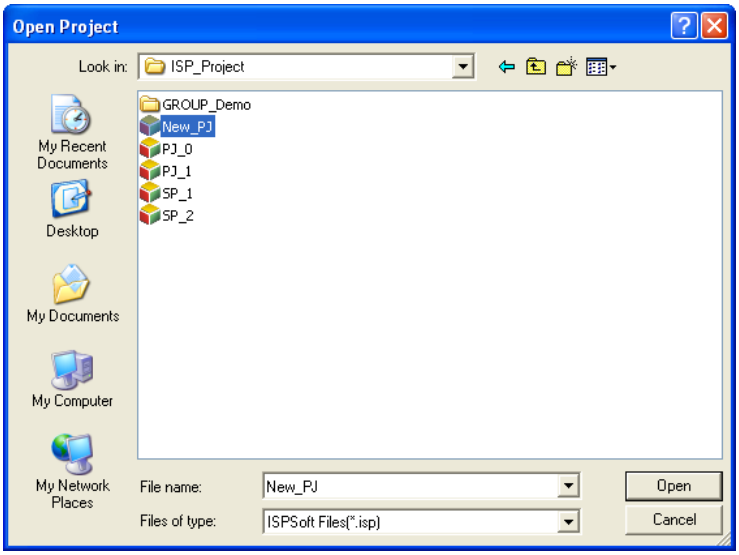


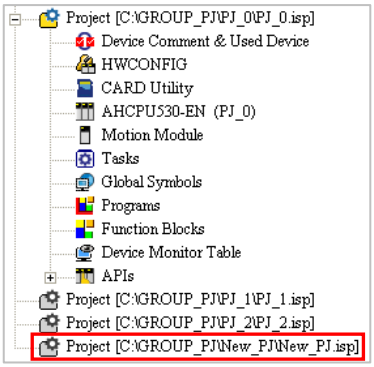
- **Importing a single project to a group**

  If an isp file is not created in a group, users can import it into the group. If users want to import a single project, they can right-click a project in the project management area, point to **Group Project** on the context menu, click **Add to Group Project**, and select a single project they want to import in the **Open Project** window.



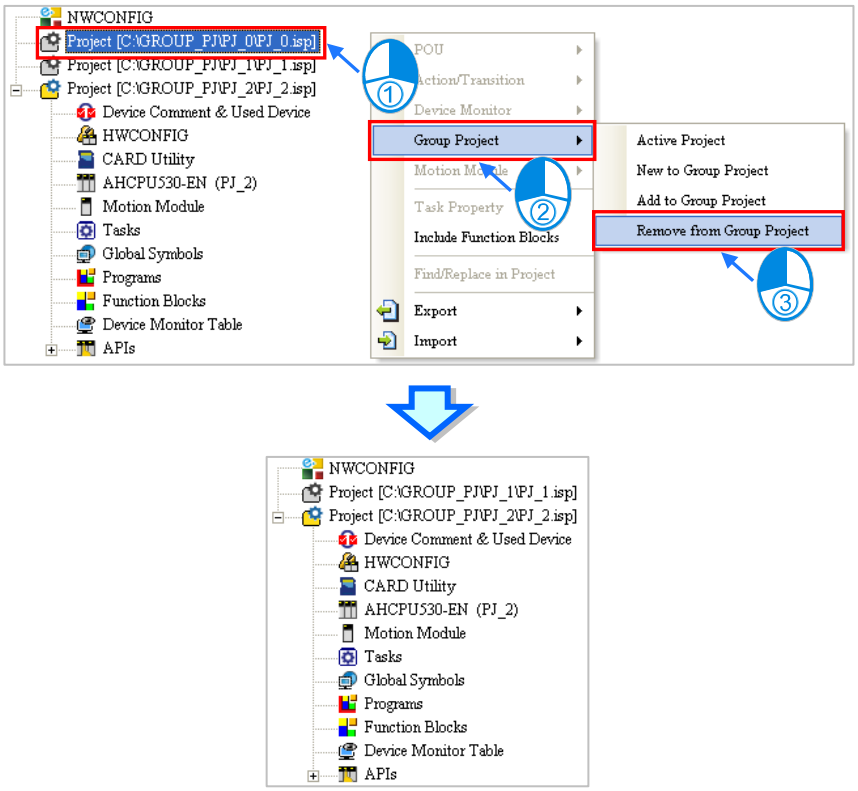
The file which can be imported is an isp file. After a project is imported into a group, the project will be copied and saved in a folder in which the group is saved, and the original project is still saved in the original folder.



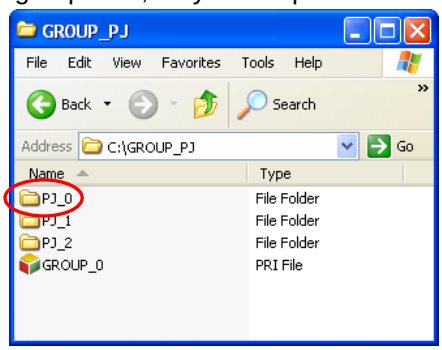


- **Removing a single project from a group**

  If users want to remove a single project from a group, they can right-click a single project they want to remove in the project management area, point to **Group Project** on the context menu, and click **Remove from Group Project**. However, if there is only one project in a group, the project can not be removed.

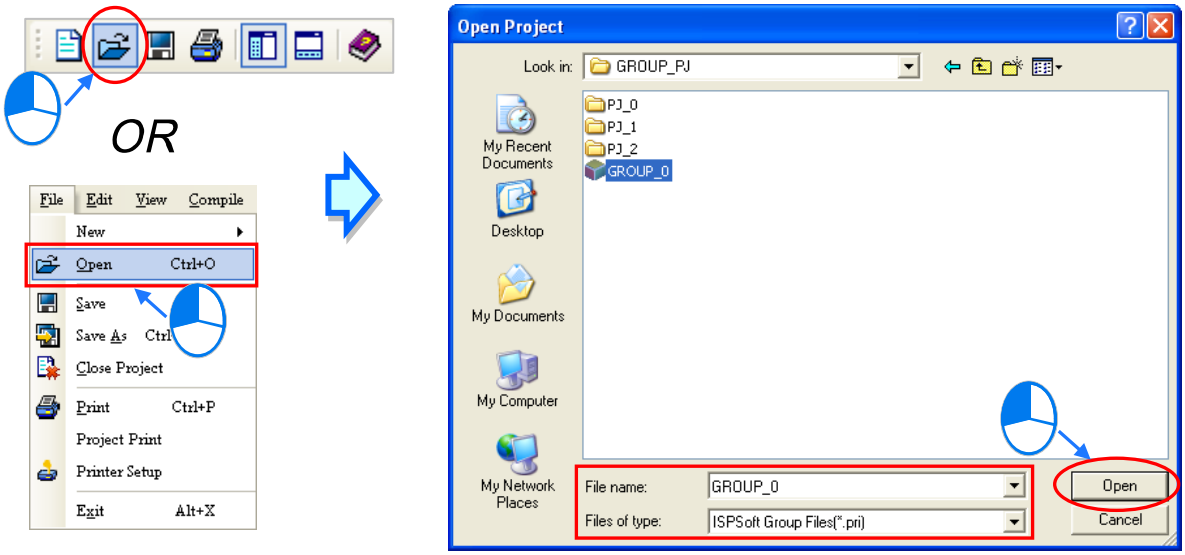


After a project is removed from a group, it will be removed from the list of projects in the project management area, and will not be deleted from the original folder in which it is saved. If users want to add the project to the group later, they can import it into the group.

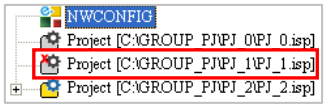


- **Opening a group of projects**

  If users want to open a group of projects created previously, they can click **Open** on the **File** menu, or on the toolbar, select **ISPSoft Group Files (*.pri)** in the **Files of type** drop-down list box in the **Open Project** window, select a file they want to open, and click **Open**.

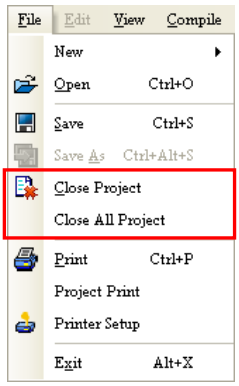


If a project in a group which is opened is marked with a red cross, the project saved in the folder in which the group is saved is abnormal. The project may be deleted or saved in a different folder, or the filename of the project is changed.



- **Closing a group of projects**

  Users can click **Close Project** on the **File** menu to close a project which is being edited. Other projects which are not activated can still be activated. If users want to close a whole group of projects, they can click **Close All Project** on the **File** menu.



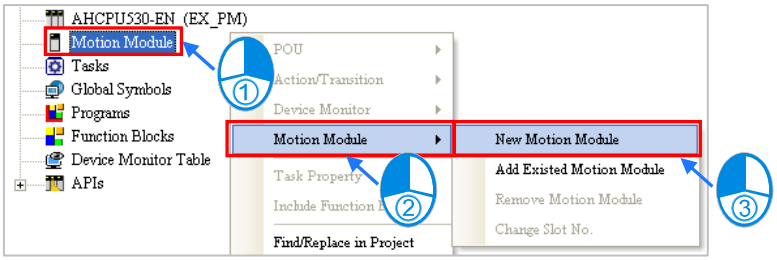
2.2.5 Managing a PMSoft Project through ISPSoft

The AH500 series modules include two motion control modules. They are AH10PM-5A and AH20MC-5A. Users can write a program for a motion control module by means of PMSoft, software developed by Delta Electronics, Inc..

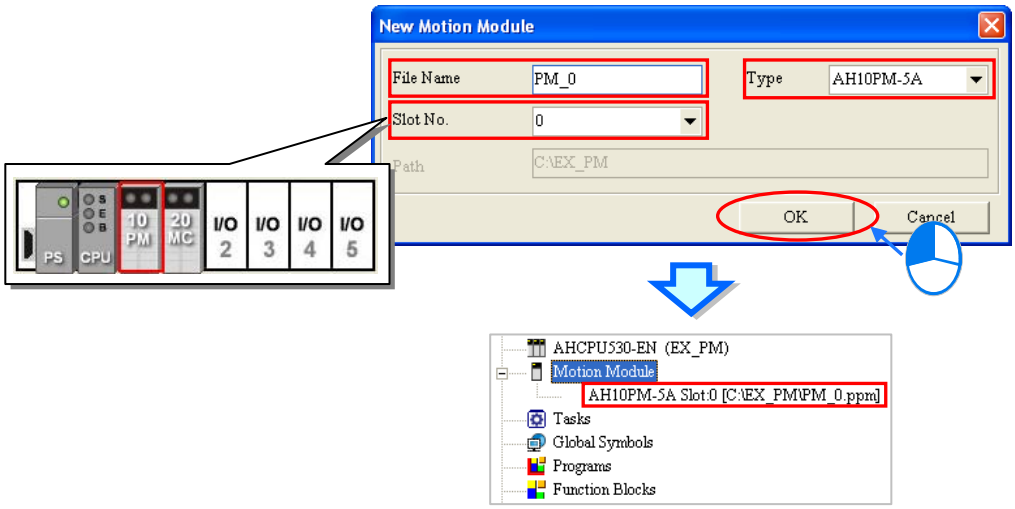
If a project is created for an AH500 series module, the **Motion Module** item will appear in the project management area. If PMSoft version 2.05 or above is installed on a computer, users can manage PMSoft projects by means of the **Motion Module** item.

- **Creating a new PMSoft project**

  If users want to create a PMSoft project with ISPSoft, they have to right-click **Motion Module** in the project management area, point to **Motion Module** on the context menu, and click **New Motion Module**.



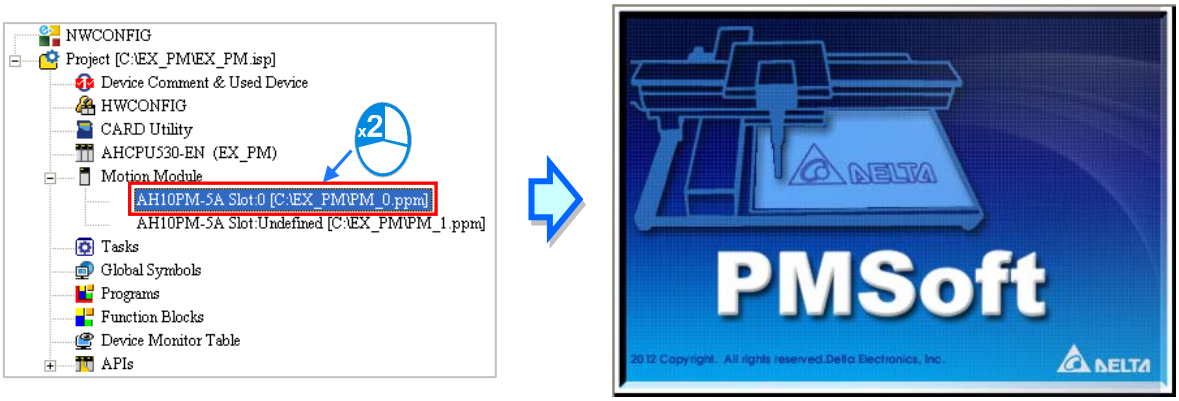
In the **New Motion Module** window, type a file name in the **File Name** box, select a model in the **Type** drop-down list box, and select a slot number in the **Slot No.** drop-down list box according to the hardware configuration carried out through **HWCONFIG**. Every slot number corresponds to a PMSoft project. If a slot number is not set, or if users want to change a slot number later, users can select **Undefined** in the **Slot No.** drop-down list box. After the setting is complete, users can click **OK**.



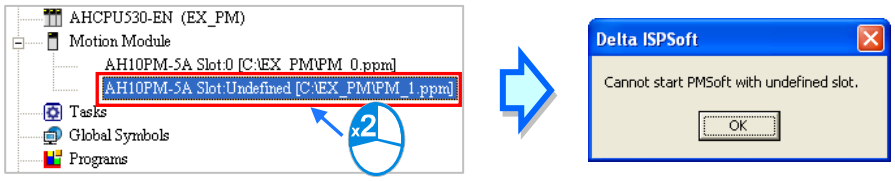
***. Please refer to chapter 3 for more information about HWCONFIG.**

- **Developing a PMSoft project**

If users want to edit a project for a motion control module which has been created, they have to double-click the project to start PMSoft. The users can write a program for the motion control module in PMSoft.



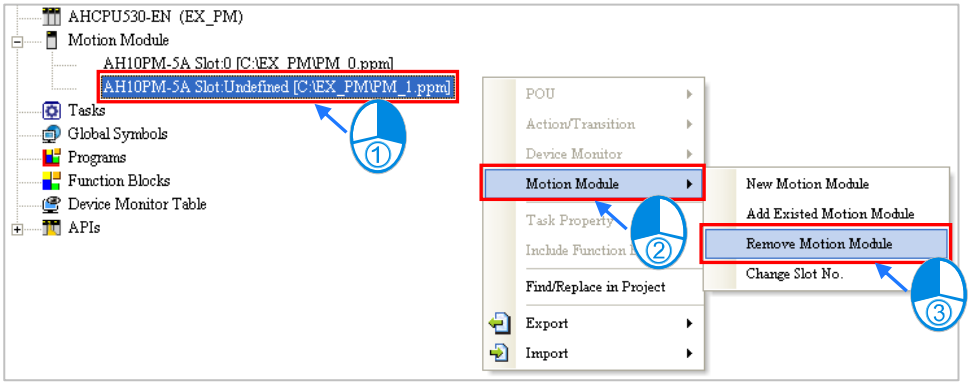
However, if users do not specify a slot for a motion control module, they are not allowed to edit the project for the motion control module.



***. Please refer to PMSoft User Manual for more information about the usage of PMSoft. If PMSoft is started through ISPSoft, users can not create a new file, open an old file, and change a model in PMSoft. Besides, users must save a program written in PMSoft by clicking Save on the File menu in PMSoft.**

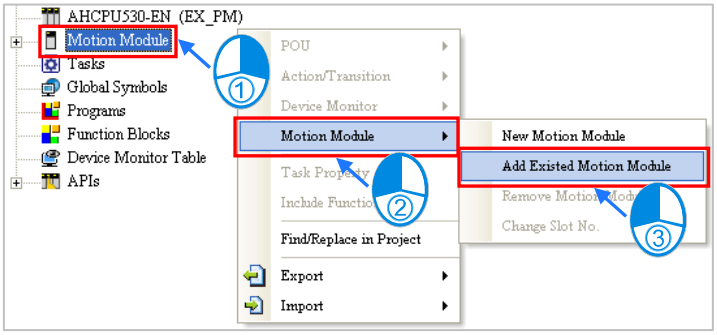
- **Deleting a PMSoft project**

If users want to delete a project for a motion control module, they can right-click a project they want to delete in the project management area, point to **Motion Module** on the context menu, and click **Remove Motion Module**. After a project for a motion control module is deleted, it will be removed from the list of projects in the project management area, and will not be deleted from the original folder in which it is saved.

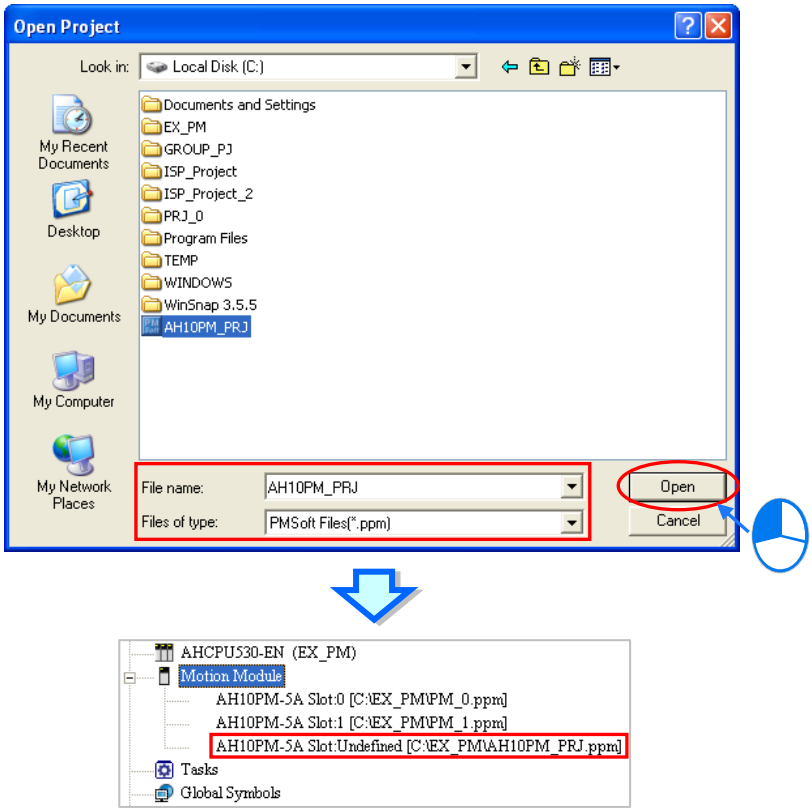


- **Importing a PMSoft project**

If users want to import a PMSoft project (a ppm file) which was created previously to an ISPSoft project, they have to right-click **Motion Module** in the project management area, point to **Motion Module** on the context menu, and click **Add Existed Motion Module**.



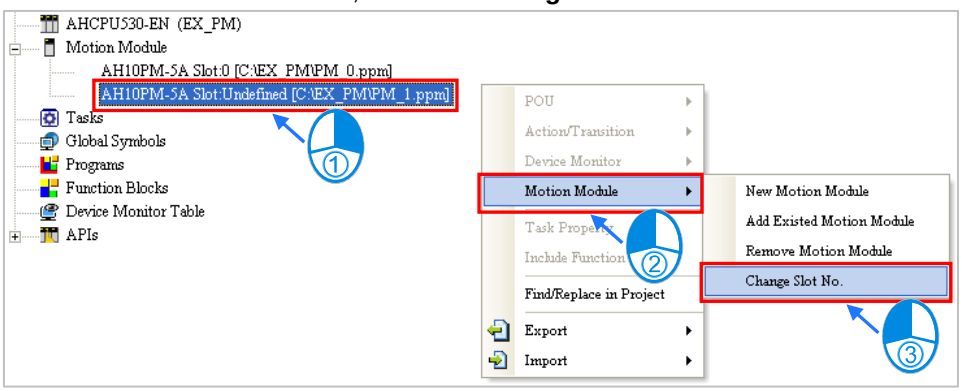
Users have to select a PMSoft project which they want to import in the **Open Project** window, and click **Open**. The slot number in a PMSoft project imported is undefined.



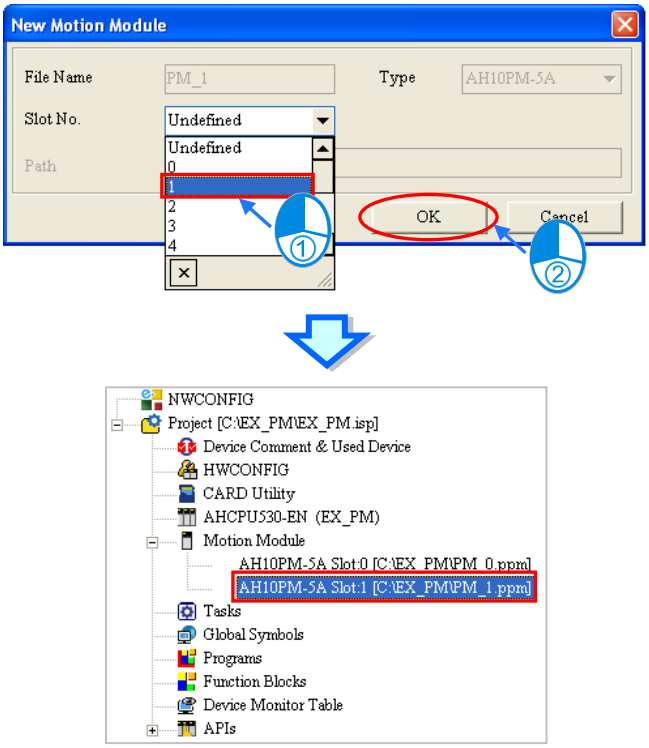
After a PMSoft project is imported into an ISPSoft project, the PMSoft project will be copied and saved in a folder in which the ISPSoft project is saved, and the original PMSoft project is still saved in the original folder. Besides, a PMSoft project which is imported must be a file created with PMSoft version 2.05 or above, and a project for a DVP-PM series PLC is not supported.

- **Changing a slot number**

  If users want to change the slot number in a project, they have to right-click the project, point to **Motion Module** in the context menu, and click **Change Slot No.**.



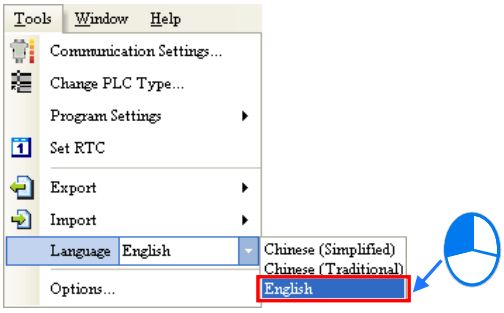
Users have to select a slot number in the **Slot No.** drop-down list box in the **New Motion Module** window. They can not select a slot number which has been selected. Finally, click **OK**.



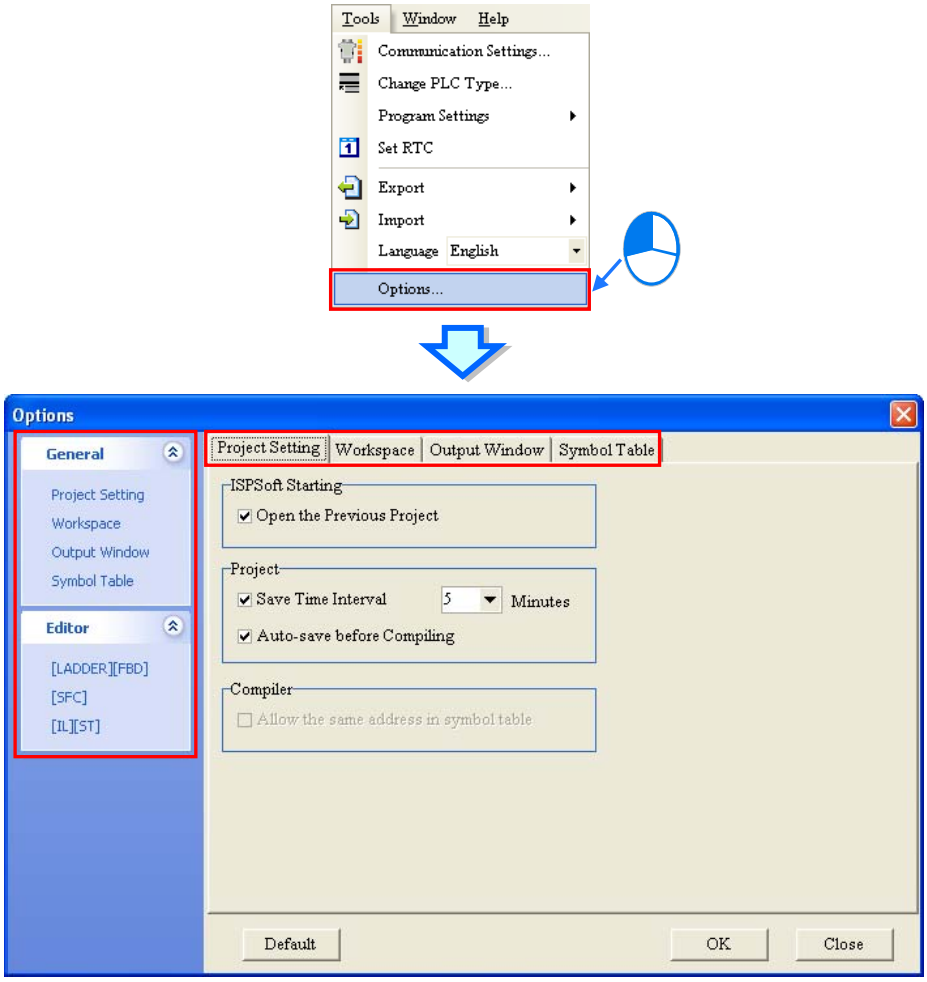
2.3 Basic Setting in ISPSoft

2.3.1 System Setting and Environment Setting

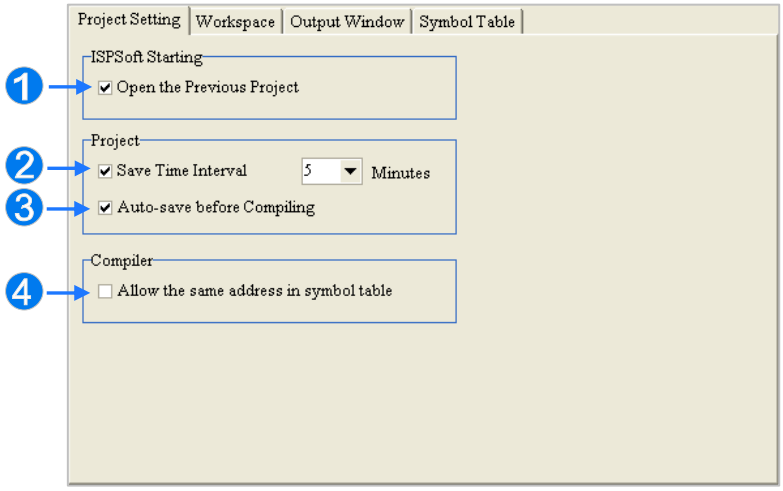
Users can select a language in the **Language** drop-down list box on the **Tools** menu.



After users click **Options…** on the **Tools** menu, the **Options** window will appear. The users can click the items at the left side of the window, and the tabs at the top of the window.



- **System setting–Project Setting**

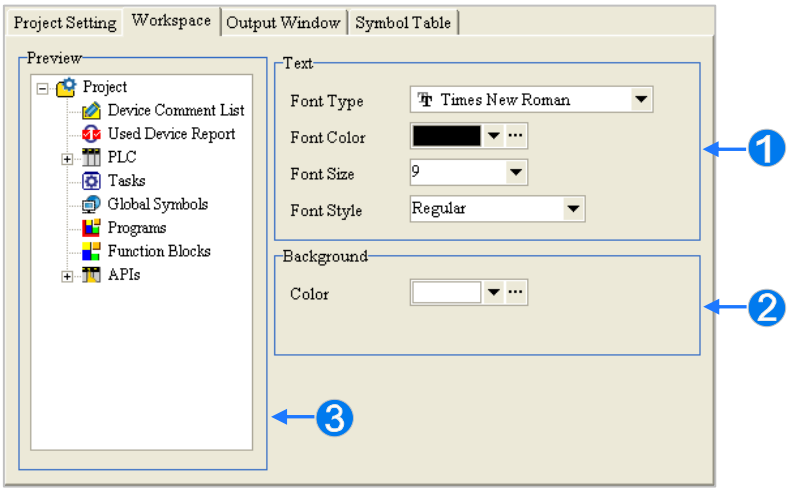


1. If this item is selected, a project which was edited last and which was not closed will be opened automatically whenever ISPSoft is started.
2. If this item is selected, a project which is being edited is backed up as a temporary file every specific period of time. (Users can select a time interval in the **Minutes** drop-down list box.) If ISPSoft is not closed normally, users will be asked whether they want to load the last backup project next time the project is opened.
3. If this item is selected, a project which is being edited is saved automatically before the

program in the project is compiled.

❹ If this item is selected, different symbols can refer to the same device. If a project is created for an AH500 series module, the item can not be selected.

- **System setting–Workspace**

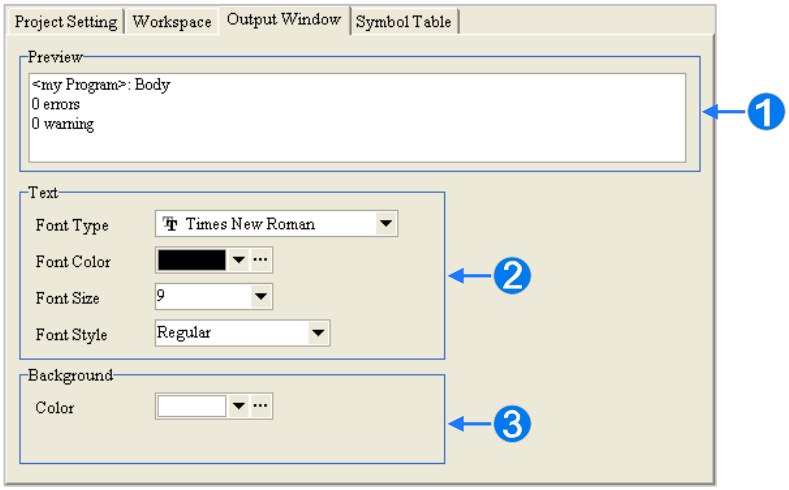


❶ Users can select a font for the project management area.

❷ Users can select a background color of the project management area.

❸ Users can preview changes here.

- **System setting–Output Window**

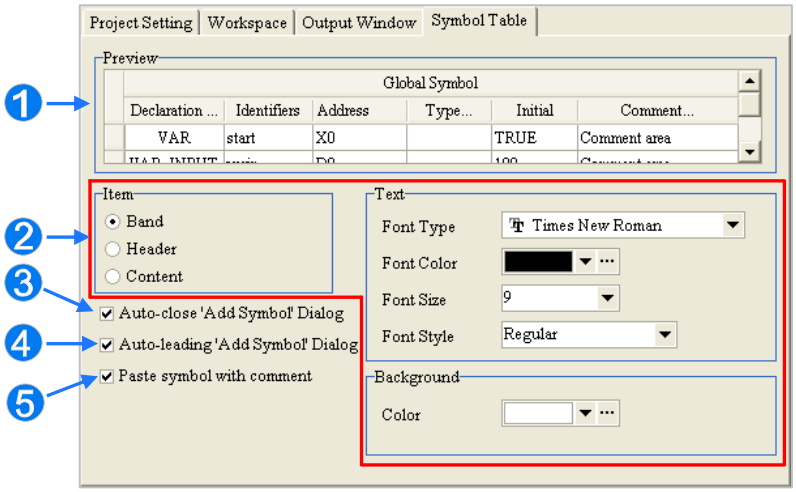


❶ Users can preview changes here.

❷ Users can select a font for the message display area.

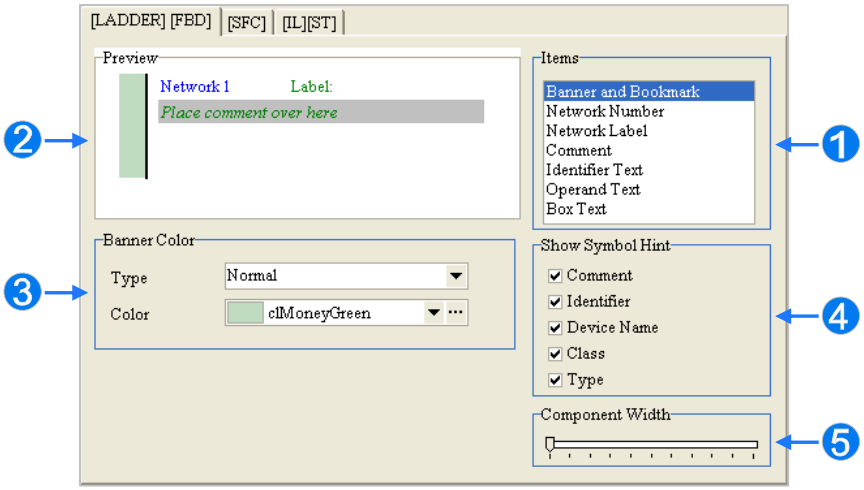
❸ Users can select a background color of the message display area.

- **System setting–Symbol Table**



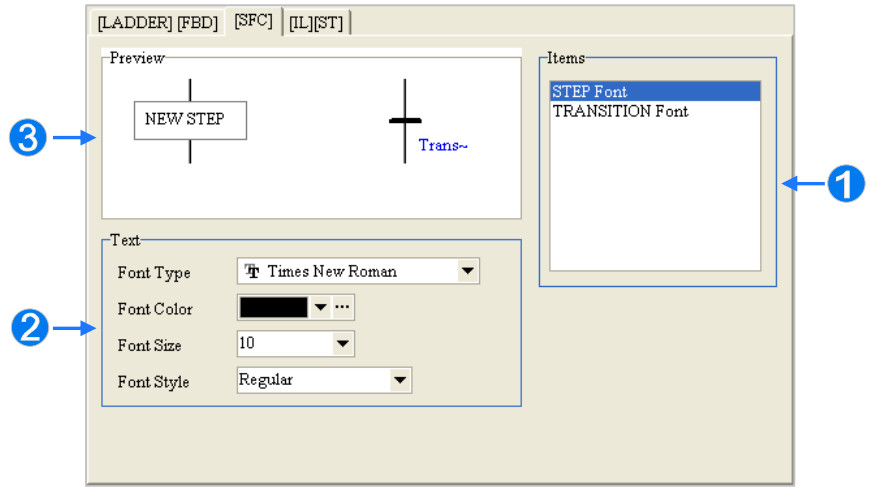
1. Users can preview changes here.
2. Users have to select an item in the **Item** section, and then select a font in the **Text** section.
3. If this item is selected, the **Add Symbol** window will close automatically after a symbol is declared.
4. If this item is selected, the **Add Symbol** window will appear automatically after users type an undefined symbol in a ladder diagram (LD), function block diagram (FBD), or a sequential function chart (SFC), and press Enter on the keyboard.
5. If a symbol is cut/copied, the symbol pasted will include a comment.

- **Editor–[LADDER] [FBD]**



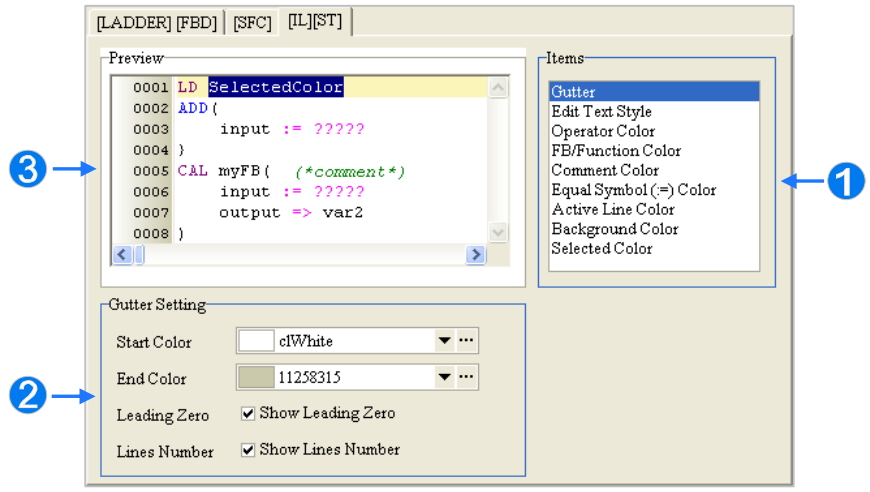
1. Users have to select an item here.
2. Users can preview changes here.
3. Users can select a type and a color.
4. After the mouse cursor stays at an object for a while, the hints users select in the **Show Symbol Hint** section will appear.
5. Users can set the widths of blocks. The setting will affect the lengths of symbols in blocks.

- **Editor–[SFC]**



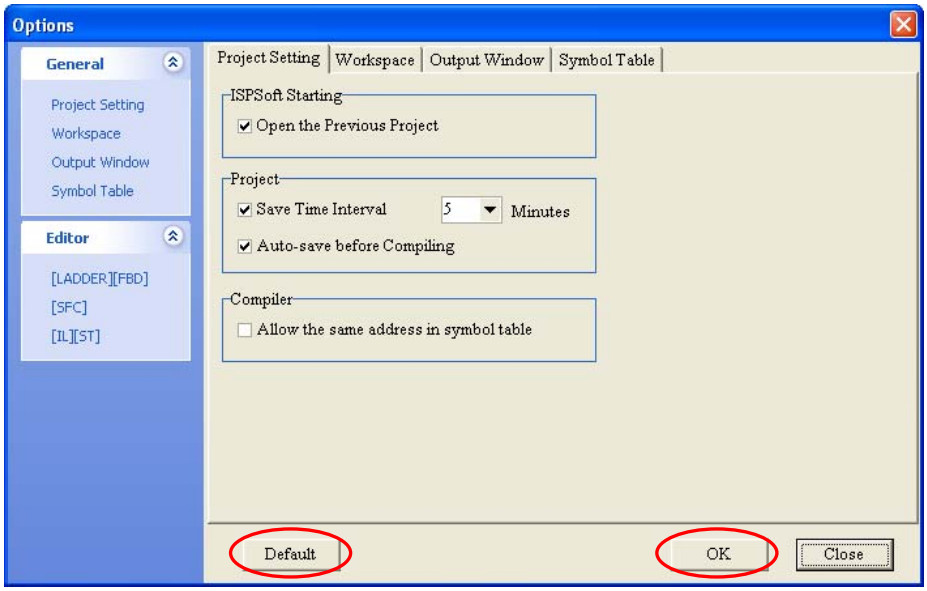
1. Users have to select an item here.
2. Users can select a font here.
3. Users can preview changes here.

- **Editor–[IL] [ST]**



1. Users have to select an item here.
2. Users can select display styles here.
3. Users can preview changes here.

After Default is clicked, all setting values will become default values. After users complete all setting, they can click **OK** to apply changes.



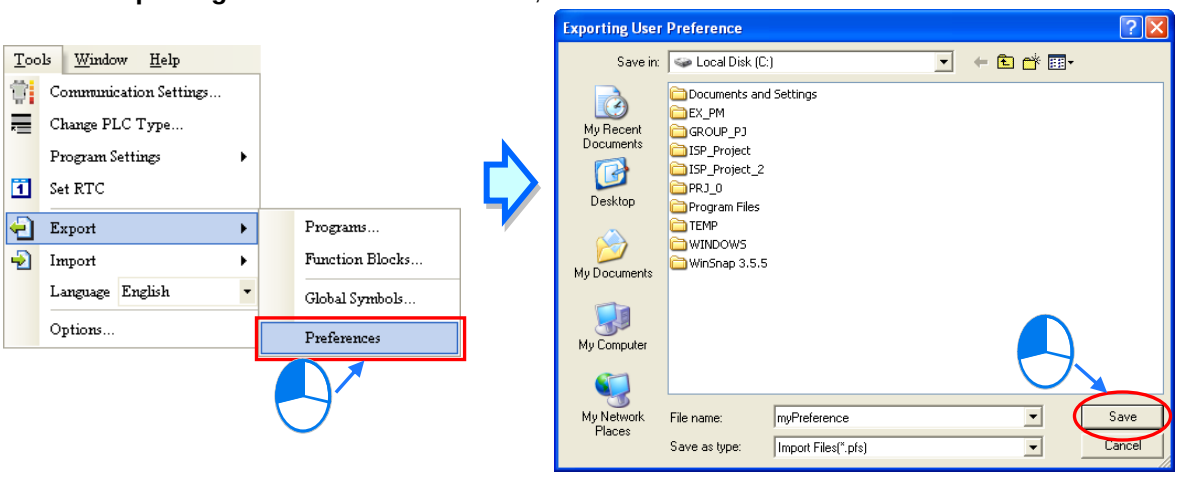
***. If several ISPSoft windows are opened, users can set them differently. However, only the environment setting of an ISPSoft window which is closed last is memorized. As a result, the setting values in an ISPSoft window which is closed last will be applied next time ISPSoft is started.**

2.3.2 Importing and Exporting Preferences

Users can export and import environment setting through the functions provided by ISPSOft.

- **Exporting preferences**

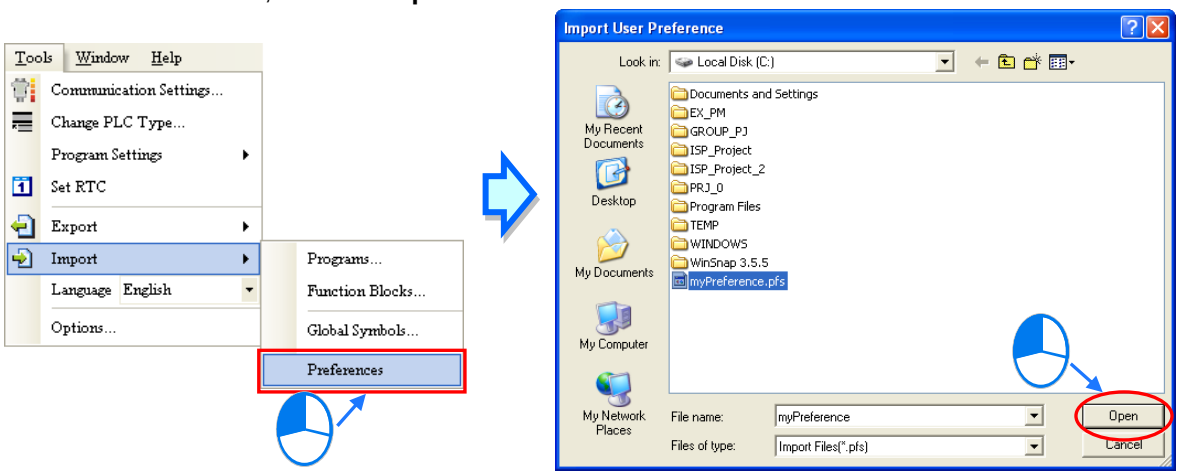
  Click the **Tools** menu, point to **Export**, click **Preferences**, type a file name in the **File name** box in the **Exporting User Preference** window, and click **Save**.



- **Importing preferences**

  Click the **Tools** menu, point to **Import**, click **Preferences**, select a file in the **Import User Preference** window, and click **Open**.

2

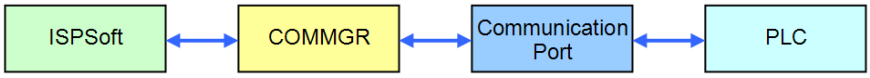


Additional remark

Owing to the fact that environment parameters vary with versions of ISPSoft, environment setting imported is not necessarily applicable. If users do not know which version of ISPSoft is a preference file (*.pfs) is exported from, they can check whether the contents of the file meets their expectation after the file is imported.

2.4 Communication Setting

The communication between ISPSoft and a Delta PLC is shown below. The communication manager **COMMGR** is a communication interface between ISPSoft and a PLC. This section introduces how to create a connection between ISPSoft and a PLC, and complete a basic test.

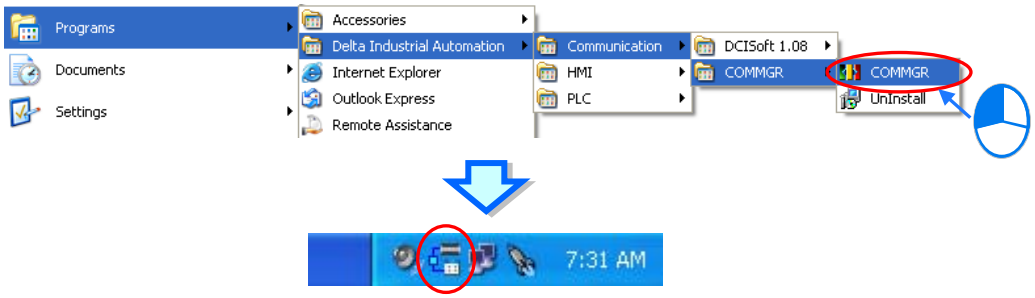


***1. Please refer to section 1.2 for more information about the installation of COMMGR.**

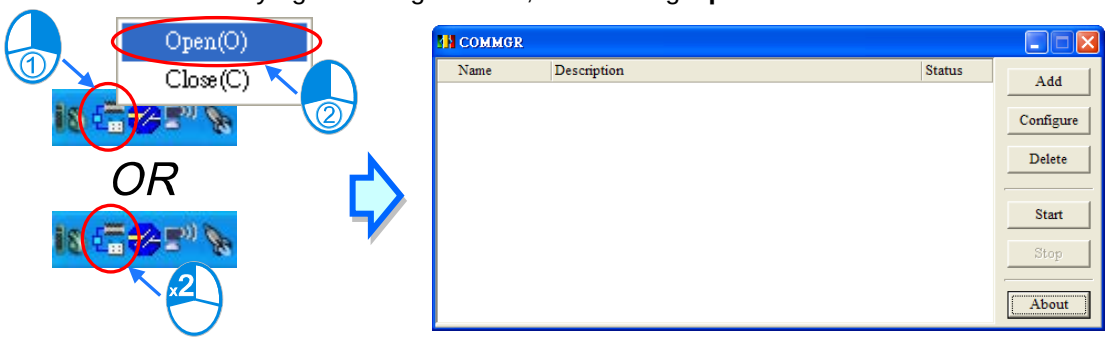
***2. COMMGR is used with ISPSoft version 2.0 and above. An older version of ISPSoft still communicates with a PLC in a traditional way.**

2.4.1 Starting / Closing COMMGR

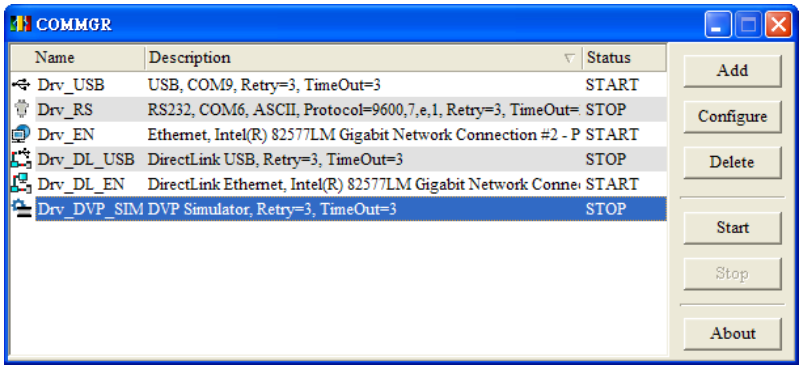
After COMMGR is installed on a computer successfully, a shortcut to COMMGR is created on the **Start** menu. Users can click the shortcut on the **Start** menu to start COMMGR. After the installation of COMMGR is complete, users have to start it by themselves. However, whenever the computer is restarted, the system starts COMMGR automatically, and the icon representing COMMGR is displayed on the system tray. If the icon representing COMMGR is not displayed on the system tray, users can start COMMGR by clicking the shortcut on the **Start** menu.



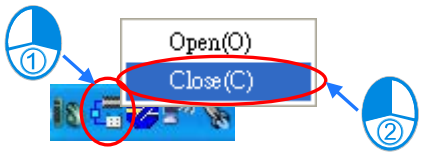
After COMMGR is started successfully, the icon representing COMMGR will be displayed on the system tray. Users can open the **COMMGR** window by double-clicking the icon. They can also open the **COMMGR** window by right-clicking the icon, and clicking **Open** on the context menu.



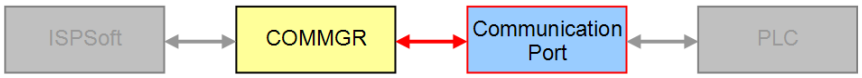
The **COMMGR** window is shown below. The drivers created are listed in the window. ISPSoft connects to COMMGR by means of specifying a driver. Users can manage the drivers through the buttons at the right side of the window. Please refer to the section below for more information about managing drivers.



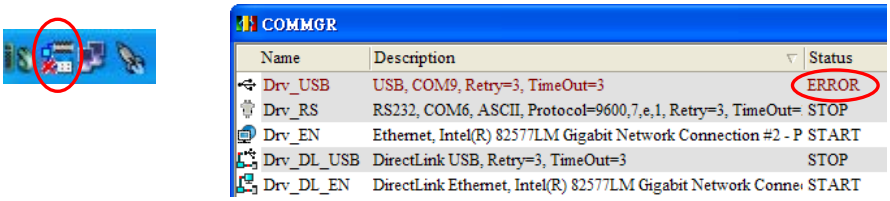
Users can close the **COMMGR** window by clicking or in the upper right corner of the window. However, the icon representing COMMGR is still displayed on the system tray. If users want to close COMMGR completely, they can right-click the icon displayed on the system tray, and click **Close** on the context menu.



2.4.2 Managing Drivers

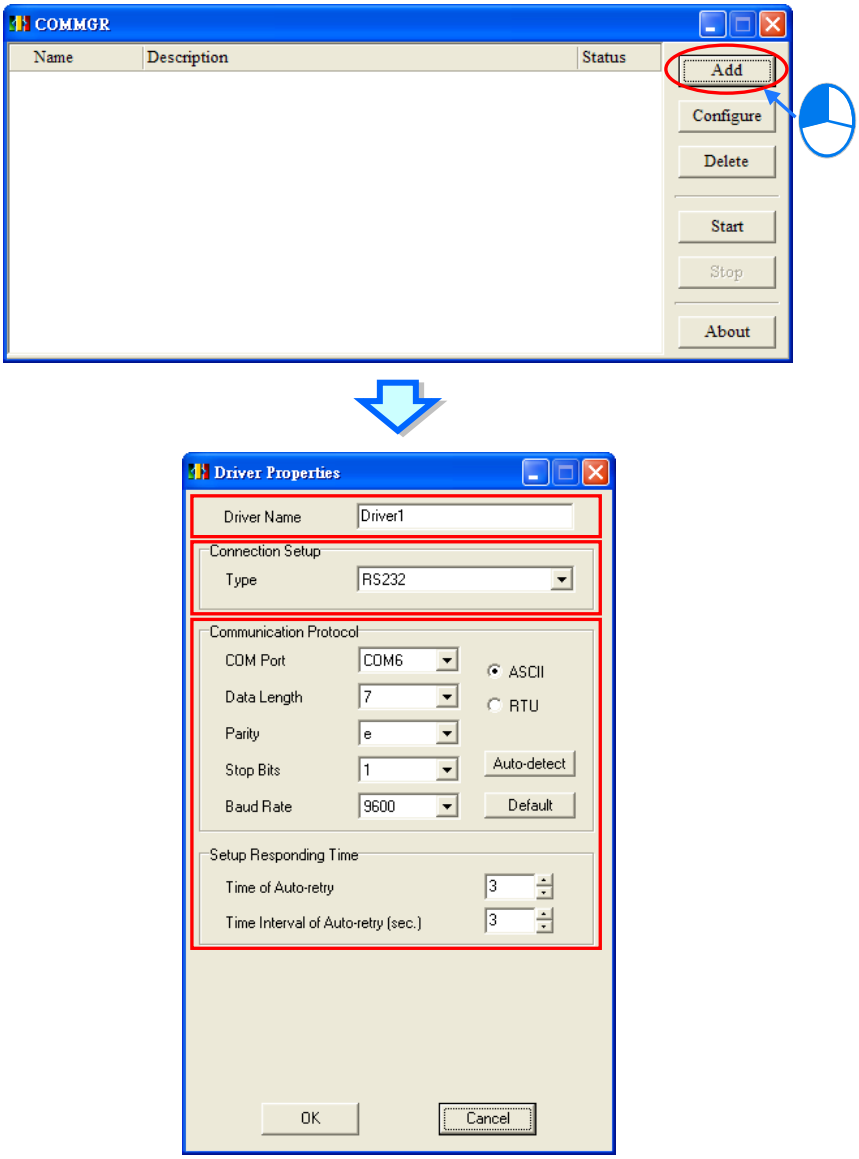


The drivers listed in the **COMMGR** window connect programs and communication ports. If the status of a driver displayed in the **COMMGR** window is **START**, COMMGR connects to the communication port specified by the driver. Whenever the computer is restarted, COMMGR starts the driver automatically. However, if COMMGR can not connect to the communication port specified by a driver, COMMGR automatically stops the driver, the status of the driver displayed in the window is **ERROR**, and the icon representing COMMGR on the system tray is marked with a red cross.



2.4.3 Creating a Connection–Creating a Driver

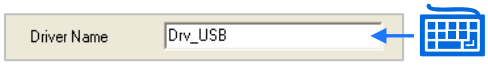
Click **Add** in the **COMMGR** window to open the **Driver Properties** window.



The steps of creating a driver are as follows.

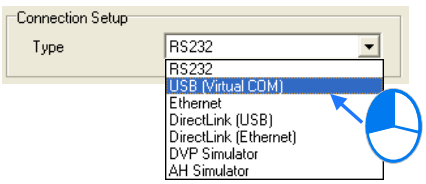
(1) **Driver name**

Users can type a driver name in the **Driver Name** box. A driver name is composed of 31 characters at most. Special marks such as *, #, ?, \, %, @, and etc. can not be used except _.



(2) **Connection type**

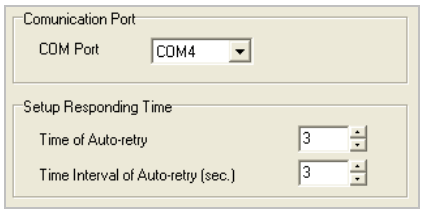
Users can select a connection type in the **Type** drop-down list box. The connection types supported by COMMGR are as follows.



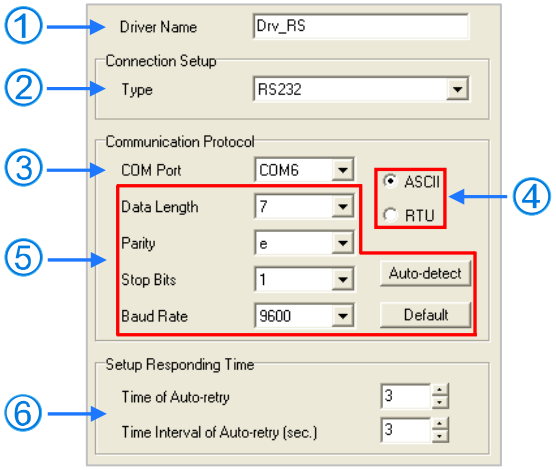
- **RS232**
  A computer communicates with a PLC through a communication port on the computer.
- **USB (Virtual COM)**
  A computer can connect to a PLC equipped with a USB port through a USB cable. However, users have to make sure that a USB driver is installed on a computer before the computer connects to a PLC equipped with a USB port. Please refer to appendix A for more information about installing a USB driver.
- **Ethernet**
  A computer communicates with a PLC through an Ethernet port on the computer.
- **DirectLink (USB) & DirectLink (Ethernet)**
  They are the connection functions provided by Delta human-machine interfaces (HMI). If a PLC connects to a HMI normally, a computer can connect to the HMI through a USB cable or Ethernet, and connect to the PLC indirectly. Please refer to manuals for Delta human-machine interfaces for more information about setting a connection.
- **DVP Simulator & AH Simulator**
  This type of driver functions as a virtual channel connecting a simulator. If users specify this type of driver in ISPSoft, COMMGR will transmit data between ISPSoft and a simulator once related connection operation is performed. It is as if COMMGR connected to a real PLC. (This mode is a framework adopted by ISPSoft version 2.00 and above.)

(3) **Communication parameters**

Communication parameters are set according to a connection type selected. Different connection types have different communication parameters. The setting of parameters for the different connection types is described in the following sections.



2.4.3.1 Setting Communication Parameters for RS232



(1) Users can type a driver name in the **Driver Name** box. Special marks can not be used except _.

(2) Select **RS232** in the **Type** drop-down list box in the **Connection Setup** section.

(3) Select a RS232 communication port in the **COM Port** drop-down list box. Each item in the **COM Port** drop-down list box is composed of a device name and a communication port number. The communication ports in the **COM Port** drop-down list box are the same as the communication ports in the **Device Manager** window.

(4) The communication format can be **ASCII** or **RTU**.

(5) The communication protocol for exchanging data through a communication port selected must be the same as the communication protocol for exchanging data through a communication port on a device connected. If users click **Default**, all communication parameters will return to the default values.

If users do not know the communication protocol for exchanging data through a communication port on a device connected, the users can connect the device to a RS232 communication port selected with a RS232 cable, and click **Auto-detect** to automatically detect the communication protocol. If the communication protocol is detected successfully, the related communication parameters in the **Driver Properties** window are set. However, when the communication protocol is detected automatically, the **COM Port** parameter and the **ASCII/RTU** parameter are not detected. As a result, the users have to set the **COM Port** parameter and the **ASCII/RTU** parameter before clicking **Auto-detect**.

(6) Users can select the number of times the sending of a command is retried if a connection error occurs in the **Time of Auto-retry** box, and select an interval of retrying the sending of a command in the **Time Interval of Auto-retry** box.

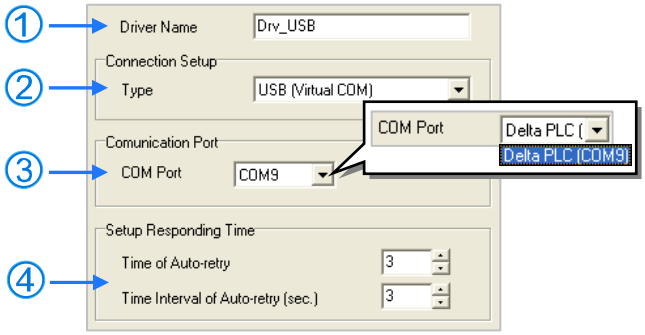
***. When the Driver Properties window is opened, the information about the communication ports in the Device Manager window is retrieved once. However, the information in the COM Port drop-down list box will not be updated. If a device is added to the computer system after the Driver Properties window is opened, the device will not be displayed in the COM Port drop-down list box. Users have to close the Driver Properties window, and open it again.**

2.4.3.2 Setting Communication Parameters for USB (Virtual COM)

If users want to connect a USB port on a computer to a PLC, they have to make sure of the items below before opening the **Driver Properties** window.

(a) A USB driver is installed on the computer.

(b) The computer is connected to the PLC through a USB cable. The computer and the PLC operate normally.



(1) Users can type a driver name in the **Driver Name** box. Special marks can not be used except _.

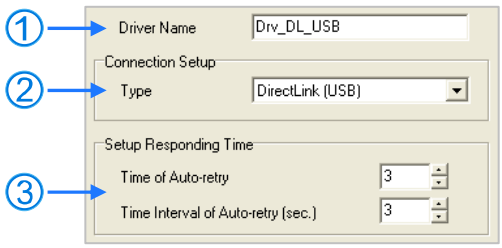
(2) Select **USB (Virtual COM)** in the **Type** drop-down list box in the **Connection Setup** section.

(3) Select a communication port in the **COM Port** drop-down list box. If users have made sure of the two items above, the PLC which is connected and its communication port will be displayed in the **COM Port** drop-down list box.

(4) Users can select the number of times the sending of a command is retried if a connection error occurs in the **Time of Auto-retry** box, and select an interval of retrying the sending of a command in the **Time Interval of Auto-retry** box.

***. Please refer to appendix A for more information about installing a USB driver.**

2.4.3.3 Setting Communication Parameters for DirectLink (USB)

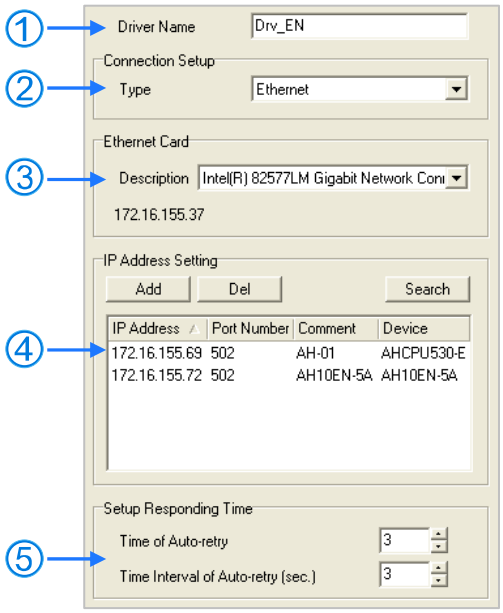


(1) Users can type a driver name in the **Driver Name** box. Special marks can not be used except _.

(2) Select **DirectLink (USB)** in the **Type** drop-down list box in the **Connection Setup** section.

(3) Users can select the number of times the sending of a command is retried if a connection error occurs in the **Time of Auto-retry** box, and select an interval of retrying the sending of a command in the **Time Interval of Auto-retry** box.

2.4.3.4 Setting Communication Parameters for Ethernet



(1) Users can type a driver name in the **Driver Name** box. Special marks can not be used except _.

(2) Select **Ethernet (USB)** in the **Type** drop-down list box in the **Connection Setup** section.

(3) Select a network interface card in the **Description** drop-down list box. An IP address assigned to a network interface card selected is displayed in the lower left corner of the **Ethernet Card** section.

(4) Owing to the characteristics of Ethernet, a computer can communicate with all devices on a network. Users can create the IP addresses of devices connected to this driver in the **IP Address Setting** section.

- Some devices support the **Search** function. After users click **Search** to search for IP addresses, results are displayed in the **IP Address Setting** section.
- After users click **Add** to add a new IP address to the list of IP addresses in the **IP Address Setting** section, they can type related information in the **IP Address** section, the **Port Number** column, and the **Comment** column.

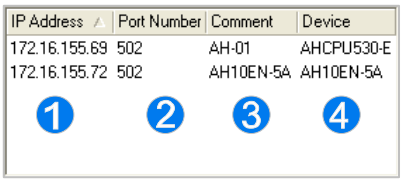
  ❶ Users can type the IP address of a device connected in this column.

  ❷ Users can type a communication port number specified. If it is not necessary to specify a

communication port number, please use the default communication port number 502.

❸ Users can type a comment in this column.

❹ Users can not type a device in this column. After users click **Search**, a device which is found is displayed in this column.

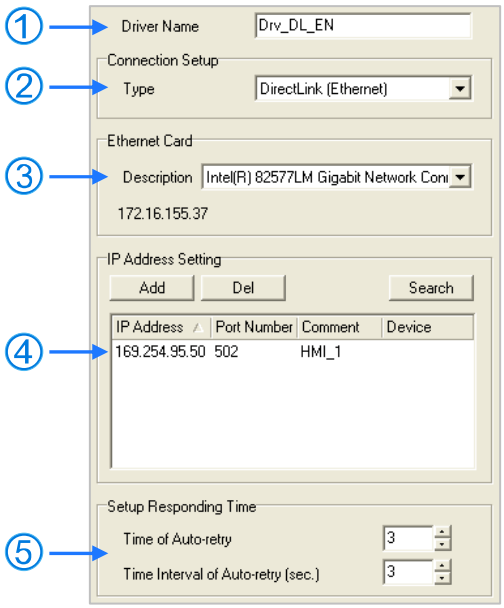


➢ After users select an IP address, they can click **Del** or press DEL on the keyboard to delete the IP address from the list.

(5) Users can select the number of times the sending of a command is retried if a connection error occurs in the **Time of Auto-retry** box, and select an interval of retrying the sending of a command in the **Time Interval of Auto-retry** box.

***. When the Driver Properties window is opened, the information about the network interface cards in the computer is retrieved once. However, the information in the Description drop-down list box will not be updated. If a network interface card is added to the computer system after the Driver Properties window is opened, the network interface card will not be displayed in the Description drop-down list box. Users have to close the Driver Properties window, and open it again.**

2.4.3.5 Setting Communication Parameters for DirectLink (Ethernet)



(1) Users can type a driver name in the **Driver Name** box. Special marks can not be used except _.

(2) Select **DirectLink (Ethernet)** in the **Type** drop-down list box in the **Connection Setup** section.

(3) Select a network interface card in the **Description** drop-down list box. An IP address assigned to a network interface card selected is displayed in the lower left corner of the **Ethernet Card** section.

(4) Owing to the characteristics of Ethernet, a computer can communicate with all devices on a network. Users can create the IP addresses of devices connected to this driver in the **IP Address Setting** section.

➢ Delta human-machine interfaces support the **Search** function. After users click **Search** to search for IP addresses, results are displayed in the **IP Address Setting** section.

➢ After users click **Add** to add a new IP address to the list of IP addresses in the **IP Address Setting** section, they can type related information in the **IP Address** section, the **Port**

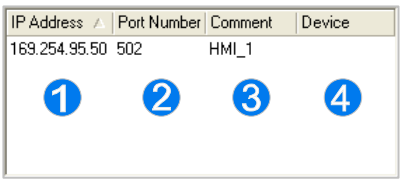
Number column, and the **Comment** column.

❶ Users can type the IP address of a device connected in this column.

❷ Users can type a communication port number specified. If it is not necessary to specify a communication port number, please use the default communication port number 502.

❸ Users can type a comment in this column.

❹ No device will be displayed in this column even if a device is found.

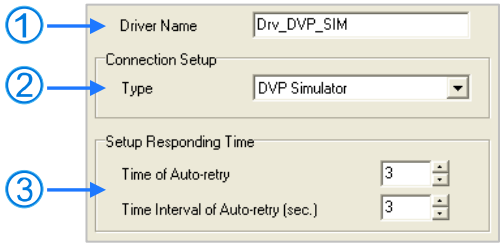


➢ After users select an IP address, they can click **Del** or press DEL on the keyboard to delete the IP address from the list.

(5) Users can select the number of times the sending of a command is retried if a connection error occurs in the **Time of Auto-retry** box, and select an interval of retrying the sending of a command in the **Time Interval of Auto-retry** box.

***. When the Driver Properties window is opened, the information about the network interface cards in the computer is retrieved once. However, the information in the Description drop-down list box will not be updated. If a network interface card is added to the computer system after the Driver Properties window is opened, the network interface card will not be displayed in the Description drop-down list box. Users have to close the Driver Properties window, and open it again.**

2.4.3.6 Setting Communication Parameters for a DVP Simulator

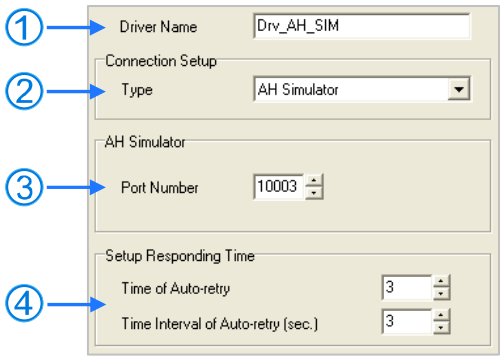


(1) Users can type a driver name in the **Driver Name** box. Special marks can not be used except _.

(2) Select **DVP Simulator** in the **Type** drop-down list box in the **Connection Setup** section.

(3) Users can select the number of times the sending of a command is retried if a connection error occurs in the **Time of Auto-retry** box, and select an interval of retrying the sending of a command in the **Time Interval of Auto-retry** box.

2.4.3.7 Setting Communication Parameters for a AH Simulator



(1) Users can type a driver name in the **Driver Name** box. Special marks can not be used except

_.

(2) Select **AH Simulator** in the **Type** drop-down list box in the **Connection Setup** section.

(3) Users can type a communication port number specified. If it is not necessary to specify a communication port number, please use the default communication port number.

(4) Users can select the number of times the sending of a command is retried if a connection error occurs in the **Time of Auto-retry** box, and select an interval of retrying the sending of a command in the **Time Interval of Auto-retry** box.

Additional remark

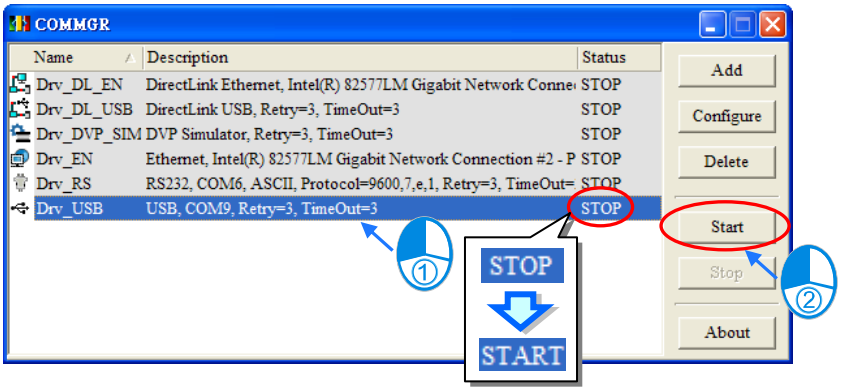
(a) Only one DVP simulator is allowed to be created. If users want to create the second DVP simulator, a warning message will appear.

(b) Users can create several AH simulator. However, no more than eight simulators can be started simultaneously. Besides, users have to assign different communication port numbers to different AH simulators.

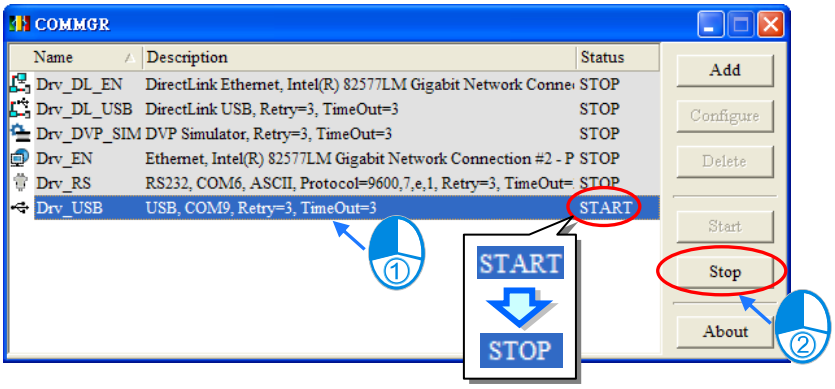
(c) Only COMMGR version 1.01 and above support AH simulators.

2.4.4 Creating a Connection–Starting / Stopping a Driver

If users want to start a driver, they can select the driver on the **COMMGR** window, and click **Start**. If the status of the driver displayed in the window becomes **START**, the driver is started successfully.



If users want to stop a driver, they can select the driver in the **COMMGR** window, and click **Stop**. If the status of the driver displayed in the window becomes **STOP**, the driver stops.



Additional remark

When users start drivers, they have to notice the following items.

(a) If the same communication port number is assigned to different drivers, only one of these drivers can be started.

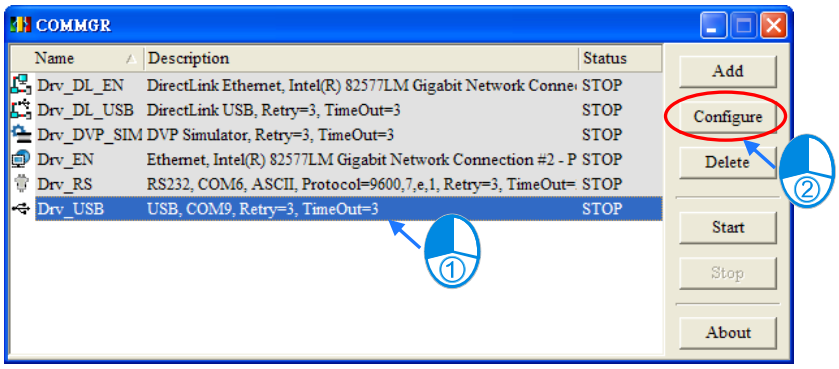
(b) If a driver to which a communication port is assigned is started, other software which is not connected through COMMGR can not use the communication port assigned to the driver. If the software wants to use the communication port, users have to stop the driver.

(c) If a communication port is used by other software not connected through COMMGR, a driver to which the communication port is assigned can not be started. If users want to start the driver,

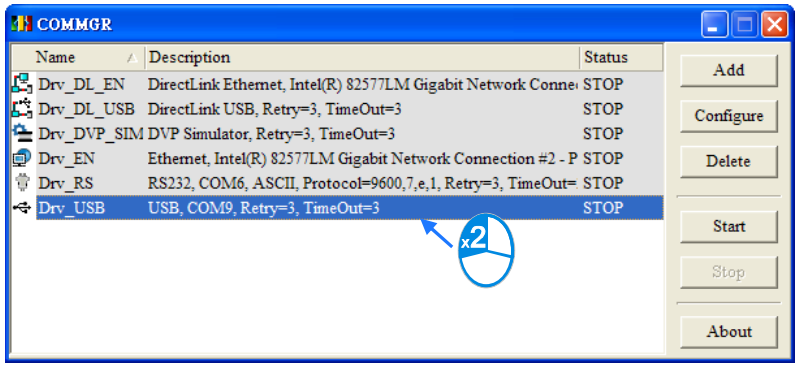
they have to close the software, or the software has to use another communication port.

2.4.5 Creating a Connection–Configuring / Deleting a Driver

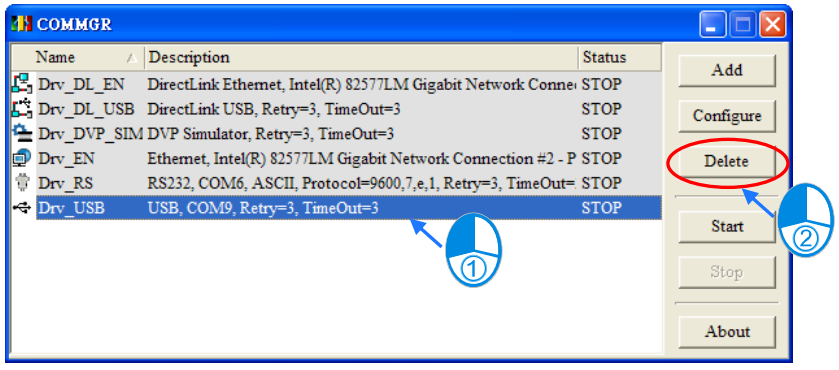
If users want to modify parameters for a driver, they have to stop the driver, and click **Configure**, or double-click the driver to open the **Driver Properties** Window. The users can set the parameters in the **Driver Properties** Window according to the description in section 2.4.3.



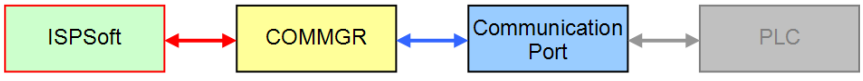
OR



If users want to delete a driver, they have to make sure that the driver stops, select the driver, and click **Delete**, or press DEL on the keyboard to delete the driver.

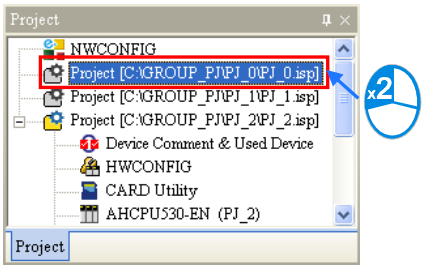


2.4.6 Creating a Connection Between ISPSoft and COMMGR

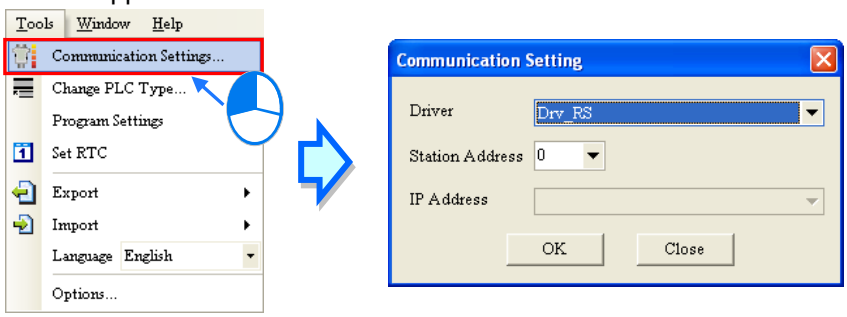


After drivers are created and started in COMMGR, users can specify drivers in ISPSoft. After the setting is complete, a connection between ISPSoft and COMMGR is created. The users have to specify a driver for every project. As a result, the users have to open the projects in a group, and specify drivers for them.

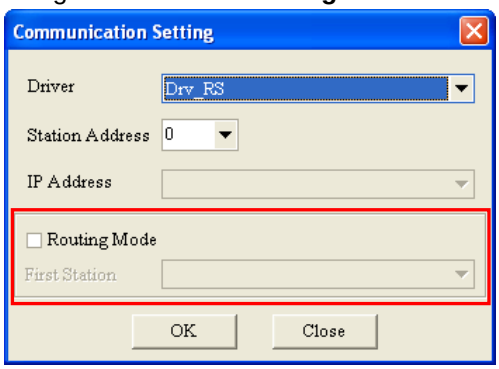
(1) First, users have to activating a project in a group. If a project is a single project, it does not need to be activated.



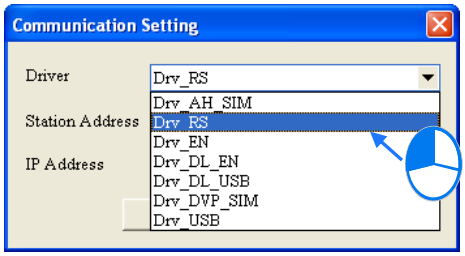
(2) After the users click **Communication Settings…** on the **Tools** menu, the **Communication Setting** window appear.



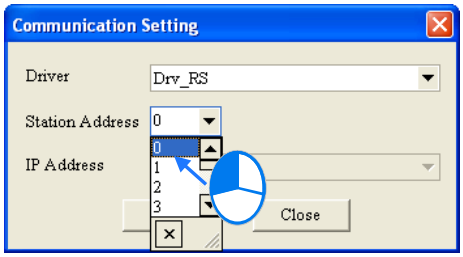
If the project is a project for an AH500 series module created in a group, the **Routing Mode** checkbox will appear in the **Communication Setting** window. If the **Routing Mode** checkbox is selected, the function of routing packets is enabled. Please refer to chapter 16 for more information about the routing function. The **Routing Mode** checkbox is generally not selected.



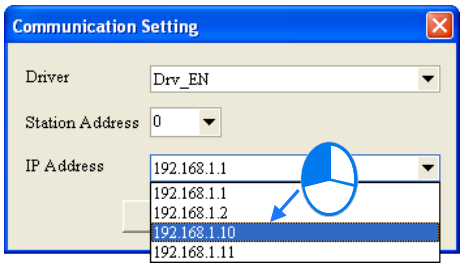
(3) Select a driver in the **Driver** drop-down list box in the **Communication Setting** window.



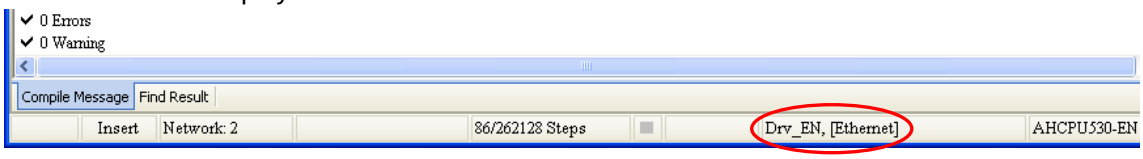
(4) Select the station address of the PLC connected to the computer in the **Station Address** drop-down list box. If the users do not know the station address, they can select 0 in the **Station Address** drop-down list box.



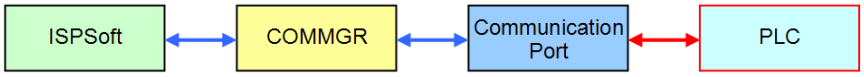
(5) If the connection type that the driver uses is Ethernet or DirectLink (Ethernet), the users have to select an IP address created in COMMGR in the **IP Address** drop-down list box.



(6) After the setting is complete, users can click **OK**. The information about the driver which is connected is displayed in the status bar.



2.4.7 Connecting a PLC and a Communication Port

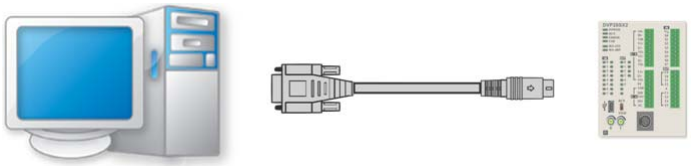


After all setting is complete, users can connect a PLC to a communication port specified through a communication cable. Some common ways to connect a PLC to a communication port, and some points for attention are listed below. Please refer to PLC manuals for more information about connecting PLCs to communication ports.

1. DVP series PLC (RS232)

   A computer is connected to a DVP series PLC through a Delta communication cable. The connection type that a driver uses is RS232.

   Note: Users have to make sure that the communication protocol for exchanging data through a driver is the same as the communication protocol for exchanging data through a communication port on a PLC before they connect the driver to the PLC.



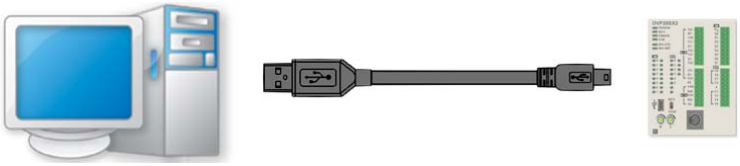
2. DVP-SX2 series PLC (USB)

   A DVP-SX2 series PLC is equipped with a type B mini USB interface. Users can connect a DVP-SX2 series PLC to a computer with a USB cable. Owing to the fact that a DVP-SX2 series PLC converts USB to RS232, the RS232 standard for serial communication is adopted. The connection type that a driver uses must be RS232.

   Note:

   (a) Users have to make sure that the USB driver for a DVP-SX2 series PLC has been installed on a computer. Please refer to appendix A for more information.

(b) Users have to make sure that the communication protocol for exchanging data through a driver is the same as the communication protocol for exchanging data through a communication port on a PLC before they connect the driver to the PLC.

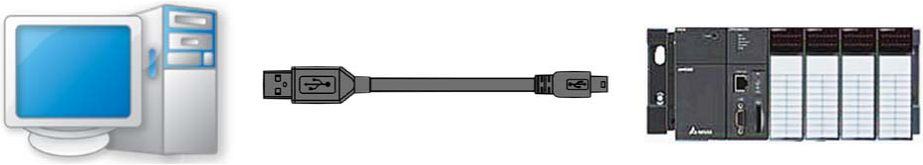


2

3. AH500 series CPU module (USB)

A computer is connected to AH500 series CPU module through a USB cable. An AH500 series CPU module is equipped with a type B mini USB interface. The connection type that a driver uses must be USB (Virtual COM).

Note: Users have to make sure that the USB driver for an AH500 series CPU module has been installed on a computer. Please refer to appendix A for more information.



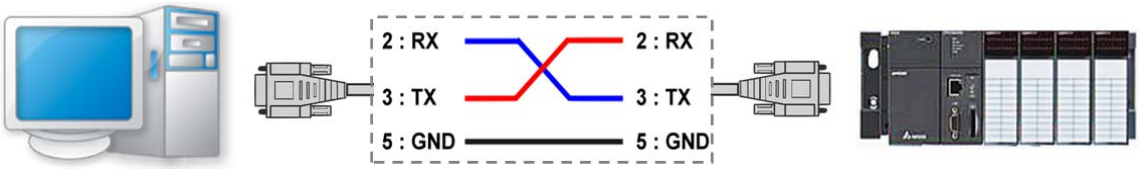
4. AH500 series CPU module (RS232)

Owing to the fact that an AH500 series CPU module is quipped with standard communication ports, users have to connect an AH500 series CPU module to a computer with a RS232 cable depicted in the figure below. The connection type that a driver uses must be RS232.

Note:

(a) Users have to make sure that the communication protocol for exchanging data through a driver is the same as the communication protocol for exchanging data through a communication port on a CPU module before they connect the driver to the CPU module.

(b) The communication ports on an AH500 series CPU module can be RS232 communication ports, RS485 communication ports, or RS422 communication ports. Users have to make sure that a communication port on an AH500 series CPU module is a RS232 port before they connect the communication port to a computer. (Please refer to the additional remark below for more information.)



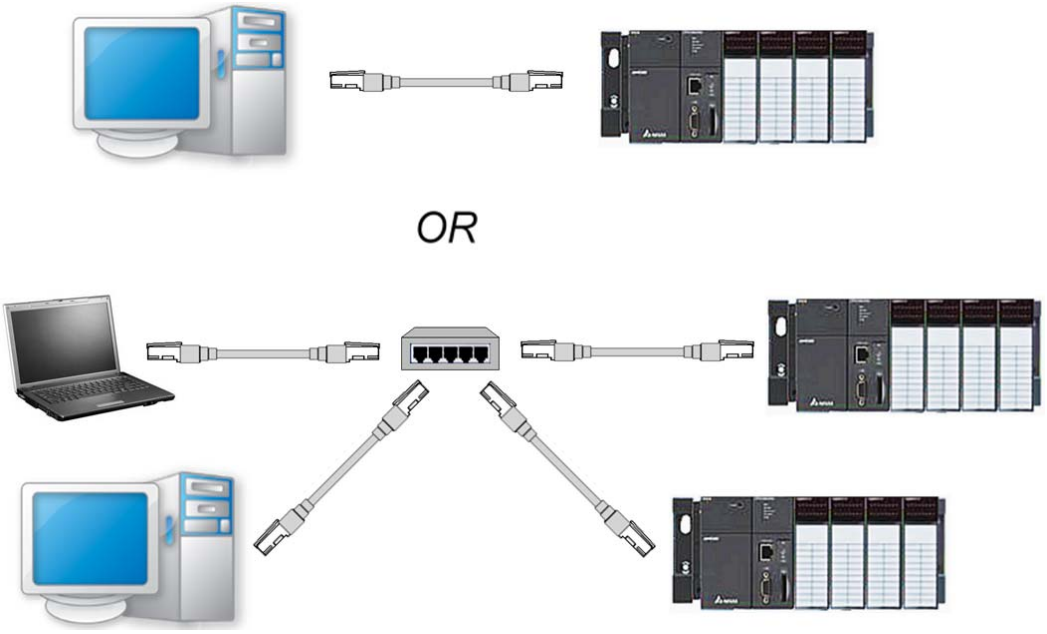
5. AH500 series CPU module (Ethernet)

An AHCPU5xx-EN series CPU module is quipped with an Ethernet port to which a RJ45 cable is connected, and can be connected to a computer through Ethernet. An AHCPU5xx-EN series CPU module and a computer can connect to the same network through a hub. An AHCPU5xx-EN series CPU module can also connect to a computer through a network cable (no jumper). The connection type that a driver uses must be Ethernet.

Note:

(a) Users have to make sure that a network created is normal before they connect a CPU module to a computer.

(b) Users have to make sure that the Ethernet parameters in a CPU module are correct. Please refer to the additional remark below for more information.



Additional remark

AH500 series CPU modules are quipped with USB ports, serial ports, and Ethernet ports. Only AHCPU5xx-EN series CPU modules are equipped with Ethernet port. Besides, the serial ports and the Ethernet ports on AH500 series CPU modules can be set by HWCONIFG in ISPSoft. Please refer to section 3.3.3 and section 3.3.4 for more information.

Compared with the serial port and the Ethernet port on an AH500 series CPU module, it is more convenient to connect the USB port on the AH500 series CPU module to a computer. If users want to connect a new CPU module to a computer, they can connect the CPU module to the computer with a USB cable, set the serial port and the Ethernet port on the CPU module through HWCONFIG in ISPSoft, and then download the communication parameters for the serial port and the Ethernet port.

The default values of the communication parameters for the serial port and the Ethernet port on an AH500 series CPU module are as follows.

- **Serial port**

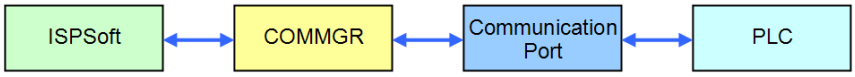
Communication port / Communication parameter	COM 1	COM 2
Connection type	RS232	RS232
Transmission mode	ASCII	ASCII
Communication speed (Baud rate)	9600	9600
Data length	7	7
Parity bit	Even	Even
Stop bit	1	1
Station address	1	3

- **Ethernet**

Communication parameter	Parameter value
IP address assignment	Static
IP address	192.168.1.1
Subnet mask	255.255.255.0
Gateway address	192.168.1.1

2.4.8 Practical Connection Test

After users complete the steps above, they can check whether a computer communicates with a PLC normally by means of a simple test.



Users have to make sure of the items below before a PLC communicates with a computer.

(a) The users have to make sure that COMMGR has been started normally, and the status of a driver which will be specified is **START**.

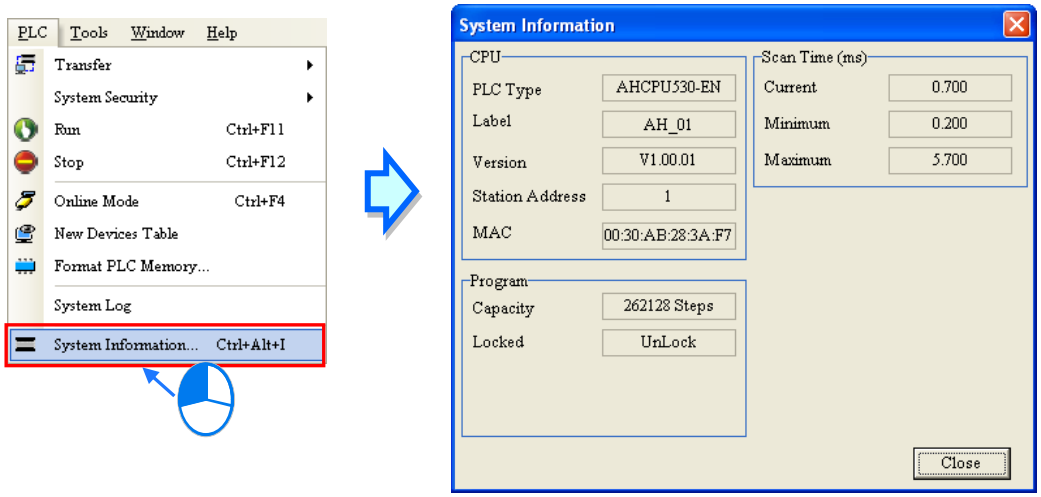
(b) The users have to make sure that the driver which will be specified is set correctly. If the connection type that the driver uses is RS232, the users have to make sure that the communication protocol for exchanging data through the driver is the same as the communication protocol for exchanging data through a communication port on the PLC.

(c) The users have to make sure that a communication port used, e.g. a network interface card in the computer, a hub, or a serial port, is in a normal state.

(d) The users have to make sure that the driver, the station address, and the IP address which are specified in the **Communication Setting** window are correct. If an IP address is specified, the users have to check whether the IP address of the PLC is the same as the IP address specified.

(e) The users have to make sure that the PLC is correctly connected to the computer through a communication cable, power is supplied to the PLC, and the PLC operates normally.

After the users make sure of the items above, they can click **System Information** on the **PLC** menu in ISPSoft. ISPSoft will retrieve related information from the PLC. If the computer communicates with the PLC normally, the **System Information** window will appear, and the related information retrieved from the PLC will be displayed in the window.



Chapter 3 Configuring and Setting a PLC System

Table of Contents

3.1 Setting Parameters for a DVP Series PLC....3-3
3.1.1 System Management Tools for a DVP Series PLC....3-3
3.1.2 Retentive Range....3-3
3.1.3 Connected Information....3-6
3.2 Hardware Configuration Tool for AH500 Series Modules—HWCONFIG 3-7
3.2.1 Introduction of the Environment of HWCONFIG....3-7
3.2.2 Configuring a Module....3-9
3.2.2.1 Adding a Module....3-9
3.2.2.2 Assigning Devices to a Module....3-12
3.2.2.3 Editing a Comment....3-16
3.2.2.4 Deleting a Module....3-17
3.2.2.5 Replacing a Module....3-18
3.2.2.6 Searching for/Replacing a Module....3-19
3.2.2.7 Copying/Pasting a Module....3-22
3.2.2.8 Cutting/Pasting a Module....3-24
3.2.2.9 Dragging a Module....3-25
3.2.2.10 Adding an Extension Rack....3-26
3.2.2.11 Deleting a Rack....3-28
3.2.2.12 Replacing a Rack....3-29
3.2.2.13 Cutting/Copying/Pasting an Extension Rack....3-30
3.2.2.14 Dragging an Extension Rack....3-32
3.2.2.15 Rearranging the Input/Output Devices....3-32
3.3 Setting the Parameters in an AH500 Series CPU Module....3-33
3.3.1 Opening the PLC Parameter Setting Window....3-33
3.3.2 Setting the Basic CPU Parameters....3-34
3.3.2.1 CPU: Name....3-34
3.3.2.2 CPU: System....3-35
3.3.2.3 CPU: Latched Device Range....3-38
3.3.3 COM Port....3-40
3.3.4 Ethernet—Basic....3-41
3.3.5 Ethernet—Advance....3-42
3.3.5.1 Ethernet—Advance: Filter....3-42
3.3.5.2 Ethernet—Advance: NTP....3-44
3.3.5.3 Ethernet—Advance: Email....3-44
3.3.5.4 Ethernet—Advance: Email Trigger....3-45
3.3.5.5 Ethernet—Advance: Email and Trigger Configuration....3-49
3.3.5.6 Ethernet—Advance: Socket....3-50
3.3.5.7 Ethernet—Advance: Web....3-52
3.3.6 Saving and Downloading/Uploading the PLC Parameters....3-53
3.4 Setting the Parameters in an AH500 Series Module....3-54
3.4.1 Managing the Version of a Module....3-54
3.4.2 Setting the Parameters in a Module....3-56

3.4.3 Exporting and Importing the Parameters in a Module 3-59
3.4.4 Setting the Parameters in an Intelligent Module 3-60
3.5 Management of the Parameters in AH500 Series Hardware and Online Diagnosis 3-61
3.5.1 Saving and Printing a Hardware Configuration 3-61
3.5.2 Purchase Order 3-62
3.5.3 Rack Information List 3-63
3.5.4 Downloading/Uploading the System Parameters 3-64
3.5.5 I/O Scan 3-65
3.5.6 Online Diagnosis 3-67
3.5.6.1 Online Mode 3-67
3.5.6.2 Module Information and Diagnosis 3-68
3.5.6.3 Changing the Status of a Module Online 3-69
3.5.6.4 Monitoring Table 3-71
3.6 Setting a Real-time Clock 3-72
3.7 Setting the Memory in a PLC 3-73

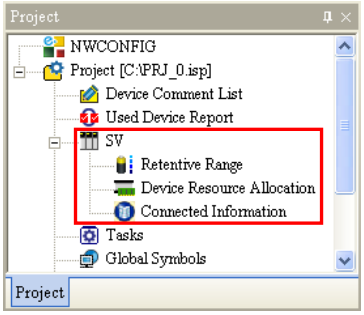
By means of the built-in tools or functions in ISPSoft, users can configure a PLC or a whole system, and set parameters. In this chapter, the configuring of a system and the setting of parameters are introduced.

3.1 Setting Parameters for a DVP Series PLC

3.1.1 System Management Tools for a DVP Series PLC

After users unfold the PLC section in the project management area in a project for a DVP series PLC, they will find the system management tools related to the PLC. Only after the PLC is connected to ISPSoft can some functions be enabled. The functions of the items are described below.

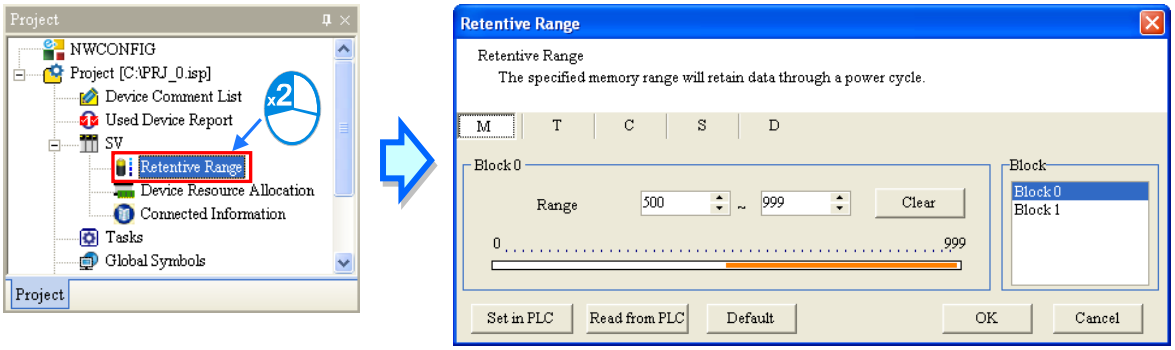
- **Retentive Range**: Users can set a range of latched devices in the PLC.
- **Device Resource Allocation:** Users can set a range of devices which will be automatically assigned to symbols.
- **Connected Information:** Users can view the present system setting and parameters in the PLC.



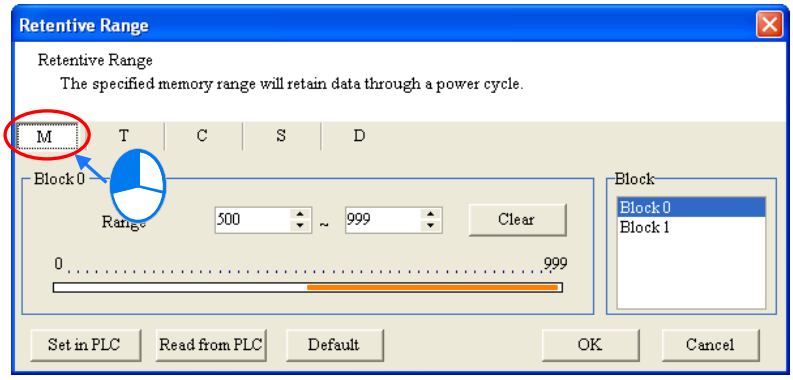
The **Device Resource Allocation** function will be introduced in section 6.2.11. The **Retentive Range** function and the **Connected Information** function will be described in section 3.1.2 and section 3.1.3.

3.1.2 Retentive Range

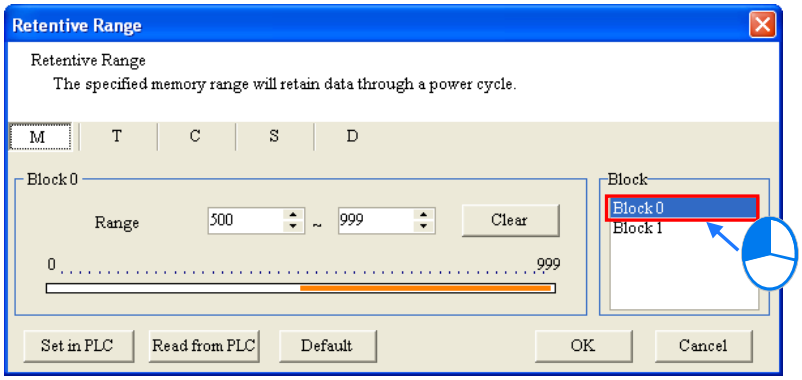
(1) After users double-click **Retentive Range**, the **Retentive Range** window will appear.



(2) Click a device type.

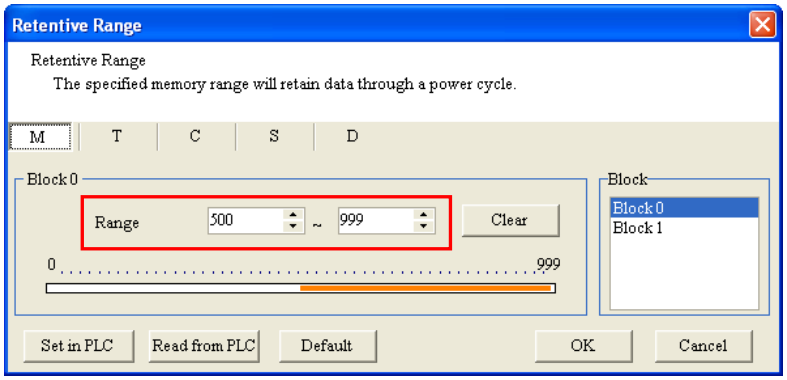


(3) Since the latched devices in a PLC may not be consecutive devices, the users have to select a block before they set a range of latched device.

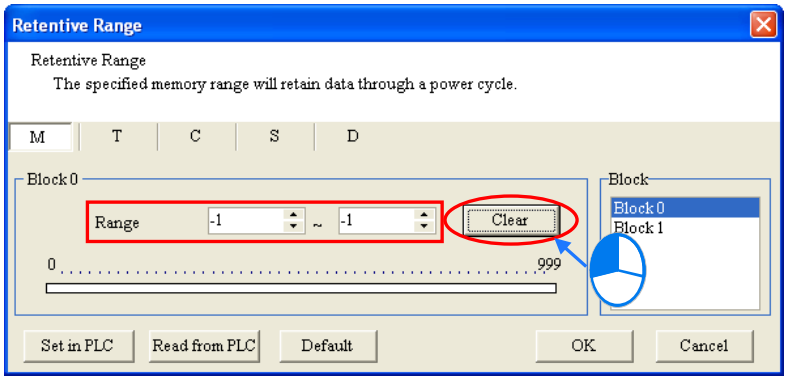


(4) There is a rectangular bar in the setting area. The values at the two ends of the bar represent a maximal value and a minimal value. The orange part represents a range of latched devices. If the setting area is gray, users can not set a range of latched devices in the block for the model selected.

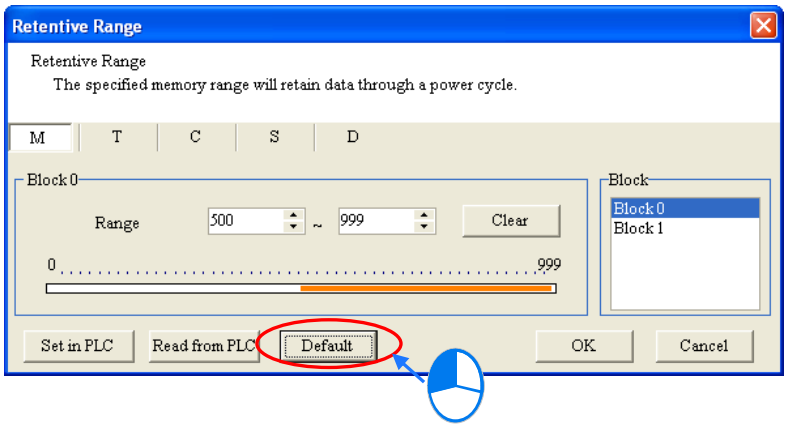
Please select values in the boxes. The value in the left box is a start address, and the value in the right box is an end address. The value in the left box must be less than the value in the right box. Besides, the value in the left box can not be less than the minimal value at the left end of the rectangular bar, and the value in the right box can not be larger than the maximal value at the right end of the rectangular bar.



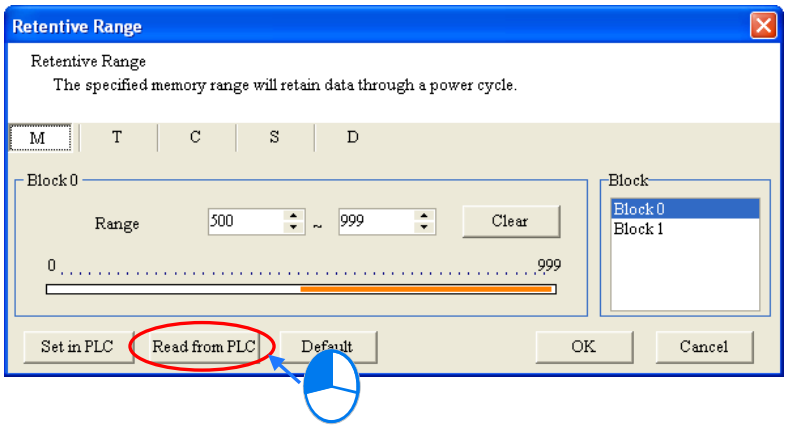
(5) If all the devices in the block are not latched devices, the users can click **Clear**. After the users click **Clear**, the range of latched devices will be -1~-1.



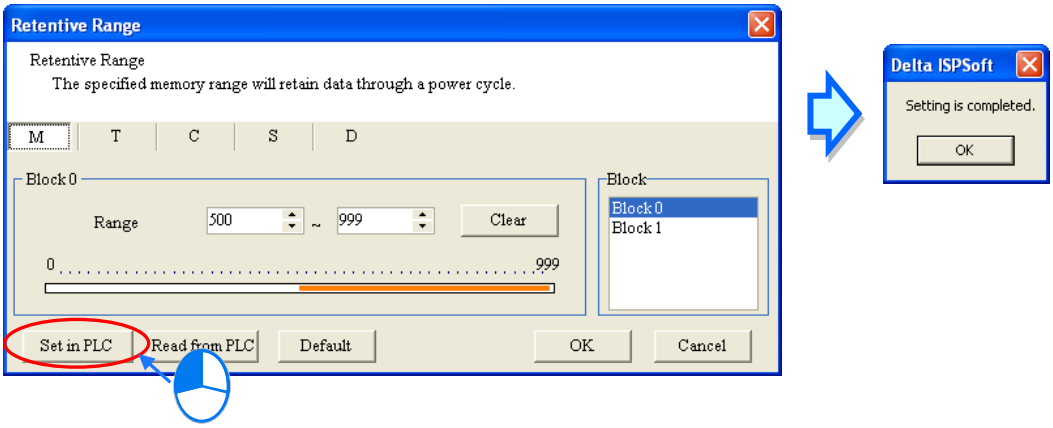
(6) If the users want to restore the setting to the default values, they can click **Default**. After the users click **Default**, the setting will be restored to the default values.



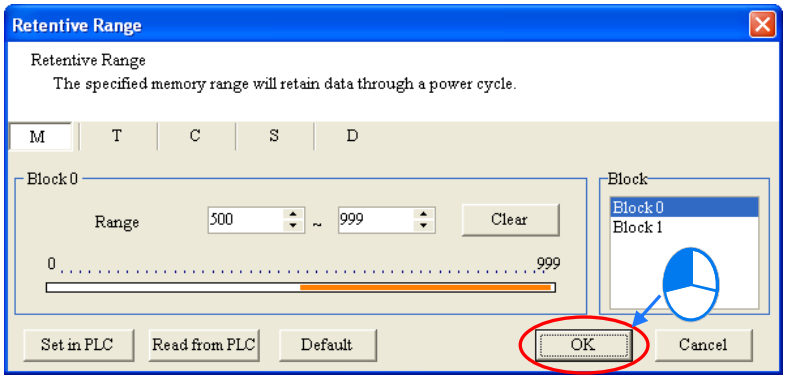
(7) If ISPSoft is connected to the PLC normally, users can click **Read from PLC** to read the setting from the PLC.



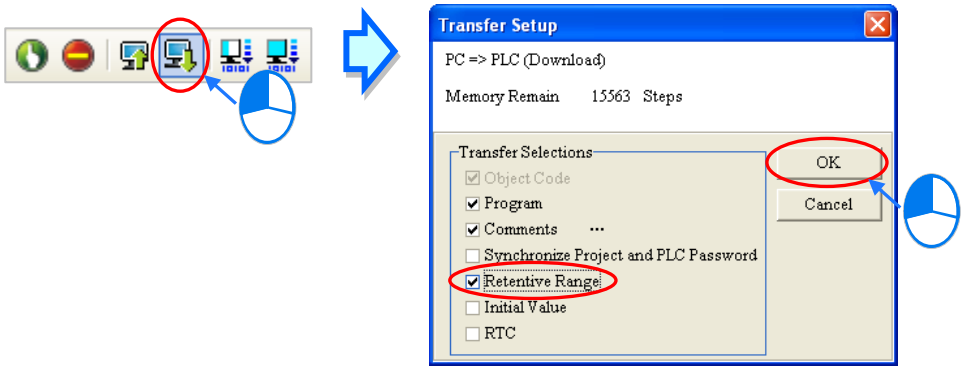
(8) After the setting is complete, the users can click **Set in PLC** to write the setting into the PLC if ISPSoft is connected to the PLC normally. The **Retentive Range** window will be closed automatically, and all the setting values will be memorized. When the ISPSoft project is saved, the setting values are also saved.



(9) If the users do not want to write the setting into the PLC for the time being, they can click **OK** to apply all the setting. When the ISPSoft project is saved, the setting values are also saved.

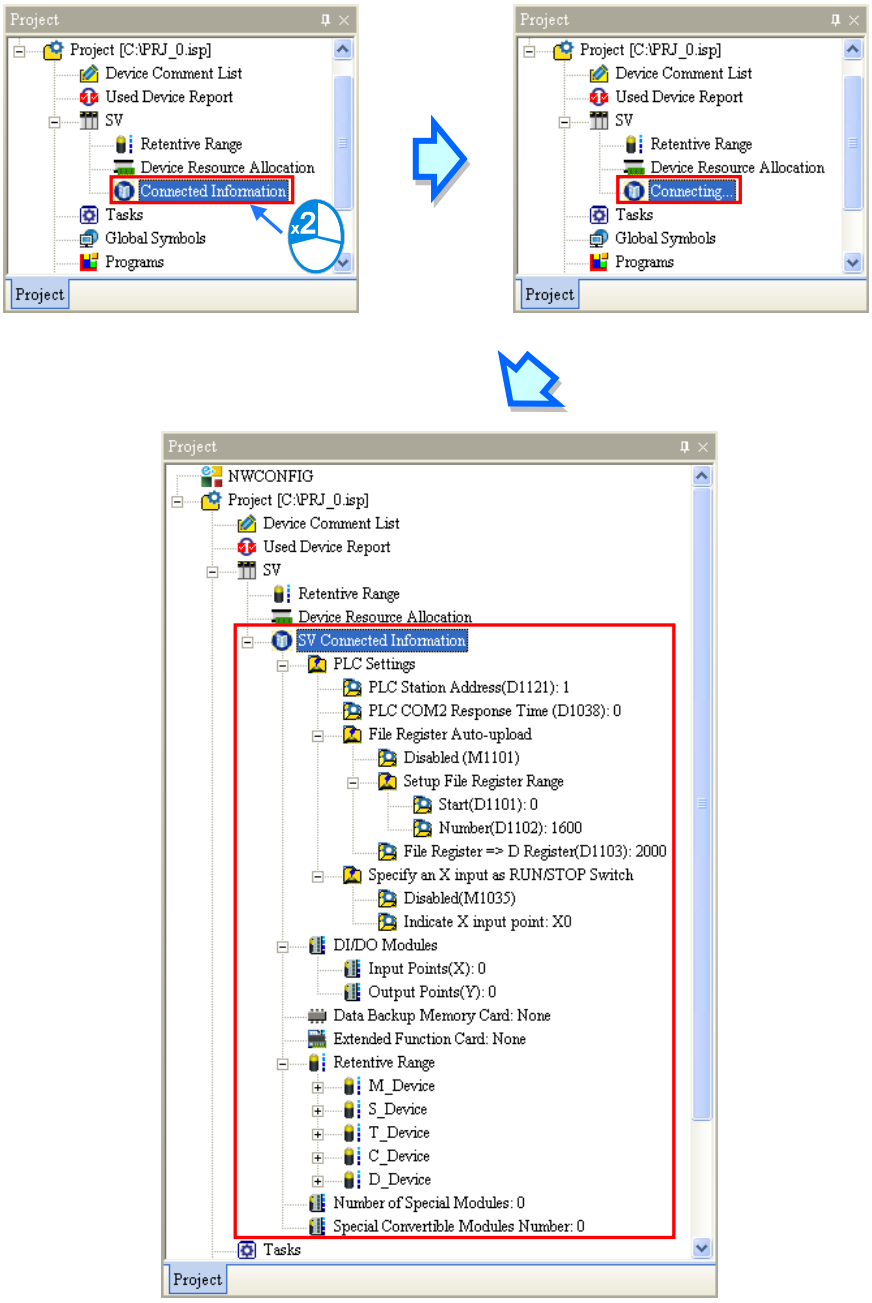


(10) In addition to writing the setting into the PLC by clicking **Set in PLC** in the **Retentive Range** window, the users can write the setting into the PLC when the program in the project is downloaded. After the users click **Download PLC** on the toolbar, they can select the **Retentive Range** checkbox in the **Transfer Setup** window, and click **OK** to download the setting to the PLC. Please refer to section 13.1.2 for more information.



3.1.3 Connected Information

If ISPSoft is connected to the PLC normally, the system will automatically retrieve the system parameters from the PLC after users double-click **Connected Information** in the project management area. The system parameters retrieved will be organized in a hierarchical structure under **Connected Information**. Users can view the information here, but they can not carry out any setting.



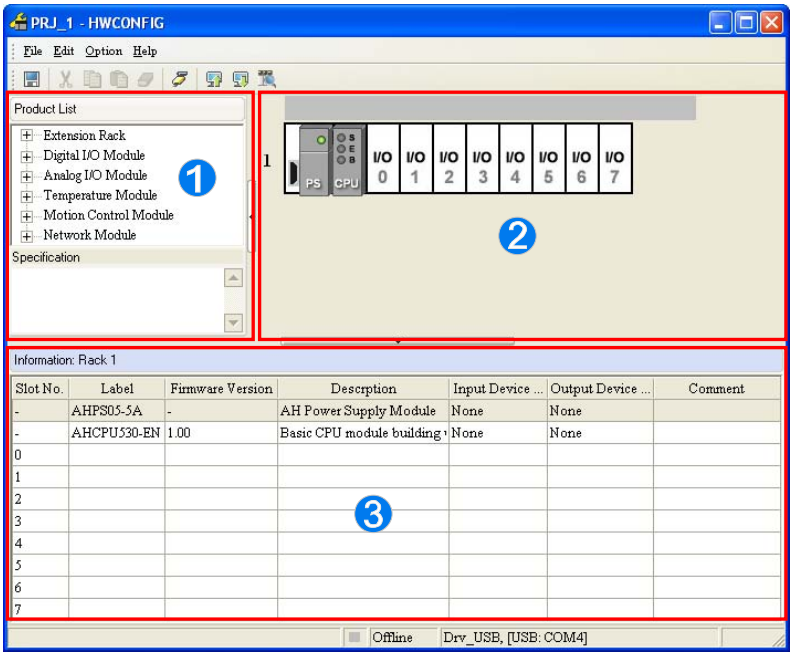
3.2 Hardware Configuration Tool for AH500 Series Modules—HWCONFIG

HWCONFIG is a built-in hardware configuration tool in ISPSoft. Users can configure racks, set CPU parameters, set module parameters, download/upload parameters, detect a hardware configuration online, and make a diagnosis through HWCONFIG.

All parameters set in HWCONFIG must be downloaded to the CPU module so that they can take effect. (Please refer to section 3.5.4 for more information.)

3.2.1 Introduction of the Environment of HWCONFIG

After users double-click **HWCONFIG** in the project management area, the **HWCONFIG** window will appear.

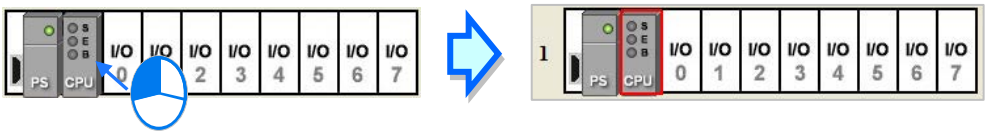


❶ **Product list**: Hardware available is listed in the catalogue.

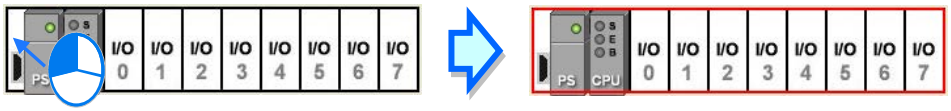
❷ **System configuration area:** It is the main working area. Users can configure and set a whole system in this area.

❸ **Information list:** The information about the present system configuration is listed in the list.

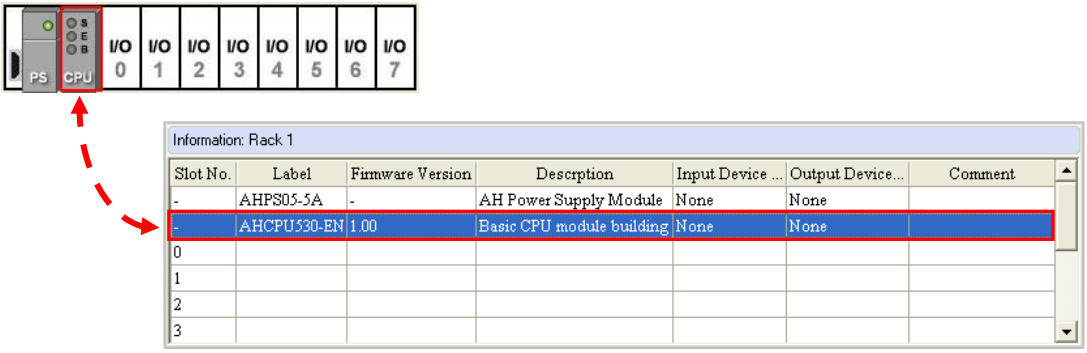
The present system configuration is displayed in the system configuration area. The number at the left side of a rack is a rack number, and the number on a slot is a slot number. Users can select a module by clicking the module.



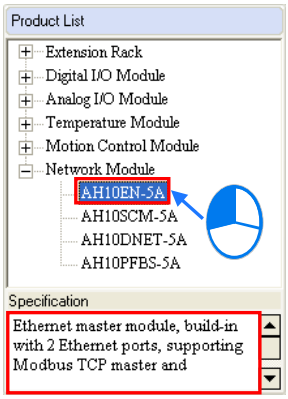
If users want to select a rack, they can move a mouse cursor to the extension port on the rack, and click the extension port.



When a rack is selected, the information about the present configuration for the rack is listed in the information list. If a module is selected, the information about the module on the list will be selected. If the information about a module is selected, the module in the system configuration area will be selected. Besides, if the configuration in the system configuration area is modified, the information on the information list will be updated.



All hardware available is listed in the product list. After a section is unfolded, all devices belongs to the section are listed under the section. If users click a device, the specifications for the device will be under the product list.

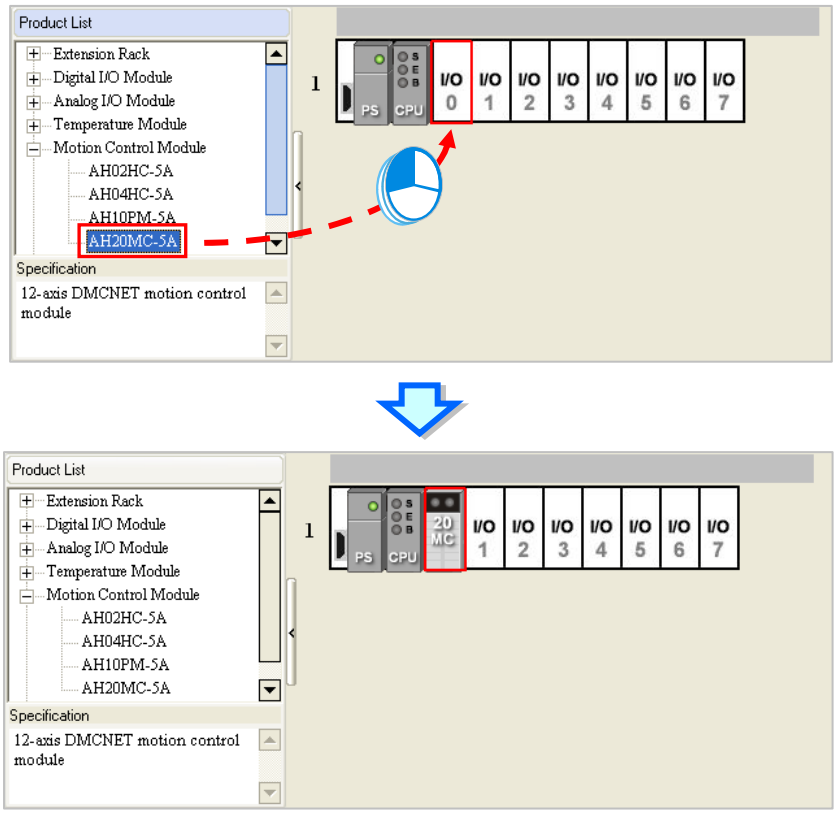


3.2.2 Configuring a Module

3.2.2.1 Adding a Module

- **Method 1**

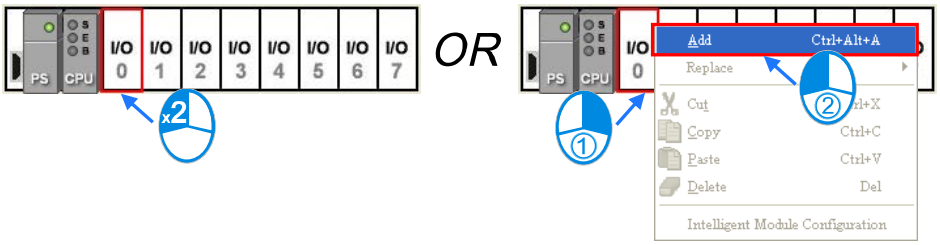
  After users select a module which will be added to the product list, they can drag it to a vacant slot.



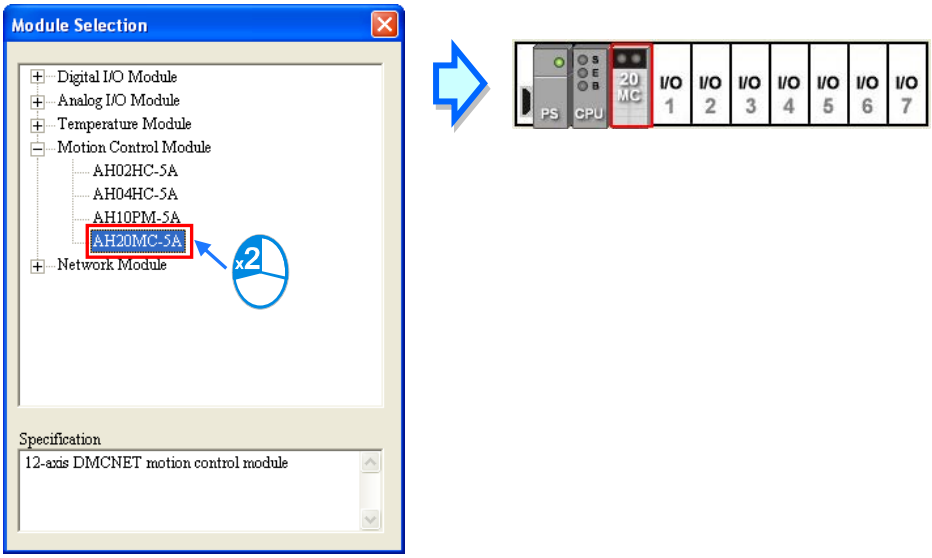
***. If users want to drag a module on the product list to the system configuration area, the module can only be put on a vacant slot. If the module is dragged to an occupied slot, the system will prohibit this operation.**

- **Method 2**

(1) After users click a vacant slot, the **Module Selection** window will appear. The users can also right-click a vacant slot, and click **Add** on the context menu to open the **Module Selection** window.



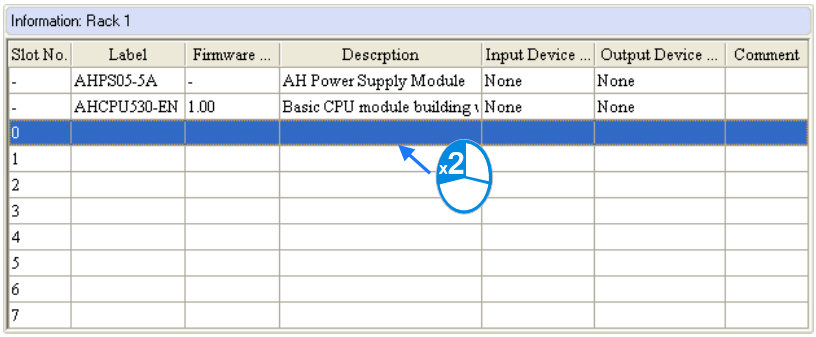
(2) If the users click a module in the **Module Selection** window, the specifications for the module will appear in the **Specification** box. After the users double-click a module in the **Module Selection** window, the module will be added.



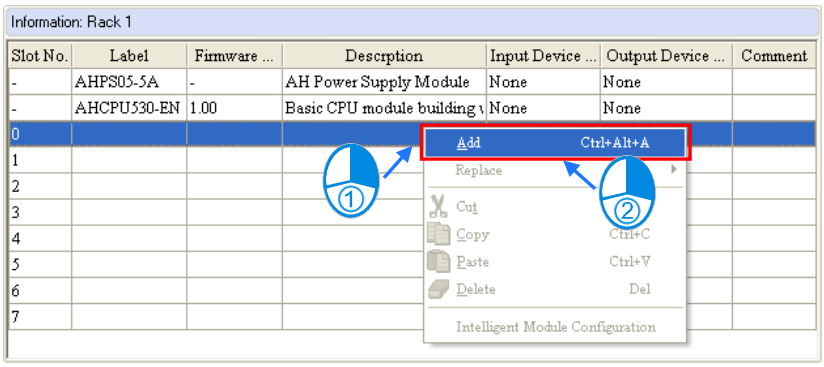
3

- **Method 3**

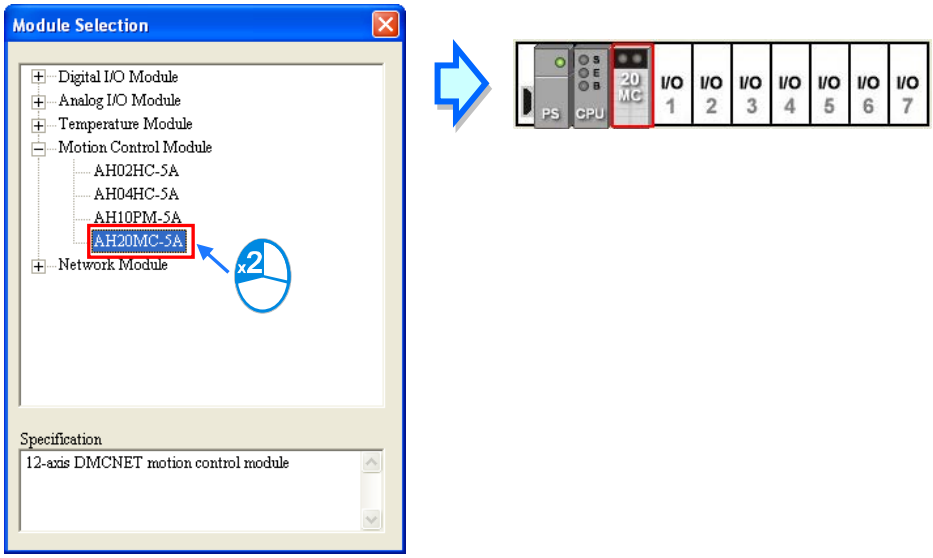
(1) After users click a blank on the information list, the **Module Selection** window will appear. The users can also double-click a blank on the information list, and click **Add** on the context menu to open the **Module Selection** window.



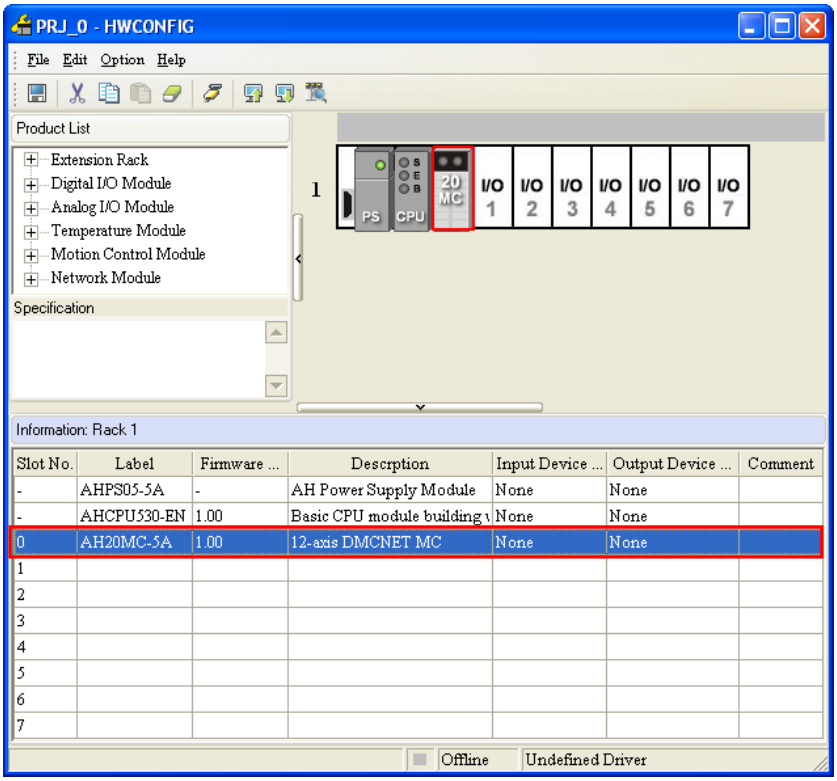
OR



(2) If the users click a module in the **Module Selection** window, the specifications for the module will appear in the **Specification** box. After the users double-click a module in the **Module Selection** window, the module will be added.

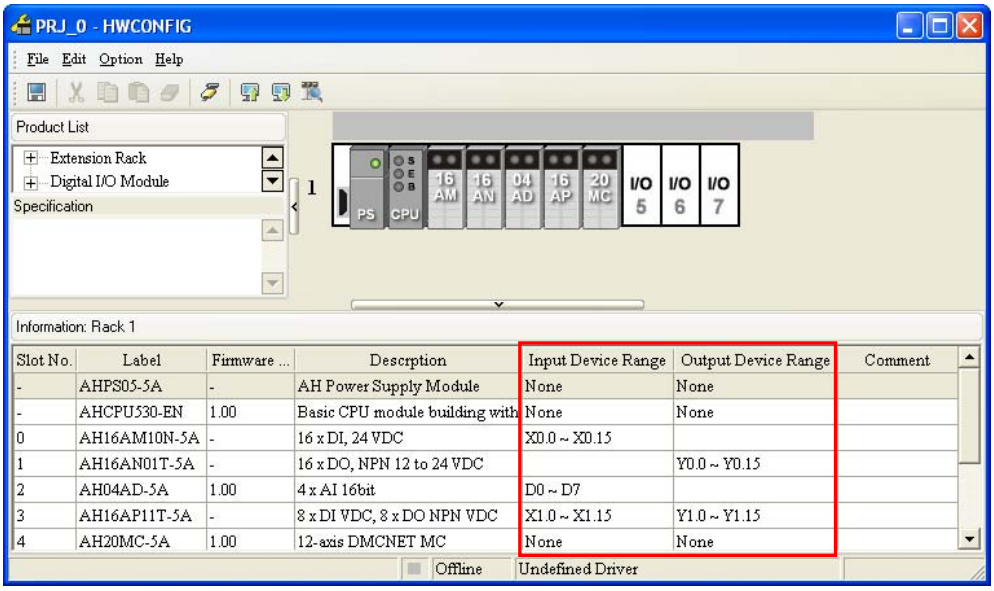


No matter which method is used to add a module, the configuration in the system configuration area and the information on the information list will be updated after a module is added.



3.2.2.2 Assigning Devices to a Module

The data in a module needs to be updated constantly. For example, analog signals received by an analog input module are updated constantly, and converted to data which can be processed by a CPU module. As a result, the system automatically assigns devices to a module so that the data in the module can be stored. The devices assigned to a module are displayed in the **Input Device Range** cell and the **Output Device Range** cell on the information list. Please refer to the following example.



- **AH16AM10N-5A (slot 0)**

  AH16AM10N-5A is a digital input module which has 16 inputs. The system assigns X0.0~X0.15 to AH16AM10N-5A so that data received by AH16AM10N-5A can be stored.
- **AH16AN01T-5A (slot 1)**

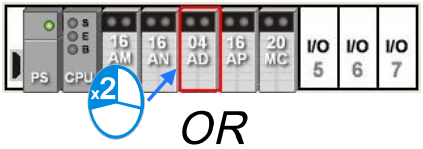
  AH16AN01T-5A is a digital output module which has 16 outputs. The system assigns Y0.0~Y0.15 to AH16AN01T-5A so that data sent by AH16AM10N-5A can be stored.
- **AH04AD-5A (slot 2)**

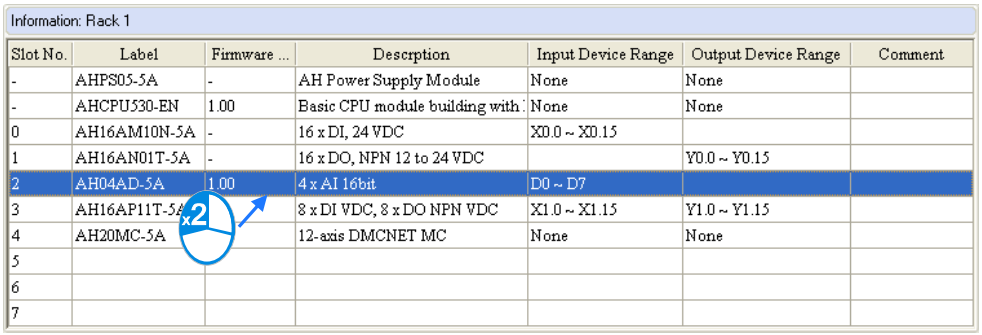
  AH04AD-5A is an analog input module which has four channels. The system assigns D0~D7 to AH04AD-5A. After analog signals received by AH04AD-5A are converted into digital data, the digital data will be stored in D0~D7.
- **AH16AP11T-5A (slot 3)**

  AH16AP11T-5A is a digital input/output module which has 8 inputs and 8 outputs. The system assign X1.0~X1.15 to AH16AP11T-5A so that data received by AH16AP11T-5A can be stored. (X1.0~X1.7 are actually used.) The system also assigns Y1.0~Y1.15 to AH16AP11T-5A so that data sent by AH16AP11T-5A can be stored. (Y1.0~Y1.7 are actually used.)
- **AH20MC-5A (slot 4)**

  AH20MC-5A is a motion control module. Owing to the fact that the data in AH20MC-5A does not need to be updated constantly, the system does not assign any device to AH20MC-5A.

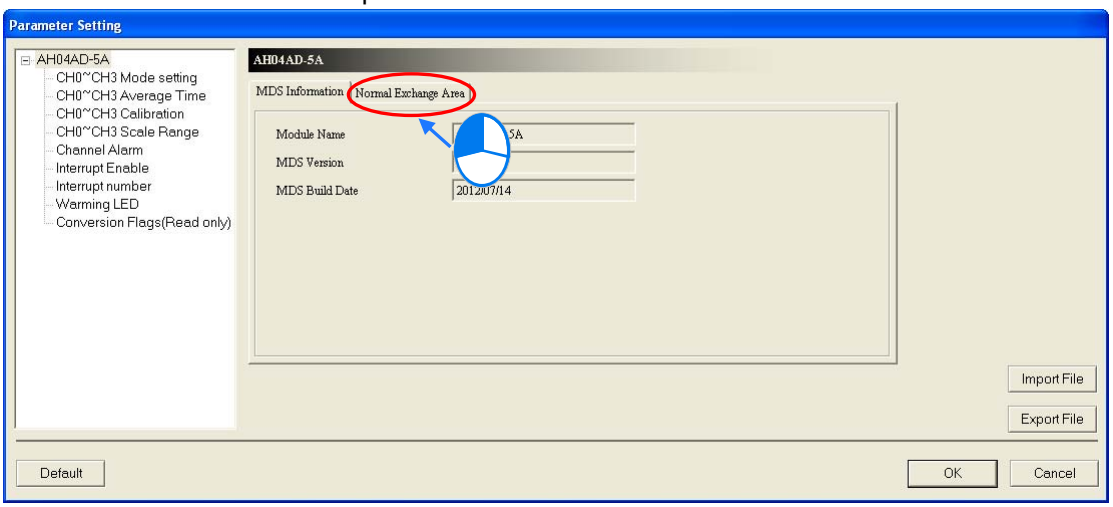
If users want to know functions to which devices assigned correspond, they can double-click the module or the information about the module on the information list to open the **Parameter Setting** window.

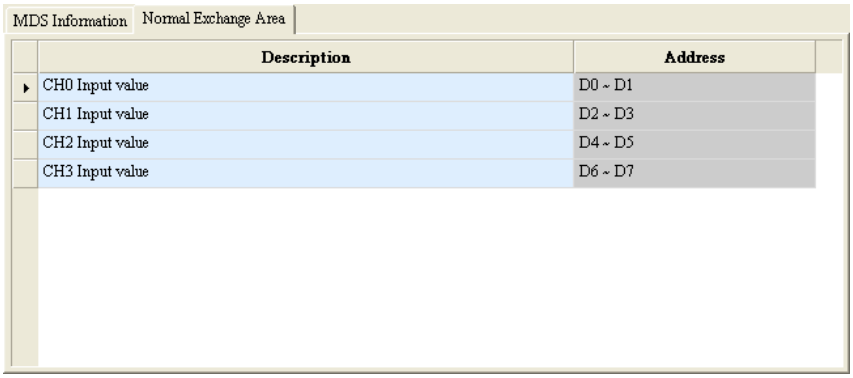




***. To avoid the cells which can be edited, if users want to double-click the information about a module on the information list, they can double-click Slot No. cell, the Label cell, or the Description cell.**

After users click the **Normal Exchange Area** tab in the **Parameter Setting** window, they can see functions to which devices correspond.

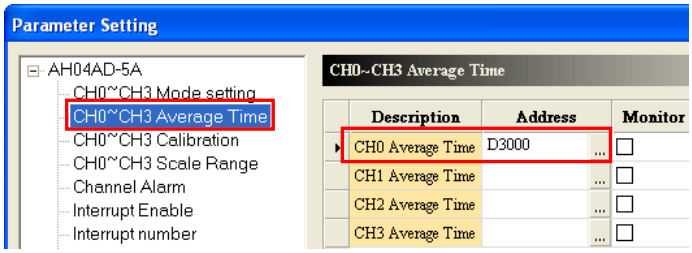


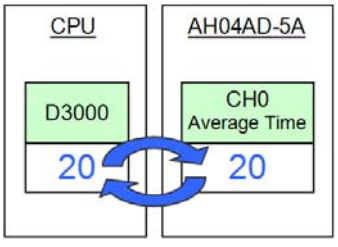


The system automatically assigns devices to a module so that the data in the module which needs to be updated constantly can be stored. The parameters in a module do not need to be updated constantly. Users assign data registers to a module so that the parameters in the module can be stored. When the system operates, the data registers in a CPU module are synchronized to the parameters in the module. As a result, the users can access the module through the data registers. Accessing a module through the data registers is more efficient than accessing the module through the instruction FROM/TO.

As the example below shows, D3000 corresponds to the parameter **CH0 Average Time** in the **Parameter Setting** window for AH04AD-5A. After the parameters in HWCONFIG are downloaded to the CPU module, users can change the value of the parameter **CH0 Average Time** by changing the value in D3000 in the CPU module.

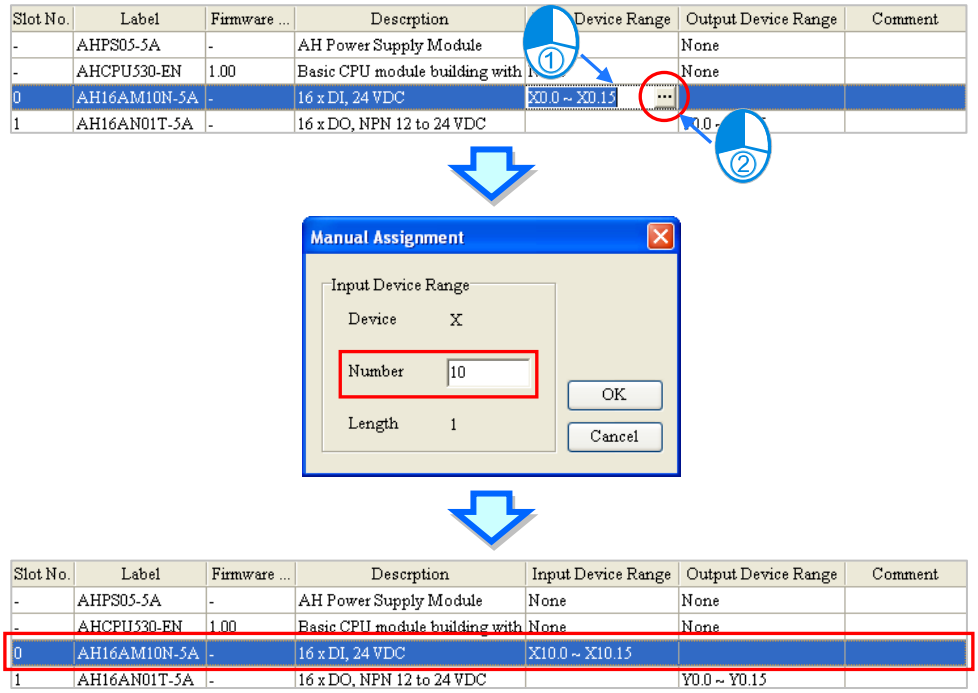
Besides, if users write a value which is not allowed by a parameter into a data register during the operation of the system, the system will restore the value in the data register and the value of the parameter to the original values.



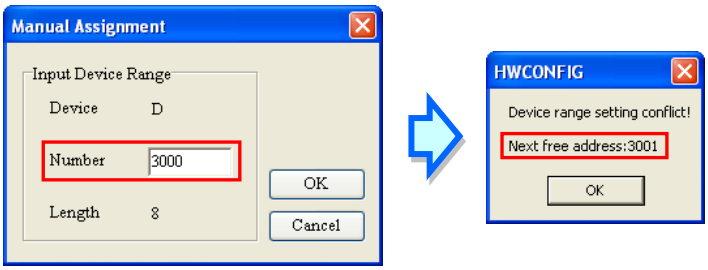


***. Please refer to section 3.4 for more information about the setting of parameters in a module. Please refer to programming manuals for more information about the instruction FROM/TO.**

The **Input/Output Device Range** column on the information list can be defined not only by the system, but also by users. Users can click a cell on the information list, and then click ···, or type a start address directly. After the **Manual Assignment** window appears, the users can type a start address, and click **OK**.

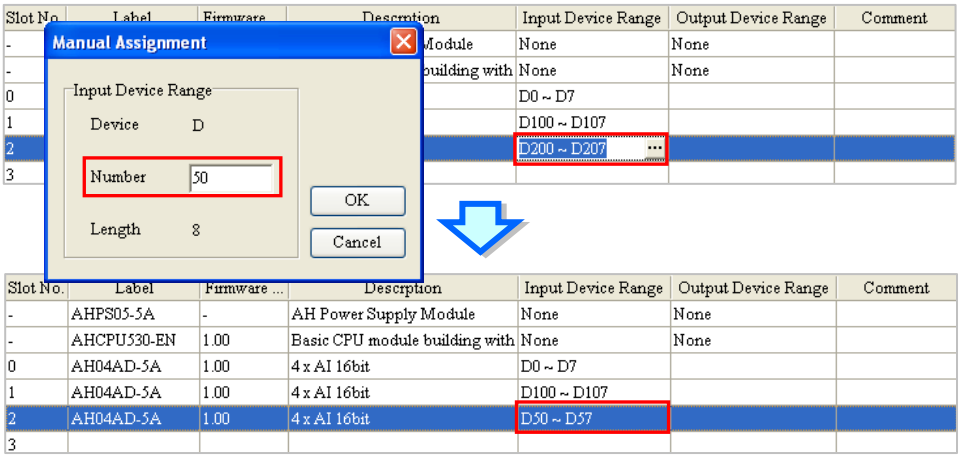


Devices that users assign to a module so that the data in the module can be stored can not overlap data registers that the users assign to a module so that the parameters in the module can be stored. If a device address typed conflicts with another device address, the system will modify the device address which is typed, and provide a device address which can be used. For example, if users want to assign D3000 which has been assigned to a module to another module, the system will provide a device address which can be used.



After the input/output device addresses assigned to a module are changed, the input/output device addresses assigned to a module added will follow the new input/output device addresses assigned to the preceding module, and the input/output device addresses which have been assigned will be skipped. Even if the new input/output device addresses assigned to the preceding module are not the largest addresses, the input/output device addresses assigned to the module added will follow the new input/output devices addresses.

Please refer to the following example. The input device addresses assigned to AH04AD-5A installed in slot 2 is changed to D50~D57.



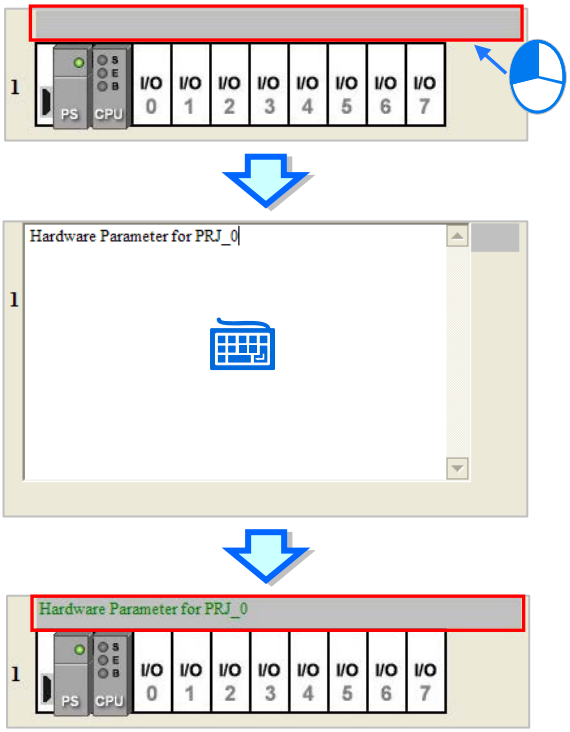
Slot No.	Label	Firmware ...	Description	Input Device Range	Output Device Range	Comment
-	AHPS05-5A	-	AH Power Supply Module	None	None	
-	AHCPU530-EN	1.00	Basic CPU module building with	None	None	
0	AH04AD-5A	1.00	4 x AI 16bit	D0 ~ D7		
1	AH04AD-5A	1.00	4 x AI 16bit	D100 ~ D107		
2	AH04AD-5A	1.00	4 x AI 16bit	D50 ~ D57		
3						

The input device addresses assigned to a module added follow D50~D57 rather than D100~D107. As a result, D58~D65 are assigned to AH04AD-5A which is installed in slot 3.

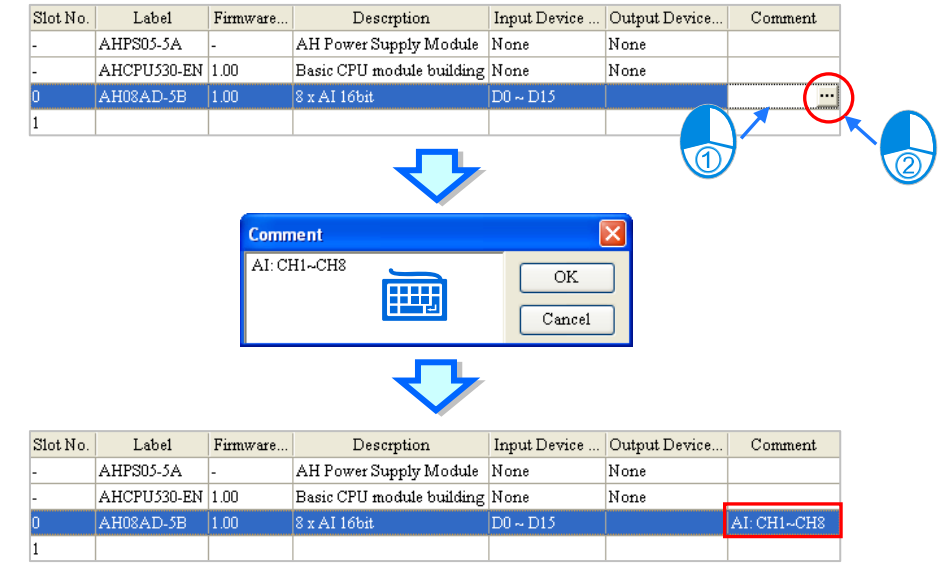
Slot No.	Label	Firmware ...	Description	Input Device Range	Output Device Range	Comment
-	AHPS05-5A	-	AH Power Supply Module	None	None	
-	AHCPU530-EN	1.00	Basic CPU module building with	None	None	
0	AH04AD-5A	1.00	4 x AI 16bit	D0 ~ D7		
1	AH04AD-5A	1.00	4 x AI 16bit	D100 ~ D107		
2	AH04AD-5A	1.00	4 x AI 16bit	D50 ~ D57		
3	AH04AD-5A	1.00	4 x AI 16bit	D58 ~ D65		
4						

3.2.2.3 Editing a Comment

After users click the gray area at the top of the system configuration area, they can type a comment about the hardware configuration in the drop-down box that appears. If users want to start a new line of text at a specific point, they can press Shift+Enter on the keyboard. After the comment is typed, users can press Enter on the keyboard.



After users click the **Comment** cell for a module on the information list, they can press a key on the keyboard, or click ··· to open the **Comment** window. The users can type a comment about the module in the **Comment** window.

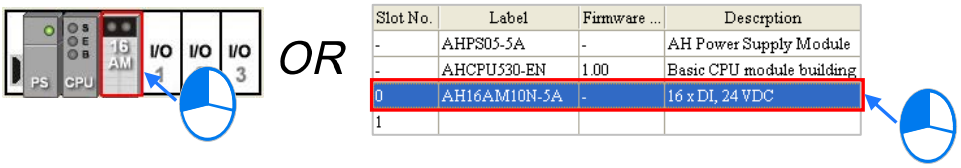


3.2.2.4 Deleting a Module

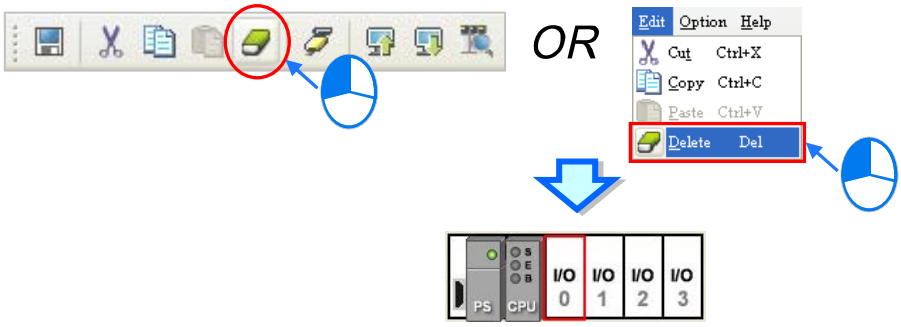
There are two ways to delete a module which has been configured. (The CPU module and the power supply module can not be deleted.)

- **Method 1**

(1) Select a module which will be deleted from the system configuration area or information list.

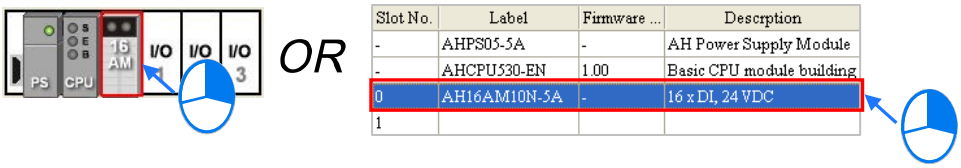


(2) Click **Delete** on the **Edit** menu, click on the toolbar, or press Delete on the keyboard.

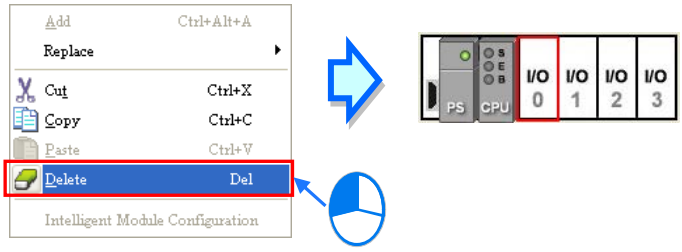


- **Method 2**

(1) Right-click a module which will be deleted from the system configuration area or information list.

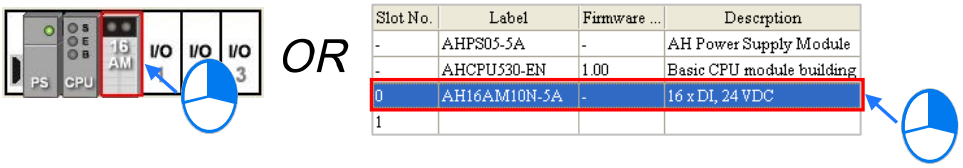


(2) Click **Delete** on the context menu.

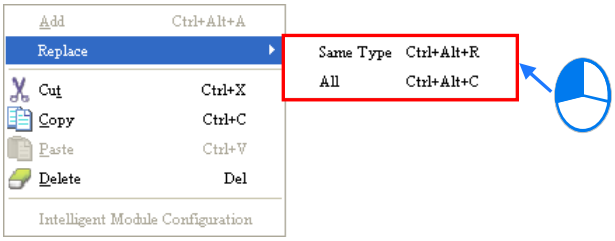


3.2.2.5 Replacing a Module

(1) Right-click a module which will be replaced in the system configuration area or on the information list. (The CPU module and the power supply module can not be replaced.)



(2) Point to **Replace** on the context menu, and then click **Sam Type** or **All**.



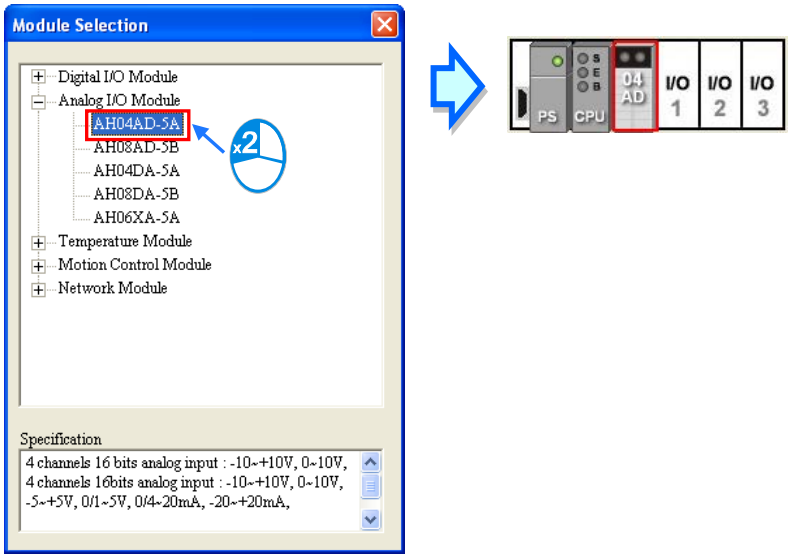
- **Same Type**

  A module selected is replaced by a same type of module. After the module is replaced, the input/output devices assigned to the new module are the same as the input/output devices assigned to the module replaced. Besides, if the parameters in the new module are not the same as the parameters in the module replaced, the setting of the parameters in the new module will be restored to the default values.

- **All**

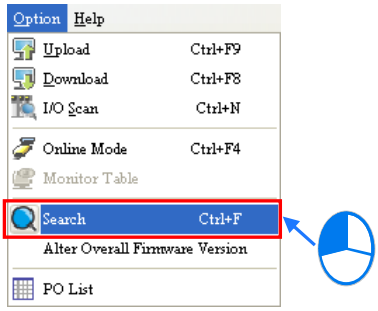
  A module selected can be replaced by any type of module. After the module is replaced, the different input/output devices will be assigned to the new module, and the setting of the parameters in the new module will be restored to the default values.

(3) After users click **Same Type** or **All**, the **Module Selection** window will appear. Items which can be selected will be displayed in the window. After the users decide on a module, they can double-click the module in the window.

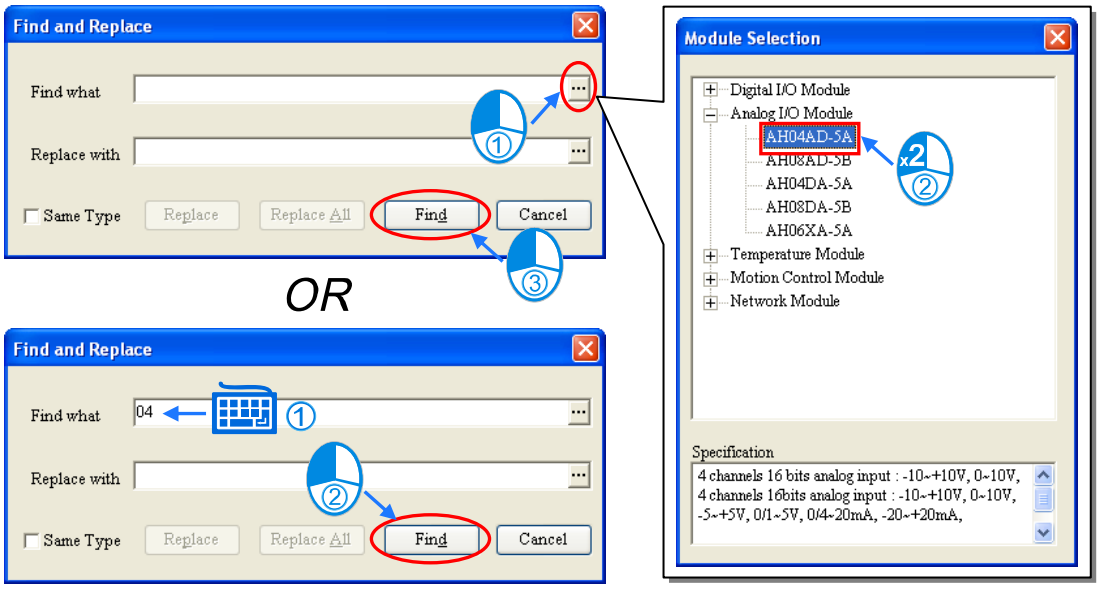


3.2.2.6 Searching for / Replacing a Module

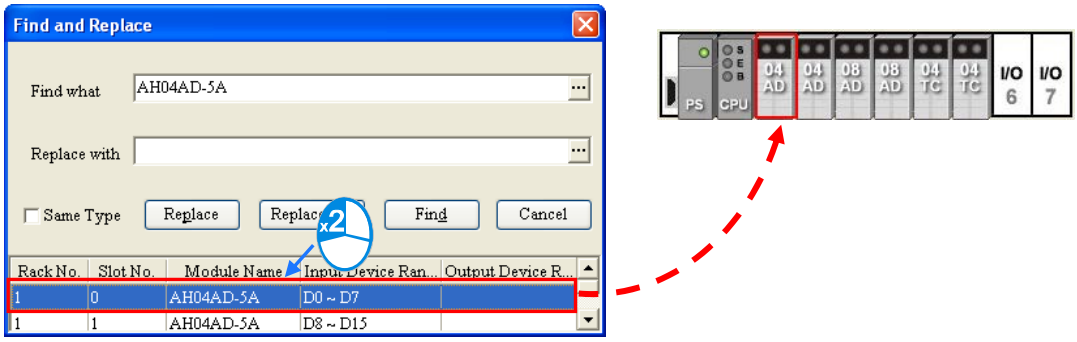
(1) After users click **Search** on the **Option** menu, the **Find and Replace** window will appear.



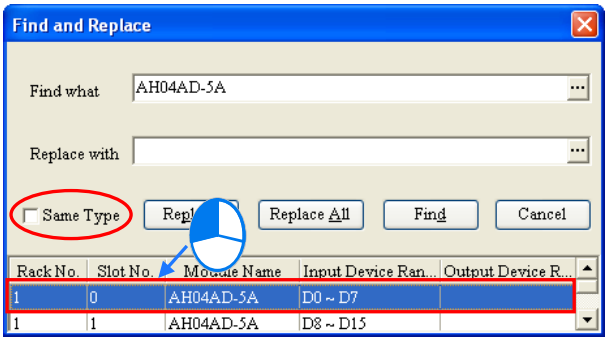
(2) Click ··· in the **Find what** box, select a module in the **Module Selection** window, and double-click the module. The users can also type part of a module model in the **Find what** box. Finally, click **Find**.



(3) After the search is complete, modules meet the search condition will be listed in the list. After the users double-click an item on the list, the module corresponding to the item in the system configuration area will be selected.



(4) If the users want to replace a module, they can click a module which will be replaced on the search list. If the users want to replace the module with a same type of module, they can select the **Same Type** checkbox.



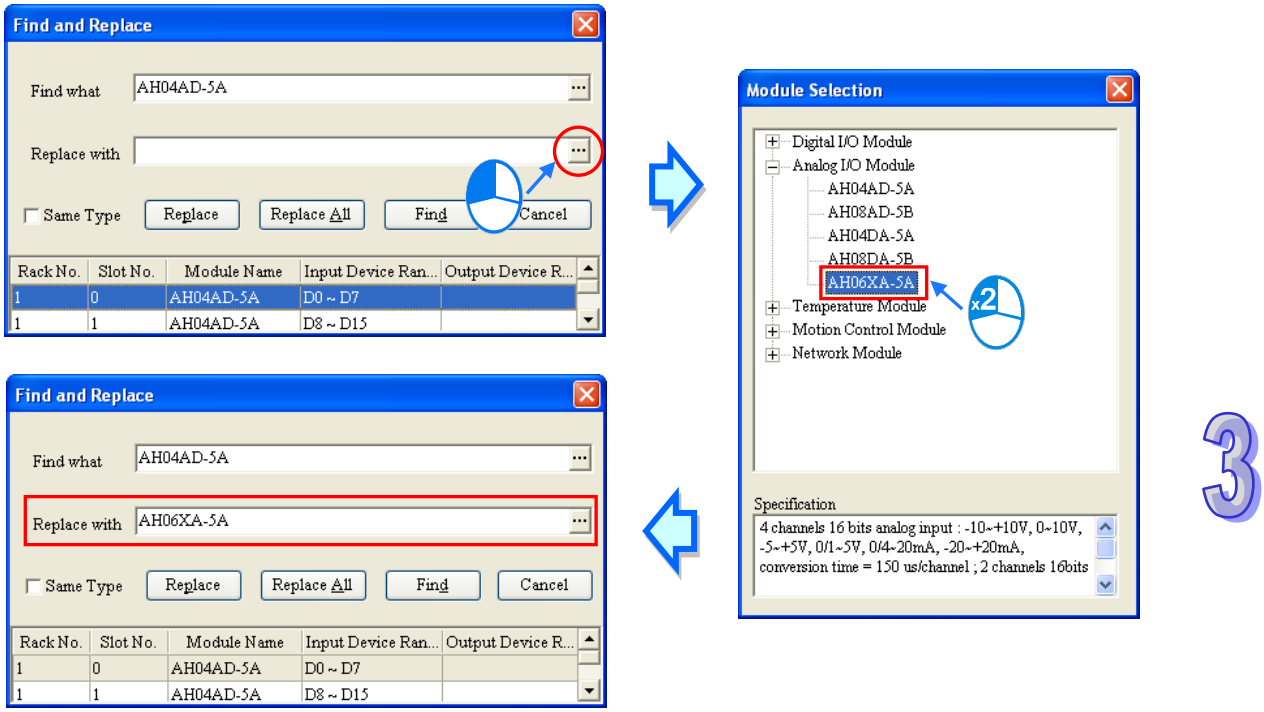
- The **Same Type** checkbox is selected.

  A module selected is replaced by a same type of module. After the module is replaced, the input/output devices assigned to the new module are the same as the input/output devices assigned to the module replaced. Besides, if the parameters in the new module are not the same as the parameters in the module replaced, the setting of the parameters in the new module will be restored to the default values.

- The **Same Type** checkbox is not selected.

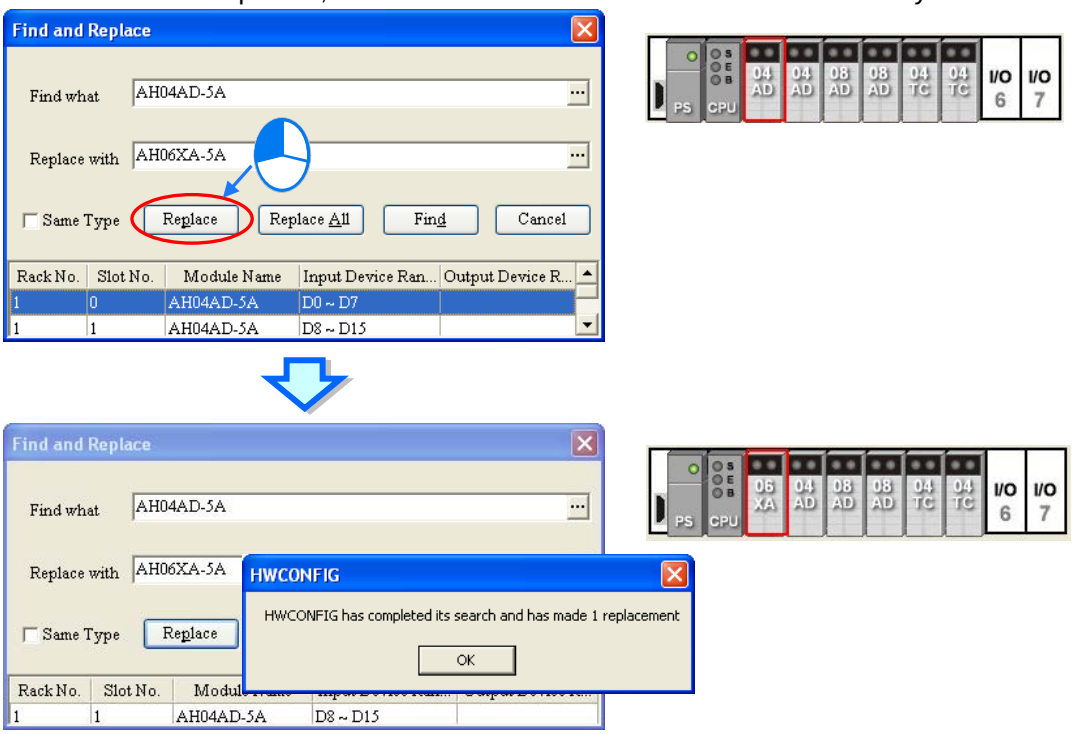
  A module selected can be replaced by any type of module. After the module is replaced, the different input/output devices will be assigned to the new module, and the setting of the parameters in the new module will be restored to the default values.

(5) Click ··· in the **Replace with** box, select a module in the **Module Selection** window, and double-click the module. Owning to the fact that a module must be replaced by a specific module, typing a module model or part of a module model in the **Replace with** box is not allowed.

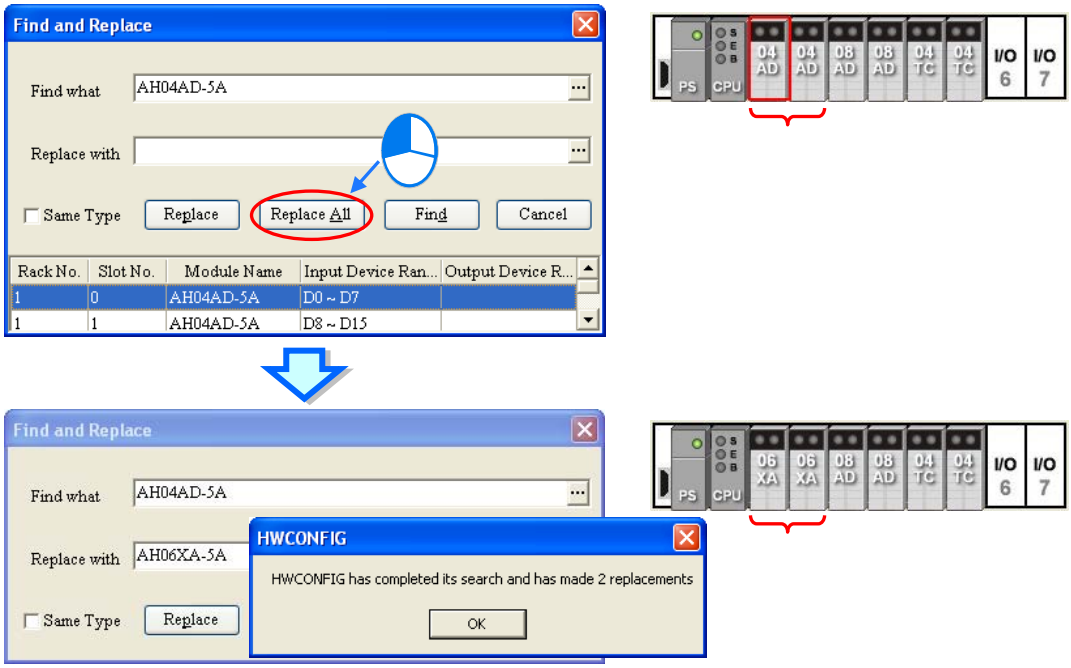


***. If the Replace with box is blank, a module selected will be deleted after Replace is clicked.**

(6) After the setting of the replacement condition is complete, the users can click **Replace** to replace the module selected with the new module. After the replacement is complete, the search list will be updated, and the next module will be selected automatically.



(7) If the users want to replace all the modules on the list, they can click **Replace All** after the setting of the replacement condition is complete. All the modules on the list will be replaced by the new module.

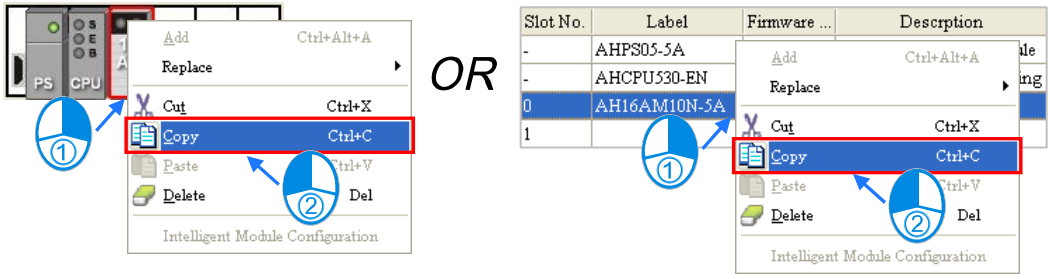


3.2.2.7 Copying / Pasting a Module

There are two ways to copy a module. (The CPU module and the power supply module can not be copied/pasted.)

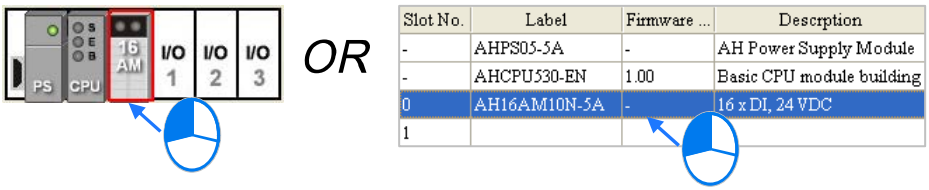
- **Method 1**

  Right-click a module which will be copied in the system configuration area or on the information list, and then click **Copy** on the context menu.

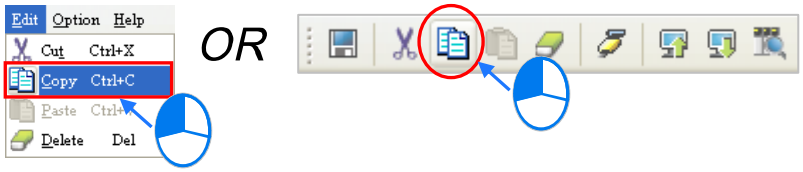


- **Method 2**

  (1) Click a module which will be copied in the system configuration area or on the information list.



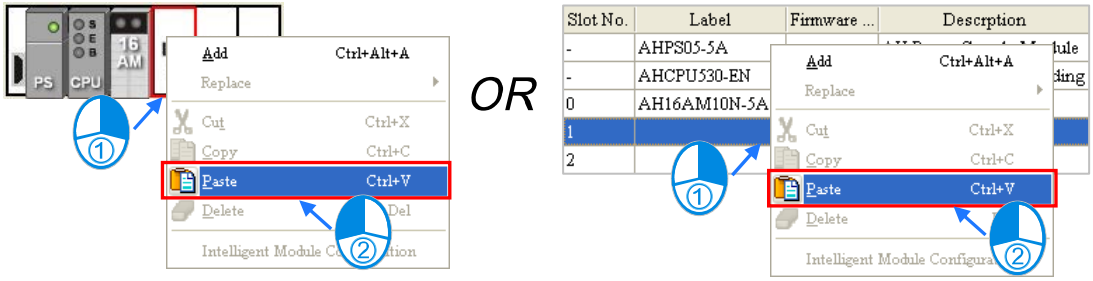
  (2) Click **Copy** on the **Edit** menu, or on the toolbar.



There are two ways to paste a module.

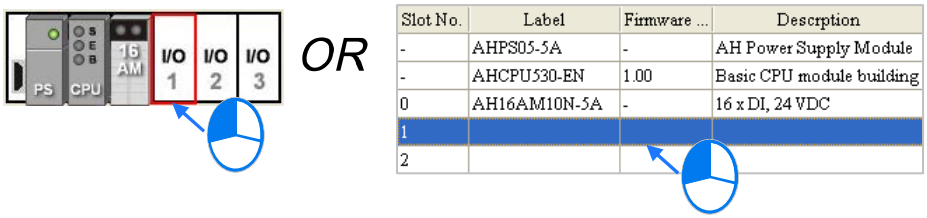
- **Method 1**

  Right-click a slot on which a module will be pasted in the system configuration area or on the information list, and then click **Paste** on the context menu.

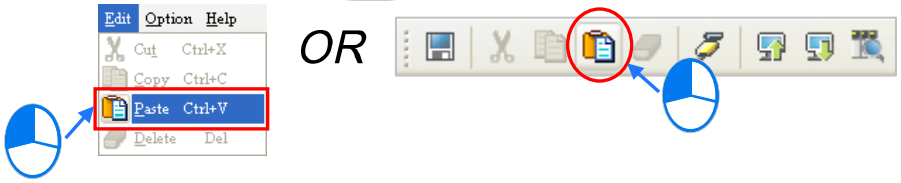


- **Method 2**

  (1) Click a slot on which a module will be pasted in the system configuration area or on the information list.



  (2) Click **Paste** on the **Edit** menu, or on the toolbar.

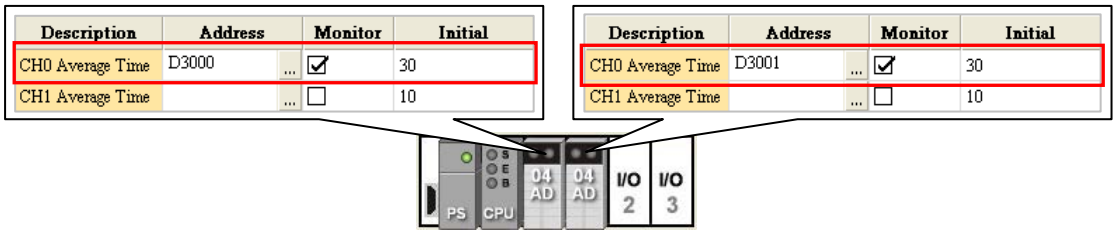


Additional remark

When a module is copied/pasted, the parameters in the module are processed as follows.

- **Input/Output device range**: The input/output devices assigned to the module cut are automatically assigned to the module pasted.
- **Comment about the module**: The comment about the module copied is copied into the module pasted.
- **Parameters in the module**: The parameters in the module cut are copied into the module pasted.
- **Data registers**: The data registers assigned to the module copied are automatically assigned to the module pasted.
- **Parameters in the intelligent module**: The setting of the parameters in the intelligent module is not copied, and is restored to the default values. The users have to set the parameters again.

As the example below illustrates, the module on slot 0 is the module which is copied, and the module on slot 1 is the module which is pasted.



Slot No.	Label	Firmware ...	Description	Input Device ...	Output Device ...	Comment
-	AHPS05-5A	-	AH Power Supply Module	None	None	
-	AHCPU530-EN	1.00	Basic CPU module building	None	None	
0	AH04AD-5A	1.00	4 x AI 16bit	D0 ~ D7		AI
1	AH04AD-5A	1.00	4 x AI 16bit	D8 ~ D15		AI
2						

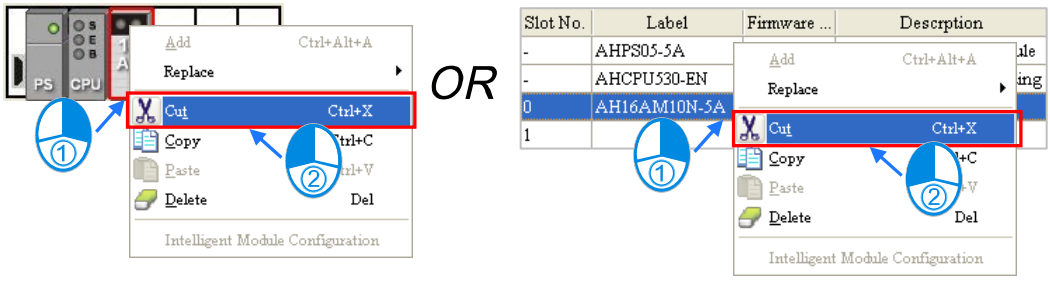
***. Some intelligent modules can be set by means of exclusive configuration tools. Please refer to section 3.4.4 for more information.**

3.2.2.8 Cutting / Pasting a Module

There are two ways to cut a module. (The CPU module and the power supply module can not be cut/pasted.)

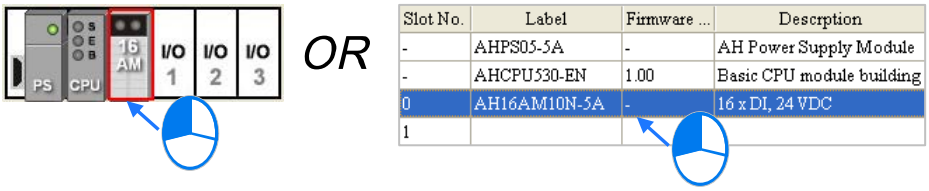
- **Method 1**

  Right-click a module which will be cut in the system configuration area or on the information list, and then click **Cut** on the context menu.



- **Method 2**

  (1) Click a module which will be cut in the system configuration area or on the information list.

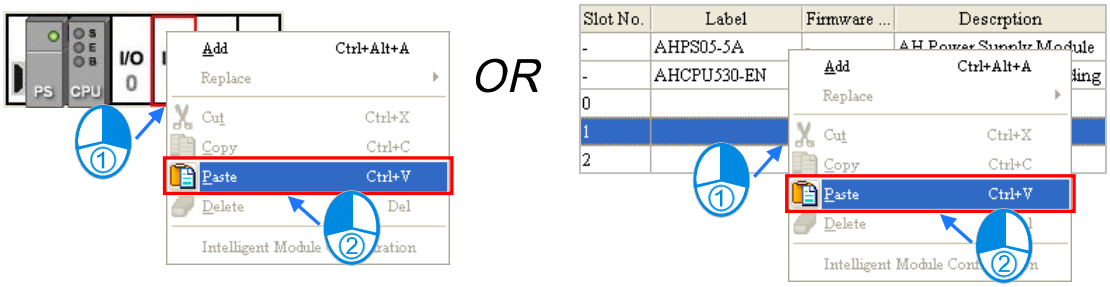


  (2) Click **Cut** on the **Edit** menu, or on the toolbar.

There are two ways to paste a module.

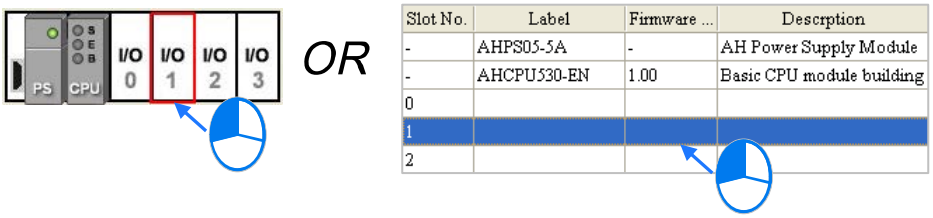
- **Method 1**

  Right-click a slot on which a module will be pasted in the system configuration area or on the information list, and then click **Paste** on the context menu.

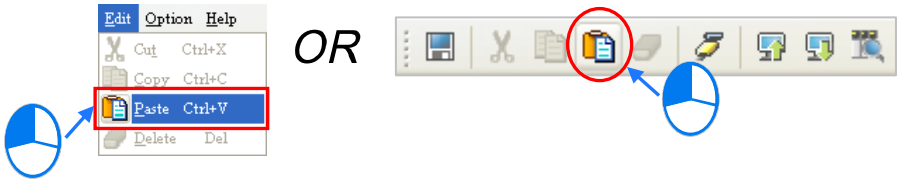


- **Method 2**

(1) Click a slot on which a module will be pasted in the system configuration area or on the information list.



(2) Click **Paste** on the **Edit** menu, or on the toolbar.



Additional remark

A module which is cut can only be pasted once. When a module is cut/pasted, the parameters in the module are processed as follows.

- **Input/Output device range**: The input/output devices assigned to the module cut are automatically assigned to the module pasted.
- **Comment about the module**: The comment about the module cut is copied into the module pasted.
- **Parameters in the module**: The parameters in the module cut are copied into the module pasted.
- **Data registers**: The data registers assigned to the module cut are automatically assigned to the module pasted.
- **Parameters in the intelligent module**: The setting of the parameters in the intelligent module is copied.

***. Some intelligent modules can be set by means of exclusive configuration tools. Please refer to section 3.4.4 for more information.**

3.2.2.9 Dragging a Module

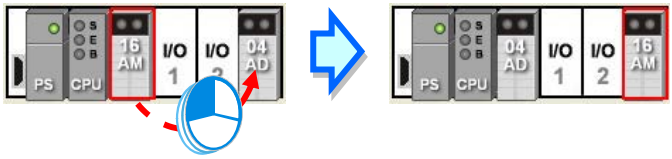
All modules in the system configuration area can be dragged by the mouse except the CPU module and the power supply module.

- **Condition 1**

Drag a module to a vacant slot when the left mouse button is held. The module will be moved to a new position.

- **Condition 2**

  Drag a module to an occupied slot when the left mouse button is held. The two modules will be interchanged.



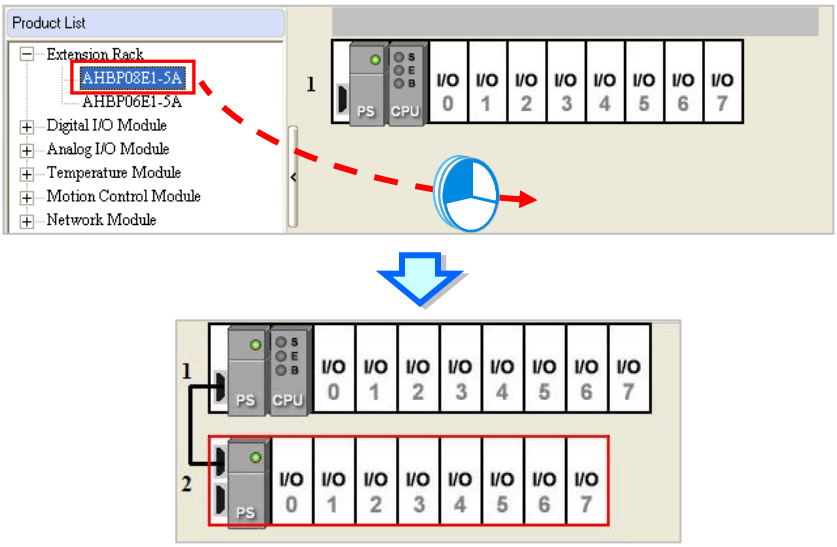
*** When a module is dragged, the input/output devices assigned to the module, the comment about the module, the parameters in the module, the data registers assigned to the module, and the parameters in the intelligent module are also dragged.**

3.2.2.10 Adding an Extension Rack

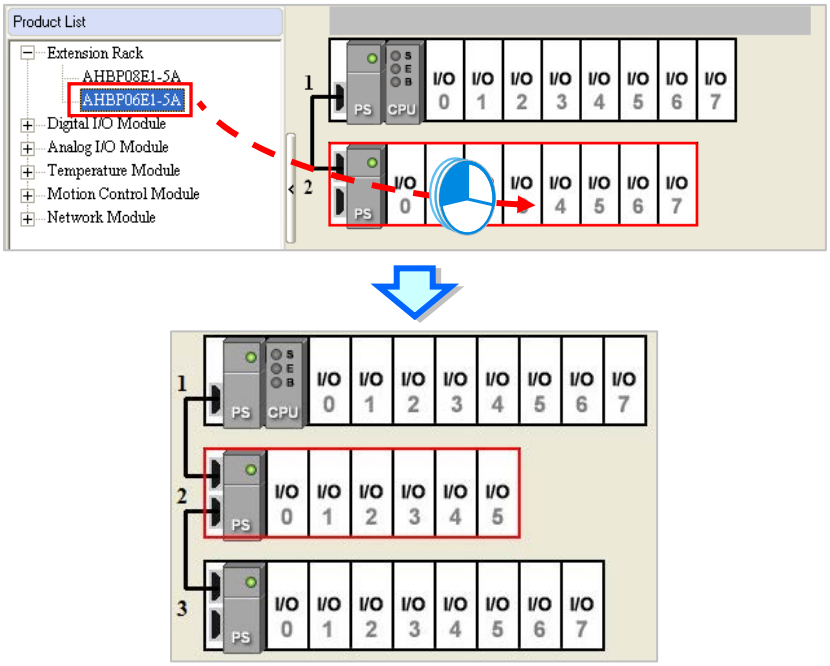
There are two ways to add an extension module.

- **Method 1**

  Drag an extension rack which will be added to the product list to the system configuration area. After the extension rack is dragged to the blank in the system configuration area, it will be under the present rack.



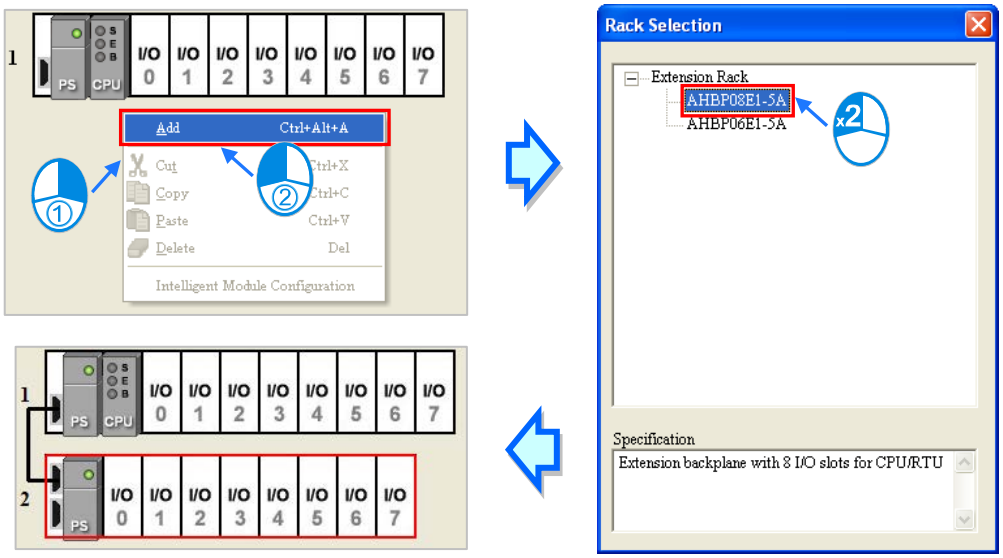
If an extension rack is dragged to a present rack, it will be put above the present rack.



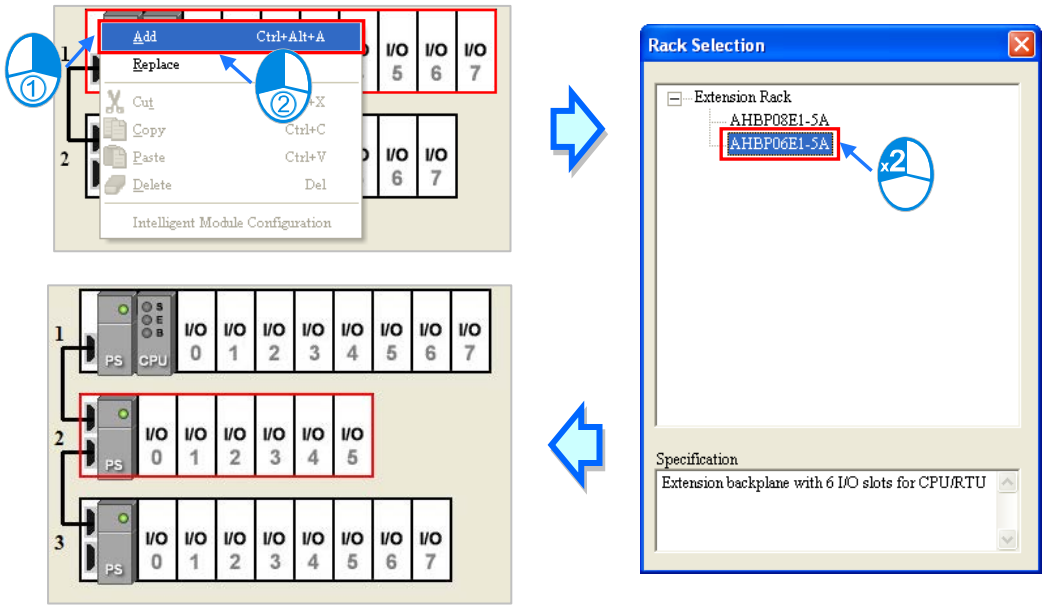
***. An extension rack can not be put above the main rack.**

- **Method 2**

  Right-click the blank in the system configuration area, click **Add** on the context menu, and double-click an extension rack which will be added in the **Rack Selection** window. The extension rack will be under the present rack.



  Right-click the left side of a rack, click **Add** on the context menu, and double-click an extension rack which will be added in the **Rack Selection** window. The extension rack will be under the rack.

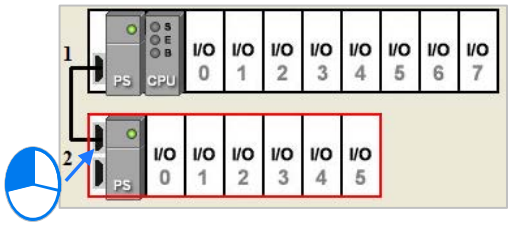


3.2.2.11 Deleting a Rack

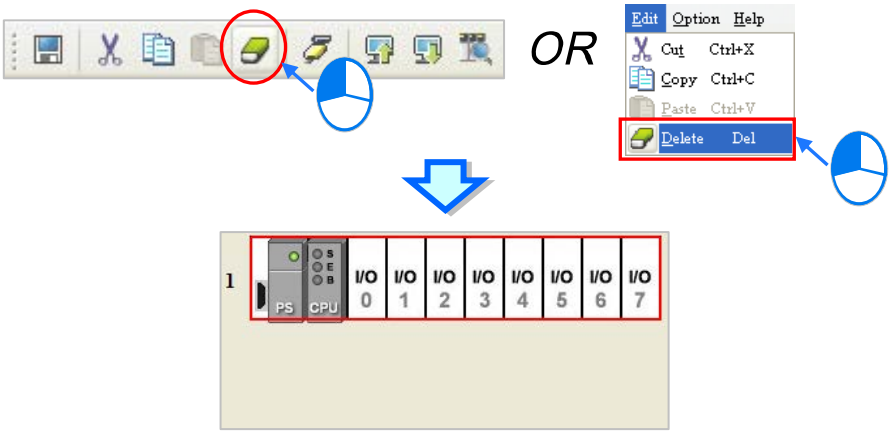
There are two ways to delete a rack. (The main extension rack can not be deleted.)

- **Method 1**

(1) Click the left side of a rack which will be deleted.

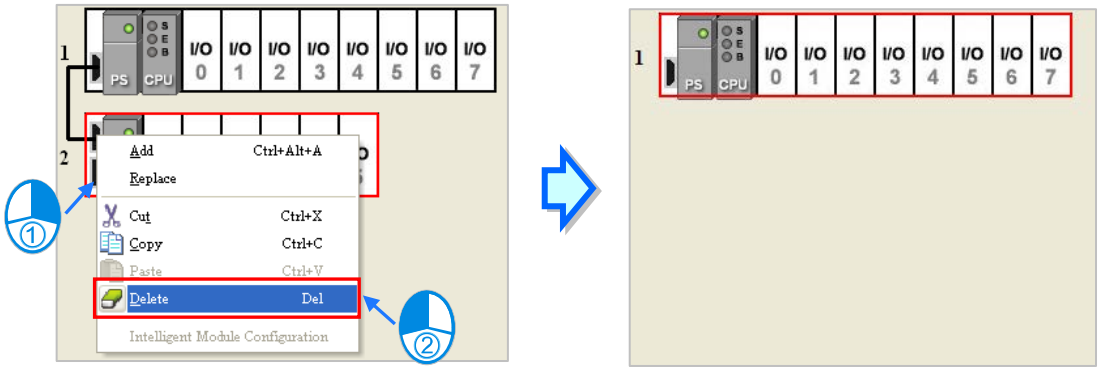


(2) Click **Delete** on the **Edit** menu, click on the toolbar, or press Delete on the keyboard.



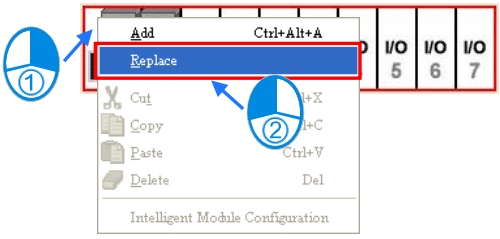
- **Method 2**

Right-click the left side of a rack which will be deleted, and then click **Delete** on the context menu.

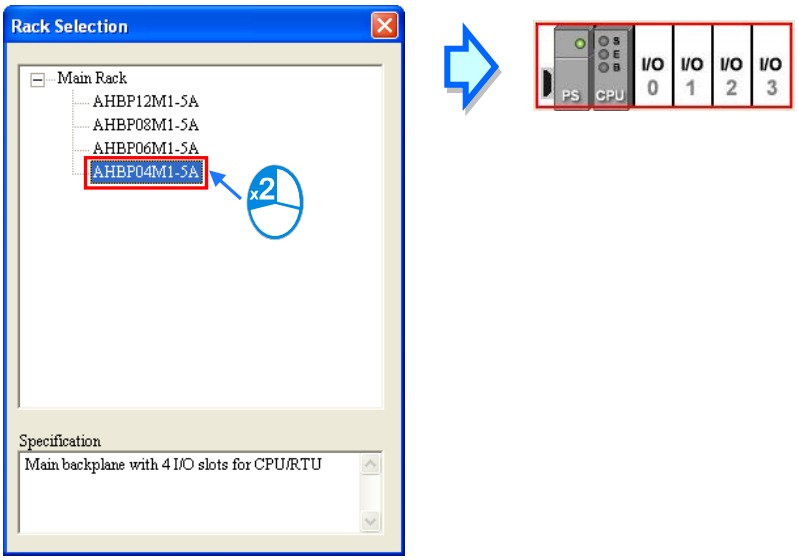


3.2.2.12 Replacing a Rack

(1) Right-click the left side of a rack which will be replaced, and then click **Replace** on the context menu.

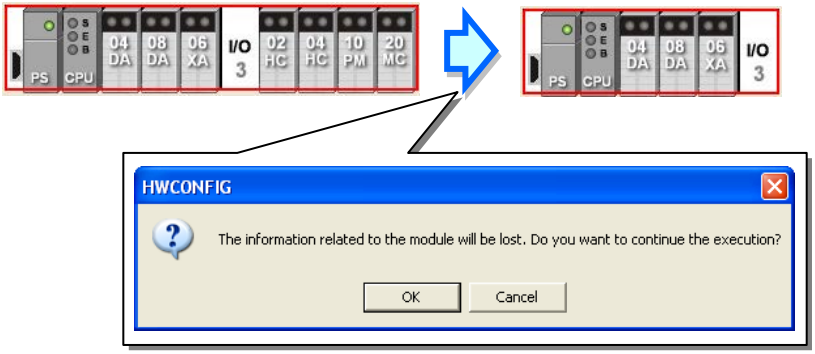


(2) Double-click a rack in the **Rack Selection** window.



Additional remark

If the number of slots on the new rack is less than the number of slots on the original rack, the modules on the original rack which can not be installed on the new rack will be deleted automatically. As a result, users have to make sure of the number of slots on the new rack before they replace a rack.

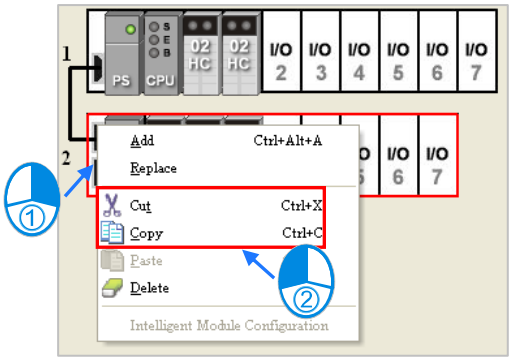


3.2.2.13 Cutting / Copying / Pasting an Extension Rack

There are two ways to copy or cut an extension rack. (The main rack can not be copied/cut/pasted.)

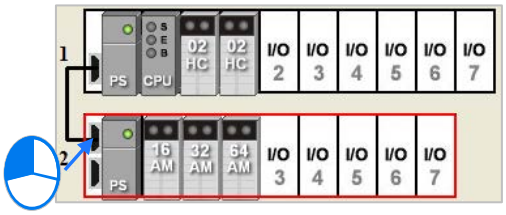
- **Method 1**

  Right-click the left side of a rack which will be copied or cut, and then click **Copy** or **Cut** on the context menu.

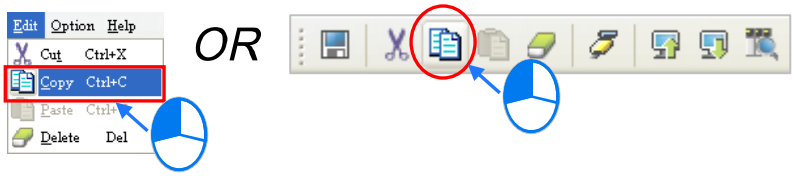


- **Method 2**

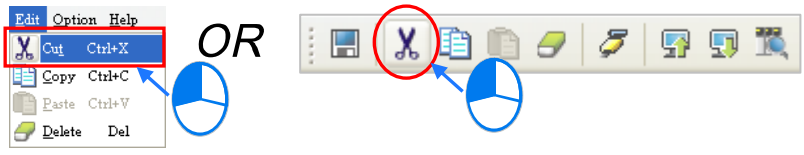
  Click the left side of a rack which will be copied or cut.



If users want to copy the rack, they can click **Copy** on the **Edit** menu, or click on the toolbar.



If users want to cut the rack, they can click **Cut** on the **Edit** menu, or click on the toolbar.

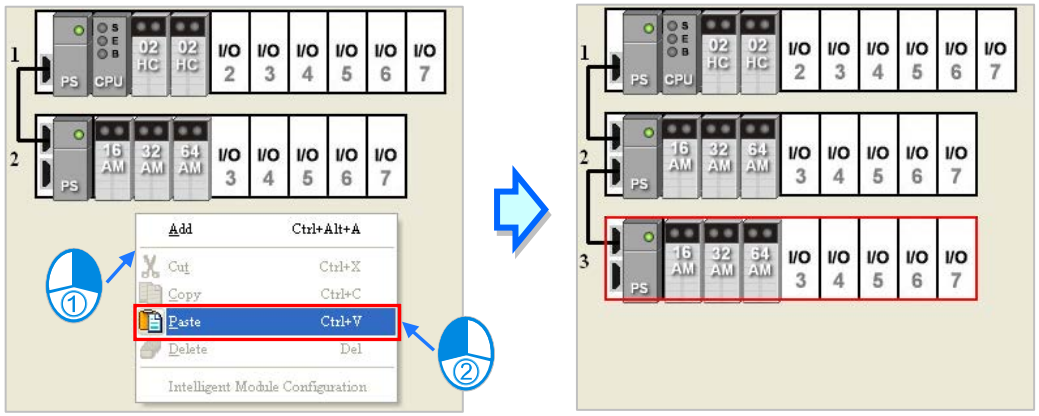


There are two ways to paste an extension rack. If an extension rack is pasted on the blank in the

system configuration area, it will be under the present racks. If an extension rack is pasted on a present extension rack, it will be put above the present extension rack.

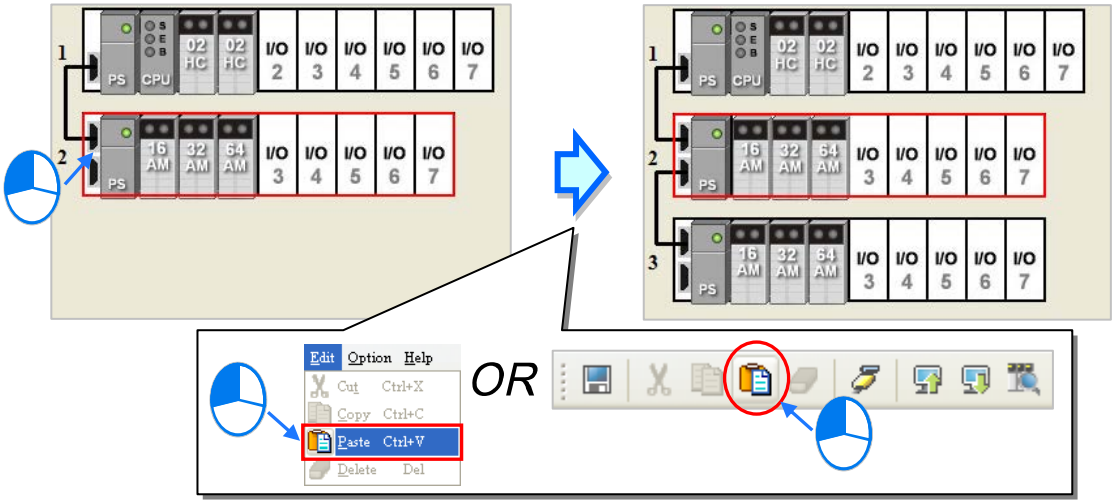
- **Method 1**

  Right-click the blank or a present rack in the system configuration area, and then click **Paste** on the context menu.



- **Method 2**

  Click the blank or a present rack in the system configuration area, and then click **Paste** on the **Edit** menu, or on the toolbar.



Additional remark

When an extension rack is cut/pasted, the parameters in the modules on the extension rack are processed as follows.

- **Input/Output device range**: The input/output devices assigned to the modules on the extension rack cut are automatically assigned to the modules on the extension rack pasted.
- **Comments about the modules**: The comments about the modules on the extension rack cut are copied into the modules on the extension rack pasted.
- **Parameters in the modules**: The parameters in the modules on the extension rack cut are copied into the modules on the extension rack pasted.
- **Data registers**: The data registers assigned to the modules on the extension rack cut are automatically assigned to the modules on the extension rack pasted.
- **Parameters in the intelligent modules**: The setting of the parameters in the intelligent modules is copied.

When an extension rack is copied/pasted, the parameters in the modules on the extension rack are processed as follows.

- **Input/Output device range**: The input/output device addresses assigned to the modules on the

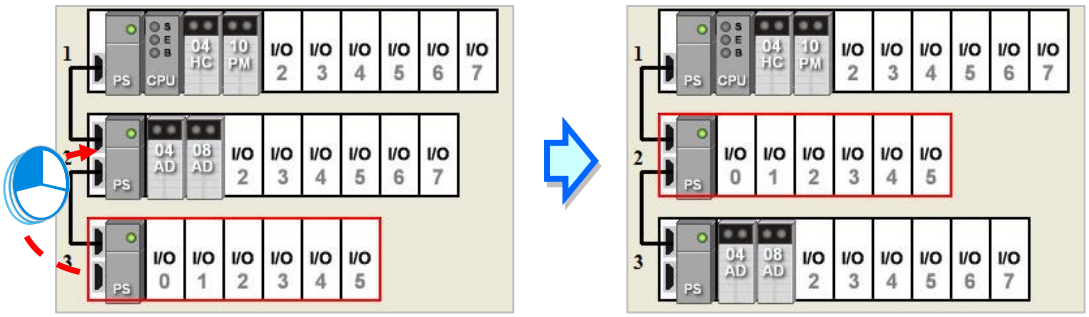
extension rack pasted follow the input/output device addresses assigned to the modules on the extension rack copied.

- **Comments about the modules**: The comments about the modules on the extension rack copied are copied into the modules on the extension rack pasted.
- **Parameters in the modules**: The parameters in the modules on the extension rack cut are copied into the modules on the extension rack pasted.
- **Data registers**: The data register addresses assigned to the modules on the extension rack pasted follows the data registers addresses assigned to the modules on the extension rack copied.
- **Parameters in the intelligent modules**: The setting of the parameters in the intelligent modules is not copied, and is restored to the default values.

***. Some intelligent modules can be set by means of exclusive configuration tools. Please refer to section 3.4.4 for more information.**

3.2.2.14 Dragging an Extension Rack

Drag an extension rack to another extension rack when the left mouse button is held. The two extension racks will be interchanged. When the extension rack is dragged, the input/output devices assigned to the modules on the extension rack, the comments about the modules on the extension rack, the parameters in the modules on the extension rack, the data registers assigned to the modules on the extension rack, and the parameters in the intelligent modules are also dragged. However, the main rack is always at the top of the system configuration area.

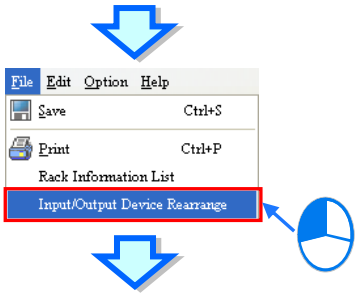


3.2.2.15 Rearranging the Input/Output Devices

During a module configuration, modules may be dragged or copied/cut/pasted several times. The input/output devices may be automatically assigned to the modules by the system, and may not be arranged in order. After users click **Input/Output Device Rearrange** on the **File** menu, the input/output devices assigned to all modules will be rearranged according to the slot numbers and the rack numbers.

The rearrangement of the input/output devices does not change the data registers assigned to the modules. During the rearrangement, if the input/output devices assigned to the modules conflicts with the data registers assigned to the modules, the system will skip the data registers assigned to the module, and provide the devices which can be used.

Slot No.	Label	Firmware ...	Description	Input Device ...	Output Device ...	Comment
-	AHPS05-5A	-	AH Power Supply Module	None	None	
-	AHCPU530-EN	1.00	Basic CPU module building	None	None	
0	AH04AD-5A	1.00	4 x AI 16bit	D64 ~ D71		
1	AH04AD-5A	1.00	4 x AI 16bit	D16 ~ D23		
2	AH04DA-5A	1.00	4 x AO 16bit		D32 ~ D39	
3	AH04DA-5A	1.00	4 x AO 16bit		D88 ~ D95	



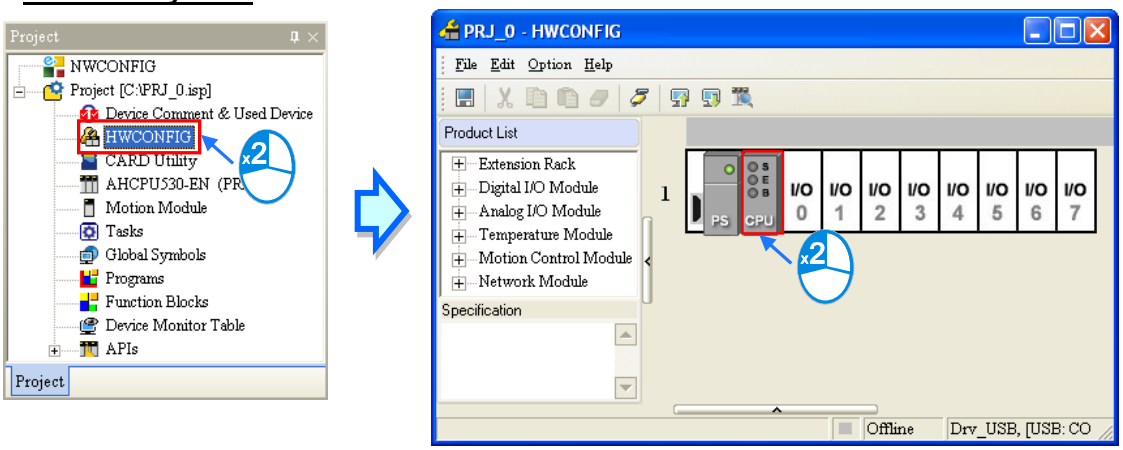
Slot No.	Label	Firmware ...	Description	Input Device ...	Output Device ...	Comment
-	AHPS05-5A	-	AH Power Supply Module	None	None	
-	AHCPU530-EN	1.00	Basic CPU module building	None	None	
0	AH04AD-5A	1.00	4 x AI 16bit	D0 ~ D7		
1	AH04AD-5A	1.00	4 x AI 16bit	D8 ~ D15		
2	AH04DA-5A	1.00	4 x AO 16bit		D16 ~ D23	
3	AH04DA-5A	1.00	4 x AO 16bit		D24 ~ D31	

3.3 Setting the Parameters in an AH500 Series CPU Module

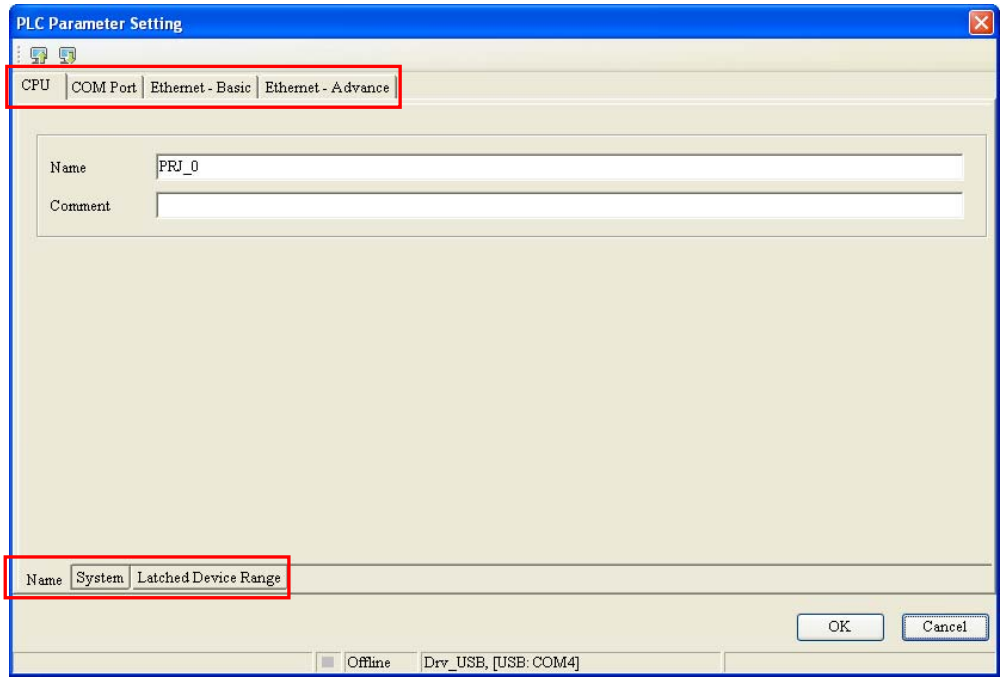
3.3.1 Opening the PLC Parameter Setting Window

After users double-click the CPU module in the system configuration area, the **PLC Parameter Setting** window will appear. The parameters which can be set vary with the models of the CPU modules.

Before the parameters in a CPU module are set, users have to refer to the operation manual for the CPU module. To prevent damage to the system or staff, the users have to make sure of the effect that the parameters which are set have on the CPU module and the whole system.



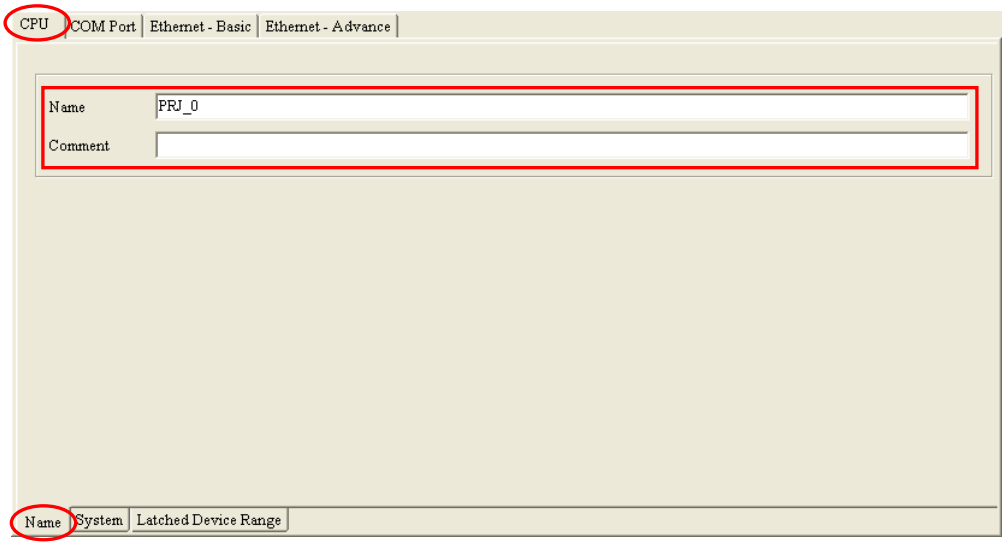
The parameters are classified into several types. The users can click the primary tabs at the top of the window, and the secondary tables at the bottom of the window to set the parameters.



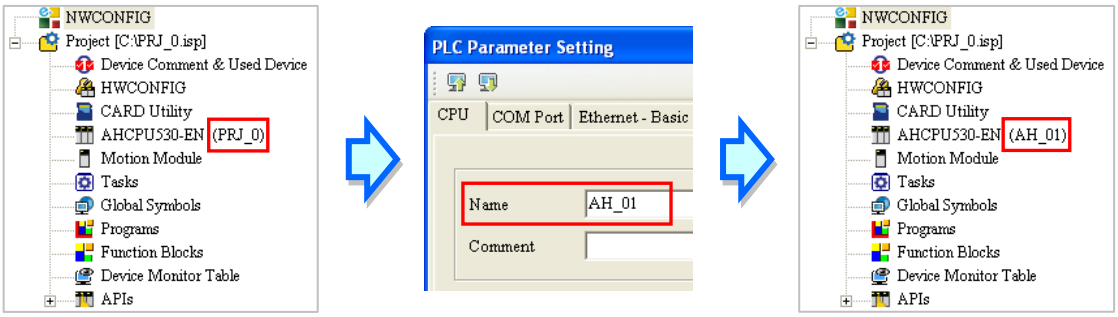
3.3.2 Setting the Basic CPU Parameters

3.3.2.1 CPU: Name

After users click the **CPU** tab at the top of the window, and the **Name** tab at the bottom of the window, they can type 32 characters at most in the **Name** box, and 60 characters at most in the **Comment** box. There is no restriction on characters which can be typed in the boxes. Characters typed in the boxes can include special marks and spaces. However, the users have to notice that a Chinese character occupies two characters.



After an ISPSoft project is created, the project name will be taken as the default name of the CPU module, and the default name of the CPU module will be attached to the model in the project management area. Users can change the default name of the CPU module in the **Nam**e box later.

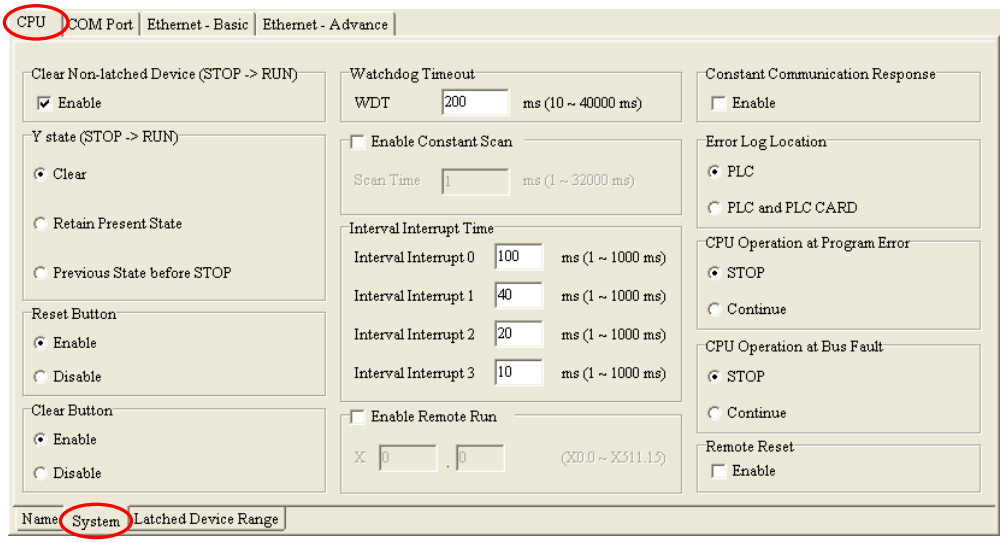


Users can identify a device by means of the name of the device. When several devices are connected on a network, users can check whether a device connected to the computer is the device they expect by means of the name of the device. To prevent unexpected effect on other CPU modules, if users want to download/upload the program, but the name of the CPU module is different from the name attached to the model in the project management area, the system will remind the users to check the name of the CPU module and the name attached to the model in the project management area.

***. The parameters set in HWCONFIG must be downloaded to the CPU module so that they can take effect. (Please refer to section 3.3.6 or section 3.5.4 for more information.)**

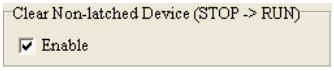
3.3.2.2 CPU: System

After users click the **CPU** tab at the top of the window, and the **System** tab at the bottom of the window, they can set the parameters in the CPU module.



- **Clear Non-latched Device (STOP→RUN)**

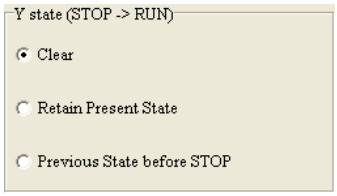
  If the **Enable** checkbox is selected, the states of the non-latched devices, and the values in the non-latched devices are cleared when the CPU module begins to run.



- **Y state (STOP→RUN)**

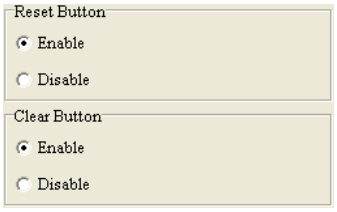
  When the CPU module begins to run, the states of the Y devices are OFF, retained, or restored to the states before the CPU module's stopping running.

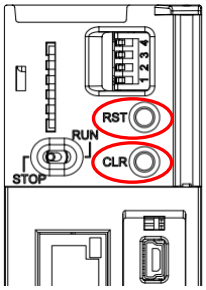
  - **Clear**: All Y devices are set to OFF.
  - **Retain Present State**: The states of the Y devices are retained.
  - **Previous State before STOP**: The states of the Y devices are restored to the states before the CPU module's stopping running.



- **Reset Button & Clear Button**

  Whether the RST button and CLR button on the CPU module is enabled or disabled depends on the setting of these parameters. Please refer to an operation manual for more information about the functions of the buttons.



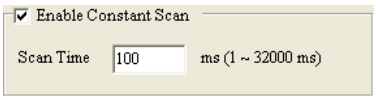


- **Watchdog Timeout**

  Users can set a timeout during which the program is scanned. Generally speaking, the watchdog timeout is reset whenever the scan of the program is complete. If the scan procedure can not be completed within the time set for some reason, the watchdog timeout occurs, and the CPU module will be in an erroneous condition. Please refer to operation manuals for more information about troubleshooting.

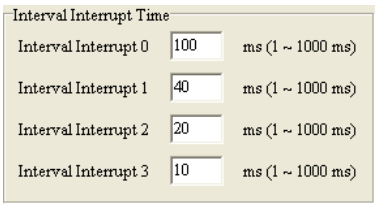
- **Enable Constant Scan**

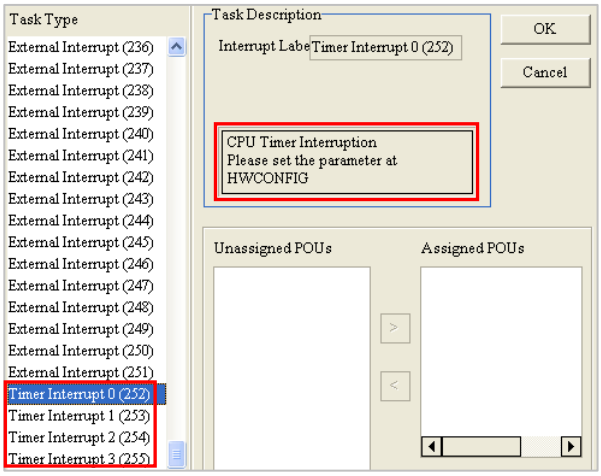
  Users can select the **Enable Constant Scan** checkbox, and set a scan time. If the actual scan time is less than setting value, the CPU module will not carry out the next scan until the setting value is reached. If the actual scan time is larger than the setting value, the CPU module will ignore the setting value, and operate according to the actual scan time. Besides, if the scan time set is larger than the watchdog timeout set, a watchdog timeout occurs when the CPU module operates.



- **Interval Interrupt Time**

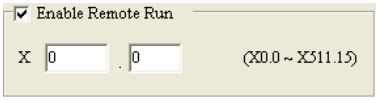
  An AH500 series CPU module provides four interrupts. Users can set intervals of triggering the interrupts. In ISPSoft, a timed interrupt is created through a task and a POU. Please refer to chapter 5 for more information.





- **Enable Remote Run**

  If the **Enable Remote Run** checkbox is selected, users can specify an X device which controls the status of the CPU module. For example, the CPU module runs when the state of X0.0 is ON, and the CPU module stops running when X0.0 is OFF.



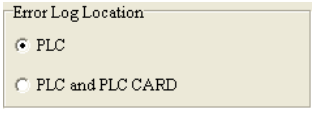
- **Constant Communication Response**

  If the **Enable** checkbox is not selected, commands received through the communication ports will not be processed until the scan cycle is complete. If the **Enable** checkbox is selected, commands received through the communication ports will be processed every specific period of time. However, the scan procedure is interrupted when the system processes commands received through the communication ports. As a result, the scan time will be prolonged if the **Enable** checkbox is not selected. Please make sure that the operation of the system is not affected when the function is used.

- **Error Log Location**

  If the **PLC** option button is selected, error logs will be stored in the special data registers in the CPU modules. If the **PLC and PLC CARD** option button is selected, error logs will be stored not only in the CPU module, but also in the memory card.

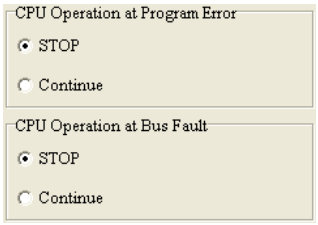
  Twenty error logs at most can be stored in an AH500 series CPU module. If there are more than twenty error logs, the oldest error log will be overwritten by the latest error log. However, if the **PLC and PLC CARD** option button is selected, and there are more than twenty error logs, the oldest error log will be backed up in the memory card before the oldest error log is overwritten.



- **CPU Operation at Program Error & CPU Operation at Bus Fault**

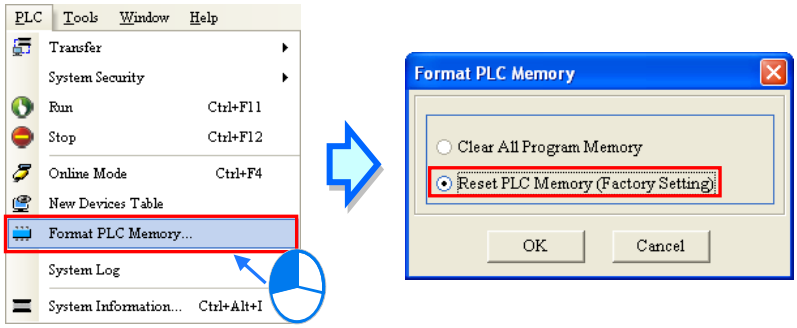
  If an error occurs, the status of the CPU may change. The status of the CPU depends on an error code generated. Users can define the status of the CPU module for some errors. They can select the **Stop** option button or the **Continue** option button in the **CPU Operation at Program Error** section, select the **Stop** option button or the **Continue** option button in the **CPU Operation at Bus Fault** section, and download the parameters to the CPU module. Please refer to operation

manuals or related technical documents for more information.



- **Remote Reset**

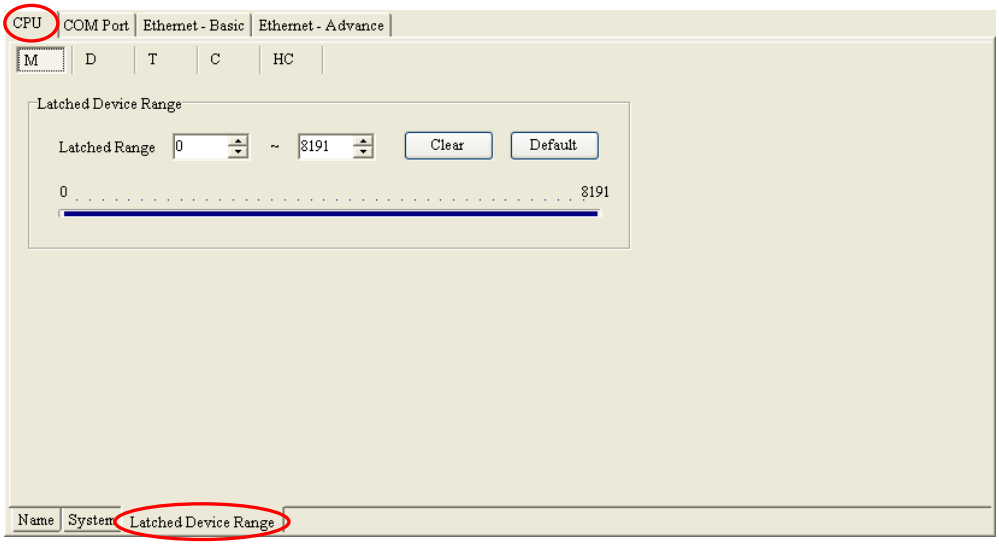
  If the **Enable** checkbox is selected, the CPU module can be reset through a remote system. After users select the **Enable** checkbox, and download the parameter to the CPU module, they can click **Format PLC Memory…** on the **PLC** menu, and select the **Reset PLC Memory (Factory Setting)** option button in the **Format PLC Memory** window.



***. The parameters set in HWCONFIG must be downloaded to the CPU module so that they can take effect. (Please refer to section 3.3.6 or section 3.5.4 for more information.)**

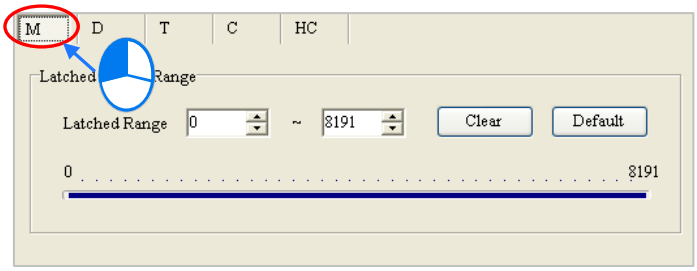
3.3.2.3 CPU: Latched Device Range

After users click the **CPU** tab at the top of the window, and the **Latched Device Range** tab at the bottom of the window, they can set a range of latched devices.

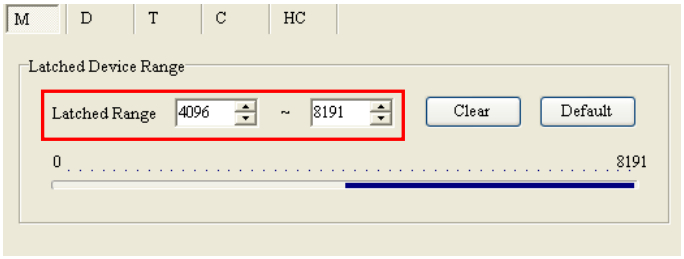


(1) After users click a device type, they can be set a range of latched devices. There is a rectangular bar in the setting area. The values at the two ends of the bar represent a maximal

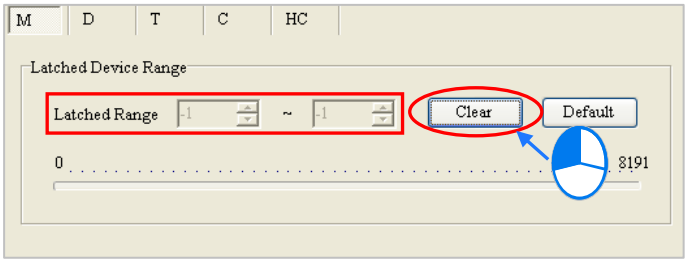
value and a minimal value. The blue part represents a range of latched devices.



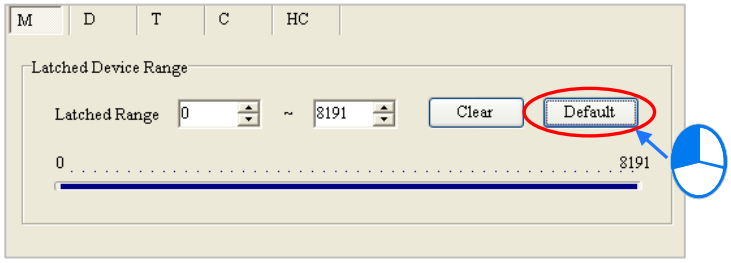
(2) Please select values in the boxes. The value in the left box is a start address, and the value in the right box is an end address. The value in the left box must be less than the value in the right box. Besides, the value in the left box can not be less than the minimal value at the left end of the rectangular bar, and the value in the right box can not be larger than the maximal value at the right end of the rectangular bar.



(3) If all the devices in the block are not latched devices, the users can click **Clear**. After the users click **Clear**, the boxes become gray, and the range of latched devices will be -1~-1.



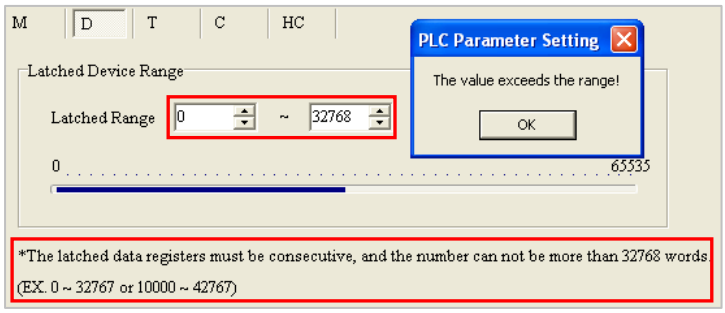
(4) If the users want to restore the setting to the default values, they can click **Default**. After the users click **Default**, the setting will be restored to the default values.



Additional remark

A range of latched devices of a certain type may not include all the devices of that type. If there is restriction on the maximum number of latched devices of a certain type, a related hint will be under the setting area. In the setting page for the data registers in AHCPU530-EN, the hint indicates that the maximum number of latched data registers is 32768. In other words, in the range between 0 and 65535, the number of latched data registers should be less than or equal to 32768, e.g. 0~32767 or 10000~42767.

In the setting page below, the range of latched data registers is D0~D32768. The number of latched data registers is 32769. When users leave this page, the **PLC Parameter Setting** dialog box appears.

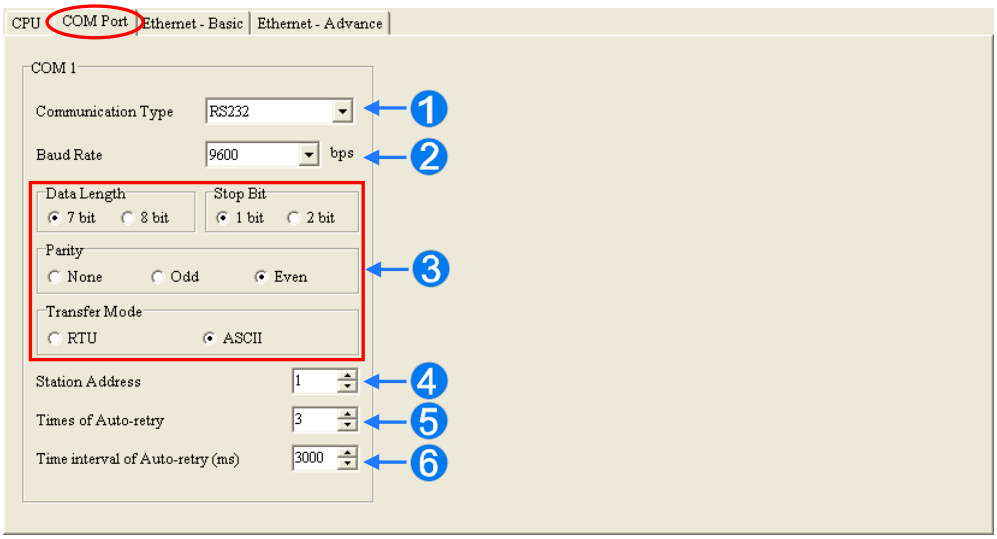


***. The parameters set in HWCONFIG must be downloaded to the CPU module so that they can take effect. (Please refer to section 3.3.6 or section 3.5.4 for more information.)**

3.3.3 COM Port

After users click the **COM Port** tab at the top of the window, they can set the communication ports in the CPU module. If the CPU module is equipped with two communication ports, there are two setting areas for the two communication ports, and the two communication ports are set individually. (AHCPU5xx-RS2 is equipped with two communication ports.)

Please refer to section 2.4.7 for more information about connecting to a CPU module through a communication port.



1. Users can select **RS232**, **RS485**, or **RS422** in the **Communication Type** drop-down list box.
2. Users can select a communication speed in the **Baud Rate** drop-down list box. If a RS-232 cable is used, the communication speed can be 9600 bps, 19200 bps, 38400 bps, 57600 bps, 115200 bps. If a RS-485 cable or a RS-422 cable is used, the communication speed can be 9600 bps, 19200 bps, 38400 bps, 57600 bps, 115200 bps, 230400 bps, 460800 bps, or 921600 bps.
3. Users can set the communication protocol parameters. If the **RTU** option button is selected in the **Transfer Mode** section, the **8 bit** option button in the **Data Length** section is automatically selected.
4. Users can set a station address. A device on a network can be identified by means of the station address of the device. The station address of a device on a network can not be the same as the station address of another device on the same network. The station address of a device must be within the range between 0 and 247.

   If the communication port functions as a slave, and there are other slaves, the station address of the communication port can not be 0. Station address 0 has the meaning of broadcasting to all slaves in a communication protocol. If a master specifies in a data packet that data must be sent to station address 0, the data will be sent to all slaves. No matter what station address of these slaves are, these slaves will receive the data packet.

5 If the sending of a command fails, the CPU module will retry the sending of the command. Users can set the number of times the sending of a command is retired in this box. The number of times the sending of a command is retired must be within the range between 0 and 20.

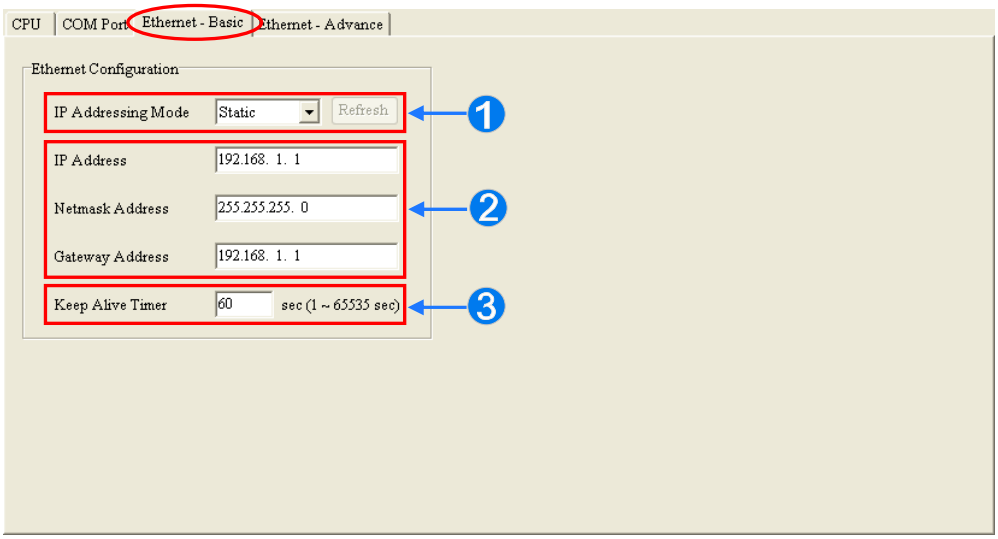
6 Users can set an interval of retrying the sending of a command. If the sending of a command fails, the CPU module will retry the sending of the command every specific period of time. The interval of retrying the sending of a command must be within the range between 100 milliseconds and 65535 milliseconds.

***. The parameters set in HWCONFIG must be downloaded to the CPU module so that they can take effect. (Please refer to section 3.3.6 or section 3.5.4 for more information.)**

3.3.4 Ethernet—Basic

If a CPU module is equipped with an Ethernet port, users can enter this page. For example, AHCPU5xx-EN is equipped with an Ethernet port. After users click the **Ethernet–Basic** tab at the top of the window, they can set the Ethernet port in the CPU module.

The users can set the communication parameters for the Ethernet port in the CPU module in this page. They can refer to section 2.4.7 for more information about connecting to a CPU module through the Ethernet port on the CPU module, and refer to other related documents or manuals for more information about Ethernet.



1 If users select **Static** in the **IP Addressing** Mode drop-down list box, they can specify an IP address. If **Dynamic** is selected in the **IP Addressing Mode** drop-down list box, an IP address is assigned by a DHCP server.

2 If **Static** is selected in the **IP Addressing Mode** drop-down list box, users can assign an IP address, a subnet mask, and a gateway address.

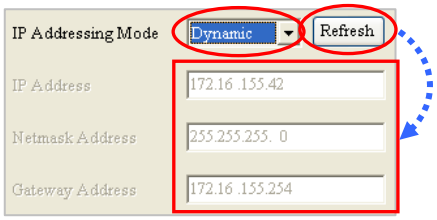
3 If no data is transmitted from the CPU module on a network, and the keepalive period has elapsed, the CPU module will be disconnected from the network automatically.

If **Dynamic** is selected in the **IP Addressing Mode** drop-down list box, the IP address, the subnet mask, and the gateway can be read from the CPU module by means of the following steps.

(1) Connect the CPU module to Ethernet, and make sure that the network and the CPU module operate normally.

(2) Select **Dynamic** in the **IP Addressing Mode** drop-down list box, and download the parameter to the CPU module.

(3) After **Refresh** is clicked, the system will read the IP address, the subnet mask, and the gateway from the CPU module. The IP address, the subnet mask, and the gateway will be displayed in the boxes.



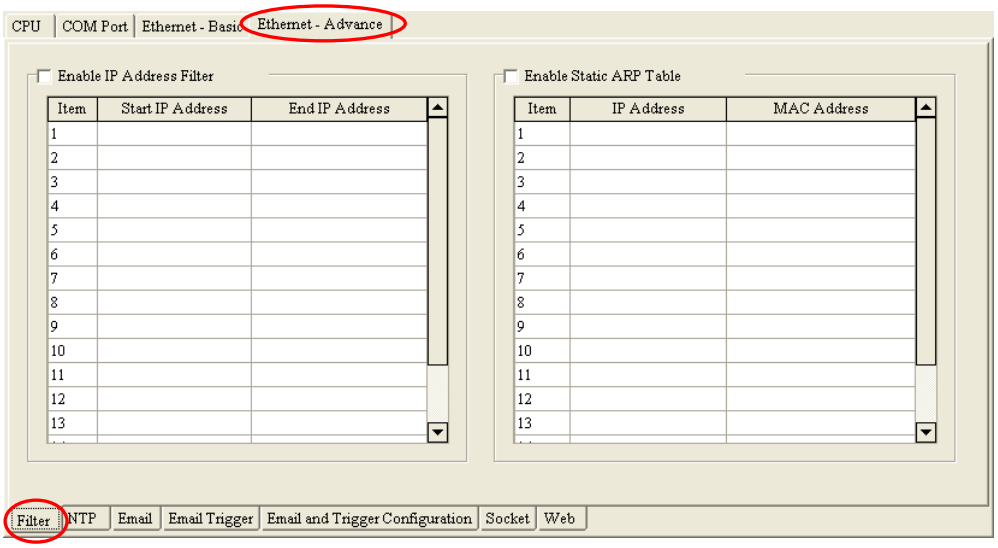
***. The parameters set in HWCONFIG must be downloaded to the CPU module so that they can take effect. (Please refer to section 3.3.6 or section 3.5.4 for more information.)**

3.3.5 Ethernet—Advance

If a CPU module is equipped with an Ethernet port, users can enter this page. For example, AHCPU5xx-EN is equipped with an Ethernet port. After users click the **Ethernet–Advance** tab at the top of the window, they can set devices on a network.

3.3.5.1 Ethernet–Advance: Filter

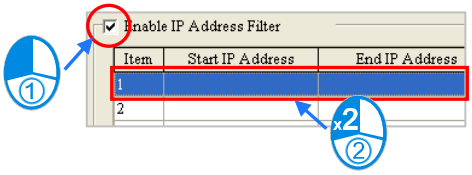
After users click the **Ethernet–Advance** tab at the top of the window, and the **Filter** tab at the bottom of the window, they can set a filter. Devices on a network are filtered. The setting here ensures that objects communicating with the CPU module are devices which are allowed.



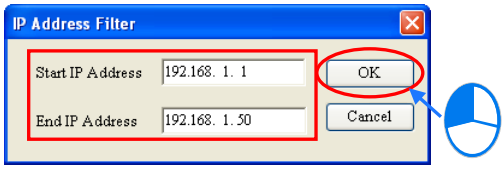
- **Enable IP Address Filter**

  If the **Enable IP Address Filter** checkbox is selected, devices whose IP addresses are listed in the table will be allowed to communicate with the CPU module, and the CPU module will discard data packets sent from devices whose IP addresses are not listed in the table. The steps of setting the function are as follows. Sixteen groups of IP address at most can be listed in the table.

  (1) Select the **Enable IP Address Filter** checkbox, and then double-click an item which will be modified or added in the table.



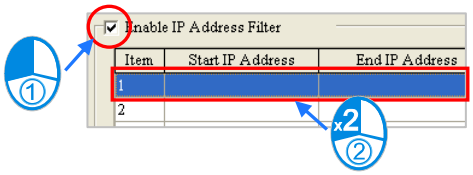
  (2) Type a start IP address and an end address in the **IP address Filter** window. The start IP address must be less than the end IP address. Finally, click **OK**.



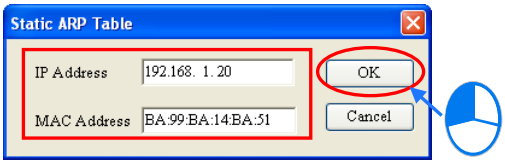
- **Enable Static ARP Table**

  Users can use this function to set MAC addresses and corresponding IP addresses. The MAC address of every device is unique. If the MAC address of device A is known, the MAC address can be bound to the IP address assigned to device A. The CPU module will regard the IP address as the exclusive address of device A. Even if device B is assigned the same IP address, the CPU module does not respond to device B. The steps of setting the function are as follows. Sixteen groups of addresses at most can be listed in the table.

  (1) Select the **Enable Static ARP Table** checkbox, and then double-click an item which will be modified or added in the table.



  (2) Type an IP address and a MAC address in the **Static ARP Table** window. Finally, click **OK**.



  Take the tables below for example. After the setting in the tables is downloaded to the CPU module, devices whose IP addresses are within the range between 192.168.1.1 and 192.168.1.50, and devices whose IP addresses are within the range between 192.168.1.100 and 192.168.1.150 are allowed to communicate with the CPU module if the CPU module operates. Besides, if the CPU module communicates with a device whose IP address of 192.168.1.20, the MAC address of the device must be BA:99:BA:14:BA:51. Likewise, if the CPU module communicates with a device whose IP address is 192.168.1.21, the MAC address of the device must be 00:10:5A:81:CA:FD.

Enable IP Address Filter

Item	Start IP Address	End IP Address
1	192.168.1.1	192.168.1.50
2	192.168.1.100	192.168.1.150
3		
4		
5		
6		
7		
8		
9		
10		
11		
12		
13		

Enable Static ARP Table

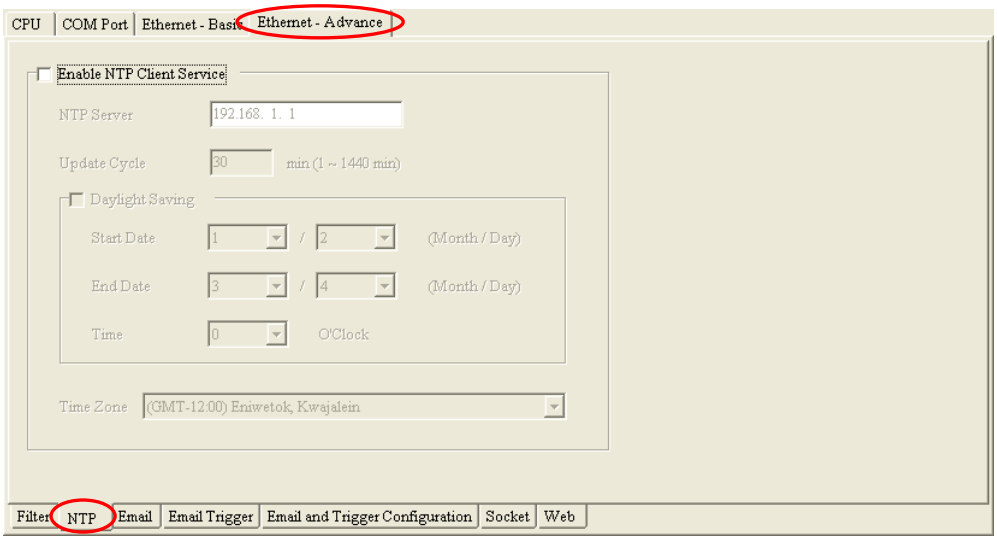
Item	IP Address	MAC Address
1	192.168.1.20	BA:99:BA:14:BA:51
2	192.168.1.21	00:10:5A:81:CA:FD
3		
4		
5		
6		
7		
8		
9		
10		
11		
12		
13		

***1. The parameters set in HWCONFIG must be downloaded to the CPU module so that they can take effect. (Please refer to section 3.3.6 or section 3.5.4 for more information.)**

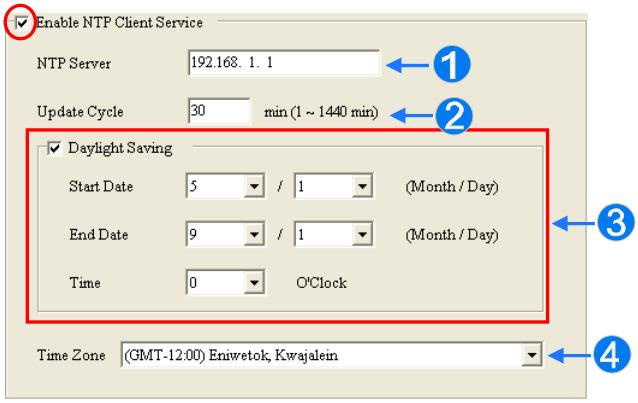
***2. If users want to know the MAC address of a CPU module, they can click Online Mode in HWCONFIG to view the information about the CPU module (section 3.5.6), or click System Information in ISPSoft to get the information about the CPU module connected and the status of the CPU module (section 14.5.1).**

3.3.5.2 Ethernet–Advance: NTP

After users click the **Ethernet–Advance** tab at the top of the window, and the **NTP** tab at the bottom of the window, they can enable the function of synchronizing the real-time clock in the CPU module to an NTP server, and carry out the related setting. NTP will not be introduced here. Please refer to related documents or manuals for more information about NTP.



Select the **Enable NTP Client Service** checkbox, and then set the related parameters.

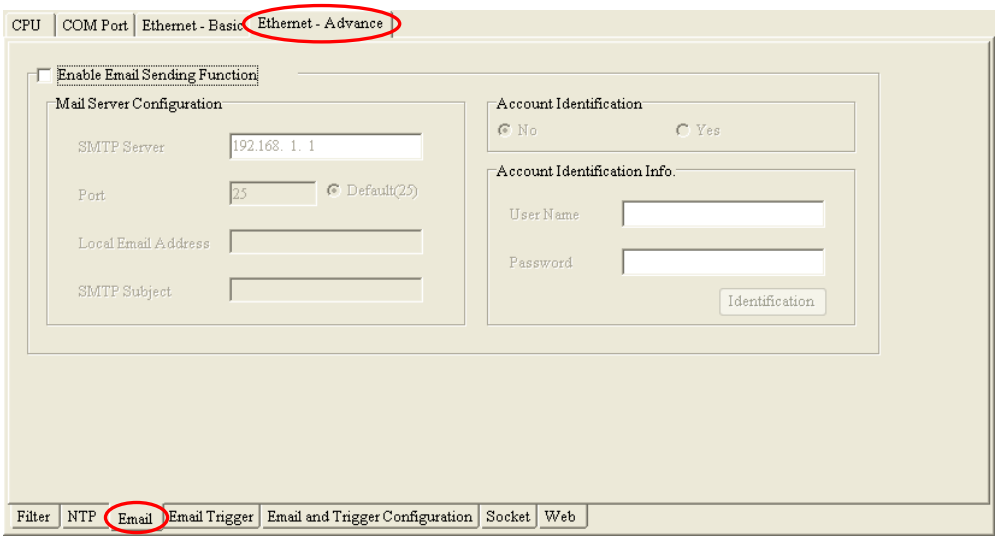


1. Users can set the IP address of an NTP server. The CPU module corrects the time inside itself by connecting to the server periodically.
2. Users can set an interval of correcting the time in the CPU module. If the interval is thirty minutes, the CPU module will connect to the NTP server every thirty minutes.
3. After users select the **Daylight Saving** checkbox, they can select a start date and an end data.
4. Users can select a time zone in the **Time Zone** drop-down list box.

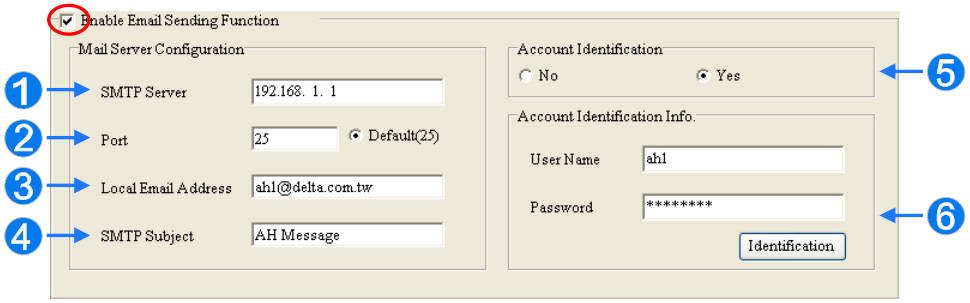
***. The parameters set in HWCONFIG must be downloaded to the CPU module so that they can take effect. (Please refer to section 3.3.6 or section 3.5.4 for more information.)**

3.3.5.3 Ethernet–Advance: Email

After users click the **Ethernet–Advance** tab at the top of the window, and the **Email** tab at the bottom of the window, they can set the mail sending function.



After users select the **Enable Email Sending Function** checkbox, they can set the related parameters.



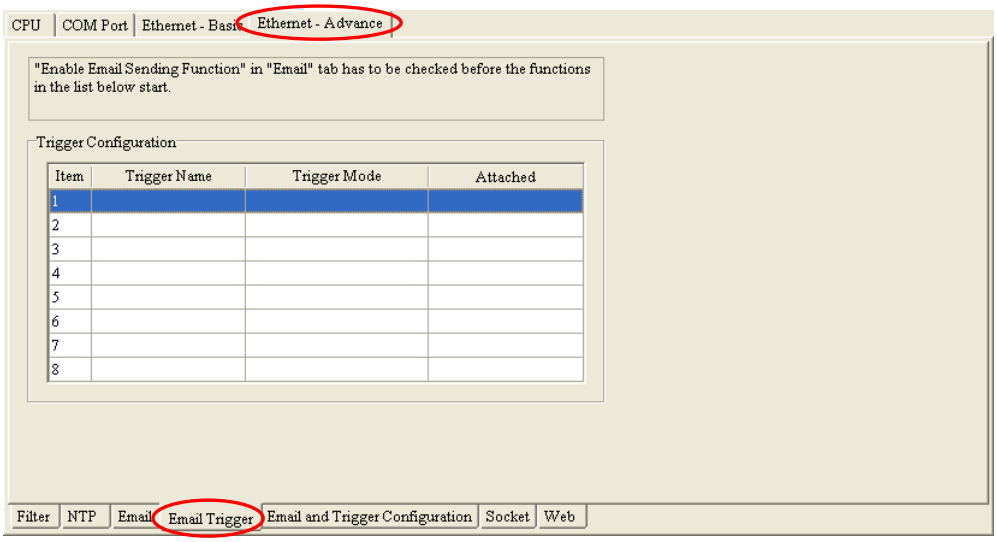
1. **SMTP Server**: Users can set the IP address of an SMTP server.
2. **Port**: Users can set a port for an SMTP server, or select the **Default (25)** option button.
3. **Local Email Address**: Users can set the address of an actual sender. A local email address is composed of 64 characters at most.
4. **SMTP Subject**: Data typed in the **SMTP Subject** box will be put at the front part of the subject of every email. Only English letters, numerical digits, and special marks can be typed in the box. The subject of an email is composed of 16 characters at most.
5. **Account Identification**: If users want to authenticate themselves with a user name and a password before logging in to an SMTP server, they can select the **Yes** option button.
6. **Account Identification Info.**: Users can type a user name and a password which are used for authentication. A user name is composed of 16 characters at most, and a password is also composed of 16 characters at most. If the parameters above are set correctly, and the computer can connect to an SMTP server normally, users can click **Identification** to test whether the user name and the password are correct.

***. The parameters set in HWCONFIG must be downloaded to the CPU module so that they can take effect. (Please refer to section 3.3.6 or section 3.5.4 for more information.)**

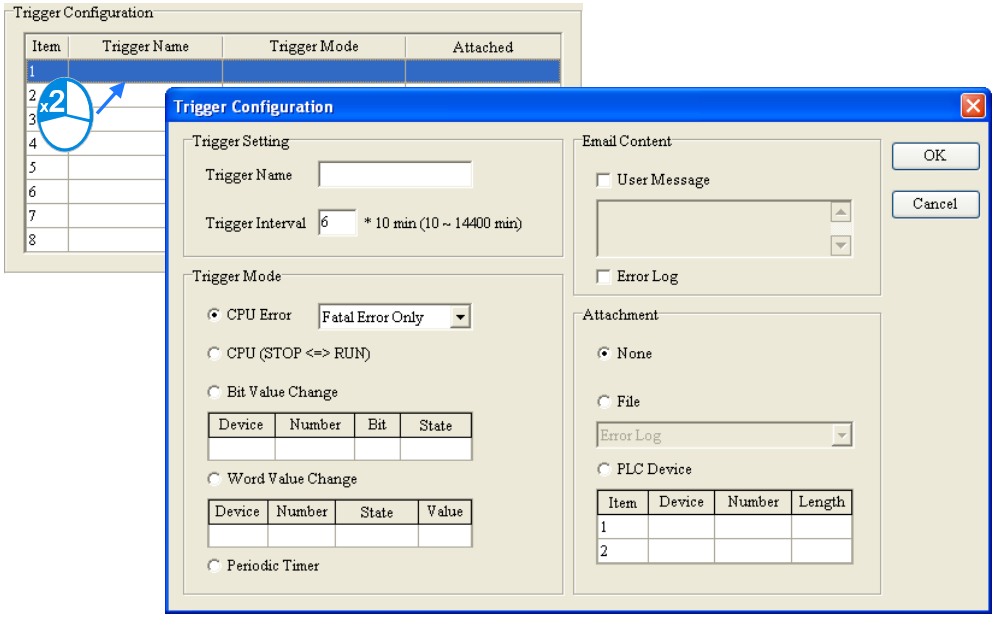
3.3.5.4 Ethernet–Advance: Email Trigger

After users click the **Ethernet–Advance** tab at the top of the window, and the **Email Trigger** tab at the bottom of the window, they can set triggers for the sending of emails. Before setting triggers, users have to click the **Email** tab, and select the **Enable Email Sending Function** checkbox.

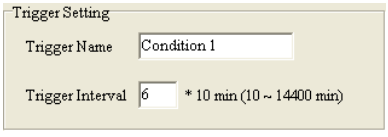
Please refer to section 3.3.5.3 for more information.



Triggers are listed in the **Trigger Configuration** table. Users can set eight triggers at most. If users want to set or view a trigger, they can double-click the trigger in the table. The **Trigger Configuration** window will appear.



(1) Type a trigger name and an interval of triggering the sending of an email in the **Trigger Setting** section.



- **Trigger Name**

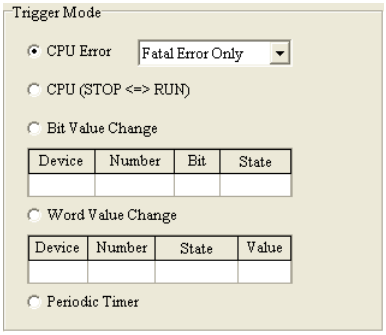
  Users can type a trigger name in the **Trigger Name** box. Only English letters, numerical digits, or special marks can be typed in the box. A trigger name is composed of 16 characters at most. If the sending of an email is due to this trigger, the trigger name will be put in the subject of this email.

- **Trigger Interval**

  Users can set an interval of triggering the sending an email. Ten minutes are taken as a unit. If the condition of triggering the sending of an email is met, the system will send the email

every specific period of time, and the email will not sent within the intervals.

(2) Select a condition of triggering the sending of an email in the **Trigger Mode** section.



- **CPU Error**

  If an error occurs in the CPU module, the condition of triggering the sending of an email is met. Please refer to operation manuals for more information about errors occurring in CPU modules. After users select the **CPU Error** option button, they have to select **Fatal Error Only** or **All Errors** in the drop-down list box at the right side of the option button.

  (a) **Fatal Error Only**: If a fatal error occurs in the CPU module, the condition of triggering the sending of an email is met.

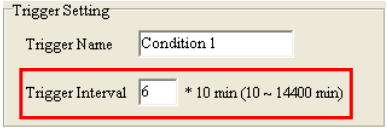
  (b) **All Errors**: If an error occurs, the condition of triggering the sending of an email is met.

- **CPU (STOP<=>RUN)**

  When the CPU module begins to run, or when the CPU module stops running, the condition of triggering the sending of an email is met.

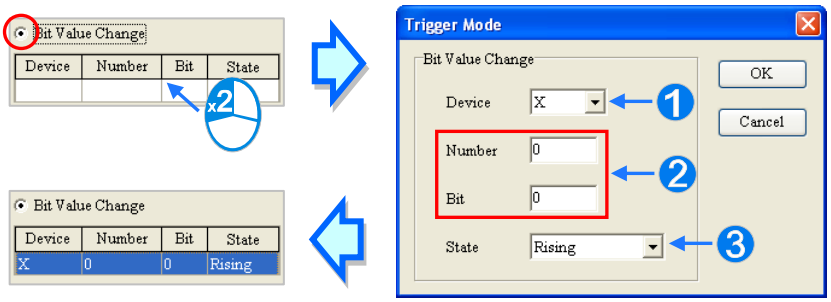
- **Periodic Timer**

  An email is sent periodically. How often an email is sent depends on the interval typed in the **Trigger Interval** box in the **Trigger Setting** section.



- **Bit Value Change**

  If the state of a bit device specified meets a condition set, the sending of an email will be triggered. For example, if X0.0 is turned from OFF to ON, the condition of triggering the sending of an email will be met. If users want to set a condition, they can double-click the table under the option button to open the **Trigger Mode** window.



  ❶ **Device**: Users can select a device type in the **Device** drop-down list box.

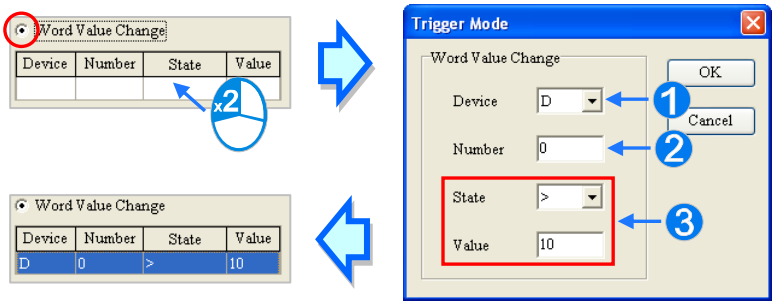
  ❷ **Number & Bit**: Users can type a device address in the **Number** box. If the device type selected is X/Y/D/L, the users have to specify a bit number.

  ❸ **State**: Users can select **Rising** or **Falling** in the **State** drop-down list box.

- **Word Value Change**

  If the value in a device specified meets a condition set, the sending of an email will be triggered. For example, if the value in D0 is larger than 10, the condition of triggering the

sending of an email will be met. If users want to set a condition, they can double-click the table under the option button to open the **Trigger Mode** window.

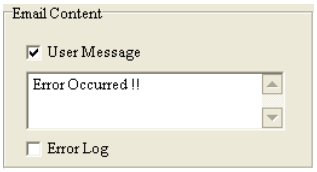


❶ **Device**: Users can select a device type in the **Device** drop-down list box.

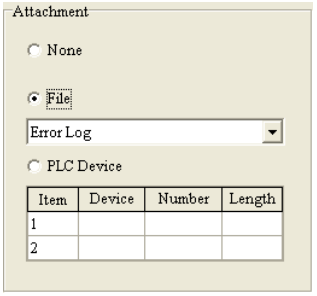
❷ **Number**: Users can type a device address in the **Number** box.

❸ **State & Value**: Users can set s condition of triggering the sending of an email here.

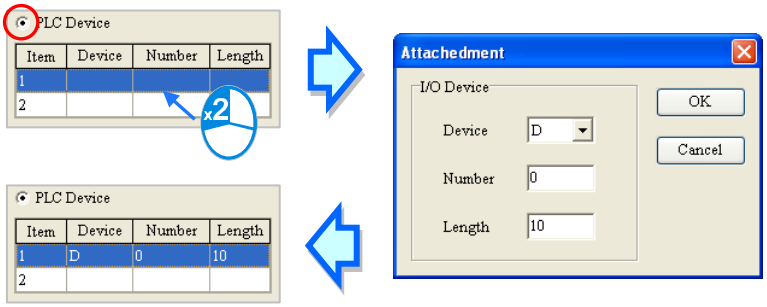
(3) Select the **User Message** box or the **Error Log** box in the **Email Content** section. If a user-define message is the text of an email, the **User Message** box is selected. If an error log in the CPU module is the text of an email, the **Error Log** box is selected. Only English letters, numerical digits, and special marks can be typed in the box under the **User Message** box.



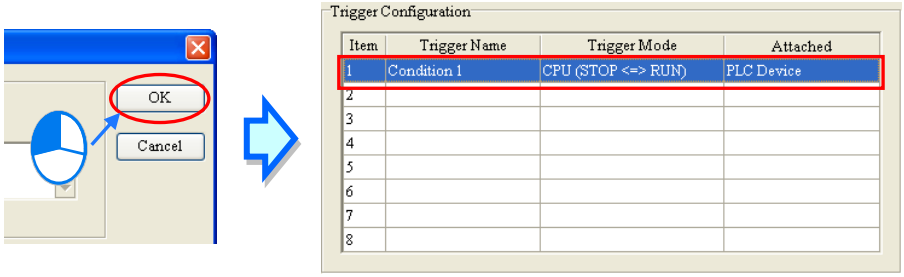
(4) Select the **None** option button, the **File** option button, or the **PLC Device** option button in the **Attachment** section. If users want to set an attachment, they have to make sure of the maximum size of an email which the CPU module allows. Please refer to operation manuals for more information.



- **None**
  If this option button is selected, no attachment will be inserted.
- **File**
  Users can select an error log in the memory card, or the system backup file in the memory card as the attachment of the email.
- **PLC Device**
  If this option button is selected, the system automatically retrieves the states of the devices, or the values in the devices listed in the table as the attachment when the email is sent. After this option button is selected, users can double-click the table to open the **Attachment** window. Two groups of devices at most can be set. For example, if the condition is met, the values in D0~D9 will be sent as an attachment.



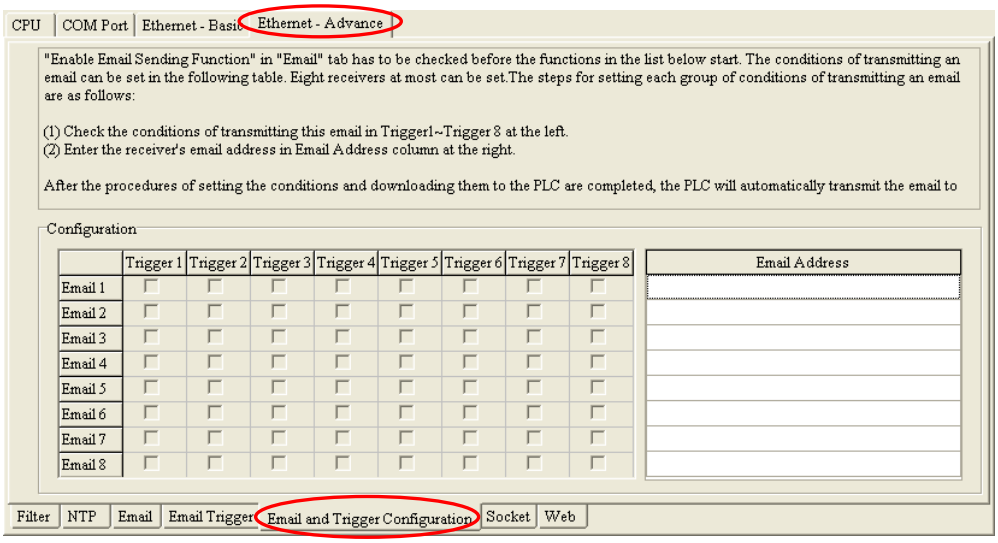
(5) After all the setting is complete, users can click **OK** to create the trigger.



***. The parameters set in HWCONFIG must be downloaded to the CPU module so that they can take effect. (Please refer to section 3.3.6 or section 3.5.4 for more information.)**

3.3.5.5 Ethernet–Advance: Email and Trigger Configuration

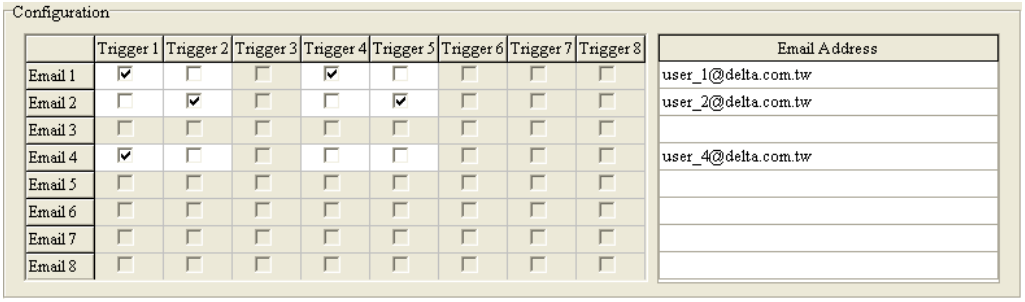
After users click the **Ethernet–Advance** tab at the top of the window, and the **Email and Trigger Configuration** tab at the bottom of the window, they can set the relation between triggers for the sending of emails and receivers. Before setting the function here, the users have to click the **Email** tab, select the **Enable Email Sending Function** checkbox, click the **Email Trigger** tab, and set at least one trigger. Please refer to section 3.3.5.3 and section 3.3.5.4 for more information.



First, type the email addresses of receivers in the **Email Address** column. Eight email addresses at most can be set for the CPU module, and every email address is composed of sixty-four characters at most. Then, select triggers corresponding to receivers. If email addresses or triggers are not set, the corresponding selection positions are gray.

For example, if the condition represented by trigger 1 or the condition represented by trigger 4 is met, the system will automatically send the emails to **user_1@delta.com.tw**. If the condition represented by trigger 2 or the condition represented by trigger 5 is met, the system will automatically send the email to **user_2@delta.com.tw**. If the condition represented by trigger 1 is met, the system will

automatically send the email to **user_1@delta.com.tw**, and **user_4@delta.com.tw**.



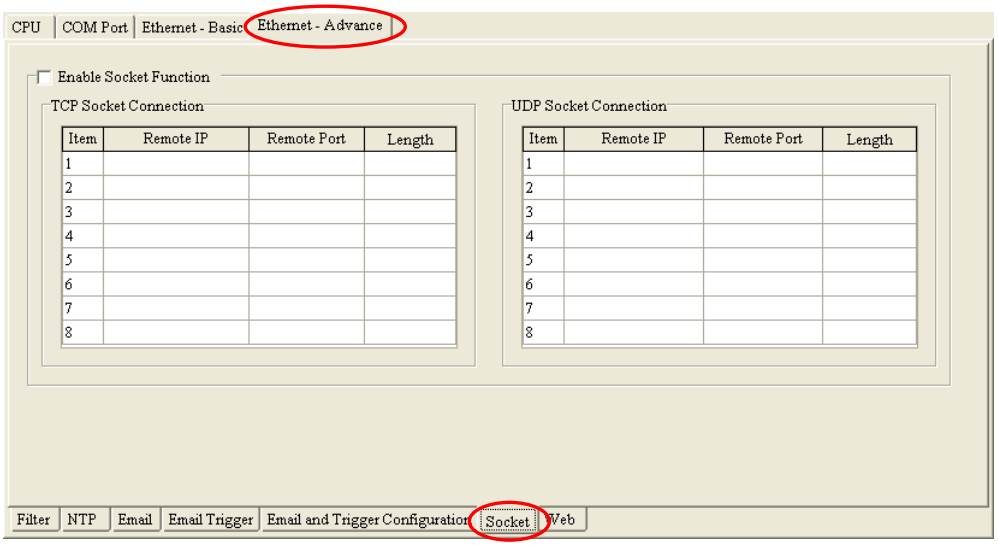
***. The parameters set in HWCONFIG must be downloaded to the CPU module so that they can take effect. (Please refer to section 3.3.6 or section 3.5.4 for more information.)**

3

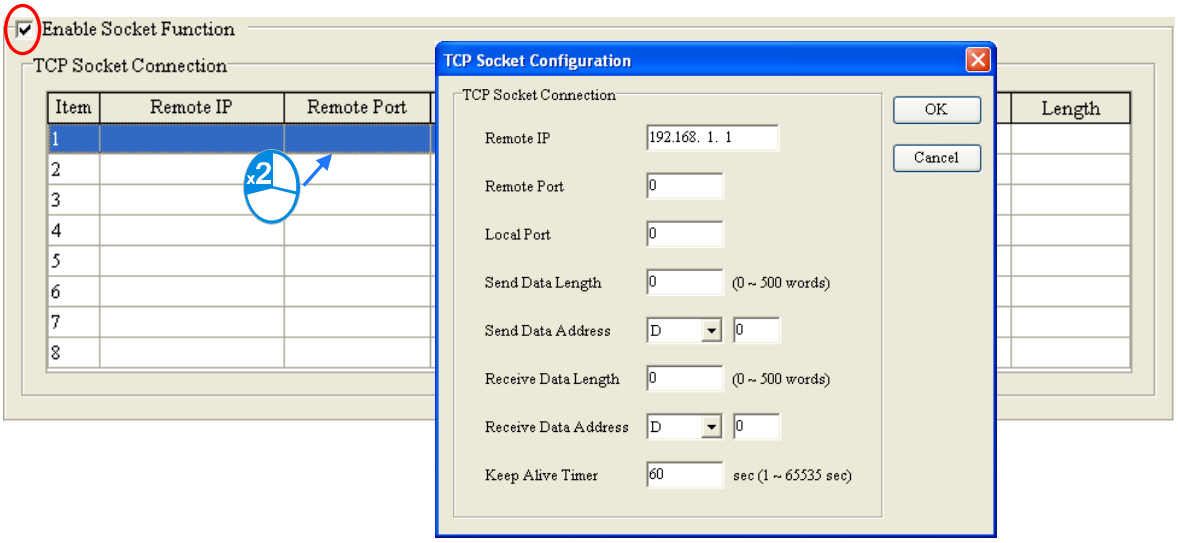
3.3.5.6 Ethernet–Advance: Socket

After users click the **Ethernet–Advance** tab at the top of the window, and the **Socket** tab at the bottom of the window, they can set the parameters for sockets through which data is transmitted. This function has to be used with specific applied instructions. Please refer to AH500 Programming Manual for more information.

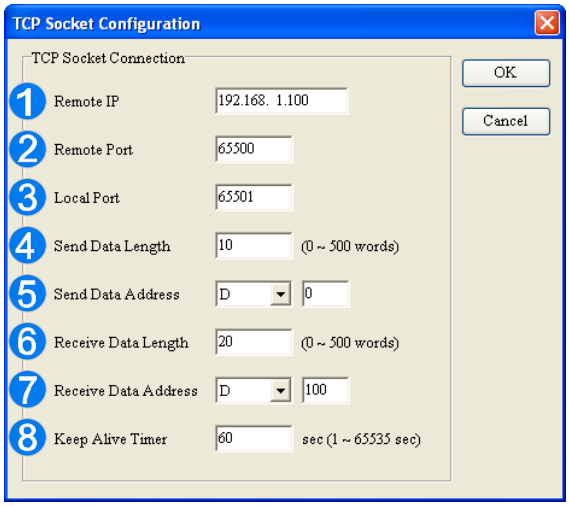
Sockets will not be introduced here. Please refer to related documents or manuals for more information about sockets.



An AH500 system supports the transmission of data between the CPU module and other CPU modules or devices by means of the creation of sockets, and supports TCP and UDP. Users can set eight TCP connections and eight UDP connections. If users want to set a TCP connection, they have to select the **Enable Socket Function** checkbox, and double-click an item in the **TCP Socket Connection** table to open the **TCP Socket Configuration** window. If the users want to set a UDP connection, they have to select the **Enable Socket Function** checkbox, and double-click an item in the **UDP Socket Connection** table to open the **UD Socket Configuration** window.



The parameters in the **TCP Socket Configuration** window are the same as the parameters in the **UDP Socket Configuration** window except that there is no **Keep Alive Timer** parameter in the UDP Socket Configuration window. The parameters in the **TCP Socket Configuration** window are described below.



1. Users can set a remote IP address.
2. Users can set a communication port used by the remote device for this TCP connection. The port number must be within the range between 0 and 65535.
3. Users can set a communication port used by the local CPU module for this connection. The port number must be within the range between 0 and 65535.
4. Users can set the length of data which will be sent by the local CPU module. The length must be within the range between 0 words and 500 words.
5. Uses can set an initial device in the CPU module where data which will be sent is stored.
6. Users can set the length of data which will be received by the local CPU module. The length must be within the range between 0 words and 500 words.
7. Uses can set an initial device in the CPU module where data which will be received is stored.
8. Users can set a maximum keepalive time for the connection. If no data is transmitted, and the keepalive period has elapsed, the CPU module will terminate the connection automatically.

The port number used by the local CPU module and the port number used by the remote device can not be the same, and the devices where data which will be sent is stored can not overlap the

devices where data which will be received is stored. If the IP address of the remote device is 192.168.1.100, the port number used by the remote device is 65500, and the port number used by the local CPU module is 65501, the remote device and the local CPU module can transmit data through this TCP connection.

If the local CPU module wants to send 10-word data to the remote device, the data will be stored in D0~D9 before the data is sent. If the local CPU module receives 20-word data from the remote device, the data will be stored in D100~D119.

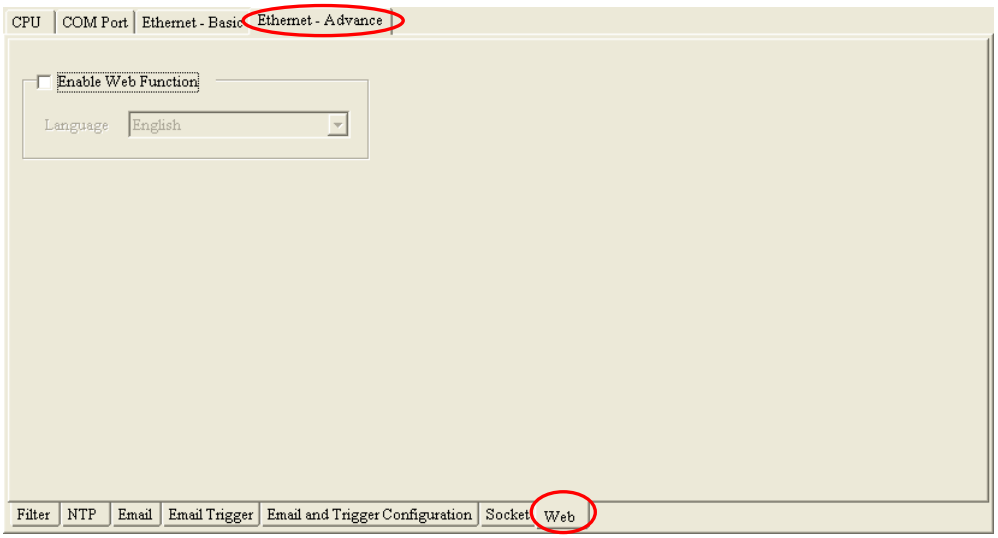
If the length of data received is larger than the length set, the first 20-word data will be stored in D100~D119, and the data after the first 20-word data will be discarded. Likewise, if the length of data received is less than the length set, the data will be stored in the devices starting from D100, and the values in devices where no new data is stored will be retained.

If no data is transmitted, and 60 seconds have elapsed, the CPU module will close the socket, and terminate the connection.

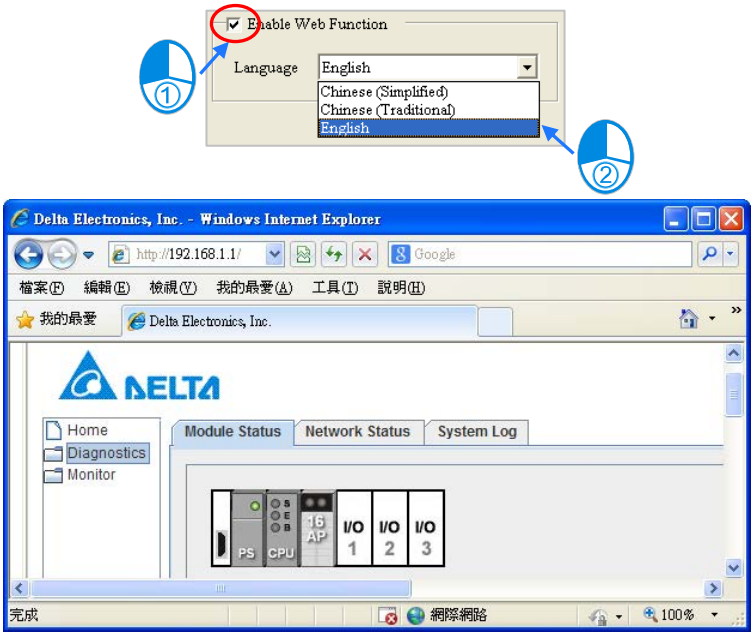
***. The parameters set in HWCONFIG must be downloaded to the CPU module so that they can take effect. (Please refer to section 3.3.6 or section 3.5.4 for more information.)**

3.3.5.7 Ethernet–Advance: Web

After users click the **Ethernet–Advance** tab at the top of the window, and the **Web** tab at the bottom of the window, they can enable the built-in web monitoring function of the CPU module.



Select the **Enable Web Function** checkbox, and the select a language which will be adopted in the drop-down list box. After the parameter is downloaded to the CPU module correctly, users can view the built-in web pages in the CPU module by means of a web browser.

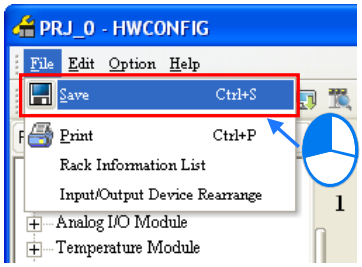


***1. The parameters set in HWCONFIG must be downloaded to the CPU module so that they can take effect. (Please refer to section 3.3.6 or section 3.5.4 for more information.)**

***2. Please refer to operation manuals for more information about the web monitoring function.**

3.3.6 Saving and Downloading / Uploading the PLC Parameters

After the setting of the PLC parameters is complete, users can click **OK** to apply the parameters. However, the parameters are still not saved as a file. If users want to save the parameters, they have to click **Save** on the **File** menu or on the toolbar after the parameters are applied. After the saving of the parameters is complete, the parameters will be saved as a para file whose primary filename is the project name in the folder in which the project (*.isp) is saved.

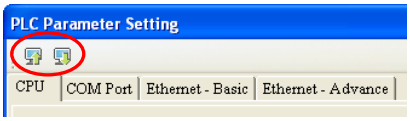


OR

There are two ways to download or upload the PLC parameters. Users can download/upload the PLC parameters through the functions on the main screen of HWCONFIG. The hardware configuration and the parameters in the modules will also be downloaded/uploaded. Users can also download/upload the PLC parameters through the functions in the **PLC Parameter Setting** window. Only the PLC parameters will downloaded/uploaded.

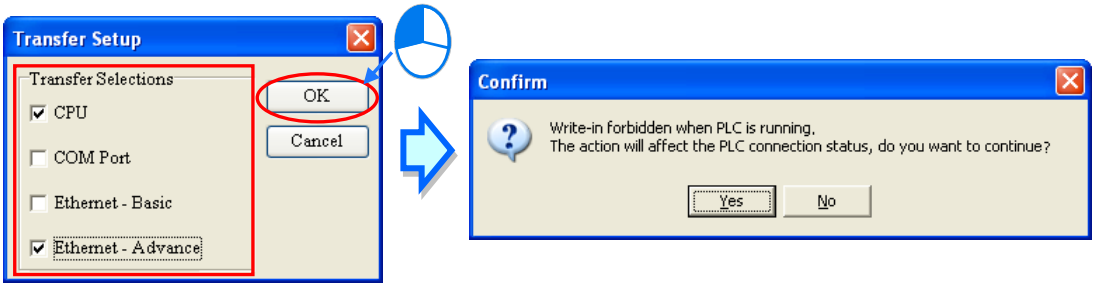
The downloading/uploading of the PLC parameters through the functions on the main screen of HWCONFIG will be introduced in section 3.5.4. The downloading/uploading of the PLC parameters through the functions in the **PLC Parameter Setting** window is introduced here.

The PLC parameters will be downloaded after in the upper left corner of the **PLC Parameter Setting** window is clicked, and the PLC parameters will be uploaded after is clicked.



After is clicked, all the PLC parameters will be uploaded. After is clicked, the **Transfer Setup** window will appear. Users can select parameter types which will be downloaded in the window. Not all parameter types need to be downloaded to the CPU module.

The **Transfer Setup** window is shown below. If the **CPU** checkbox or the **Ethernet–Advance** are selected, the CPU module must stop running during the downloading of the parameters. As a result, if the CPU module does not stop running, a dialog box will appear before the parameters are downloaded.



***1. If the name of the CPU module is different from the name attached to the model in the project management area, a dialog box will appear before the parameters are downloaded or uploaded. Please refer to section 3.3.2.1 for more information.**

***2. HWCONFIG adopts the communication setting in ISPSoft. Please make sure that ISPSoft is connected to the CPU module normally before the parameters are downloaded/uploaded. Please refer to section 2.4 for more information.**

3.4 Setting the Parameters in an AH500 Series Module

3.4.1 Managing the Version of a Module

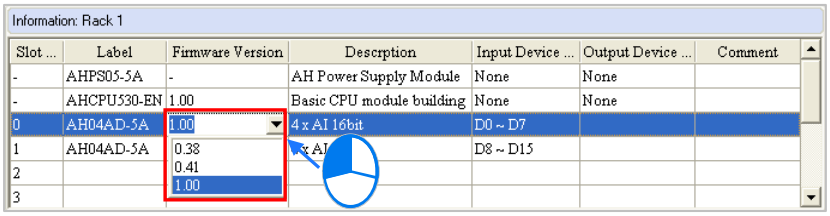
The functions of a module or the parameters in the module vary with the versions of the module. In HWCONFIG, the functions of the modules and the parameters in the modules are defined in a document called **MDS**. When ISPSOft is installed, the latest MDS document is also installed. If a new version of the MDS document is released, users can download it or get the related file from an agent.

An MDS document provides the information about the firmware versions of the modules. If users want to configure modules in HWCONFIG, they can select a suitable configuration according to the firmware versions of the actual modules. Besides, the parameters which can be set in the modules vary with the firmware versions of the modules which are set. If the firmware version of a module set in HWCONFIG is older than the firmware version of the actual module, the module will operate normally after the parameters are downloaded. If the firmware version of a module set in HWCONFIG is newer than the firmware version of the actual module, the module will not operate normally after the parameters are downloaded. Please refer to the descriptions in the table below for more information.

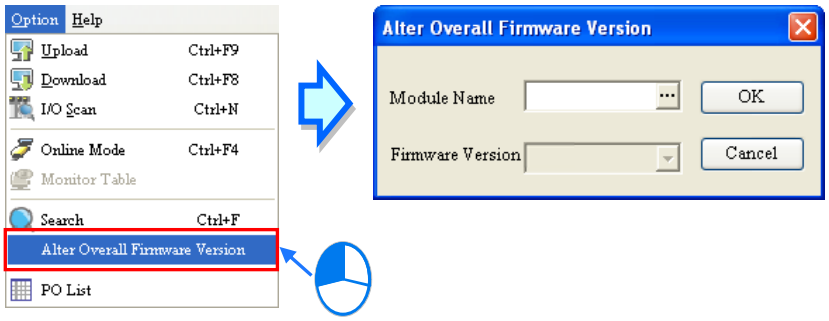
Difference	Compatibility
The firmware version set is older than the actual firmware version.	The module operates normally. Functions or parameters not defined in HWCONFIG are assigned the default values.
The firmware version set is newer than the actual firmware version.	The module can not operate, and is in an erroneous condition. Users have to select a firmware version which is the same as the firmware version of the actual module in HWCONFIG, check the parameters, and download the parameters again.

***. The compatibility mechanism described above only applies to modules whose firmware version is 1.0 or above. A firmware version below 1.0 is not compatible with firmware version 1.0 or above.**

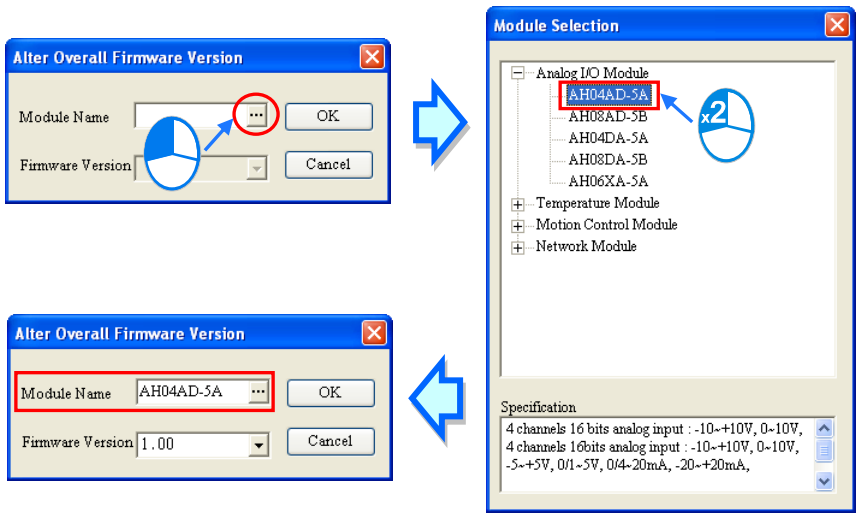
If users want to set the firmware version of a module, they can select a version number in the **Firmware Version** drop-down list cell on the information list. If the firmware version of a module is unknown, no version number will be displayed.



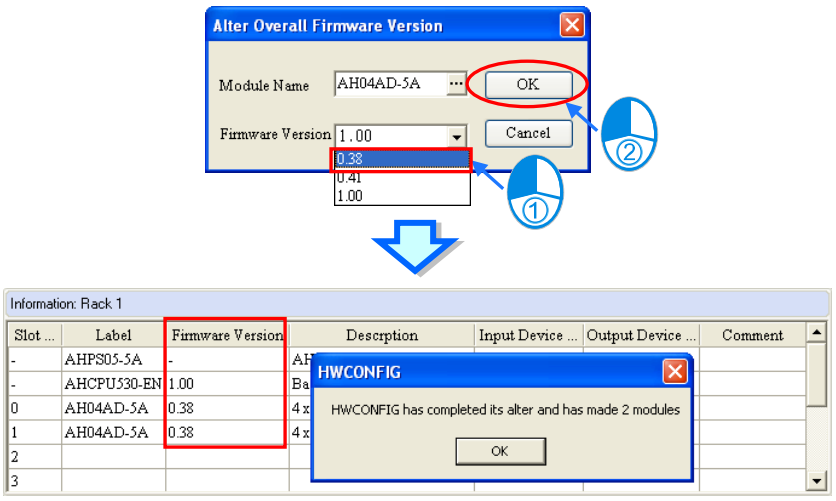
If users want to change the firmware versions of the modules of a certain model, they can click **Alter Overall Firmware Version** on the **Option** menu. The **Alter Overall Firmware Version** window will appear.



Click ··· in the **Module Name** box, select a module model in the **Module Selection** window, and double-click the module model.



Select a version number in the **Firmware Version** drop-down list cell. After the setting is complete, users can click **OK** to change the firmware versions of the modules specified.

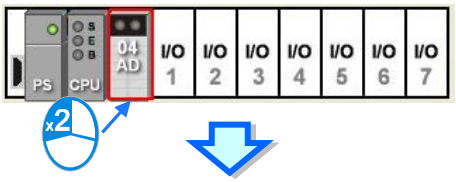


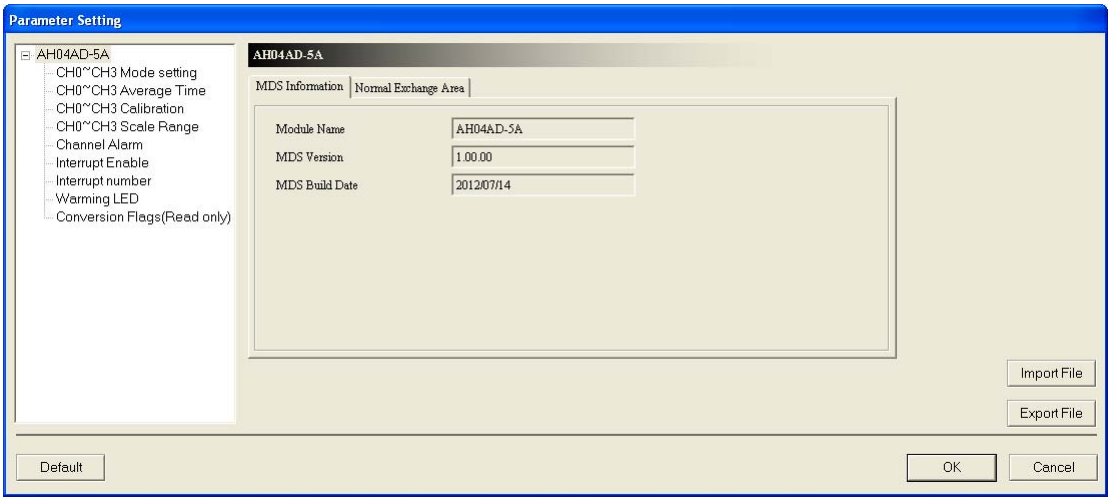
Additional remark

Users can know the firmware version of an actual module by clicking **I/O Scan** on the **Option** menu. If ISPSfot is connected to the CPU module, users can get the information about the firmware version of an actual module in the **Module Information** window. If users want to know the firmware version of an actual module by click **I/O Scan** on the **Option** menu, or view the information about the firmware version of an actual module in the **Module Information** window, ISPSoft must be connected to the CPU module normally, and the installation of modules must be completed. Please refer to section 3.5.5 and section 3.5.6 for more information.

3.4.2 Setting the Parameters in a Module

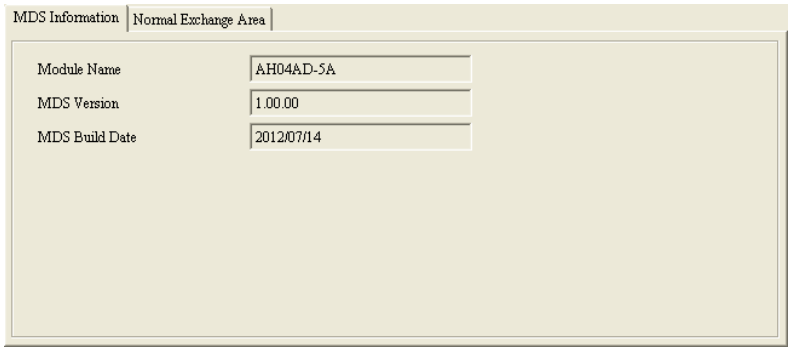
Users can set the parameters in every module configured in HWCONFIG. The functions and characters of every module depend on the setting of the parameters. If users want to set the parameters in a module, they can double-click the module in the system configuration area. The system will automatically open the **Parameter Setting** window.



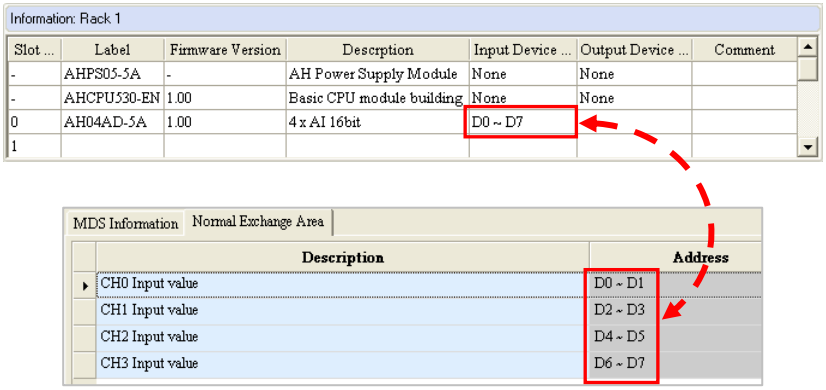


The parameters are set according to the functions supported by the module. Before users set the parameters, they have to refer to AH500 Module Manual for more information. To prevent damage to the system or staff, the users have to make sure of the effect that the parameters which are set have on the module and the whole system.

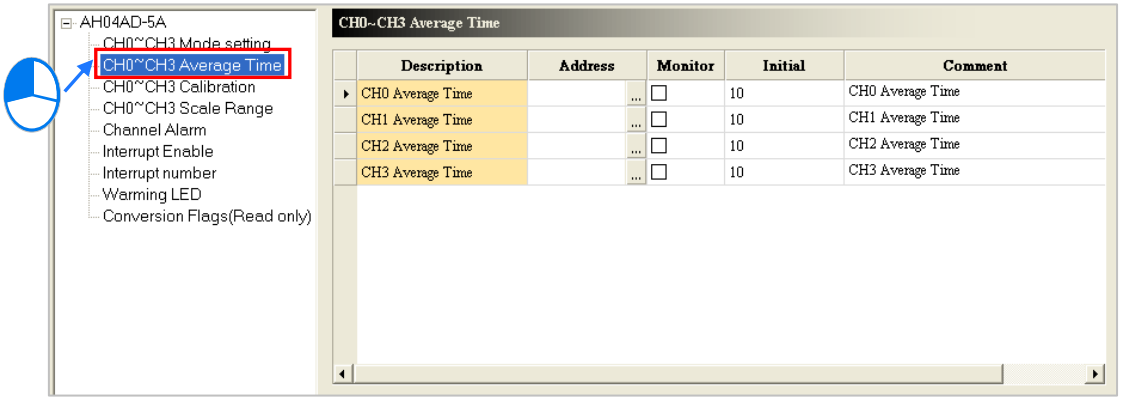
There are two tabs in the window. They are the **MDS Information** tab and the **Normal Exchange Area** tab. If users click the **MDS Information** tab, the version of the MDS document will be displayed. The number of parameters which can be set in the module and the range of setting values depend on the version of the MDS document. Besides, if the parameters in HWCONFIG in a project are created by a newer MDS document, the system asks users to update the MDS document when the users open the **HWCONFIG** window in the project.



If users click the **Normal Exchange Area** tab, the functions to which the input/output devices assigned to the module correspond will be displayed.



Users can select a parameter type at the left side of the window, and then set the parameters in the table at the right side of the window. The formats of the tables for the parameter types are the same, but the contents of the tables are different. Besides, the number of parameters which can be set and the range of setting values depend on the firmware version of the module which is set.



The description of the columns in a table is as follows. If the color of a table cell is gray, the cell can not be edited.

- **Description**

  The parameter names or descriptions are shown in this column.

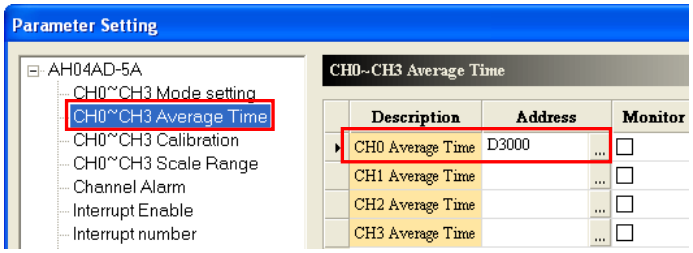
Description	Address	Monitor	Initial	Comment
CH0 Average Time			10	CH0 Average Time
CH1 Average Time			10	CH1 Average Time
CH2 Average Time			10	CH2 Average Time
CH3 Average Time			10	CH3 Average Time

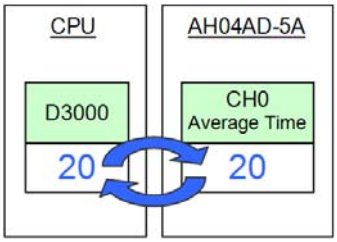
- **Address**

Users can set data registers in this column. The parameters in the module which do not need to be updated constantly are stored in the data registers. If users specify data registers for the parameters in the module, and the parameters in HWCONFIG are downloaded to the CPU module correctly, the data registers in the CPU module are synchronized to the parameters in the module when the system operates. As a result, the users can access the module through the data registers. Accessing a module through the data registers is more efficient than accessing the module through the instruction FROM/TO. Besides, the number of data registers that a module has is limited. When users set data registers, they have to refer to the instructions for the module.

As the example below shows, D3000 corresponds to the parameter **CH0 Average Time** in the **Parameter Setting** window for AH04AD-5A. After the parameters in HWCONFIG are downloaded to the CPU module, users can change the value of the parameter **CH0 Average Time** by changing the value in D3000 in the CPU module.

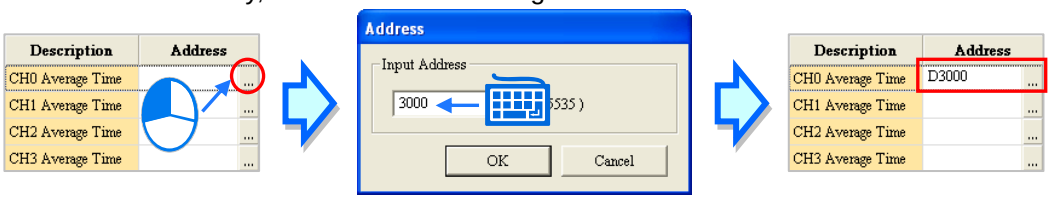
Besides, if users write a value which is not allowed by a parameter into a data register during the operation of the system, the system will restore the value in the data register and the value of the parameter to the original values.





***. Please refer to section 3.4 for more information about the setting of parameters in a module. Please refer to programming manuals for more information about the instruction FROM/TO.**

If users want to set a data register, they can click ... in the **Address** cell for a parameter, and type a data register address which will be used in the **Address** window. Please notice that a data register address which has been assigned to the module can not be used. If users want to delete the data register address in the **Address** cell for a parameter, they can open the **Address** window in the same way, and delete the data register address in the window.



If the color of the **Address** cell for a parameter is gray, users can not set a data register for the parameter. In the example below, the values of these parameters are binary values. The states of certain bits in the data register set in the **Address** cell for the first parameter correspond to these parameters. Please refer to the documents for the modules for more information about the relation between parameters and the states of the bits in data registers.

Description	Address	Monitor	Initial	Com
Warn of CH0 Input value Out of Physical Range			flashing	Warn of CH0 Input v
Warn of CH1 Input value Out of Physical Range			flashing	Warn of CH1 Input v
Warn of CH2 Input value Out of Physical Range			flashing	Warn of CH2 Input v
Warn of CH3 Input value Out of Physical Range			flashing	Warn of CH3 Input v

- **Monitor**

After users click **Monitor Table** on the **Option** menu in HWCONFIG, they can access a module through the data registers in the **Monitor Table** window. As a result, if a data register is set for a

parameter in a module, and the checkbox in the **Monitor** cell for the parameter is selected, the data register specified will be displayed in the **Monitor Table** window after the **Monitor Table** window is opened. The checkbox in the **Monitor** cell for a parameter to which no data register is assigned can not be selected.

In the example below, D3000~D3003 are assigned to the parameters, but only D3000 and D3001 are monitored. As a result, only D3000 and D3001 will be displayed in the **Monitor Table** window after the **Monitor Table** window is opened. Please refer to section 3.5.6 for more information about **Monitor Table** in HWCONFIG.

	Description	Address	Monitor	Initial	Comment
	CH0 Average Time	D3000 ...	☑	20	CH0 Average Time
▸	CH1 Average Time	D3001 ...	☑	10	CH1 Average Time
	CH2 Average Time	D3002 ...	☐	10	CH2 Average Time
	CH3 Average Time	D3003 ...	☐	10	CH3 Average Time

- **Initial**

  Users can set initial values in this column. When the parameters in HWCONFIG are downloaded, the initial values set are written into the module. Users may need to type initial values in the **Initial** columns for some of the parameter types, and select initial values in the **Initial** columns for the other parameter types. Please refer to the instructions for a module for more information about the range of values which can be set, and the processing mechanism after the downloading of the initial values.

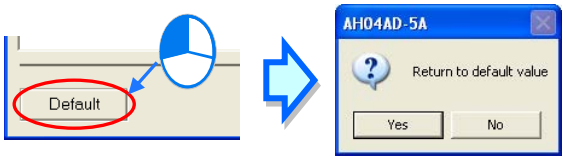
	Description	Address	Monitor	Initial	Comment
	CH0 Average Time	D3000 ...	☑	20	CH0 Average Time
▸	CH1 Average Time	D3001 ...	☑	10	CH1 Average Time
	CH2 Average Time	D3002 ...	☐	10	CH2 Average Time
	CH3 Average Time	D3003 ...	☐	10	CH3 Average Time

- **Comment**

  After users click the **Comment** cell for a parameter, they can type a comment on the parameter. A comment is composed of 32 characters at most, and a Chinese character occupies two characters. The default contents of the **Comment** column for a parameter type are usually the same as the contents of the **Description** column for the parameter type.

	Description	Address	Monitor	Initial	Comment
▸	CH0 Average Time	D3000 ...	☑	20	Comment...
	CH1 Average Time	D3001 ...	☑	10	CH1 Average Time
	CH2 Average Time	D3002 ...	☐	10	CH2 Average Time
	CH3 Average Time	D3003 ...	☐	10	CH3 Average Time

After **Default** in the lower left corner of the **Parameter Setting** window is clicked, the values of the parameters in the module will be restored to the default values.

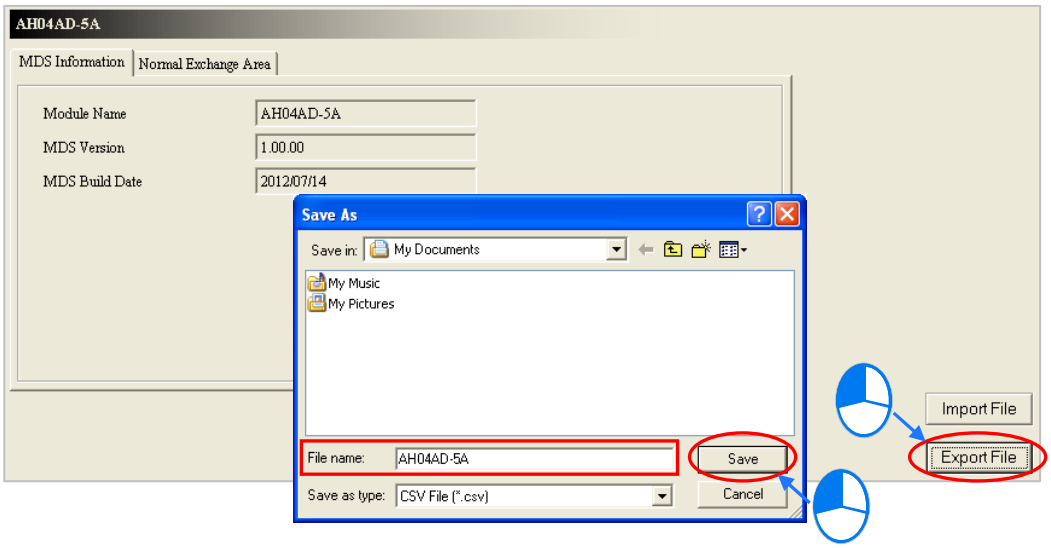


3.4.3 Exporting and Importing the Parameters in a Module

After **Export File** in the **Parameter Setting** window is clicked, the parameters in the module will be saved as a CSV file. A file which was exported previously can also be imported.

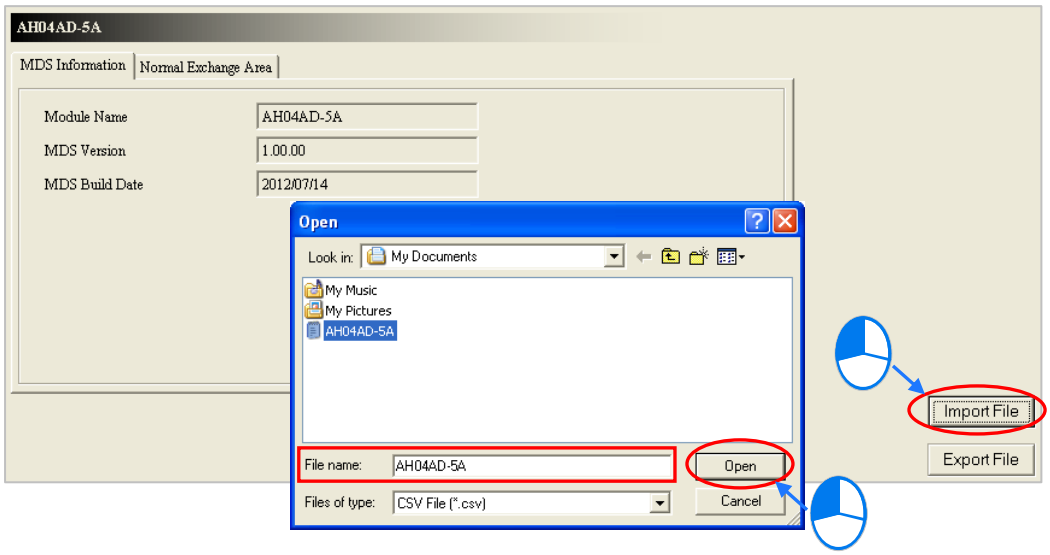
- **Exporting the parameters in a module**

  Click **Export File** in the **Parameter Setting** window, select a folder in the **Save in** drop-down list box in the **Save As** window, type a filename in the **File name** box, and click **Save**.



- **Importing the parameters in a module**

  Click **Import File** in the **Parameter Setting** window, select a file which will be imported in the **Save As** window, and click **Open**.



  Before the parameters are imported, the system will check the file format and the module model. If the file format or the module model is not correct, the system will forbid the import of the parameters. If the data registers specified in the original file exported have been used in the present configuration, different data registers will be specified in the file after the file is imported.

3.4.4 Setting the Parameters in an Intelligent Module

Delta Electronics, Inc. provides the exclusive configuration tools for some modules. Through the software, the modules can be configured further, and the parameters in the modules can be set further. Some of the configuration tools can be opened through HWCONFIG. The configuration tools supported by HWCONFIG are listed in the table below. The table suggests that the configuration tools for modules which are not listed in the table can not be opened through HWCONFIG. It does not suggest that there are no configuration tools for modules which are not listed in the table.

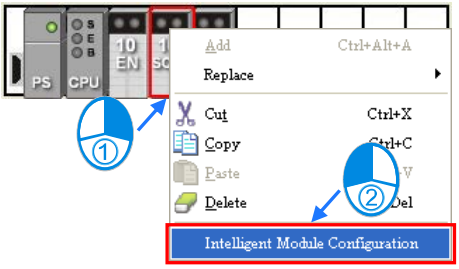
Module type	Module model	Software supported and its version
Network module	AH10EN-5A	The configuration tool is in HWCONFIG, and does not need to be installed.
	AH10SCM-5A	DCISoft version 1.08 or above needs to be installed.

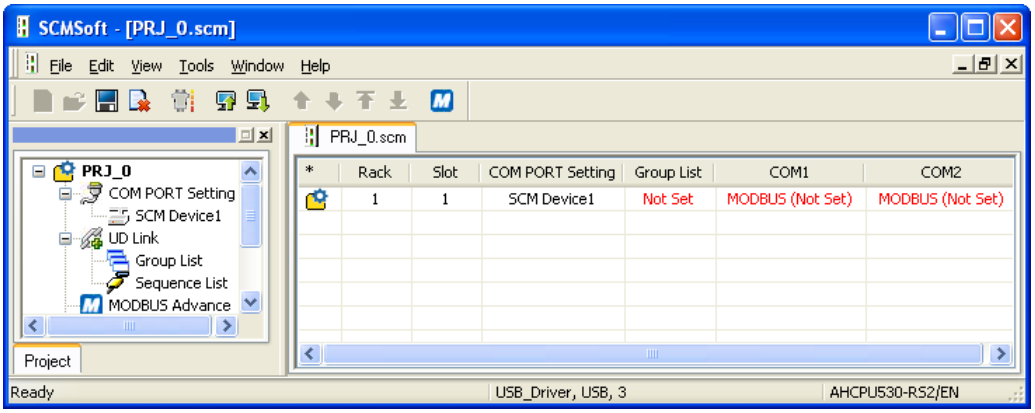
***1. DCISoft version 1.08 or above can be opened through HWCONFIG. Users can select a version of DCISoft according to the specifications for AH10SCM-5A and their requirements.**

***2. The configuration tool for AH10SCM-5A is SCMSoft. After DCISoft version 1.08 or above is installed, SCMSoft can be used.**

If users want to set the parameters in an intelligent module, they can right-click the module in the system configuration area, and click **Intelligent Module Configuration** on the context menu. If the configuration and the setting of the parameters in HWCONFIG have not been saved, the system will ask the users to save the configuration and the setting before the corresponding software is started. If the corresponding software is started, the users can not perform any editing tasks in HWCONFIG before the software is closed.

Please refer to instructions for software and modules for more information about the usage of the software.



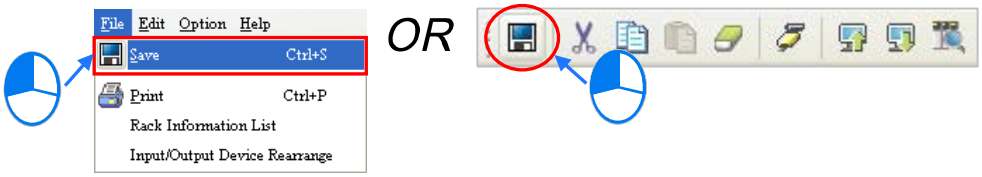


3.5 Management of the Parameters in AH500 Series Hardware and Online Diagnosis

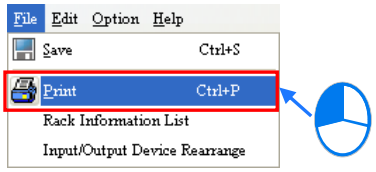
3.5.1 Saving and Printing a Hardware Configuration

After a hardware configuration is complete, the setting in HWCONFIG can be saved with the whole project. The contents which are saved include the hardware configuration, the parameters in the modules, and the parameters in the CPU module, but do not include the parameters in the intelligent modules. The parameters in the CPU module are saved as a para file whose primary filename is the project name. The hardware configuration and the parameters in the modules are saved as an hw file whose primary filename is the project name. The files are saved in a folder in which the project (*.isp) is saved.

If users want to save the setting in HWCONFIG, they can click **Save** on the **File** menu, or on the toolbar.

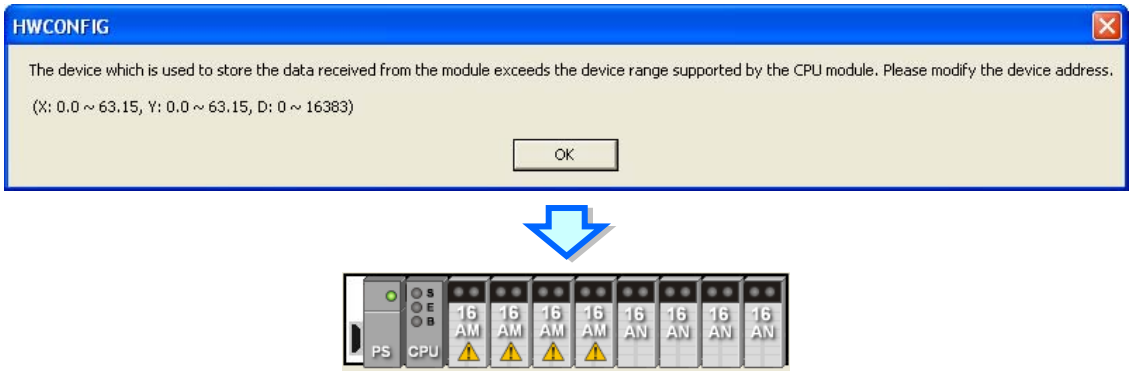


If users want to perform a printing task, they click **Print** on the **File** menu to open the **Print Preview** window. The contents which are printed only include the configuration in the configuration area. Please refer to appendix C for more information about the **Print Preview** window.



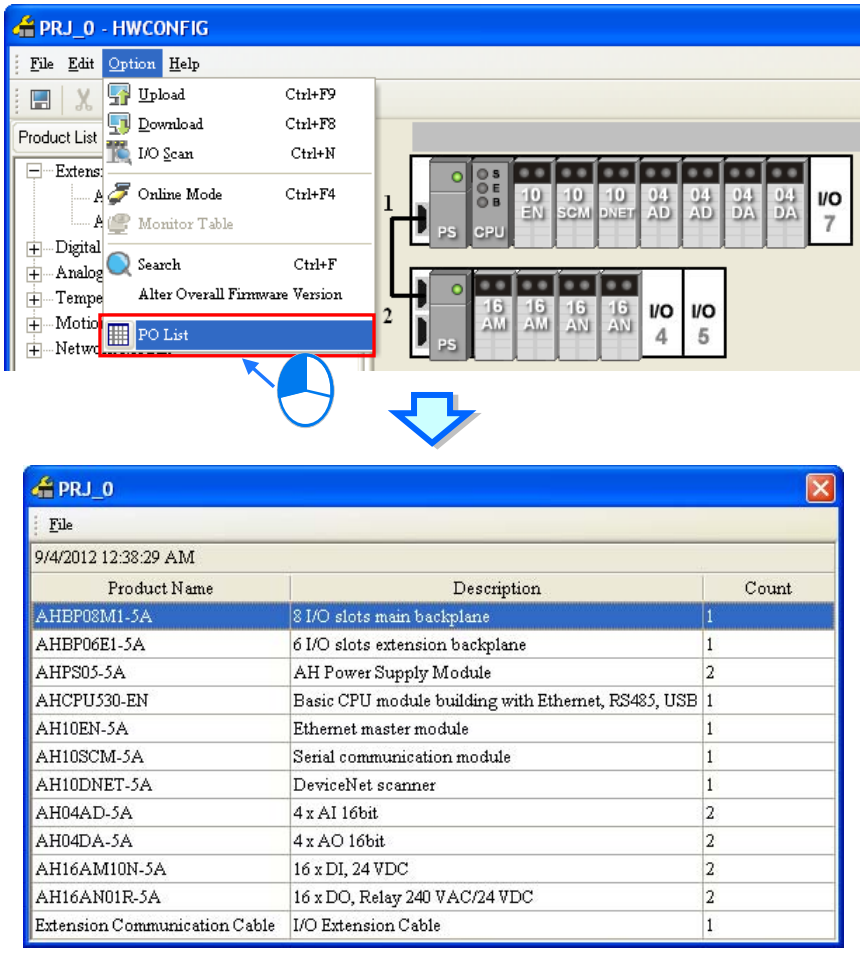
Additional remark

If the CPU module is changed in ISPSoft after the setting of the parameters in the hardware is complete, and the new CPU module can not completely support the range of devices set for the original hardware, a warning message appears when the **HWCONFIG** window is opened again. The problematic modules are marked with triangular error signs in HWCONFIG. These modules are in an abnormal condition. The **Download** function and the **Online Mode** function can not be enabled. After the incorrect device addresses are modified, users can proceed with the operation. Besides, the triangular error signs will disappear after the incorrect devices addresses are modified.

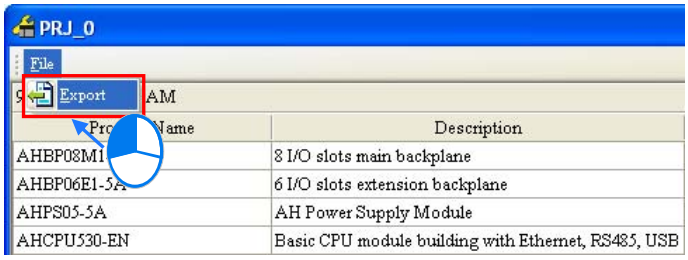


3.5.2 Purchase Order

After a hardware configuration is complete, users can click **PO List** on the **Option** menu. The system will collect the hardware which are configured, and list it in a list.

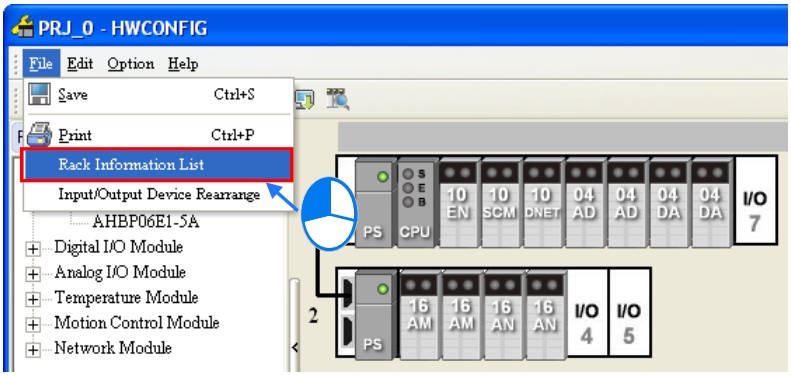


After **Export** on the **File** menu in the upper left corner of the window is clicked, the list will be saved as a CSV file. The data which is exported can be edited or managed through Microsoft Excel.



3.5.3 Rack Information List

After a hardware configuration is complete, users can click **Rack Information List** on the **File** menu. The users can view the configuration in the **Rack Information List** window. The modules in the window are sorted according to the backplanes on which they are installed, and arranged in a hierarchical tree structure. The users can fold or unfold a rack section to increase the convenience of viewing the configuration



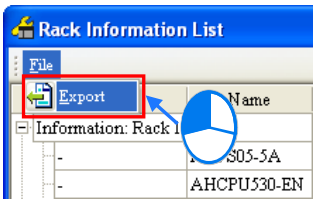
3

Rack Information List

File

Slot No.	Name		Description	Input Device Range	Output Device Range	Comment
Information: Rack 1						
-	AHPS05-5A	-	AH Power Supply Module	None	None	
-	AHCPU530-EN	1.00	Basic CPU module building with	None	None	
0	AH10EN-5A	1.00	Ethernet master module	D0 ~ D19	D20 ~ D39	
1	AH10SCM-5A	1.00	Serial communication module	D40 ~ D57		
2	AH10DNET-5A	1.00	DeviceNet scanner	None	None	
3	AH04AD-5A	1.00	4 x AI 16bit	D58 ~ D65		
4	AH04AD-5A	1.00	4 x AI 16bit	D66 ~ D73		
5	AH04DA-5A	1.00	4 x AO 16bit		D74 ~ D81	
6	AH04DA-5A	1.00	4 x AO 16bit		D82 ~ D89	
7						
Information: Rack 2						
-	AHPS05-5A	-	AH Power Supply Module	None	None	
0	AH16AM10N-5	-	16 x DI, 24 VDC	X0.0 ~ X0.15		
1	AH16AM10N-5	-	16 x DI, 24 VDC	X1.0 ~ X1.15		
2	AH16AN01R-5	-	16 x DO, Relay 240 VAC/24 VDC		Y0.0 ~ Y0.15	
3	AH16AN01R-5	-	16 x DO, Relay 240 VAC/24 VDC		Y1.0 ~ Y1.15	
4						
5						

After **Export** on the **File** menu in the upper left corner of the window is clicked, the list will be saved as a CSV file. The data which is exported can be edited or managed through Microsoft Excel.



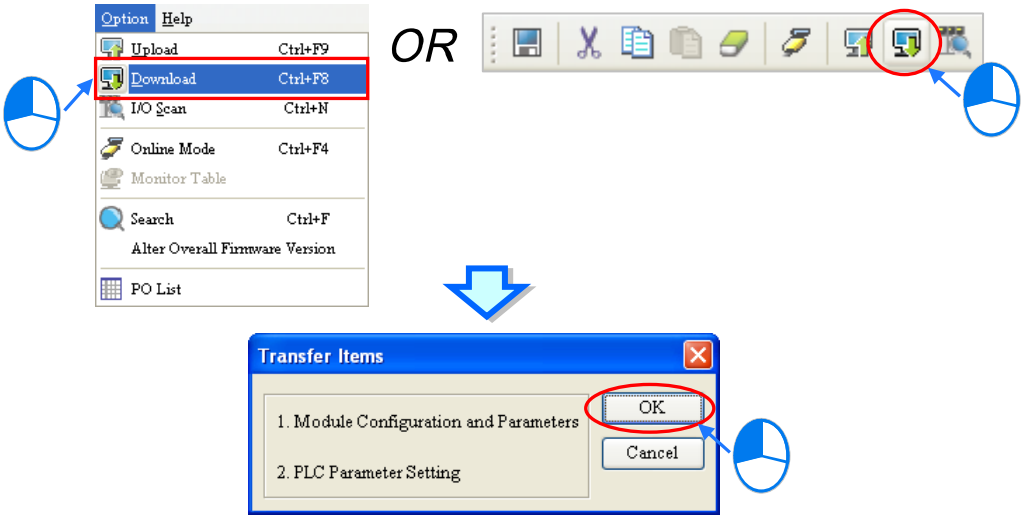
3.5.4 Downloading / Uploading the System Parameters

The parameters set in HWCONFIG must be downloaded to the CPU module so that they can take effect. Owing to the fact that HWCONFIG adopts the communication setting in ISPSoft, users have to make sure that ISPSoft is connected to the CPU module normally before the parameters are downloaded/uploaded. Please refer to section 2.4 for more information about communication setting. If users want to download/upload the program, but the name of the CPU module is different from the name attached to the model in the project management area, the system will remind the users to check the name of the CPU module and the name attached to the model in the project management area. Please refer to section 3.3.2.1 for more information.

- **Downloading the system parameters**

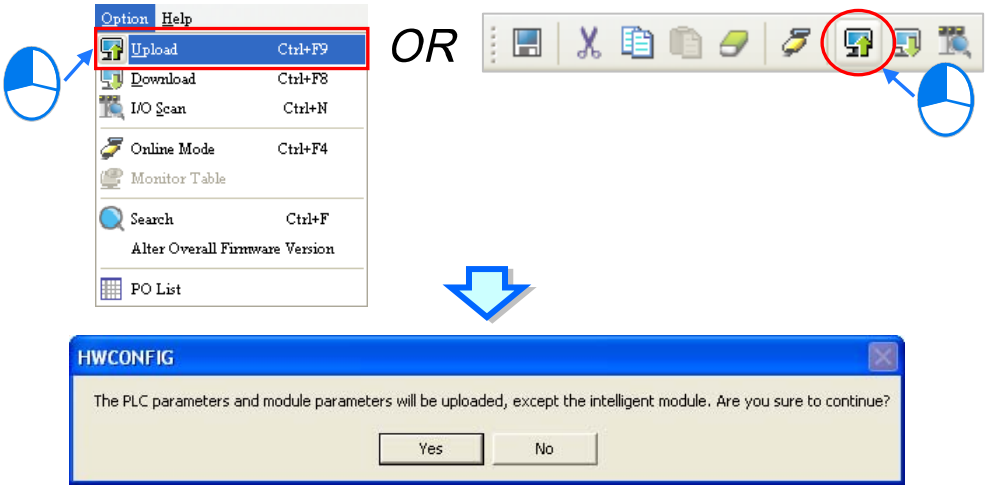
  After users click **Download** on the **Option** menu, or on the toolbar, the **Transfer Items**

window will appear. The items which will be downloaded include the hardware configuration, the parameters in the modules, and the parameters in the CPU module, but do not include the parameters in the intelligent modules. After the users click **OK**, the parameters will be downloaded to the CPU module.



- **Uploading the system parameters**

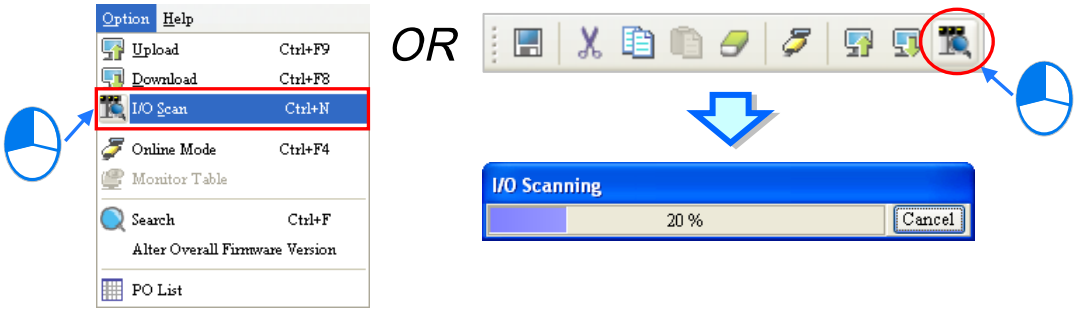
  After users click **Upload** on the **Option** menu, or on the toolbar, the system parameters in the CPU module will be uploaded to HWCONFIG. The items which will be uploaded include the hardware configuration, the parameters in the modules, and the parameters in the CPU module, but do not include the parameters in the intelligent modules.



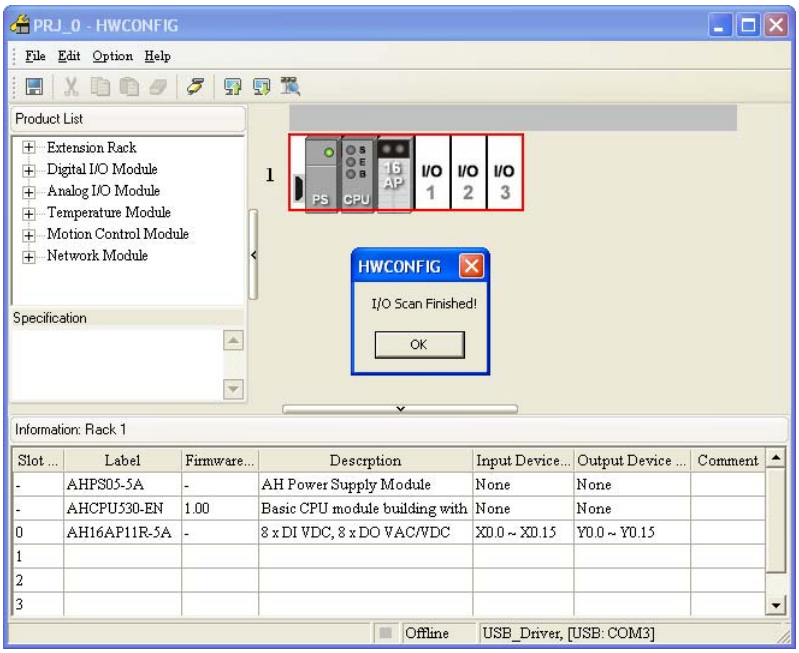
3.5.5 I/O Scan

Users can configure modules in the way mentioned above. If users have gotten related hardware, they can install the hardware, click **I/O Scan** in HWCONFIG, and scan the actual hardware configuration through communication. The time of configuring the modules will be saved. Users have to make sure that ISPSoft has connected to the CPU module normally before they carry out the operation. Please refer to section 2.4 for more information about communication setting.

Users can click **I/O Scan** on the **Option** menu, or on the toolbar to carry out the operation. The contents which will be scanned include the module models, the hardware configuration, and the firmware versions of the modules, but do not include the parameters in the modules and the parameters in the CPU module. After **I/O Scan** is clicked, the original hardware configuration and the parameter setting will be cleared. However, the parameters in the CPU module will not be affected.



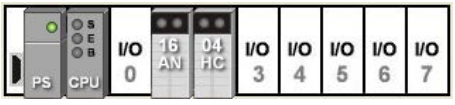
After the scan is complete, the actual hardware configuration will be displayed in the system configuration area, the system will assign input/output devices to the modules, the parameters in the modules are assigned the default values, and the firmware versions set in HWCONFIG will be the actual firmware versions of the modules. If the firmware version of a module which is scanned is higher than the firmware version of the module in the MDS document, the highest version which can be selected will be displayed in the **Firmware Version** drop-down list cell for the module, and the actual firmware version of the module will be displayed in the **Comment** cell for the module.

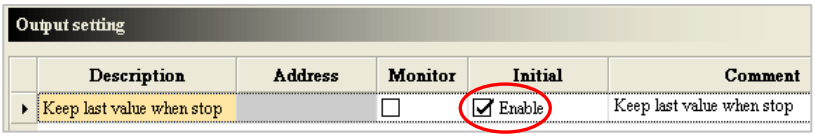


***. Please refer to section 3.4.1 for more information about MDS documents and firmware versions of modules.**

The difference between **I/O Scan** and **Upload** can be illustrated with the following example. If **I/O Scan** is clicked, the hardware configuration will be scanned, and the parameters in the modules will not be scanned. If **Upload** is clicked, the hardware configuration in the CPU module will be read. The hardware configuration uploaded to HWCONFIG may be different from the actual hardware configuration.

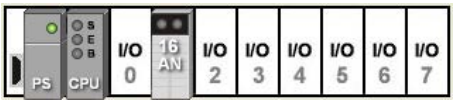
The parameters in the hardware shown below are downloaded to the CPU module first. The parameter in AH16AN01R-5A (installed in slot 1) is as follows.

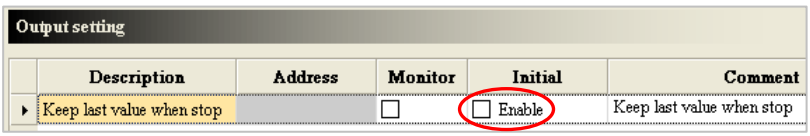




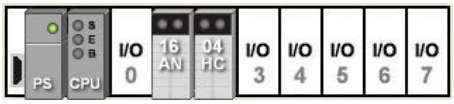
Then, AH04HC-5A installed in slot 2 is removed from the actual backplane. The BUS FAULT LED indicator on the CPU module will be ON, that is to say, the hardware configuration in the CPU is

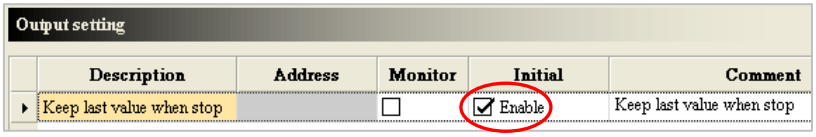
different from the actual hardware configuration.
Next, **I/O Scan** is clicked. The scan result will be the same as the actual hardware configuration. After AH16AN01R-5A (installed in slot 1) is double-clicked to open the **Parameter Setting** window, users will find that the parameter in the module is assigned the default value. The value of the parameter is different from the value of the parameter previously downloaded to the CPU module, that is to say, only the hardware configuration will be scanned after **I/O Scan** is clicked.





Then, **Upload** is clicked. After the uploading is complete, users will find that the hardware configuration and the parameters in the modules which are uploaded are the same as the hardware configuration and the parameters in the modules which were previously downloaded to the CPU module. In other words, after **Upload** is clicked, the system parameters in the CPU module will read, and the actual hardware configuration will not be uploaded.





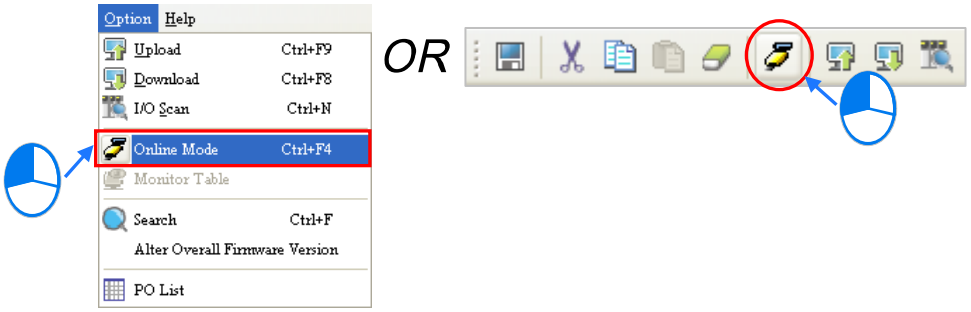
Next, **I/O Scan** is clicked, and the scan result is downloaded to the CPU module. The BUS FAULT LED indicator on the CPU module will be OFF, that is to say, the hardware configuration in the CPU module is the same as the actual hardware configuration.

3.5.6 Online Diagnosis

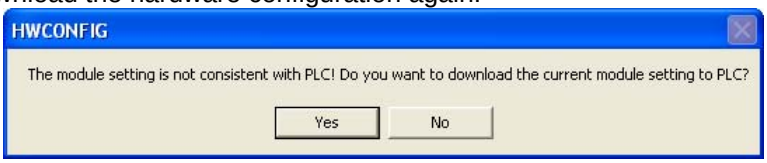
Users not only can configure modules in HWCONFIG offline, but also can operate and inspect the system through the **Online Mode** function if ISPSoft is connected to the CPU module. Owing to the fact that HWCONFIG adopts the communication setting in ISPSoft, users have to make sure that ISPSoft is connected to the CPU module normally before the **Online Mode** function is enabled. Please refer to section 2.4 for more information about communication setting. If users want to enable the **Online Mode** function, but the name of the CPU module is different from the name attached to the model in the project management area, the system will remind the users to check the name of the CPU module and the name attached to the model in the project management area. Please refer to section 3.3.2.1 for more information.

3.5.6.1 Online Mode

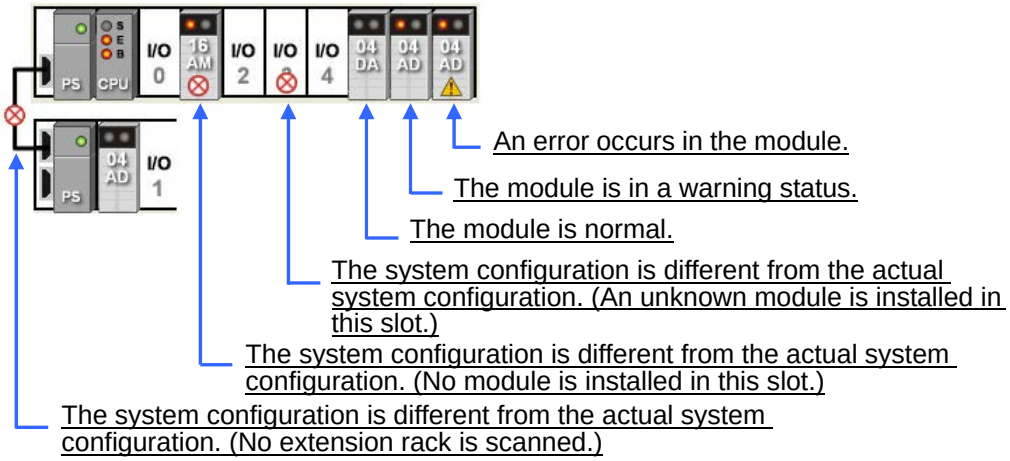
(1) After users click **Online Mode** on the **Option** menu, or on the toolbar, the hardware configuration will be in the online mode. If the users click again, the hardware configuration will not be in the online mode.



(2) Before the hardware configuration enters the online mode, the system checks whether the hardware configuration stored in the CPU module is the same as the hardware configuration in the system configuration area. If the hardware configuration stored in the CPU module is different from the hardware configuration in the system configuration area, the system will ask users to download the hardware configuration again.

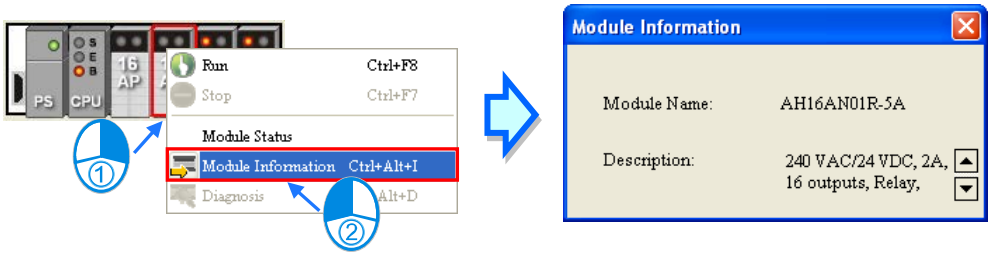


(3) After the hardware configuration enters the online mode, the statuses of the modules displayed in the system configuration area will vary with the actual statuses of the modules. The statuses of the LED indicators on the CPU module displayed in the system configuration area are the same as the actual statuses of the LED indicators on the CPU module. The LED indicators are the RUN LED indicator, the ERROR LED indicator, and the BUS FAULT LED indicator from top to bottom.

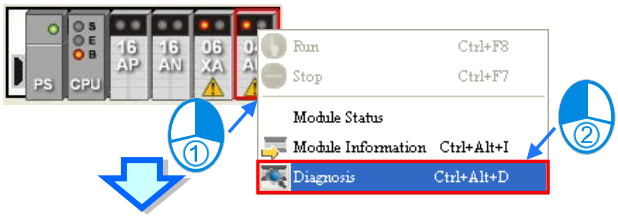


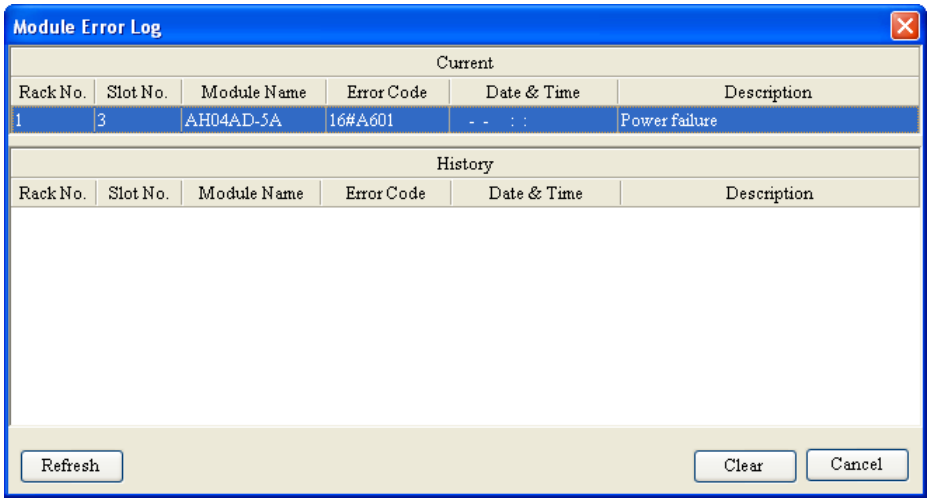
3.5.6.2 Module Information and Diagnosis

If the hardware configuration is in the online mode, users can right-click the CPU module or a module, and click **Module Information** on the context menu. The users can get the information related to the CPU module or the module through the connection, and the information is displayed in the **Module Information** window.



If users right-click a module, and click **Diagnosis** on the context menu, the **Module Error Log** window will appear.





In the **Module Error Log** window, the current error is displayed in the **Current** column, and the errors which occurred before are displayed in the **History** column. The numbers of error logs which are stored in the modules are different. If the errors which occurred in a module before are not stored in the module, the **History** column will be blank. Besides, owing to the fact that the digital I/O modules do not generate any error code, users can not use the **Diagnosis** function.

After the errors occurring in a module are eliminated, users can click **Clear** to clear the errors in the window, and the error stored in the module. The module will not be in the erroneous condition. After users click **Refresh**, the system will retrieve the data in the module again. The data will be displayed in the window.

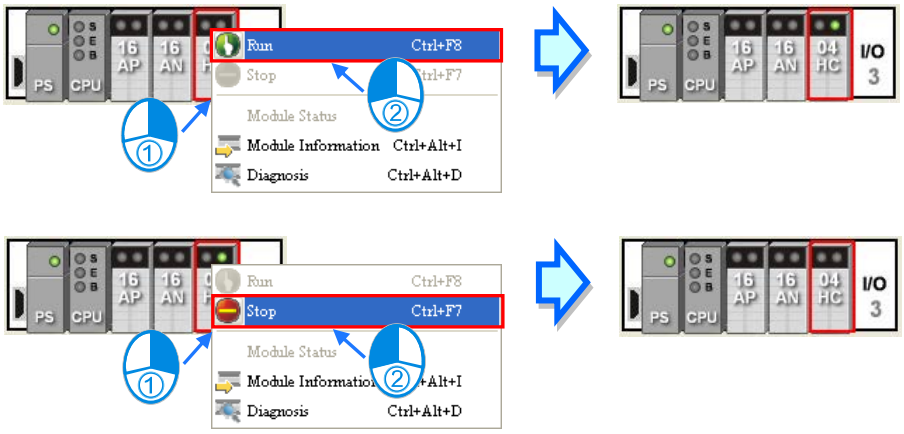
3.5.6.3 Changing the Status of a Module Online

If the hardware configuration is in the online mode, users can change the status of a module, carry out a simple test.

⚠ **Before the status of a module is changed, users have to make sure that the operation does not cause damage to the system or staff.**

- **Changing the operating state of a module**

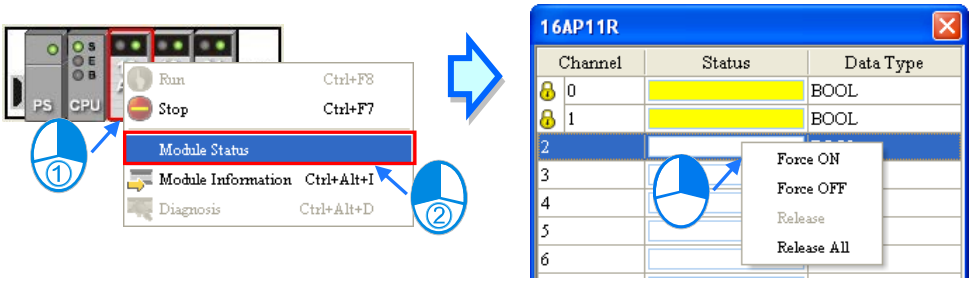
  If the hardware configuration is in the online mode, users can right-click a module, and click **Run** on the context menu. The module will begin to run. When the module runs, the green light on the module is ON. If the users click **Stop** on the context menu, the module will stop running. Besides, if the operating state of the CPU module is changed, the operating states of all the modules are also changed.



3

- **Changing the input/output state of a module**

  If the modules which are configured include a digital I/O module, an analog I/O module, or a temperature measurement module, users can right-click the module, and click **Module Status** on the context menu. The users can monitor the input/output state of the module in the window which appears. In the window for a digital I/O module, users can right-click an input/output channel, and click **Force ON** or **Force OFF** on the context menu. Before **Force ON** or **Force OFF** on the context menus is clicked, the CPU module and the digital I/O module must run to produce the actual outputs.

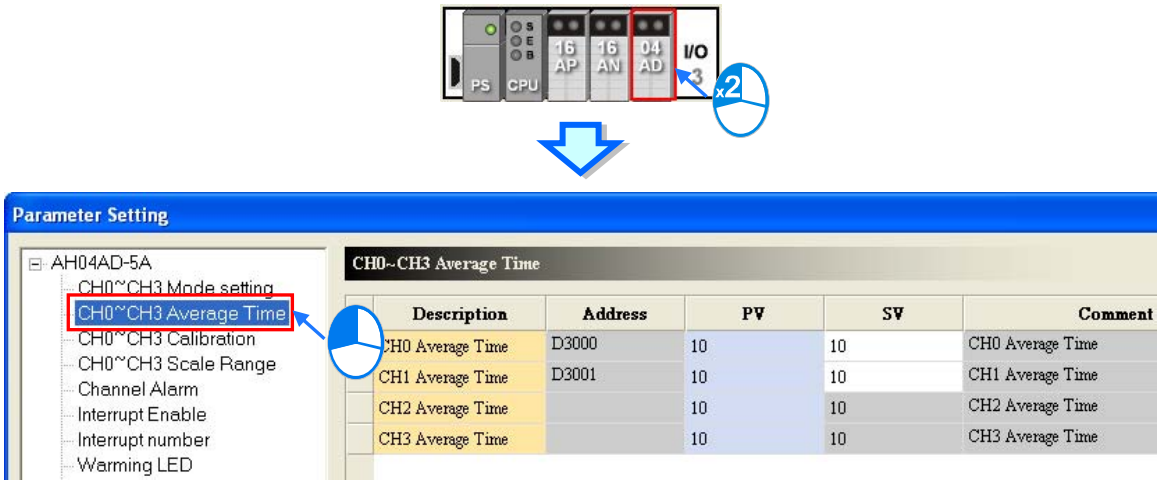


  - **Force ON**: The channel selected is forced ON.
  - **Force OFF:** The channel selected is forced OFF.
  - **Release**: The channel selected is not forced ON/OFF.
  - **Release all**: All the channels are not forced ON/OFF.

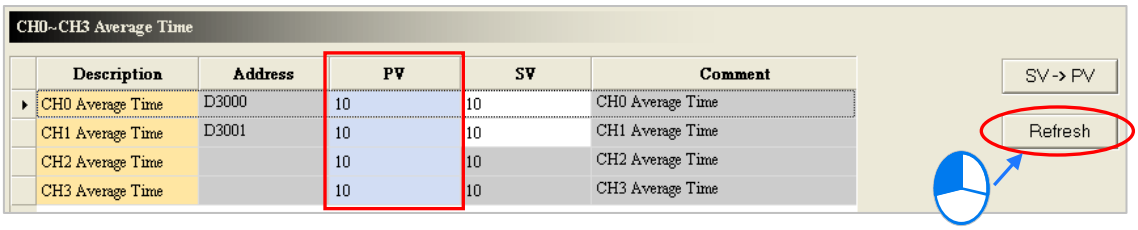
- **Changing the values of the parameters in a module**

  If a module is assigned data registers, users can change the values in the data registers after the hardware configuration enters the online mode. However, on account of the design of the module, the CPU module and the module may need to run to produce the values of the parameters before the values in the data registers are changed.

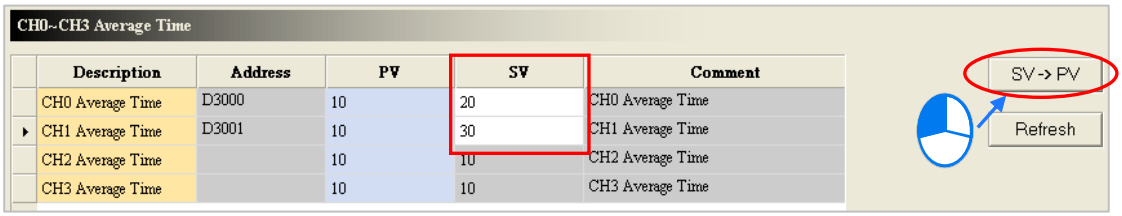
  (1) If the hardware configuration is in the online mode, users can double-click a module to open the **Parameter Setting** window, and click a parameter type which will be set at the left side of the window. The users will find that the appearance of the table in the window is quite different from the appearance of the table in the offline mode.



(2) The present values of the parameters are displayed in the **PV** column, but they are not updated timely. After users click Refresh in the window, the system will retrieve the present values again, and the values retrieved will be displayed in the **PV** column.



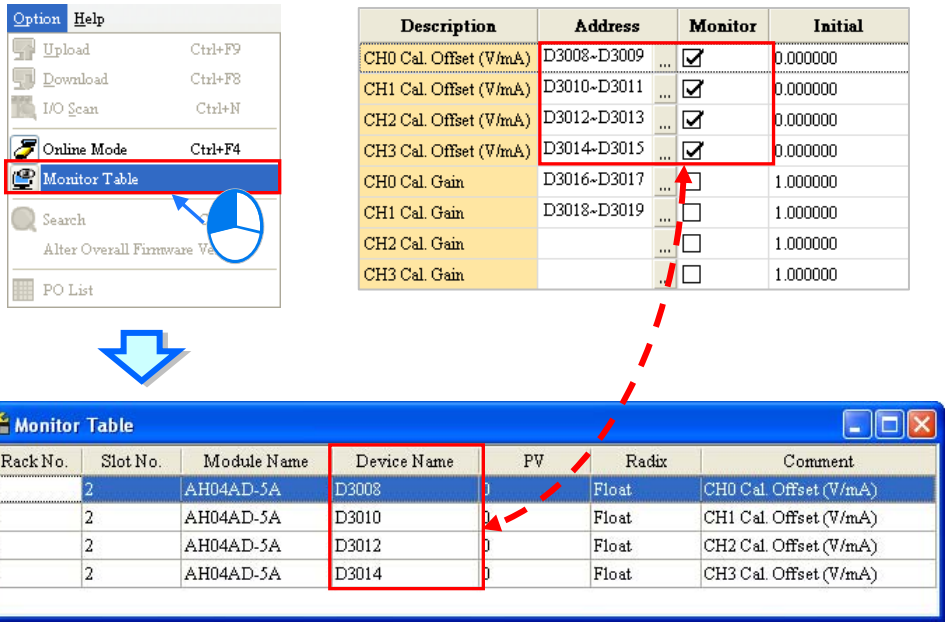
(3) If users want to change the values in the data registers assigned to the parameters, they can type the setting values in the **SV** column. Users can only type the setting values in the **SV** cells for the parameters to which the data registers are assigned. If users want to change the values of parameters, they can type the setting values in the **SV** cells for the parameters, and click SV-> PV after the typing of the setting values is complete. The values of the parameters will be written into the data registers assigned to the parameters.



***. Before the value of a parameter is changed, users have to refer to the instructions for the module to make sure of the range of values which are allowed.**

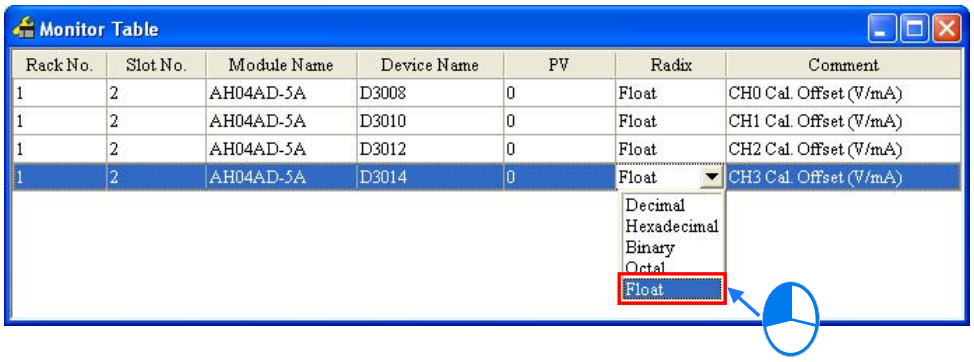
3.5.6.4 Monitoring Table

If the hardware configuration is in the online mode, users can view the values in the data registers in the **Monitor Table** window after they click **Monitor Table** on the **Option** menu. Users can not add any item to the **Monitor Table** window. After users select the checkboxes in the **Monitor** cells for parameters, the data registers assigned to the parameters will be listed in the **Monitor Table** window. Please refer to section 3.4.2 for more information.



***. If the setting in the Monitor column in the Parameter Setting window is changed, the setting can take effect without having to be downloaded to the CPU module.**

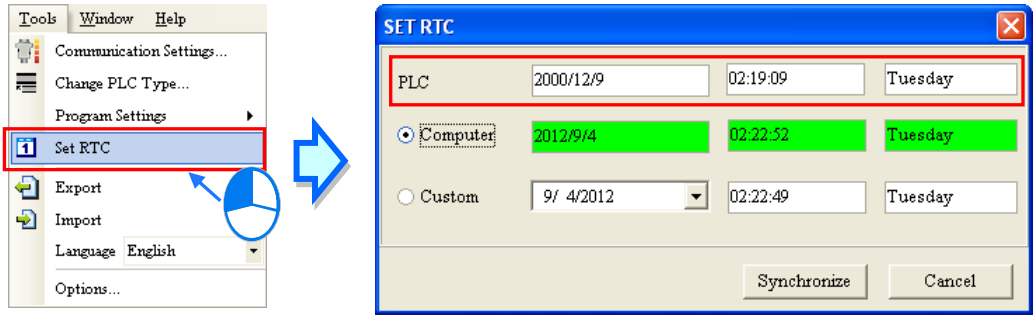
Users can view the values in the data registers in the **Monitor Table** window, but they can not change the values. If users want to change the values in the data registers in the **Monitor Table** window, they can follow the instructions in section 3.5.6. Besides, after users click the **Radix** cell for a data register, they can select a method of representing the value in the data register on the drop-down list.



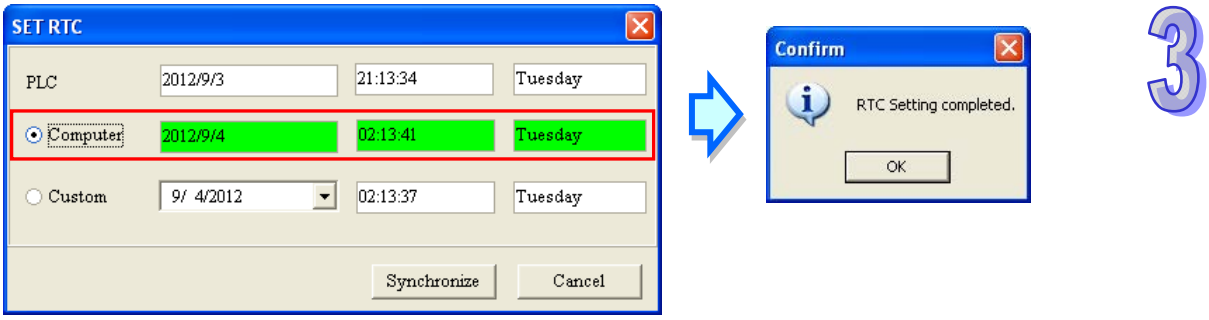
3.6 Setting a Real-time Clock

The real-time clock in a PLC can be set by a tool provided by ISPSoft. Before the real-time clock in a PLC is set, users have to make sure that ISPSoft is connected to the PLC normally. Please refer to section 2.4 for more information about communication setting.

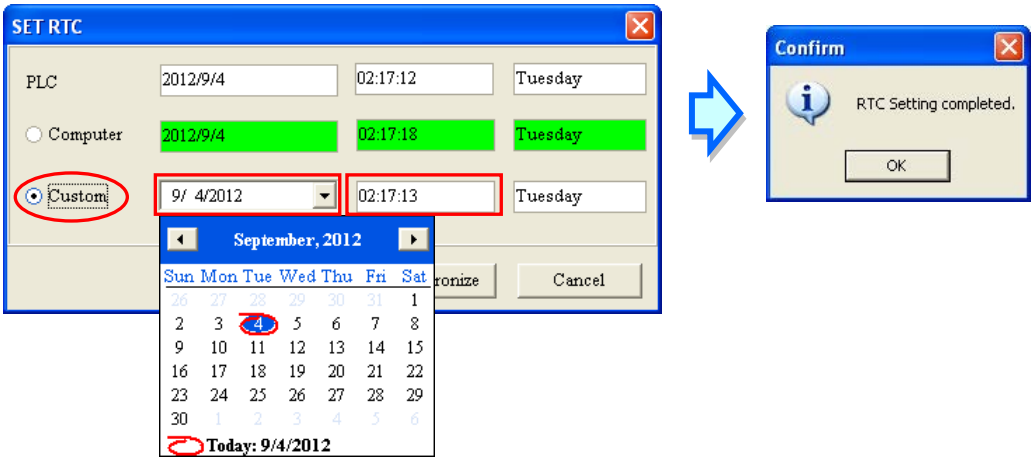
(1) After users click **Set RTC** on the **Tools** menu, the **SET RTC** window will appear. The date displayed at the top of the window is the date retrieved from the PLC when the window is opened, and the time displayed at the top of the window is the time retrieved from the PLC when the window is opened.



(2) If the time on the computer's clock is the time on the real-time clock, users can select the **Computer** option button, and click **OK**.



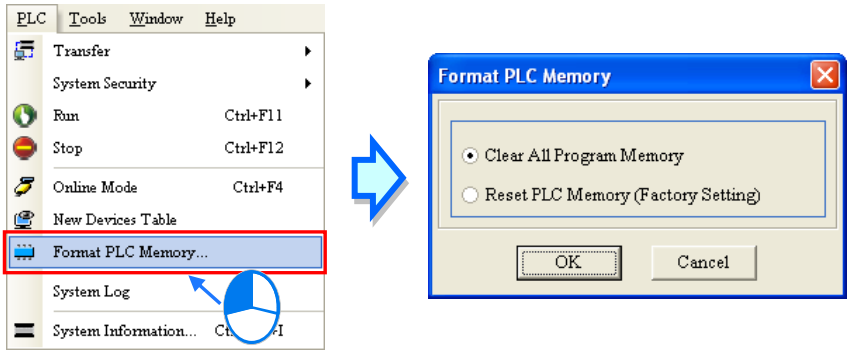
(3) If users want to set the real-time clock by themselves, they can select the **Custom** option button, and type a date and a time in the boxes. If users click ▼ in the box, an auxiliary tool will appear. Click **OK** after the setting of the real-time clock is complete



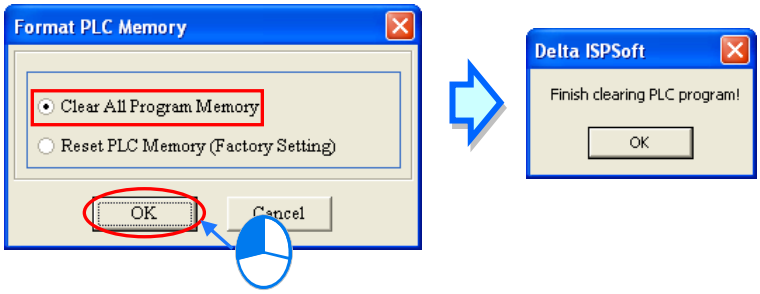
3.7 Setting the Memory in a PLC

By means of ISPSoft, users can clear the memory, or restore the setting in the PLC to the factory setting. Users have to make sure that ISPSoft has connected to the CPU module normally before they carry out the operation. Please refer to section 2.4 for more information about communication setting.

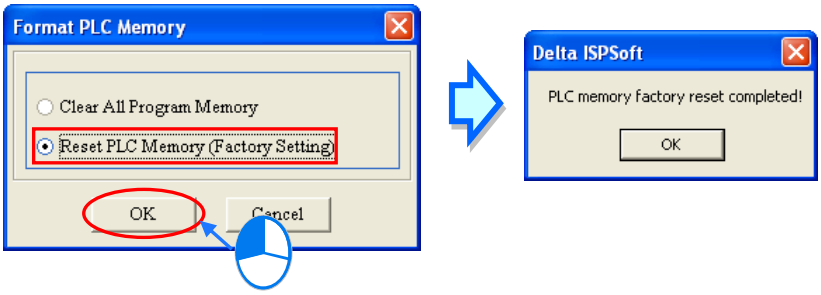
(1) After users click **Format PLC Memory…** on the **PLC** menu, the **Format PLC Memory** will appear.



(2) Select the **Clear All Program Memory** option button, and then click **OK**. The system will clear the memory in the PLC.



(3) If the **Reset PLC Memory (Factory Setting)** option button is selected, the setting in the PLC will be restored to the factory setting after **OK** is clicked. On account of the design of the PLC, after the setting in the PLC is restored to the factory setting, users may need to repower the PLC so that they can proceed with the operation.



If ISPSoft is connected to an AH500 series CPU module, the setting in the CPU module is not allowed to be restored to the factory setting until the **Enable Remote Run** checkbox in the **PLC Parameter Setting** window is selected and the parameters are downloaded. Please refer to section 3.3.2.2 for more information.

Chapter 4 Quick Start

Table of Contents

4.1 Quick Start....4-2
4.1.1 Example....4-2
4.1.2 Hardware....4-2
4.1.3 Program....4-2
4.2 Procedure for Creating a Project in ISPSoft....4-3
4.3 Creating a Project....4-3
4.4 Hardware Configuration....4-4
4.4.1 Configuring a Module....4-4
4.4.2 Setting the Parameters in a CPU Module and a Module....4-6
4.5 Creating a Program....4-8
4.5.1 Adding a Ladder Diagram....4-9
4.5.2 Basic Editing—Creating a Contact and a Coil....4-10
4.5.3 Basic Editing—Inserting a Network and Typing an Instruction....4-13
4.5.4 Basic Editing—Selection of a Network and Operation....4-15
4.5.5 Basic Editing—Connecting a Contact in Parallel....4-17
4.5.6 Basic Editing—Editing a Comment....4-18
4.5.7 Basic Editing—Inserting an Applied Instruction....4-19
4.5.8 Basic Editing—Creating a Comparison Contact and Typing a Constant....4-21
4.5.9 Writing a Program....4-22
4.5.10 Checking and Compiling a Program....4-23
4.6 Testing and Debugging a Program....4-24
4.6.1 Creating a Connection....4-24
4.6.2 Downloading a Program and Parameters....4-27
4.6.3 Connection Test....4-29

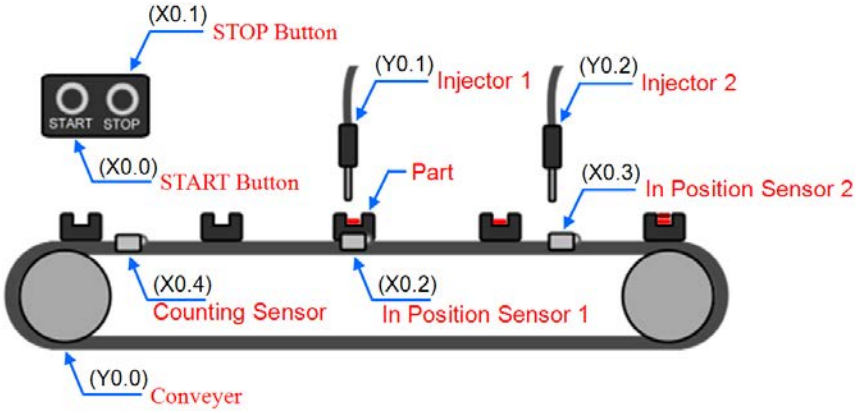
4.1 Quick Start

The chapter provides a simple example, and leads users to create a traditional ladder diagram in ISPSoft in a short time. However, in order to help users who are not familiar with IEC 61131-3 understand the functions provided by ISPSoft, and create a traditional ladder diagram, programming concepts related to IEC 61131-3 are not introduced in this chapter. For example, POUs, function blocks, variables, and etc. are not introduced. The concepts related to IEC 61131-3 will be introduced in the following chapters.

4.1.1 Example

When the equipment operates, the parts on the conveyer are conveyed from left to right. If a sensor senses that a part is under an injector, the PLC will send a trigger signal to the injector, and the injector will injects the glue. How long the part will be injected is set externally, and is not controlled by the program in the PLC. However, the program in the PLC must be able to turn the trigger signal OFF so that the trigger signal can be sent next time. There are two injectors above the conveyer, and the two injectors inject glue in the same way.

Besides, there is a sensor at the left side of the conveyer. When a part passes the sensor, the sensor value increases by one increment. If the sensor value is 100, the internal completion flag will be set to ON. The state of the flag can be used by other procedures later. However, the use of the state of the flag is not introduced in this example.



4.1.2 Hardware

In this example, the AH500 series CPU module used is **AHCPU530-EN**, the digital I/O module used is **AH16AP11R-5A**, and the main backplane used is **AHBP04M1-5A**. The table below is an I/O allocation table.

Type	ID	Description
Digital input	X0.0	START button
Digital input	X0.1	STOP button
Digital input	X0.2	In position sensor 1
Digital input	X0.3	In position sensor 2
Digital input	X0.4	Counting sensor
Digital output	Y0.0	Conveyer
Digital output	Y0.1	Trigger signal for injector 1
Digital output	Y0.2	Trigger signal for injector 2

4.1.3 Program

(1) When the START button (X0.0) is turned from OFF to ON, the internal operation flag is set to ON, and the conveyer (Y0.0) starts to run. When the STOP button (X0.1) is turned from OFF to ON, an error occurs (the error flag is ON), the operation flag is reset to OFF, and the conveyer stops running.

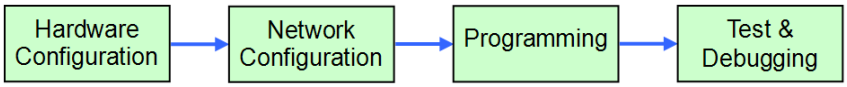
(2) When in position sensor 1 (X0.2) is ON, the trigger signal for injector 1 (Y0.1) is set to ON. When in position sensor 1 is OFF, the trigger signal for injector 1 is reset to OFF.

(3) When in position sensor 2 (X0.3) is ON, the trigger signal for injector 2 (Y0.2) is set to ON. When in position sensor 2 is OFF, the trigger signal for injector 2 is reset to OFF.

(4) When the counting sensor (X0.4) is turned from OFF to ON, the sensor value increases by one increment. If the sensor value is larger than or equal to 100, the internal completion flag will be set to ON.

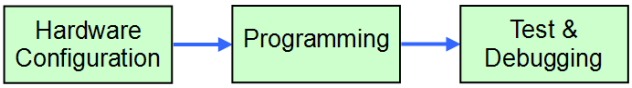
4.2 Procedure for Creating a Project in ISPSoft

The procedure for creating a project in ISPSoft is as follow. Users can adjust the procedure according to the practical application and their habits.



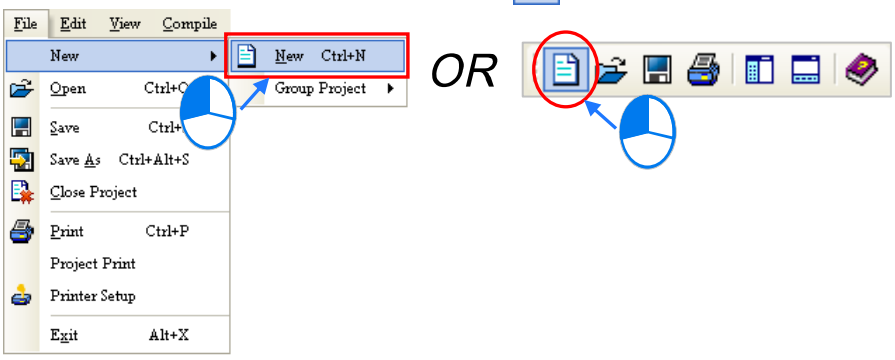
- **Hardware configuration**
  Users can set the parameters such as a range of latched devices and a port number in a PLC. Besides, the users have to configure modules used with an AH500 series CPU module, and set the parameters in these modules.
- **Network configuration**
  If a system used adopts network architecture, or devices need to exchange data, users can configure a network, a PLC Link, or an Ether Link easily through the network configuration tool **NWCONFIG** in ISPSoft.
- **Programming**
  After users write a program in ISPSoft, they can compile the program. If the compiling of a program is unsuccessful, the messages in the **Compile Message** page can lead users to the places where errors occur to check the program code.
- **Test and debugging**
  Users can download a program which is compiled, a hardware configuration, and a network configuration to a PLC. Besides, the users can test and debug the program online by means of the functions provided by ISPSoft.

Owing to the fact that the example introduced in this chapter does not discuss a network configuration, only the following procedure is carried out. The procedure will be introduced in the following section.



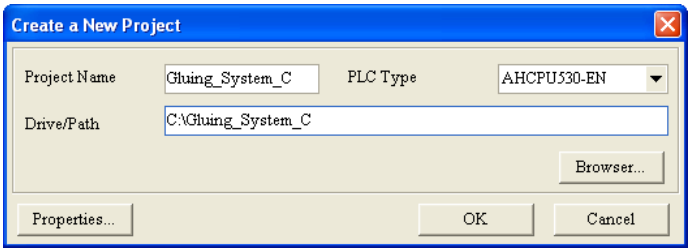
4.3 Creating a Project

After ISPSoft is started, users can click the **File** menu, point to **New**, and click **New** to create a new project. They can also create a new project by clicking on the toolbar after ISPSoft is started.

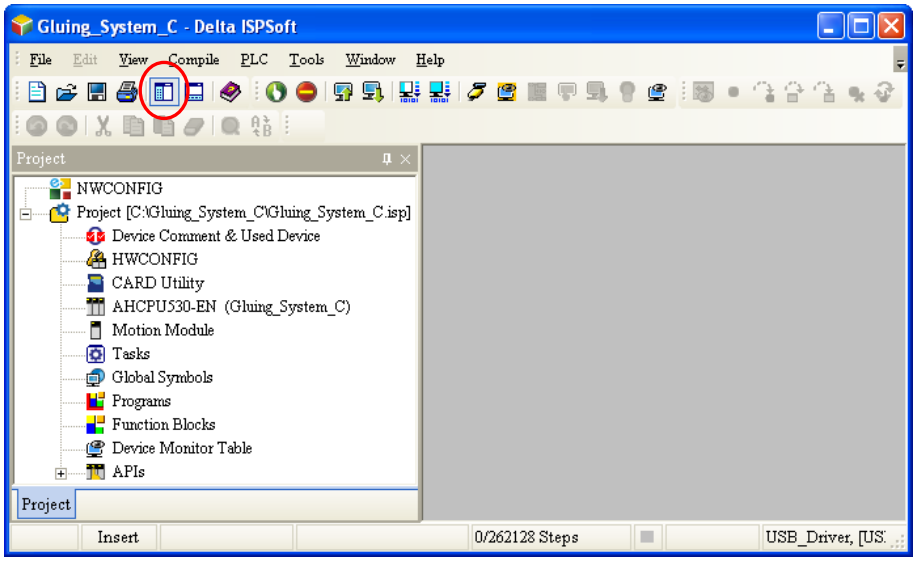


In the **Create a New Project** window, type a project name in the **Project Name** box and a path in the **Drive/Path** box, select a PLC in the **PLC Type** drop-down list box, and click **OK**. (The PLC used

in this example is AHCPU530-EN.)

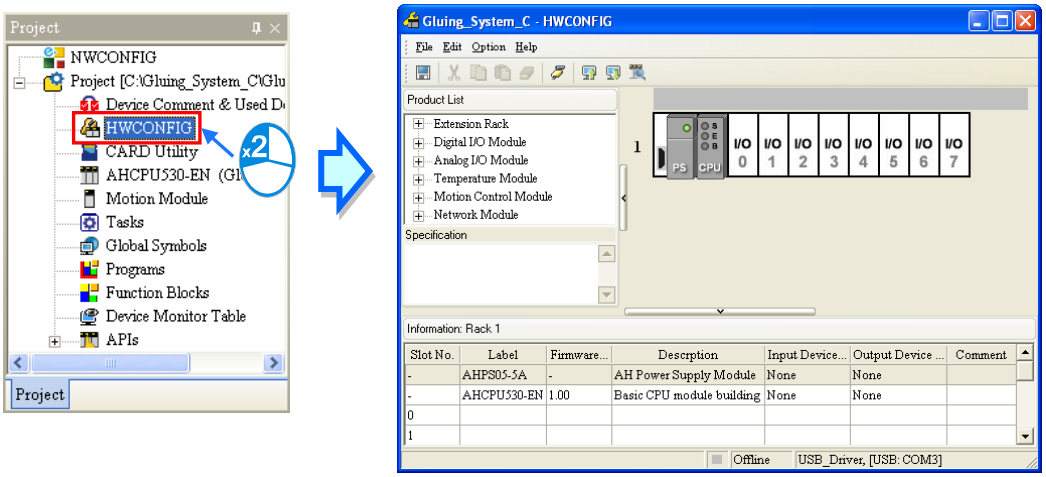


After the project is created successfully, a project management area will appear at the left side of the main screen. The relation between the items listed in the project management area is represented by a hierarchical tree structure. If the project management area does not appear, the users can click **Workspace** on the **View** menu, or click on the toolbar.



4.4 Hardware Configuration

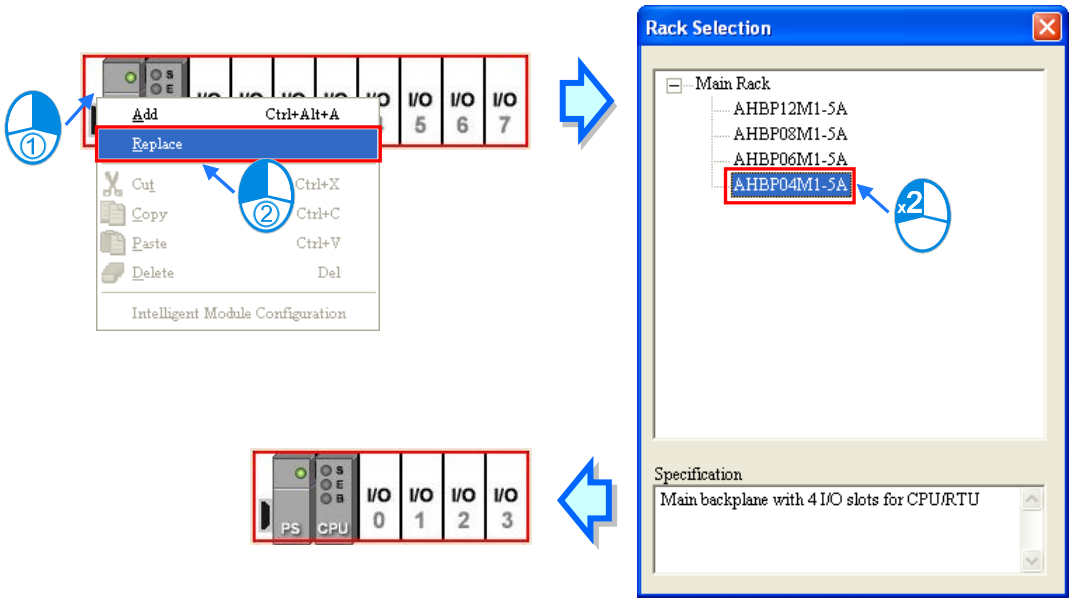
After users double-click **HWCONFIG** in the project management area, the **HWCONFIG** window will appear.



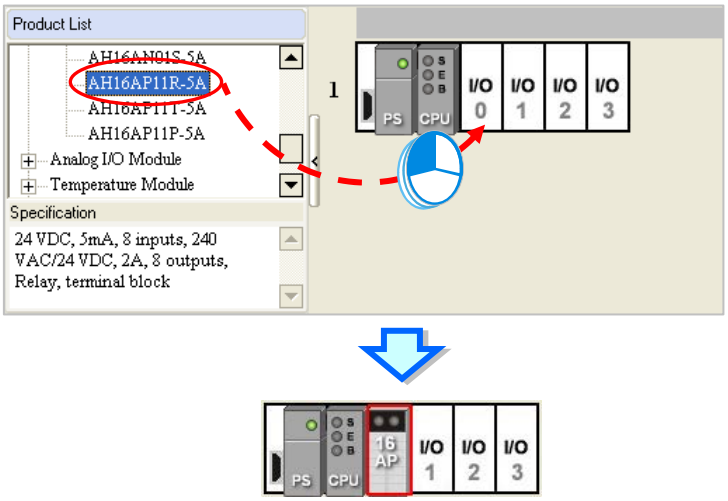
4.4.1 Configuring a Module

In the **HWCONFIG** window, there is an eight-slot backplane on which a CPU module and a power supply module are installed. However, the backplane used in this example is the four-slot backplane **AHBP04M1-5A** on which the digital I/O module **AH16AP11R-5A** is installed.

If users want to replace the backplane, they can right-click the left side of the rack in the system configuration area, click **Replace** on the context menu, and double-click **AHBP04M1-5A** in the Rack Selection window.



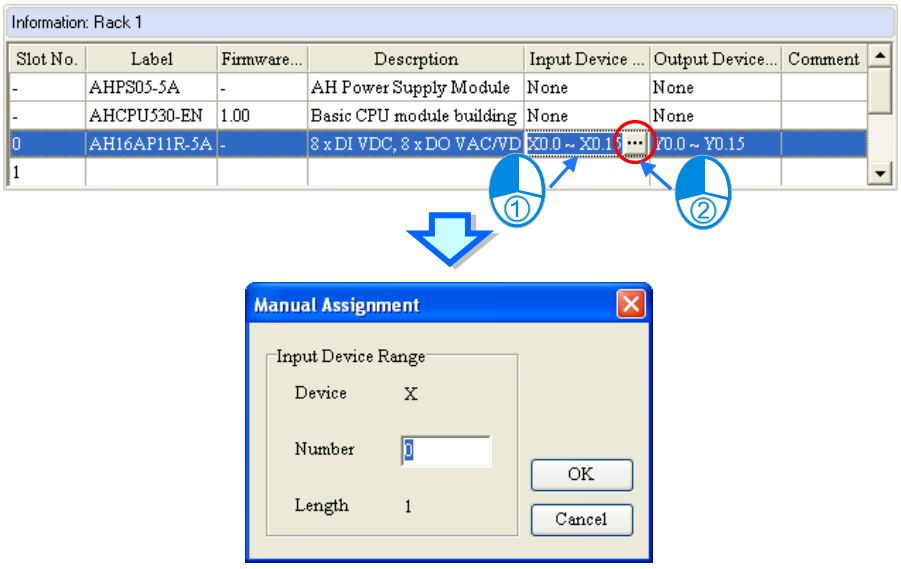
Unfold the **Digital I/O Module** section on the product list, find **AH16AP11R-5A**, and drag the module to a vacant slot on the backplane in the system configuration area. After the module is added successfully, the related information and the devices assigned to the module will be listed in the table at the bottom of the window.



Information: Rack 1

Slot No.	Label	Firmware...	Description	Input Device ...	Output Device...	Comment
-	AHPS05-5A	-	AH Power Supply Module	None	None	
-	AHCPU530-EN	1.00	Basic CPU module building	None	None	
0	AH16AP11R-5A	-	8 x DI VDC, 8 x DO VAC/VD	X0.0 ~ X0.15	Y0.0 ~ Y0.15	
1						

The system automatically assigns devices to a module which is added. If the devices assigned to a module do no conform to what is expected, users can click the **Input/Output Device Range** cell for the module, click ··· in the cell, and type a device address in the **Manual Assignment** window.

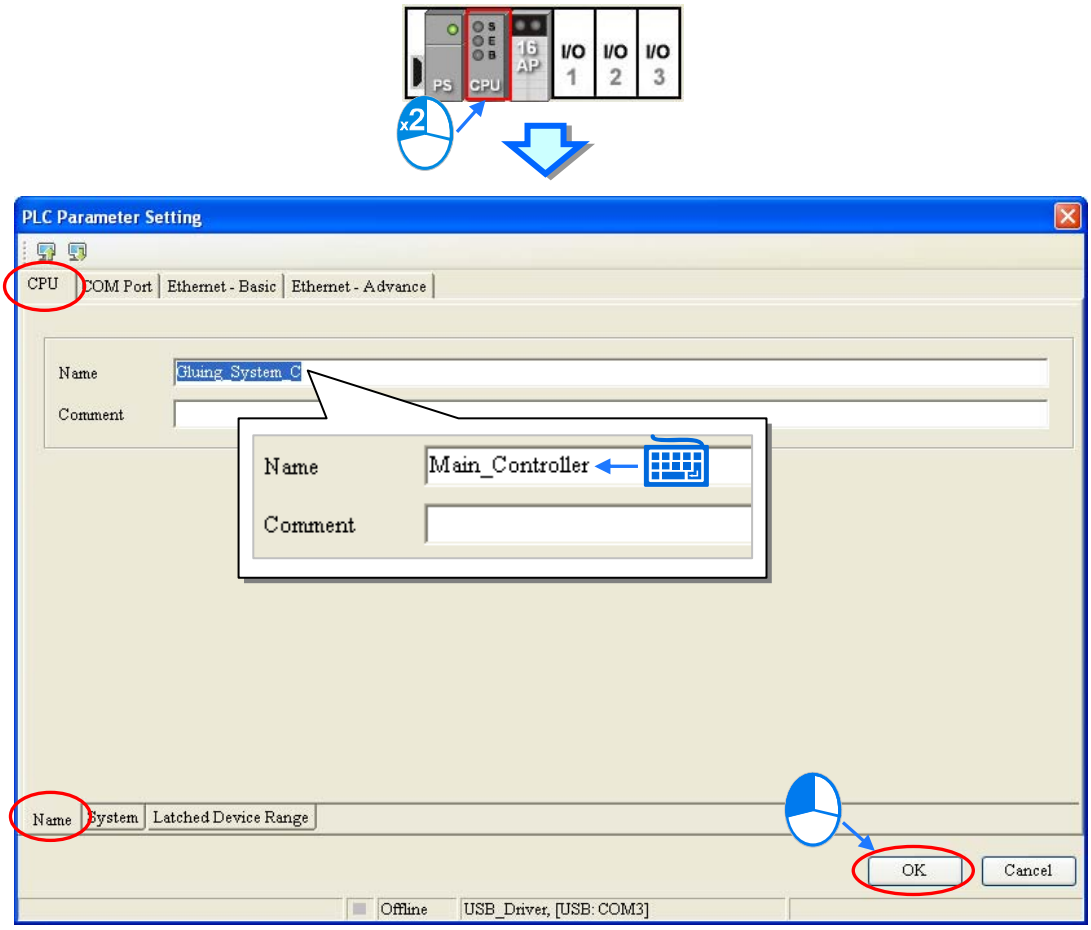


4.4.2 Setting the Parameters in a CPU Module and a Module

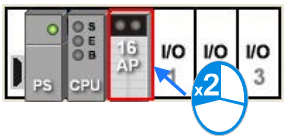
After **AH16AP11R-5A** is configured, users can set the parameters in the CPU module and the parameters in the extension module. After the users double-click the CPU module or the extension module, a corresponding window will appear.

After the CPU module is double-clicked, the **PLC Parameter Setting** window will appear. The users can click the primary tabs at the top of the window, and the secondary tables at the bottom of the window to set the parameters. In this example, the users only need to define the name of the CPU module.

After the users click the **CPU** tab at the top of the window, and the **Name** tab at the bottom of the window, they can type a name in the **Name** box. The users will find the default name in the **Name** box is the same as the project name. Delete the default name, type “Main_Controller” in the **Name** box, and click **OK**.

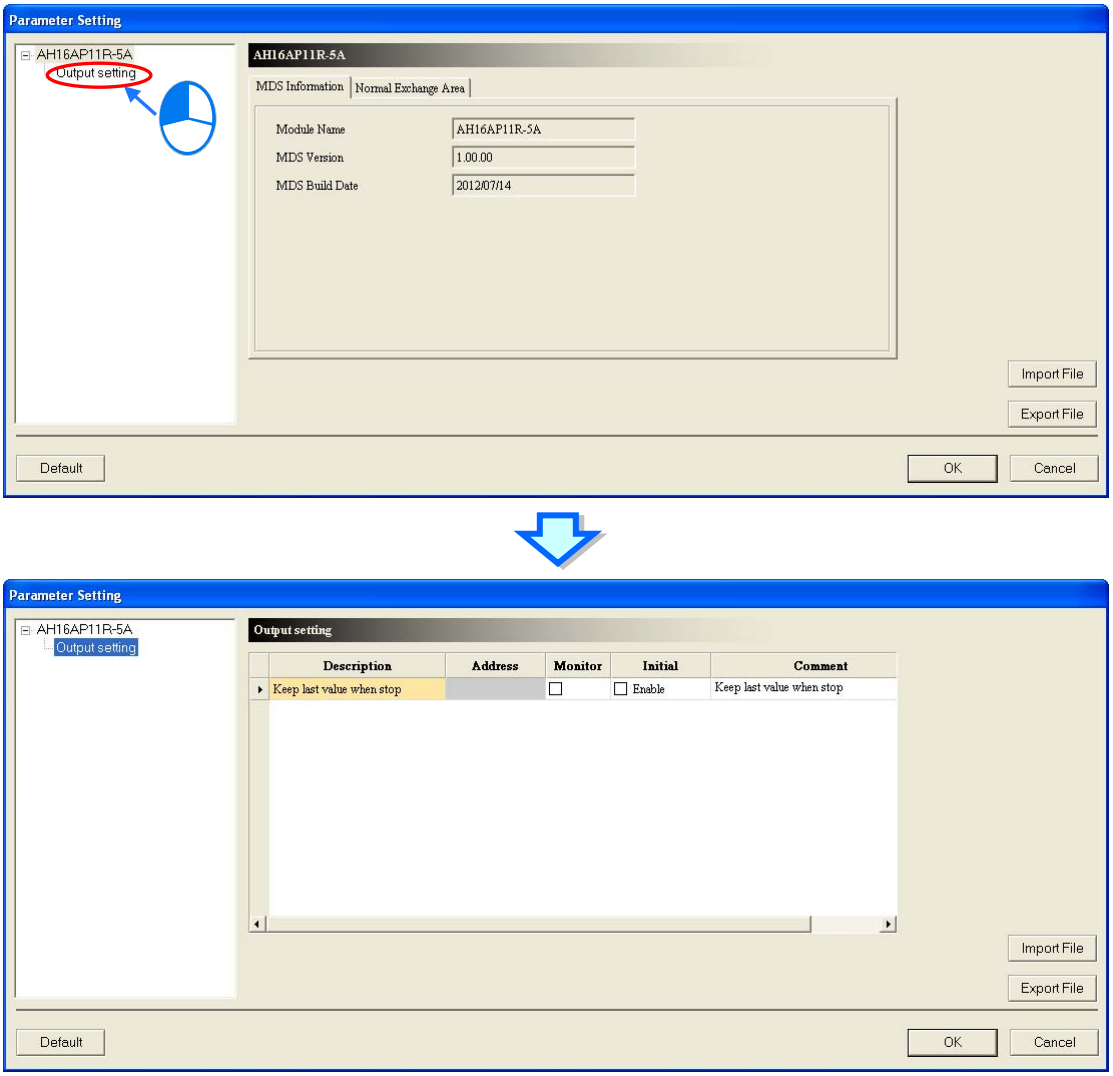


After the users double-click AH16AP11R-5A, the **Parameter Setting** window will appear.



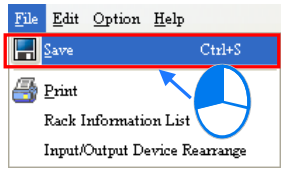
After the **Parameter Setting** window is opened, the users can view the information related to the module. The users can select the parameter type at the left side of the window, and then set the parameter in the table at the right side of the window.

In this example, the default values are retained. Therefore, the step of setting the parameter in AH16AP11R-5A is skipped.



The hardware configuration is not complete until the parameters in the CPU module and AH16AP11R-5A are set. However, the configuration and the setting must be downloaded to the CPU module so that they can take effect. The configuration and the setting are saved here, and will be downloaded with the program in the project later.

If the users want to save the configuration and the setting, they can click **Save** on the **File** menu, or on the toolbar. After the configuration and the setting are saved, the users can close the **HWCONFIG** window.



OR

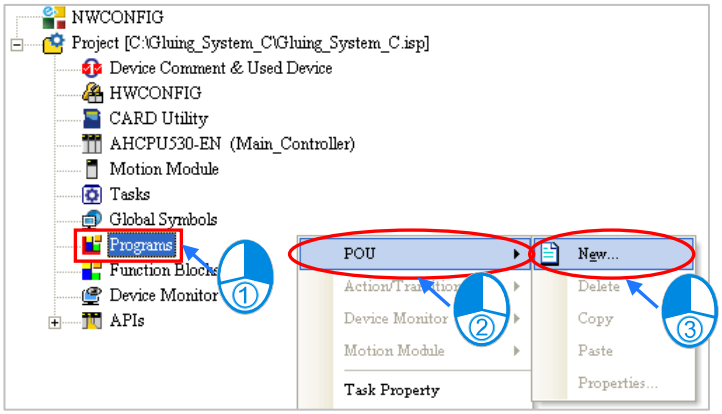
***. Please refer to chapter 3 for more information about HWCONFIG.**

4.5 Creating a Program

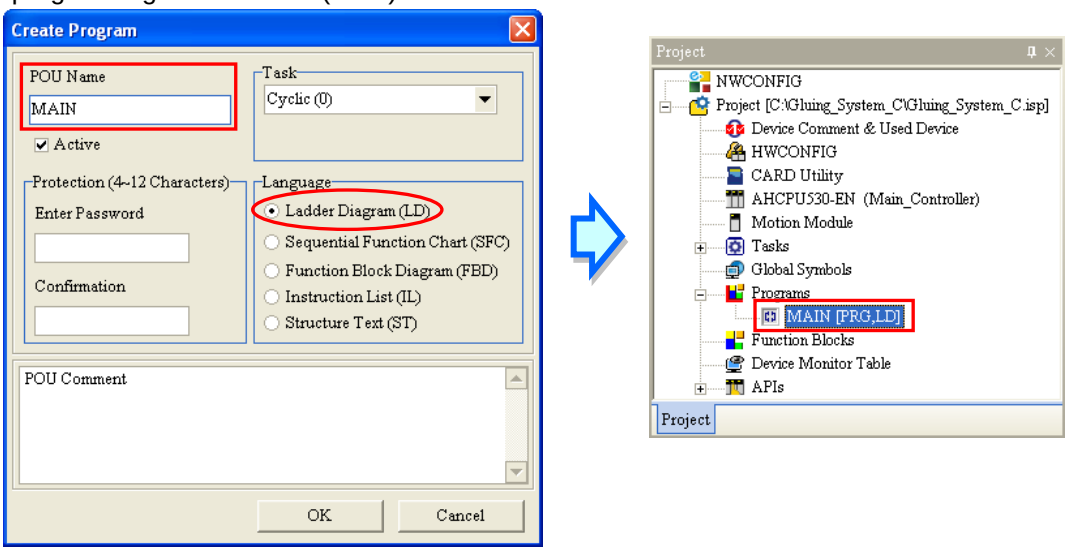
The following sections will lead users to create a traditional ladder diagram in ISPSoft. The contents of the following sections include creating a POU, editing a traditional diagram, and compiling a program. The users are expected to equip themselves with the basic abilities to create a traditional ladder diagram in a short time.

4.5.1 Adding a Ladder Diagram

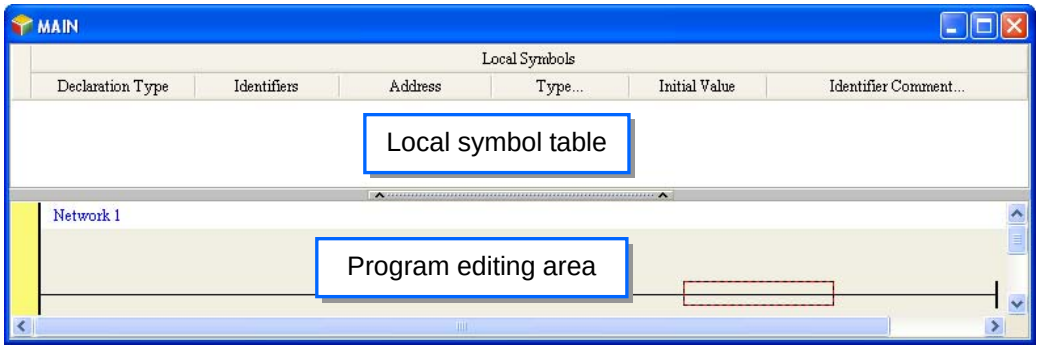
(1) Right-click **Programs** in the project management area, point to **POU** on the context menu, and click **New…**.



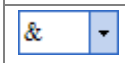
(2) Type a program name in the **POU Name** box, select the **Ladder Diagram (LD)** option button in the **Language** section, and retain the other default values. Click **OK** after the setting is complete. An item will be under **Programs** in the project management area. The item is a program organization unit (POU).



(3) After the POU is added, a program editing window will appear in the main working area.



After the program editing window is opened, the corresponding toolbar will appear in the window. The functions are described below.

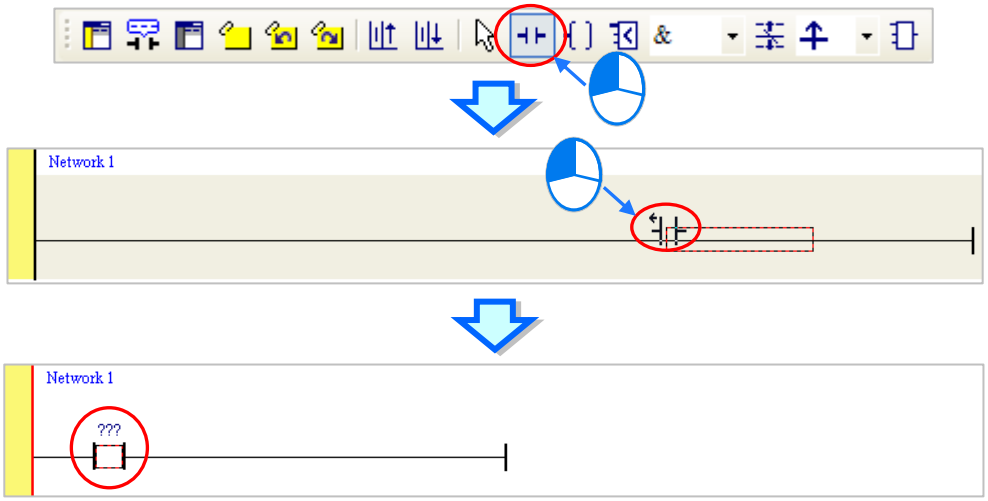
Icon	Keyboard shortcut	Function
	Shift+Ctrl+C	Displaying/Hiding the comments on the networks
	None	Displaying/Hiding the commands on the devices
	Shift+Ctrl+A	Activating/Inactivating the network selected
	Shift+Ctrl+B	Adding a bookmark to the network selected or deleting a bookmark from the network selected
	Shift+Ctrl+P	Going to the previous bookmarked position
	Shift+Ctrl+N	Going to the next bookmarked position
	Ctrl+I	Putting a network above the network selected
	Shift+Ctrl+I	Putting a network under the network selected
	ESC	Selection
	Typing an instruction	Inserting a contact
	Typing an instruction	Inserting a coil
	Typing an instruction	Inserting a comparison contact
	Typing an instruction	Selecting a type of comparison contact
	Typing an instruction	Inserting a block logic instruction (NP/PN/INV/FB_NP/FB_PN)
	Typing an instruction	Selecting a type of block logic instruction (NP/PN/INV/FB_NP/FB_PN)
	Shift+Ctrl+U	Inserting an instruction or a function block

***. Please refer to section 4.5.3 for more information about typing an instruction.**

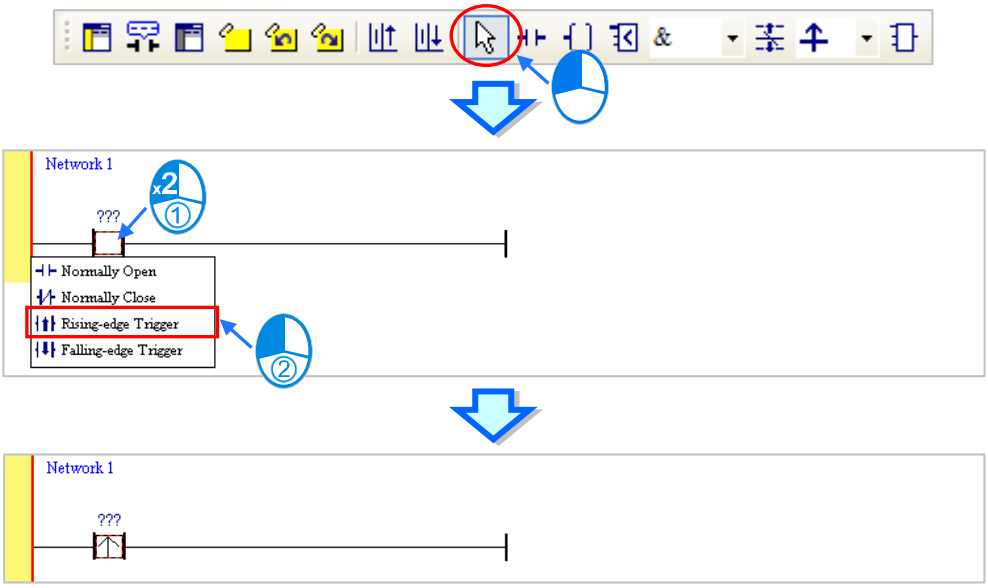
4.5.2 Basic Editing—Creating a Contact and a Coil

(1) Click on the toolbar, and then move the mouse cursor to the red frame in network 1. The mouse cursor appears as a contact when the mouse cursor is moved to the left side of the red frame, the right side of the red frame, or the bottom of the red frame. Users can decide where to insert a contact. If a ladder diagram is edited, the mouse cursor must be near a position which is edited. Besides, an object inserted is arranged by the system automatically. Users can not move the object at will.

In this example, users do not need to decide where to insert the contact. Therefore, the mouse cursor can be near the red frame, and the users can click the left mouse button.

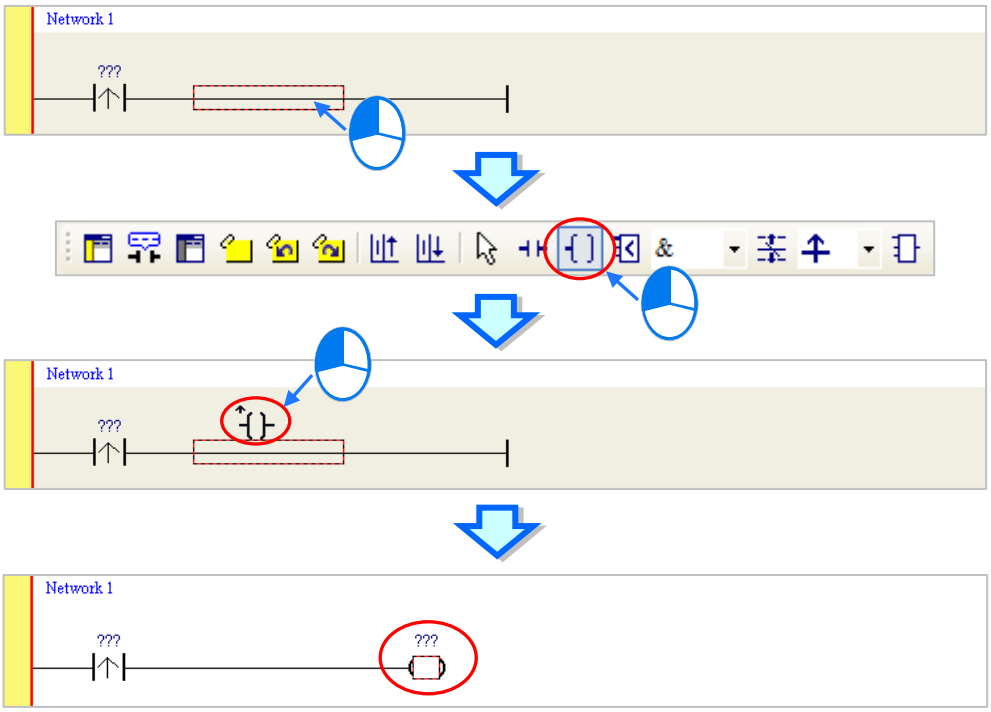


(2) Click on the toolbar, or press Esc on the keyboard. After the contact is double-clicked, a drop-down list will appear. The items on the drop-down list are **Normally Open**, **Normally Close**, **Rising-edge Trigger**, and **Falling-edge Trigger**. In this example, **Rising-edge Trigger** is selected.

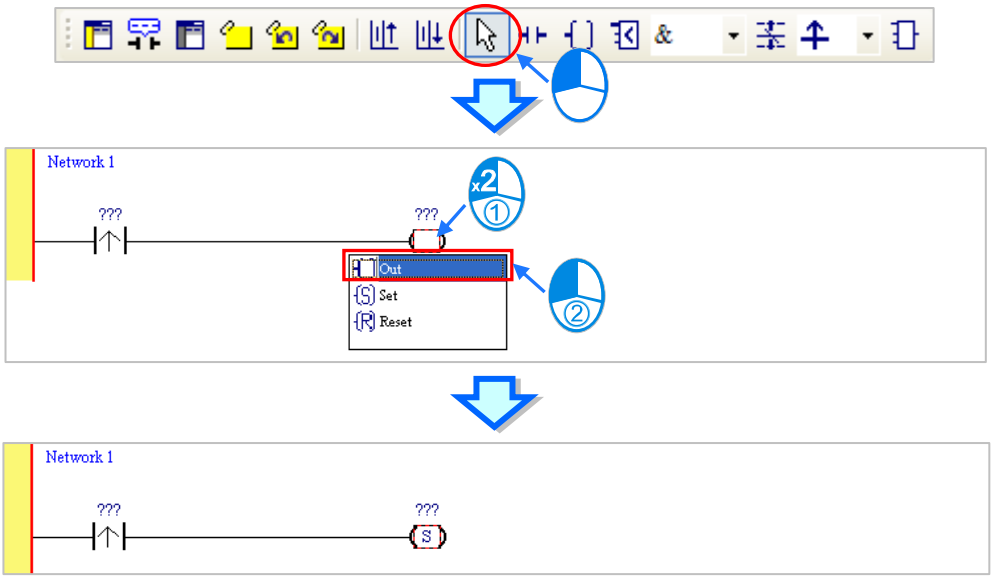


(3) Click the line at the right side of the contact, click on the toolbar, and move the mouse cursor to the red frame. Likewise, the mouse cursor appears as a coil when the mouse cursor is above or under the red frame. Users can decide where to insert the coil.

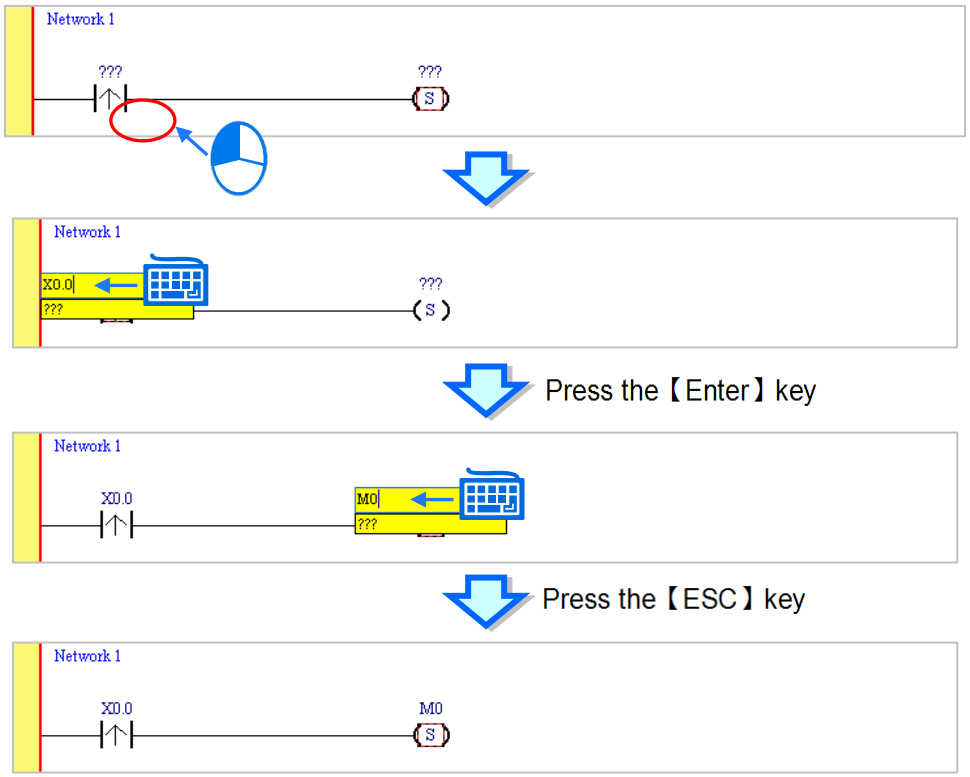
In this example, the users do not need to decide where to insert the coil. Therefore, the mouse cursor can be near the red frame, and the users can click the left mouse button.



(4) Click on the toolbar, or press Esc on the keyboard. After the coil is double-clicked, a drop-down list will appear. The items on the drop-down list are **Out**, **Set**, and **Reset**. In this example, **Set** is selected.



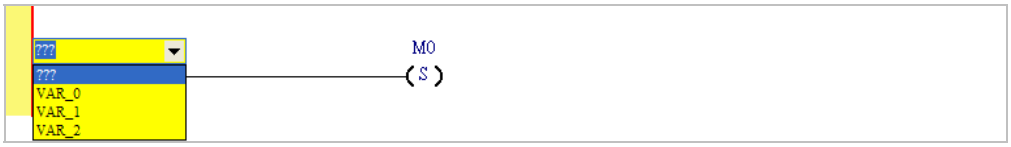
(5) Click ??? above the contact, type a device address in the box, and press Enter on the keyboard to jump to the next box in the network. After a device address is typed in the box, the users can press Esc on the keyboard to complete the editing. In this example, X0.0 is typed in the box for the contact, and M0 is typed in the box for the coil.



Additional remark

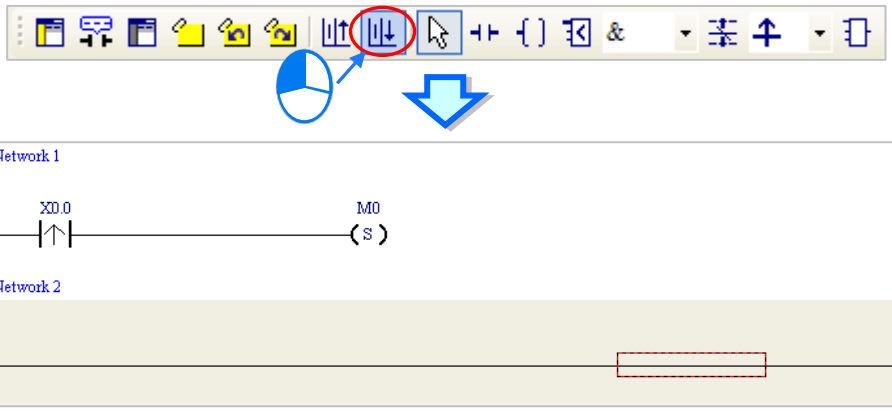
After users click a network and press Enter on the keyboard, they can edit a box. The users can edit the next box in the network after they press Enter on the keyboard. Besides, the next network is selected after the users press Tab on the keyboard. The users can edit a box with the keyboard. After the editing is complete, the users can press Enter on the keyboard to jump to the next box. If the users want to end the editing, they can press Esc on the keyboard.

If the users have declared symbols, they can click in a box, or press Page Down on the keyboard when they edit the box. The symbols on the drop-down list are the symbols which can be assigned to the object. The users can select a symbol by the mouse or the up/down key on the keyboard. Please refer to chapter 6 for more information about symbols.



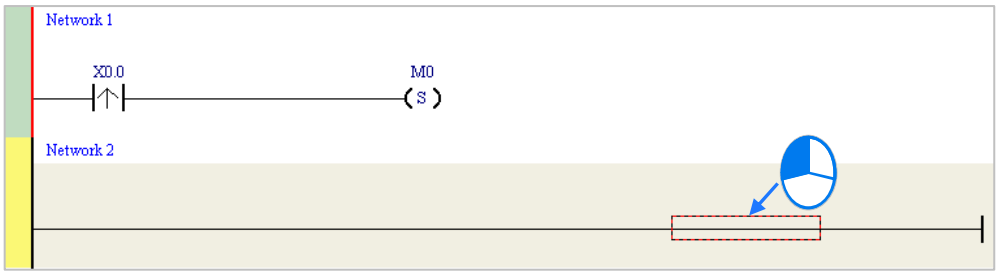
4.5.3 Basic Editing—Inserting a Network and Typing an Instruction

After on the toolbar is clicked, a network will be under the network selected. After on the toolbar is clicked, a network will be put above the network selected. In this example, a network is under network 1.

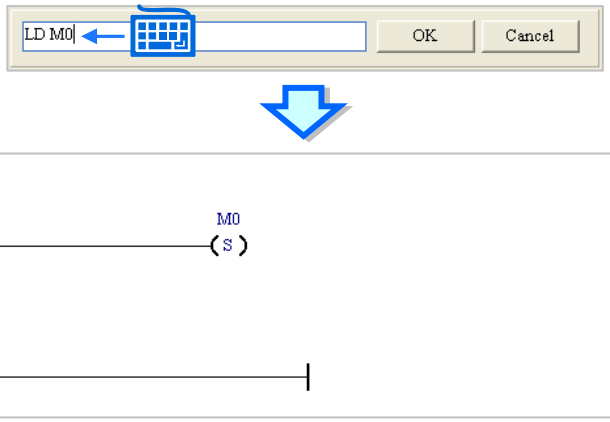


A contact and a coil can be created not only by clicking and on the toolbar, but also by typing instructions.

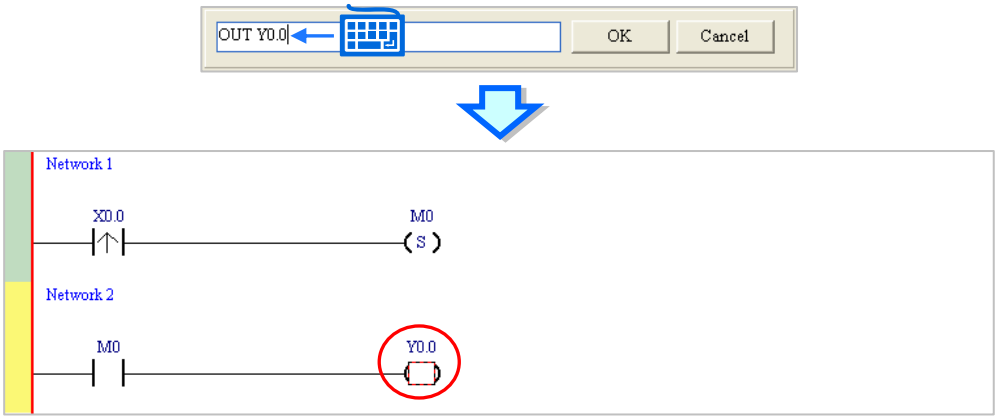
(1) Click the line in network 2.



(2) Type the IL instruction “LD M0”. (The instruction is case-insensitive.) As soon as the IL instruction is typed, a box which can be edited appears. After the typing of the IL instruction is complete, users can press Enter on the keyboard or click **OK** at the right side of the box.



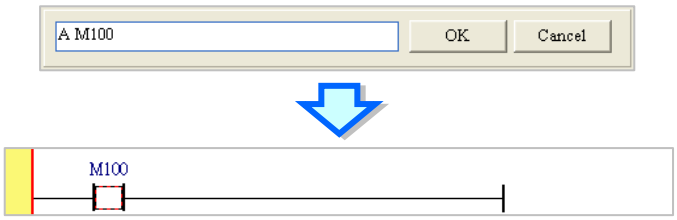
(3) Type the IL instruction “OUT Y0.0”, and write the program shown below.



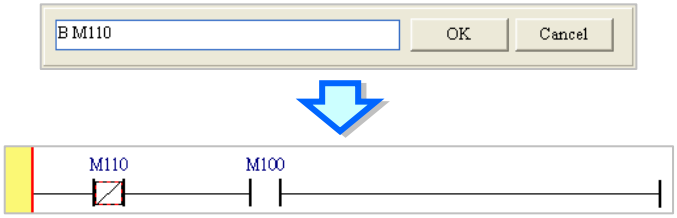
Additional remark

A contact and a coil can be created by typing simple instructions. Please refer to the description below. (The instructions typed are case-insensitive.)

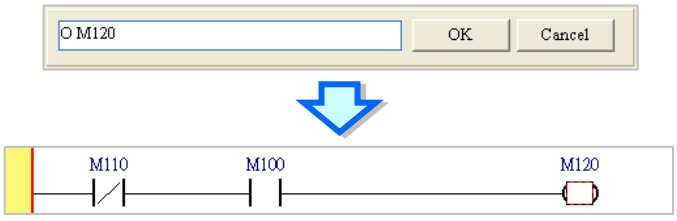
- Inserting a normally-open contact (contact A): “A Device address”



- Inserting a normally-closed contact (contact B): “B Device address”

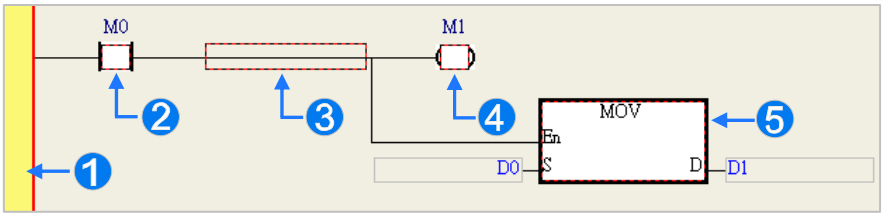


- Inserting an output coil (OUT): “O Device address”



4.5.4 Basic Editing—Selection of a Network and Operation

Before an object in a network is selected, users have to press Esc on the keyboard, or click on the toolbar. After the cursor appears as a small arrow, the users can click the object in the network. The basic selection is shown below.



❶ Selecting the network

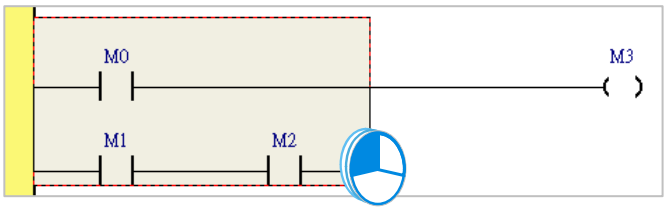
❷ Selecting the input contact

❸ Selecting the network

❹ Selecting the output coil

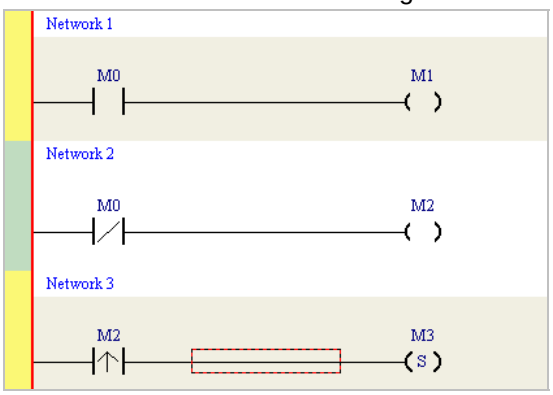
❺ Selecting the block

If users want to select a group of devices, they can click a device, and drag it to draw a frame round the group of devices. The users can also select the group of devices by clicking the first device, pressing Ctrl+B on the keyboard, clicking the last device, and pressing Ctrl+B on the keyboard. Users must draw a frame round devices which are in the same network, and the devices must be adjacent to one another. Besides, input devices and output devices can not be in the same frame.



4

If users want to select several networks, they can press Ctrl on the keyboard, and click the networks. The users can also select a range of networks by pressing Shift on the keyboard, clicking the first network within the range, and the last network within the range.

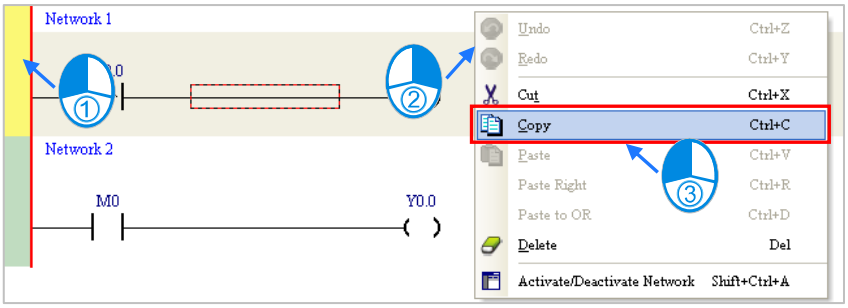


If users right-click an object after the object is clicked, they can click an item on the context menu.

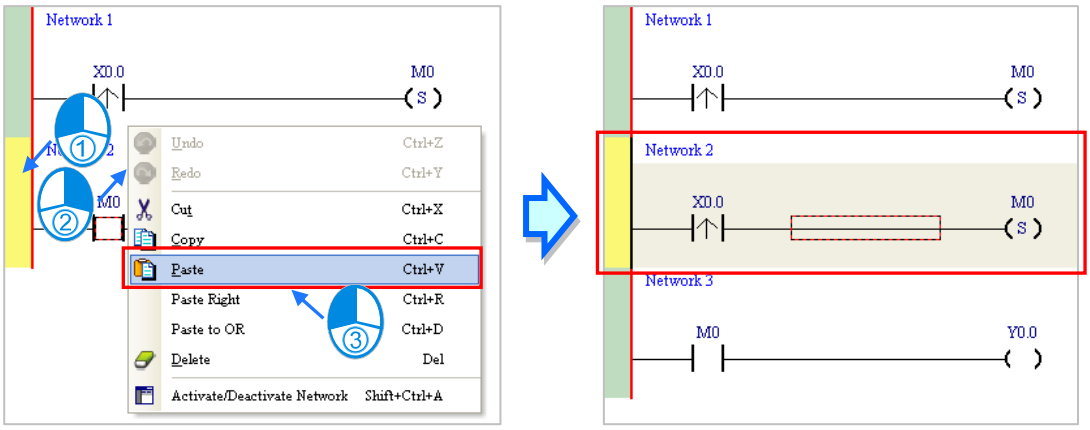
Item	Function
Undo	Undoing the last action (The number of previous actions that can be undone is 20.)
Redo	Redoing an action which has been undone
Cut	Cutting a device, a block, or a network
Copy	Copying a device, a block, or a network
Paste	Paste an object which has been copied or cut on the present position
Paste right	Pasting an object at the right side of the position selected (The object will be connected to the position selected in series.)
Paste under	Pasting an object under the position selected (The object will be connected to the position selected in parallel.)
Delete	Deleting a device, a block, or a network
Activate/Inactivate Network	Activating or Inactivating the network selected (The network which is inactivated is ignored when the program is compiled.)

Users can proceed with the operation in the example.

(1) Select network 1, right-click network 1, and click **Copy** on the context menu.

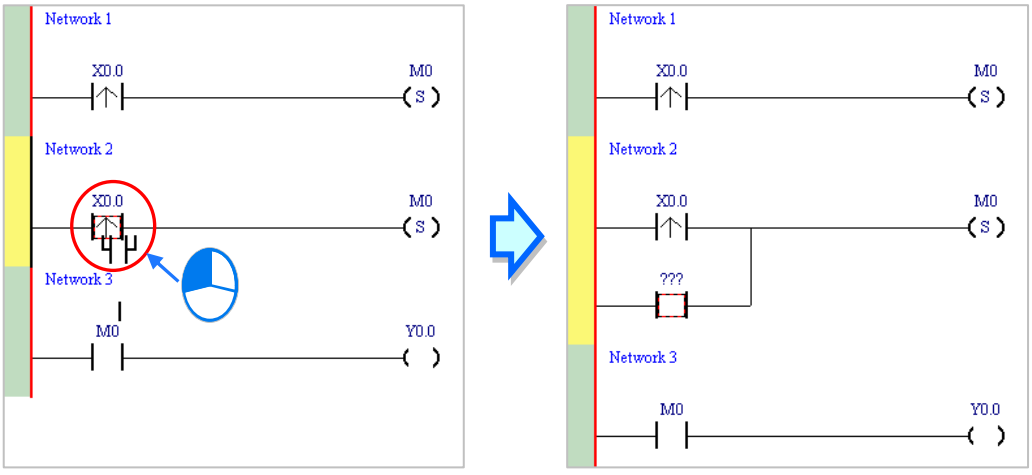


(2) Select network 2, right-click network 2, and click **Paste** on the context menu. A copy of network 1 will be put above network 2, and network 2 will become network 3.

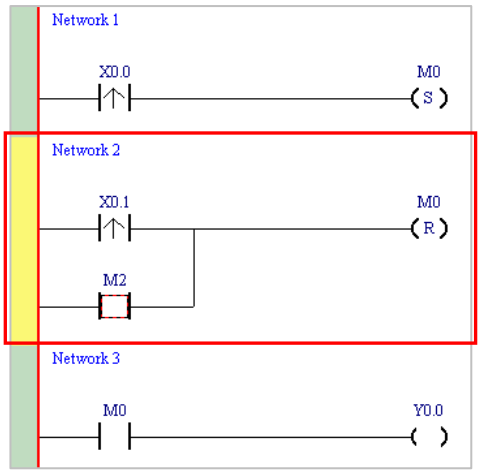


4.5.5 Basic Editing—Connecting a Contact in Parallel

(1) Click on the toolbar, and then move the mouse cursor to the input contact in network 2. The mouse cursor will appear as a contact. Move the mouse cursor to the button of the input contact in network 2. After the mouse cursor appears as ┤├, users can click the left mouse button. A contact will be connected to the input contact in network 2 in parallel.

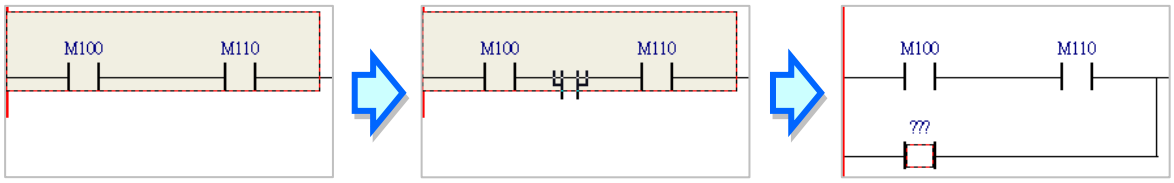


(2) Write the program in network 2 shown below in the way described above.



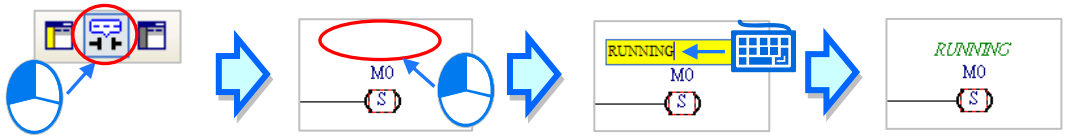
Additional remark

After users select a group of contacts, they can connect a contact to the group of contacts in the way described above.

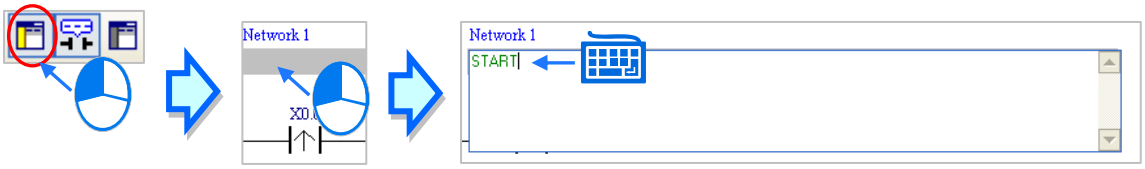


4.5.6 Basic Editing—Editing a Comment

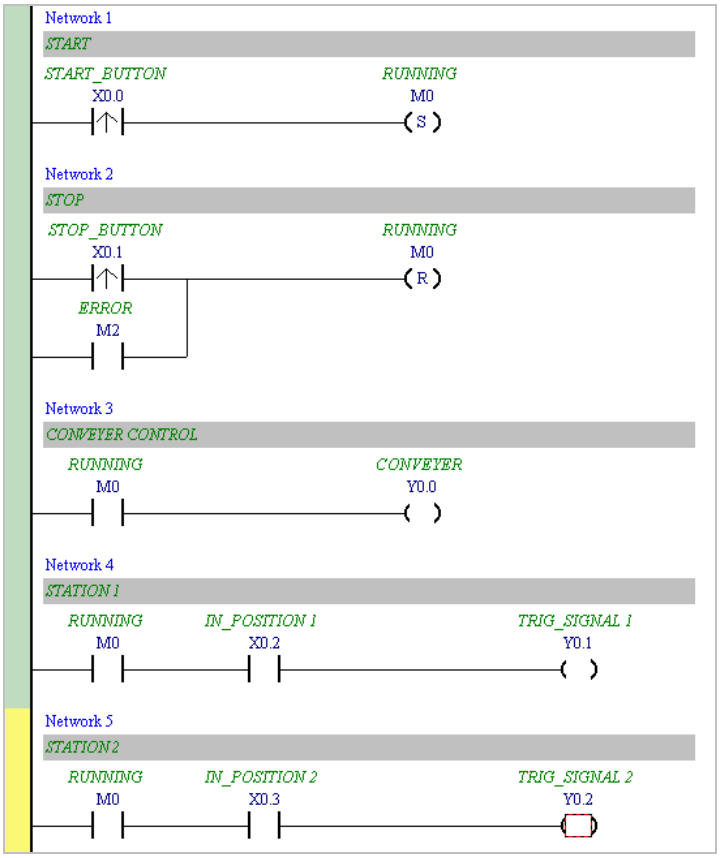
(1) Make sure that on the toolbar is pressed. Click the position above a device name, type a comment in the box, and press Enter on the keyboard.



(2) Make sure that on the toolbar is pressed. Click the position under a network number, and then type a comment in the box. If users want to start a new line of text at a specific point, they can press Shift+Enter on the keyboard. Press Enter on the keyboard after the editing is complete.

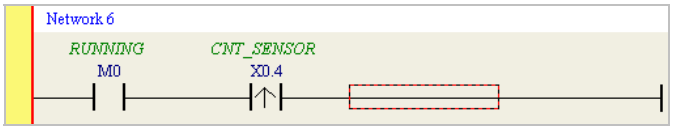


(3) Write the program shown below in the way described above.



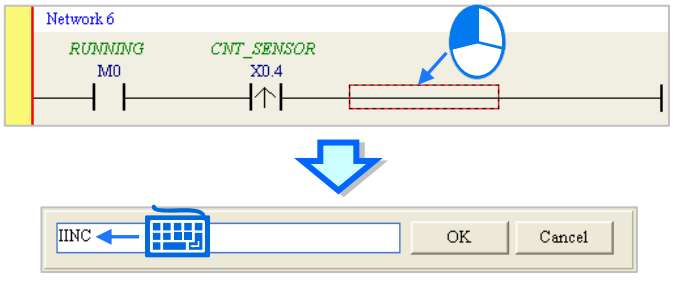
4.5.7 Basic Editing—Inserting an Applied Instruction

Put network 6 under network 5, and then write the program shown below. Users can insert an applied instruction in one of the three ways described below.



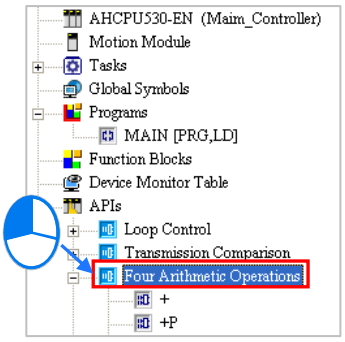
- Method 1

  Click the position where an instruction will be inserted, type the instruction (INC in this example), and press Enter on the keyboard.

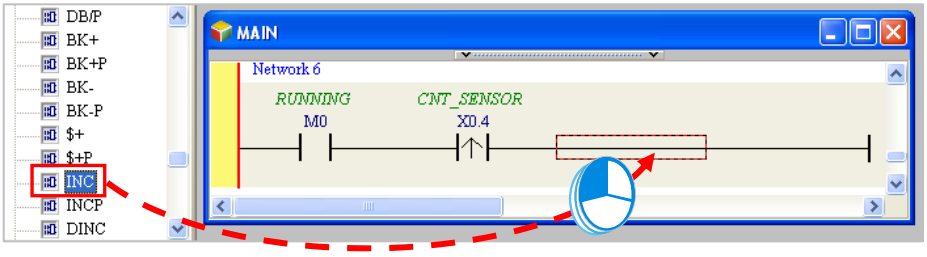


- Method 2

  Unfold the **APIs** section in the project management area, find the instruction type, and unfold the instruction type section.



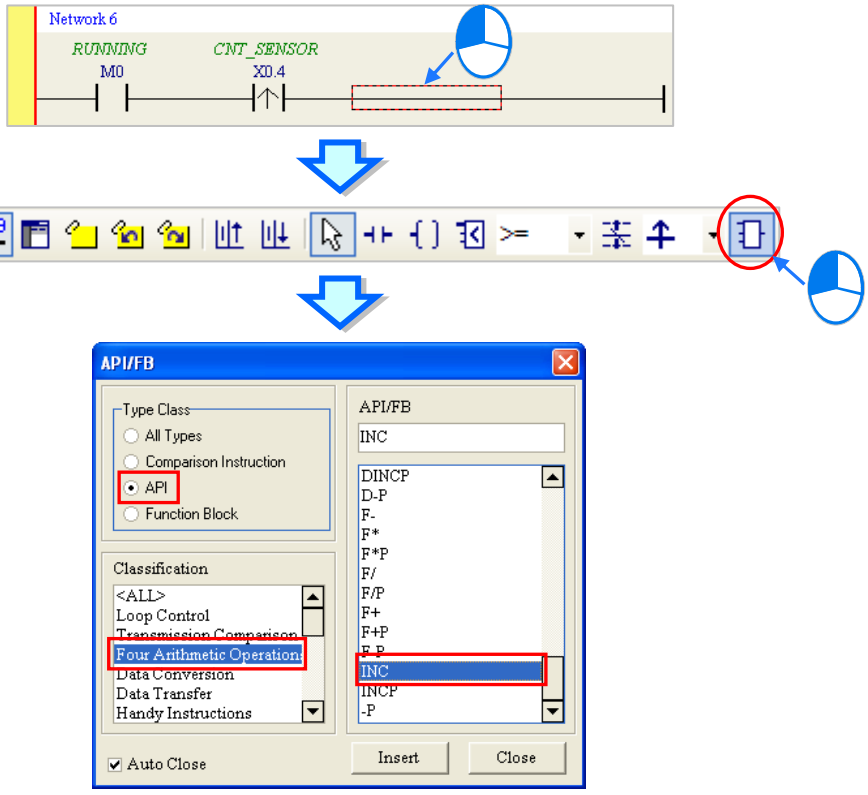
Select the instruction (INC in this example) which will be inserted, and then drag it to the position where it will be inserted.



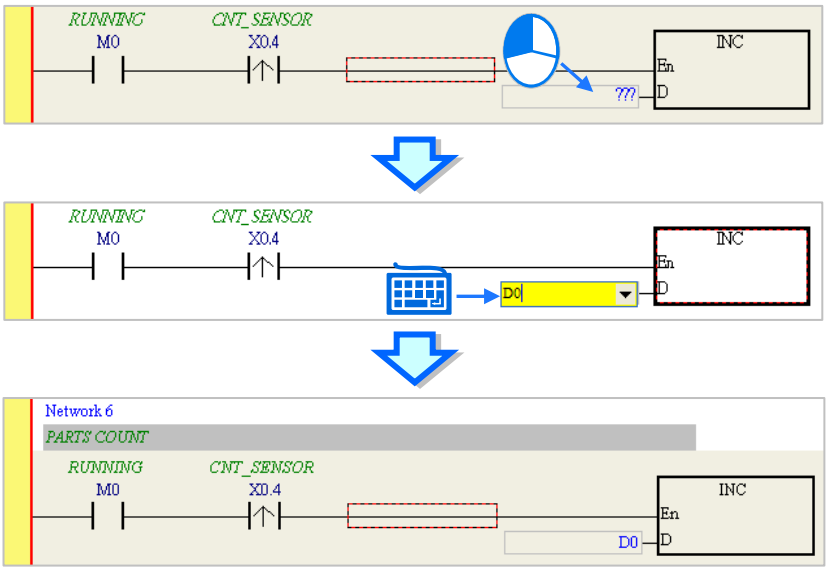
4

- Method 3

Click the position where an instruction will be inserted, click on the toolbar, select the instruction (INC in this example) which will be inserted in the **API/FB** window, and click **Insert**.

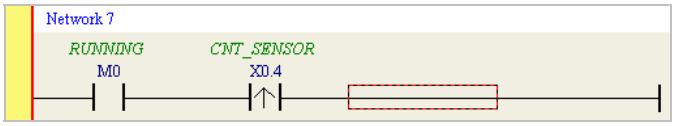


After the instruction is inserted successfully, the users can assign a device address to the operand, and write the program shown below.

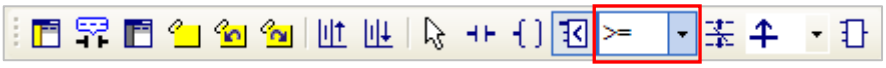


4.5.8 Basic Editing—Creating a Comparison Contact and Typing a Constant

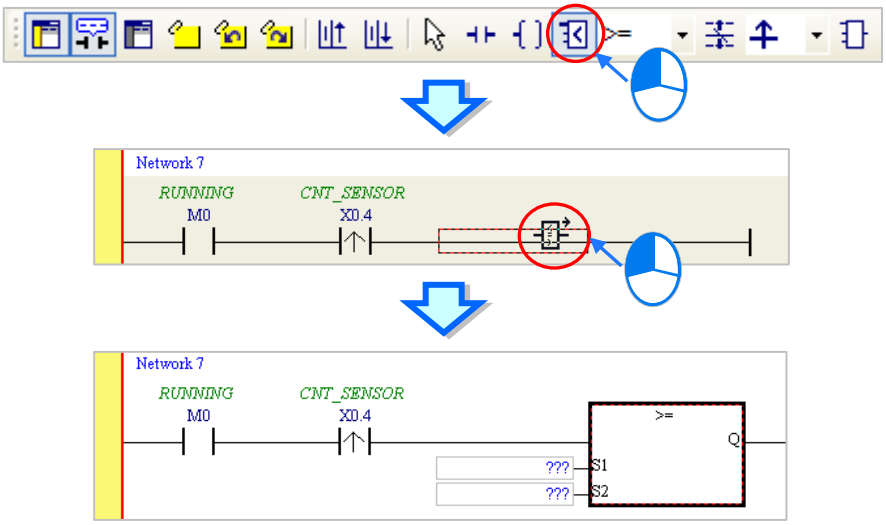
A comparison contact can be inserted not only in one of the three ways described in section 4.5.7, but also by means of the following steps. Users need to put network 7 under network 6, and write the program shown below.



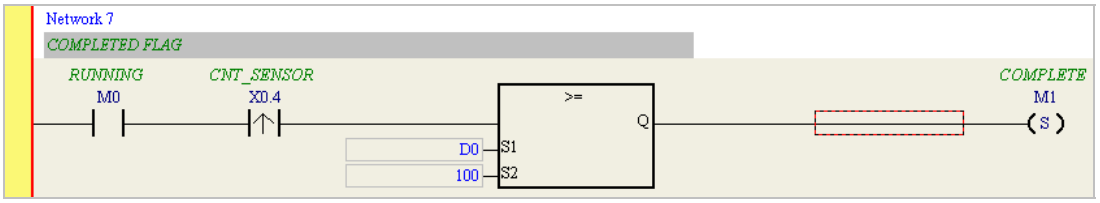
(1) Click on the toolbar, and then select a type (>= in this example).



(2) Click on the toolbar, and then move the mouse cursor to the position where the comparison contact will be inserted. The mouse cursor appears as a comparison contact when the mosue is moved to the left side of the red frame, the right side of the red frame, or the bottom of the red frame. The users can decide where to insert the comparison contact. After the users decided on a position, they can click the left mouse button to insert the comparison contact.

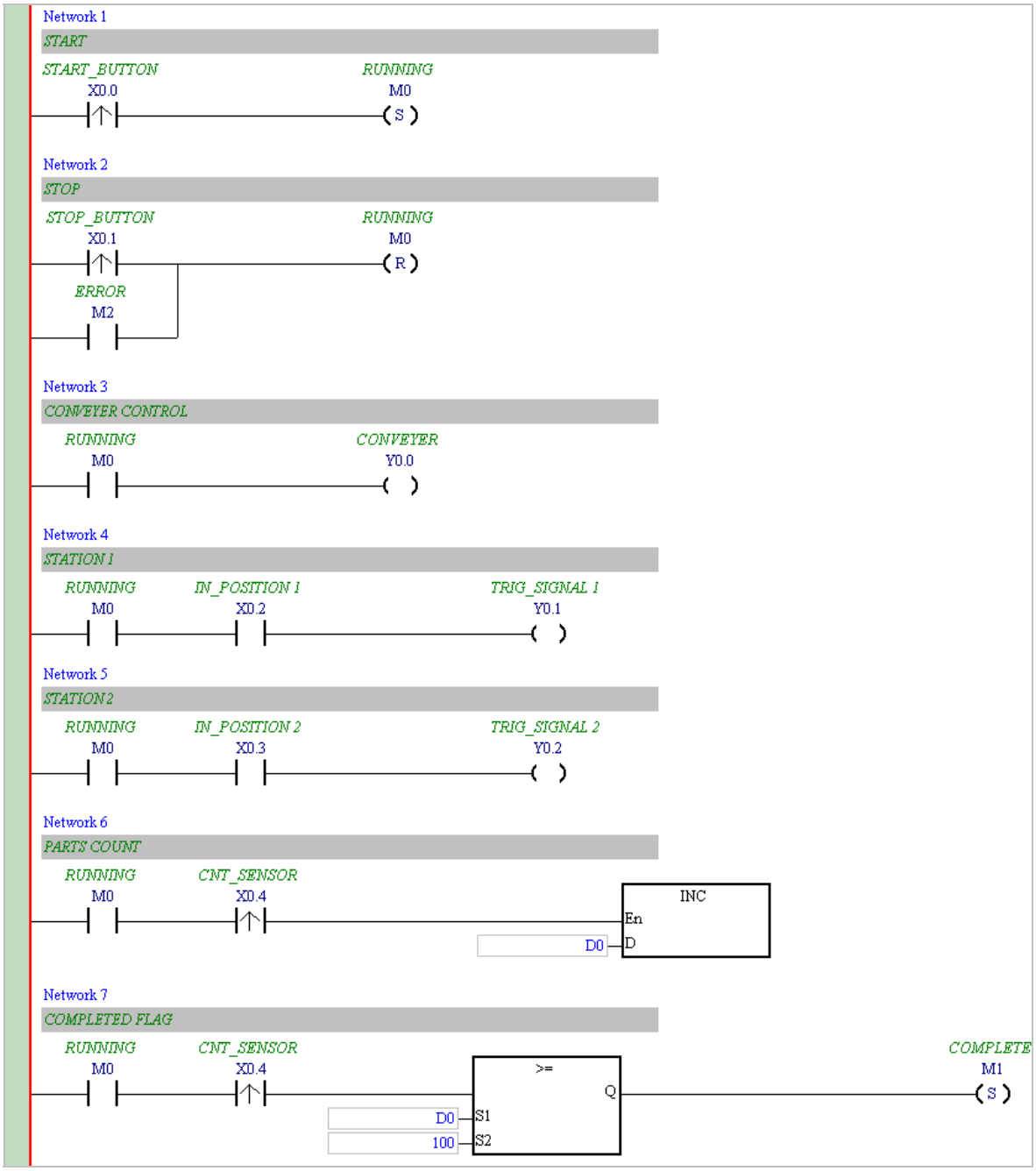


Write the program shown below in the way described above. In WPLSoft, a decimal value is preceded by K, and a hexadecimal value is preceded by H. If users want to type a decimal value in ISPSoft, they can type it directly. If users wan to type a hexadeicmal value in ISPSoft, they have to type “16#” and the hexadecimal value, e.g. 16#7FFF. In ISPSoft, an octal value is preceded by 8#, and a binary value is preceded by 2#.



4.5.9 Writing a Program

The creation of a traditional ladder diagram in ISPSoft has been introduced. Users can write the program shown below in the way described in the previous sections. Owning to the fact that the program has not been compiled, the mother line at the left side of the ladder diagram is red during the writing of the program. The following sections will introduce how to compile the program, and how to download the program which has been compiled to the CPU module to test the program.



***1. The program above is saved in the folder denoted by …\ISPSoft x.xx\Project\Example\Gluing_System_C.**

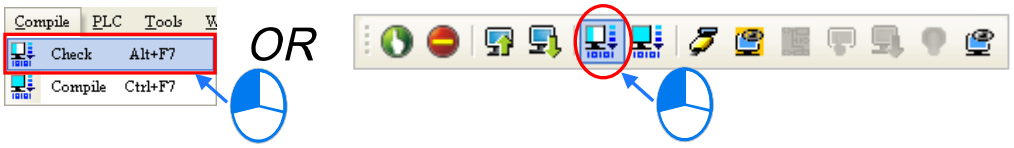
***2. Please refer to chapter 8 for more information about creating a ladder diagram.**

4.5.10 Checking and Compiling a Program

After users write a program, they can check the syntax of the programming language or compile the program. The syntax and the structure in the present window will be checked after the **Check** function is enabled. The whole project will be checked after the **Compile** function is enabled. If there is no error in the project, an execution code will be generated automatically. After the program is compiled successfully, the mother line at the left side of the ladder diagram will become black.

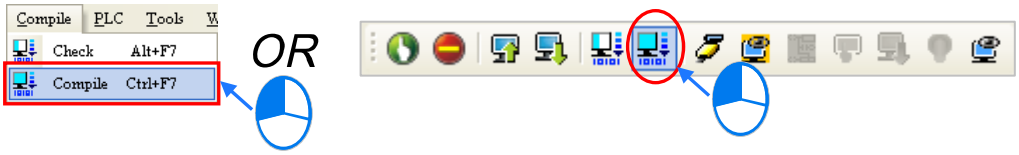
- **Check**

  Click **Check** on the **Compile** menu, or on the toolbar.

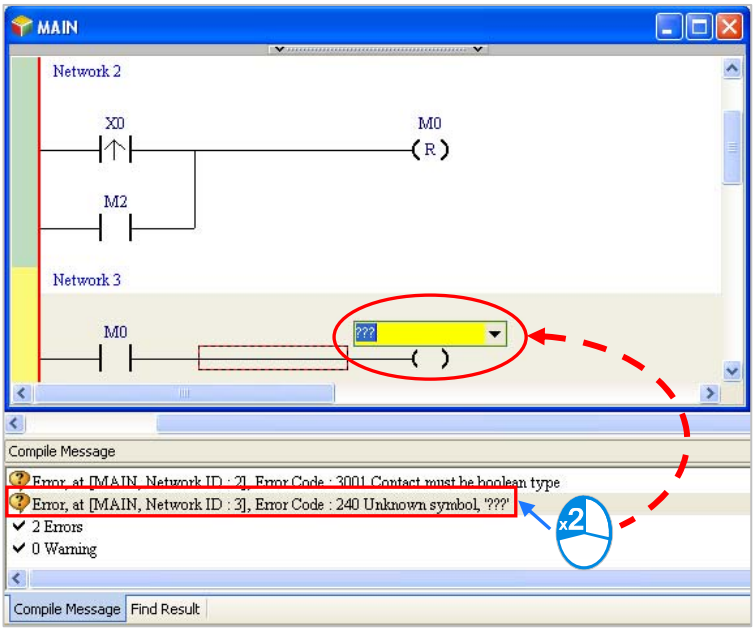


- **Compile**

  Click **Compile** on the **Compile** menu, or on the toolbar.



After the check is complete, the **Compile Message** page shows the result related to the check. If there is any error in the project, the **Compile Message** page will show the related message. After the message is clicked, the system will automatically lead users to the place where the error occurs. The users can enable the **Check** function or the **Compile** function after the error is eliminated.



4.6 Testing and Debugging a Program

4.6.1 Creating a Connection

Before a program and parameters are downloaded to a PLC or monitored online, ISPSoft must be connected to the PLC. In this example, ISPSoft is connected to the CPU module AHCPU530-EN through a USB cable. Please refer to section 2.4 for more information about connecting ISPSoft to a PLC in other ways. Please refer to operation manuals for more information about wiring.

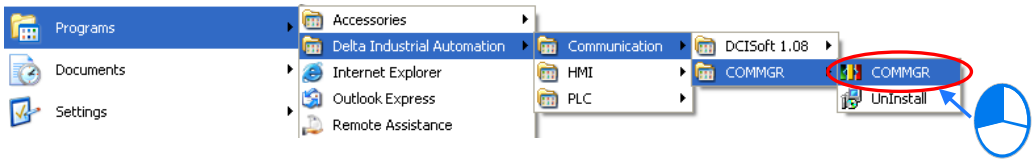
Those who have connected ISPSoft to a PLC successfully in accordance with the contents of section 2.4 can skip this section.

(1) Install the modules on the main backplane in accordance with the hardware configuration in HWCONFIG. Make sure that the wiring is correct, and then power the CPU module.

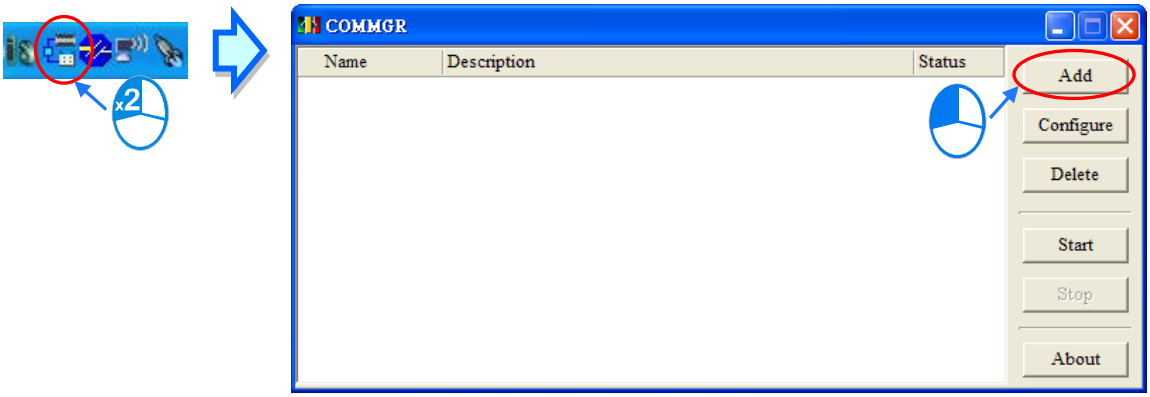
(2) Connect the CPU module to the computer through a USB cable. If the USB driver for the AH500 series CPU module has been installed on the computer, **Delta PLC** will appear in the **Device Manager** Window, and a port number will be assigned to **Delta PLC**. Please refer to appendix A for more information about installing a USB driver.

(3) Make sure that COMMGR is started, and the icon representing COMMGR is displayed on the system tray. If the icon representing COMMGR is not displayed on the system tray, users can start COMMGR by clicking the shortcut on the **Start** menu (**Start**>**Programs**>**Delta Industrial**

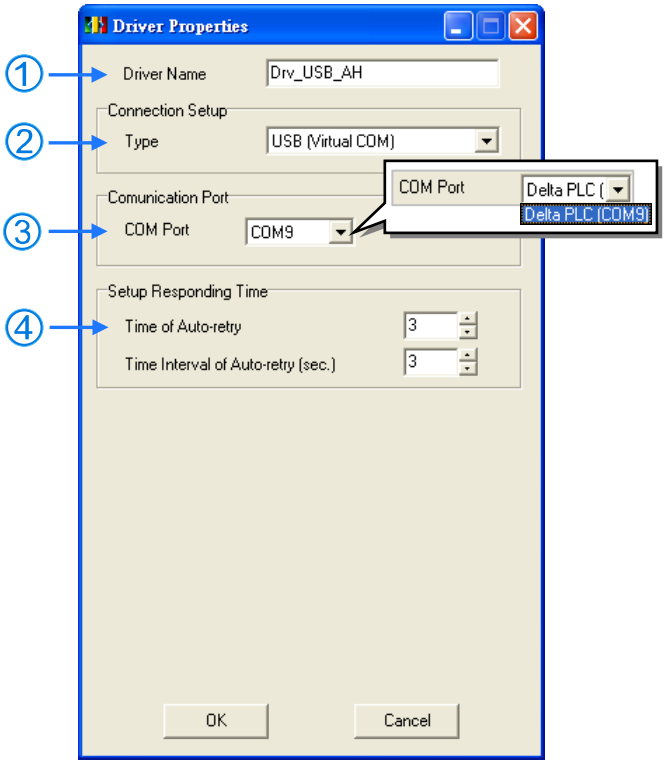
Automation>Communication>COMMGR).



(4) Double-click the icon representing COMMGR on the system tray to open the **COMMGR** window. Click **Add** in the **COMMGR** window to create a driver.



(5) Set the parameters in the **Driver Properties** window, and then click **OK**.



① Type a driver name in the **Driver Name** box.

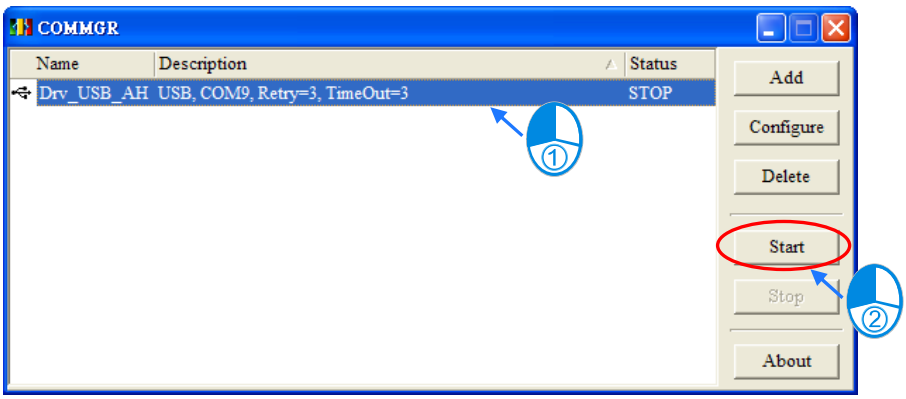
② Select **USB (Virtual COM**) in the **Type** drop-down list box in the **Connection Setup** section.

③ Select a communication port in the **COM Port** drop-down list box. If the first two steps are complete, the PLC which is connected and its communication port will be displayed in the **COM Port** drop-down list box.

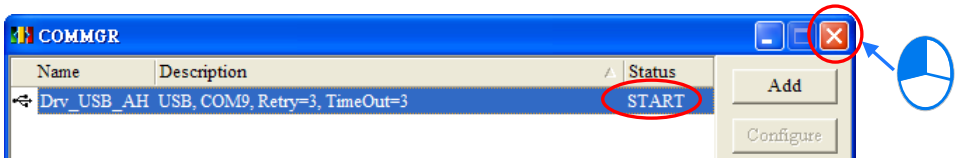
④ Users can select the number of times the sending of a command is retried if a connection error occurs in the **Time of Auto-retry** box, and select an interval of retrying the sending of

a command in the **Time Interval of Auto-retry** box.

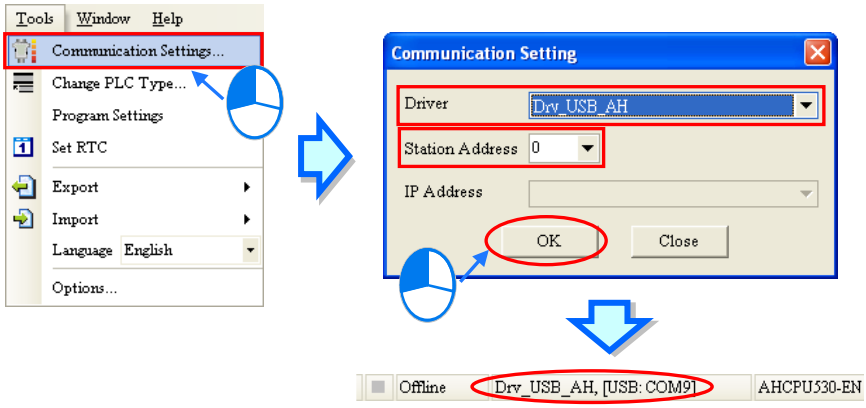
(6) Click the driver created in the **COMMGR** window, and then click **Start**.



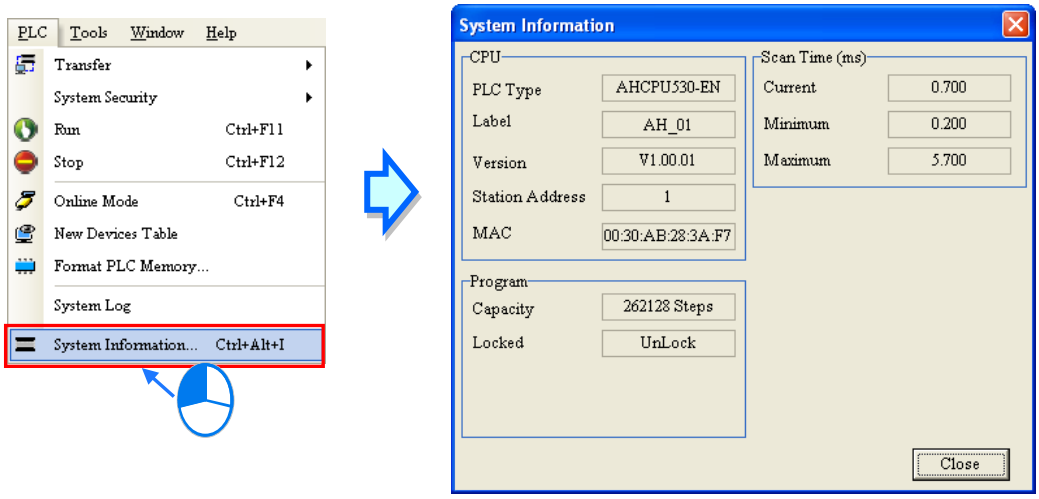
(7) After the status of the driver displayed in the window becomes **START**, the window can be closed. The icon representing COMMGR will still be displayed on the system tray.



(8) Start ISPSoft, and then click **Communication Settings…** on the **Tools** menu. In the **Communication Setting** window, select the driver which has been created in the **Driver** drop-down list box, appear, and select 0 in the **Station Address** drop-down list box, and click **OK**. The information about the driver will be displayed in the status bar in ISPSoft.



(9) Click **System Information** on the **PLC** menu. ISPSoft will retrieve related information from the PLC. If the computer communicates with the CPU module normally, the related information retrieved from the PLC will be displayed in the **System Information** window.

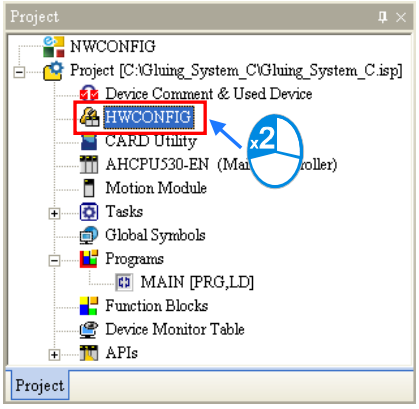


4.6.2 Downloading a Program and Parameters

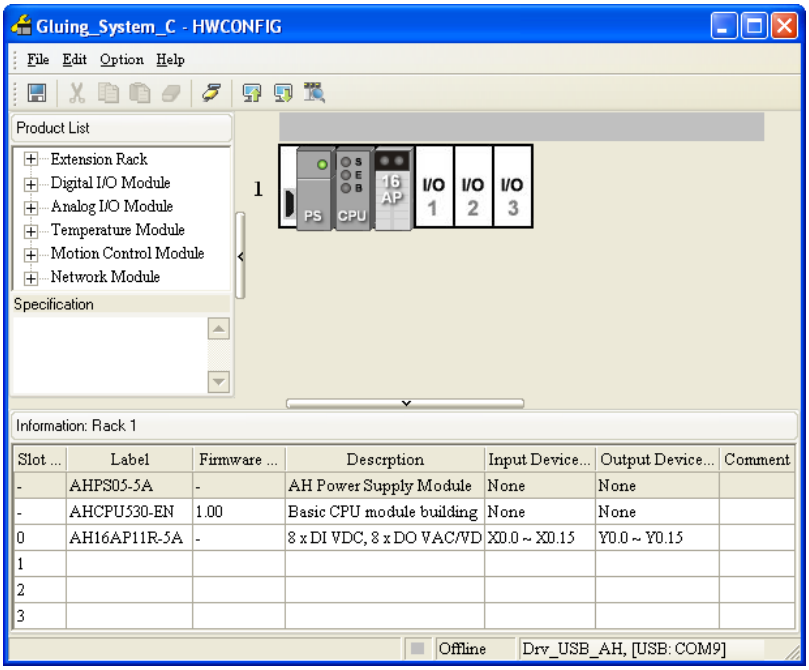
If ISPSoft is connected to a PLC normally, the parameters and the program in the project can be downloaded to the PLC. First, start ISPSoft and open the project created in the previous sections. In this example, two types of parameters are downloaded to the CPU module. They are the hardware configuration and the program.

- **Downloading the hardware configuration**

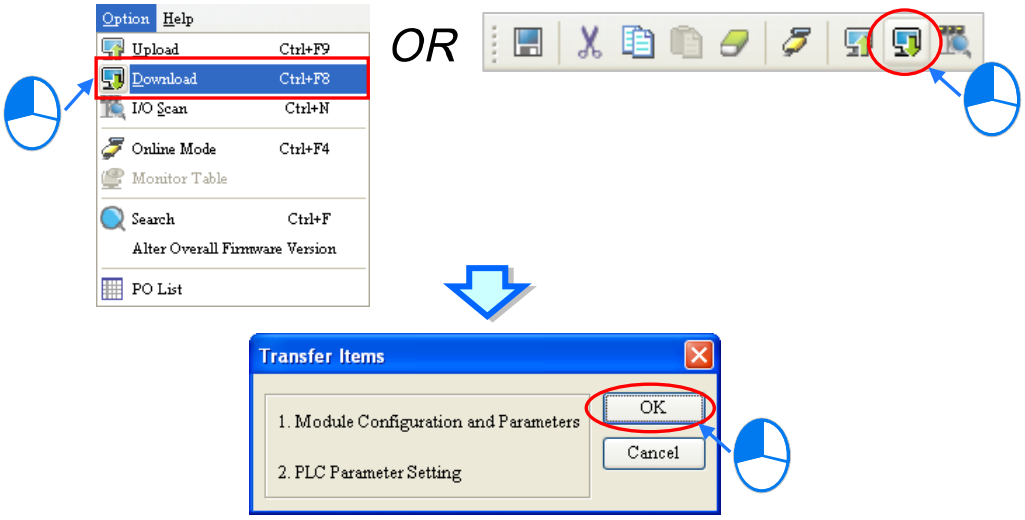
(1) Double-click **HWCONFIG** in the project management area to open the **HWCONFIG** window.



(2) The hardware configuration is displayed in the window. Before the hardware configuration is downloaded to the CPU module, users have to make sure that the actual hardware configuration is the same as the hardware configuration in the window.



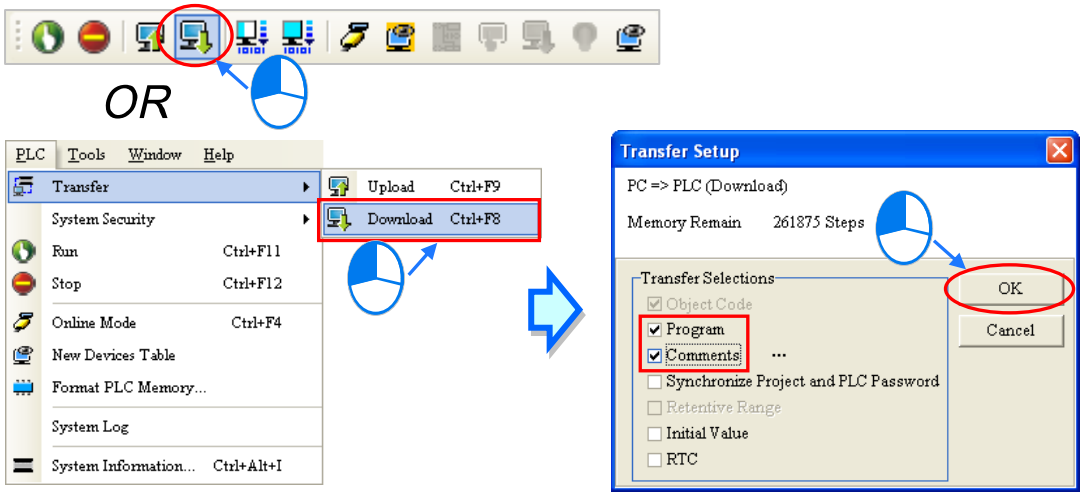
(3) After the users click **Download** on the **Option** menu, or on the toolbar, the **Transfer Items** window will appear. The hardware configuration will be downloaded to the CPU module after **OK** is clicked.



(4) After the hardware configuration is downloaded to the CPU module successfully, the BUS FAULT LED indicator on the CPU module will be OFF. The users can close the **HWCONFIG** window. If the BUS FAULT LED indicator on the CPU module is still ON or blinking, the CPU module is in an abnormal state. Please make sure that the actual hardware configuration is the same as the hardware configuration in the **HWCONFIG** window again, or refer to the operation manual for more information about eliminating the error. Please refer to chapter 3 for more information about HWCONFIG.

- **Downloading the program**

  After the program is compiled successfully, the users can click the **PLC** menu, point to **Transfer**, and click **Download**. The users can also click on the toolbar after the program is compiled successfully. Select the **Program** checkbox and the **Comments** checkbox in the **Transfer Setup** window so that the program in the CPU module can be uploaded later, and then click **OK**.



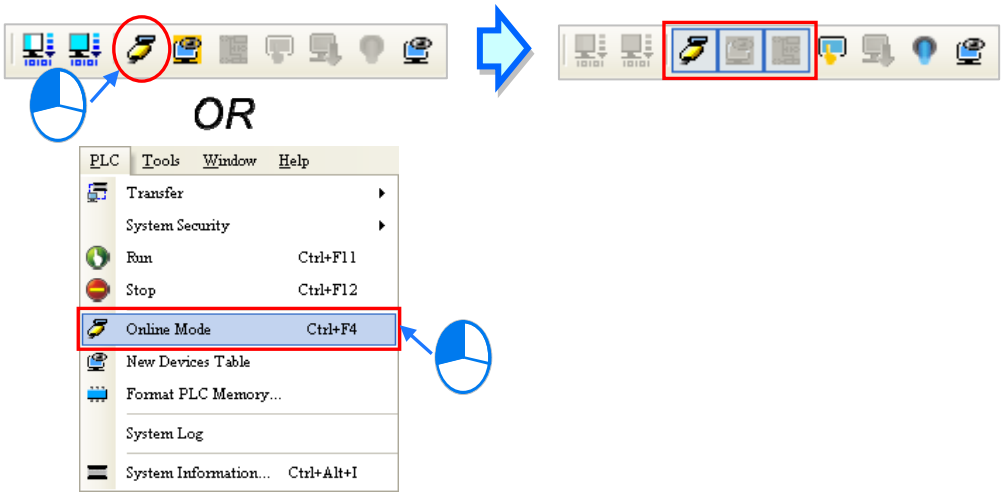
4.6.3 Connection Test

After a program is downloaded to a PLC, users can monitor the execution status of the PLC through ISPSoft. There are two monitoring modes that ISPSoft provide. One is the device monitoring mode, and the other is the program monitoring mode.

Monitoring mode	Description
Device monitoring mode	Users can monitor the statuses of the devices in the PLC through the monitoring table. In this mode, ISPSoft only needs to update the statuses of the devices. The present program in ISPSoft does not need to be the same as the program in the PLC.
Program monitoring mode	In this mode, the operating status of the program is displayed in the program editing window. As a result, the present program in ISPSoft must be the same as the program in the PLC.

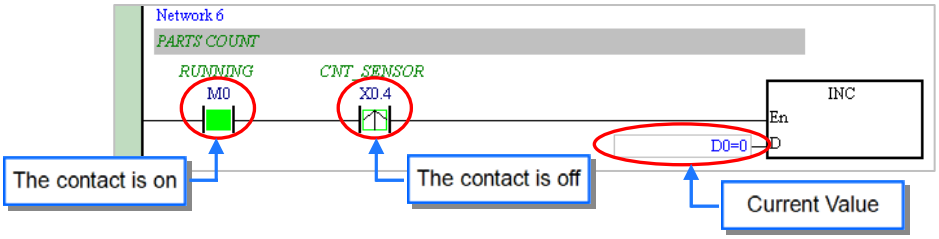
***. The device monitoring function can be enabled independently. However, if the program monitoring function is enabled, the device monitoring function is also enabled.**

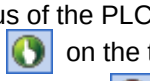
After users click **Online Mode** on the **PLC** menu, or on the toolbar, the online monitoring function will be enabled. The system will also enable the device monitoring mode and the program monitoring mode.

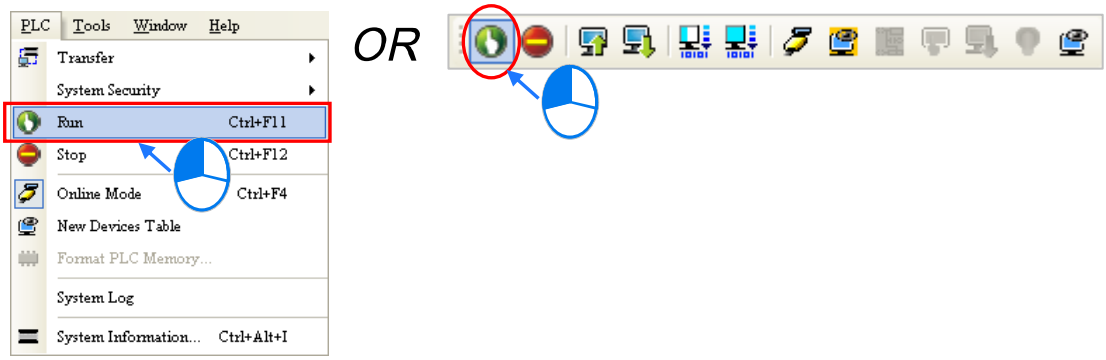


In the online monitoring mode, users can view the present scan time, the communication status, and the status of the PLC in the status bar in ISPSoft.

Besides, the present statuses of the devices will be displayed in the original program editing window after the program monitoring function is enabled.

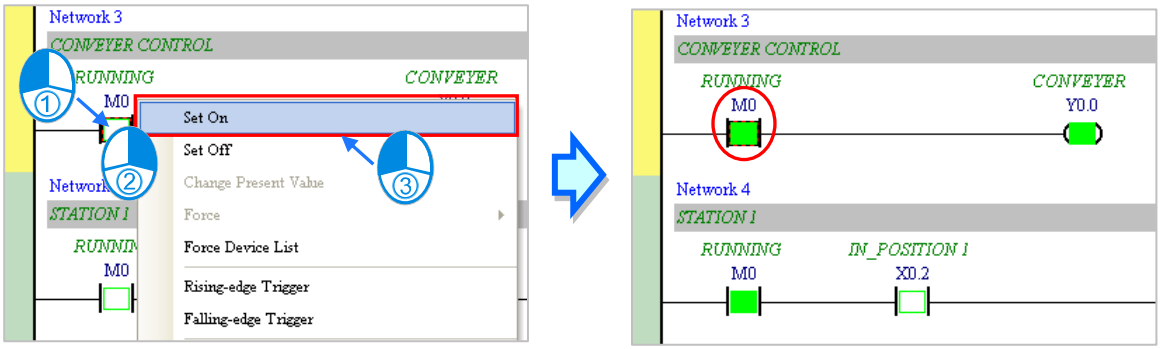


Users can change the operating status of a PLC by the RUN/STOP switch on the PLC. They can also change the operating status of the PLC through the functions provided by ISPSoft. After users click **Run** on the **PLC** menu or  on the toolbar, the PLC will begin to run. The PLC will stop running after **Stop** on the **PLC** menu or on the toolbar is clicked.



In the online monitoring mode, users can select a device, right-click the device, and click an item on the context menu. During a test, users can change the status of a device or the value in a device by clicking an item on the context menu.

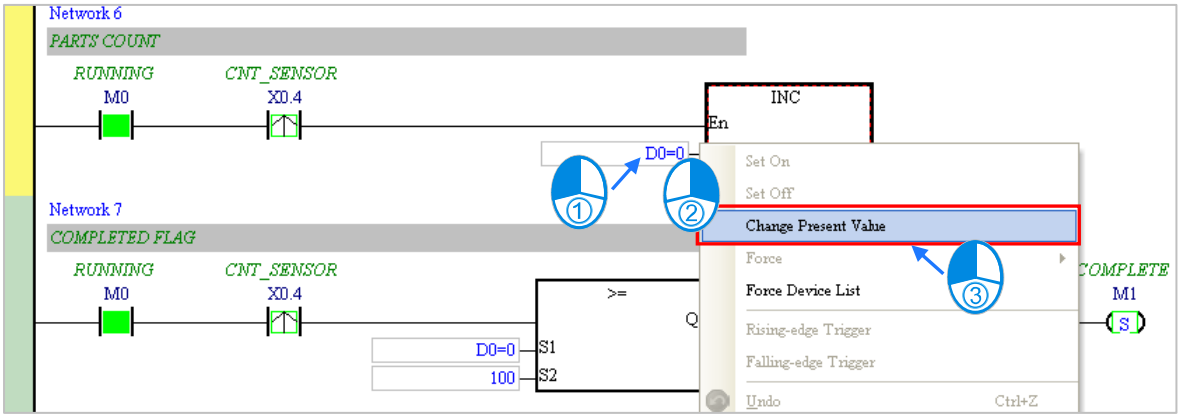
Before the status of a device is changed, users have to make sure that the operation does not cause damage to the system or staff.



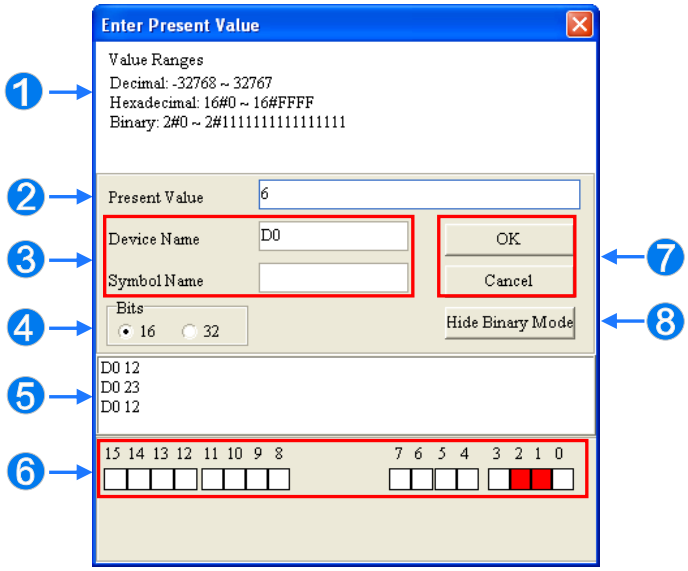
The items on the context menu are described below. **Force** on the context menu only applies to input contacts and output contacts.

Item	Description
Set On	Setting the contact selected to ON
Set Off	Setting the contact selected to OFF
Rising-edge Trigger	No matter what the state of the contact selected is, the system set the contact to OFF, and then set it to ON.
Falling-edge Trigger	No matter what the state of the contact selected is, the system set the contact to ON, and then set it to OFF.
Force	Forcing an input contact or output contact ON or OFF
Force Device List	Forcing several input contacts or output contacts in the tables ON or OFF

If users want to change the value in a device, they can click the device, right-click the device, click **Change Present Value** on the context menu, and set a present value in the **Enter Present Value** window.

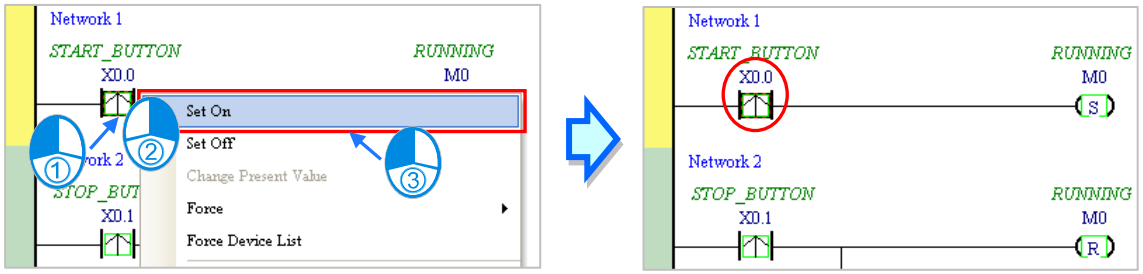


The **Enter Present Value** window is described below.

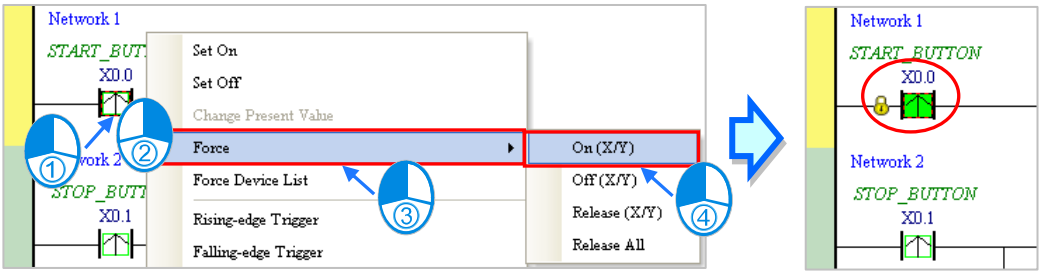


1. Message
2. Users can type a value in the **Present Value** box.
3. Name of a device or a symbol whose present value will be changed
4. Users can type a 16-bit value or a 32-bit value.
5. Value change history (Format: Device name Value)
6. In the binary mode, users can set the states of the bits through the mouse.
7. The setting values will be applied after **OK** is clicked. The window will be closed after **Cancel** is clicked.
8. Users can display or hide the binary mode.

In this example, X0.0~X0.15 and Y0.0~Y0.15 are input devices and output devices assigned to the digital I/O module AH16AP11R-5A. After the parameters in the hardware are downloaded to the CPU module, the states of X0.0~X0.15 will be the same as the states of the inputs on the actual module. Even if users set X0.0~X0.15 to ON or OFF in the program editing window, the states of X0.0~X0.15 will be updated by the actual input signals.



However, an input contact can be forced ON or OFF during a test. Users can click an input contact or output contact which will be set, right-click the contact, point to **Force** on the context menu, and select **On (X/Y)**, **Off (X/Y)**, **Release (X/Y)**, or **Release All**. If an input contact or output contact is forced ON or OFF, a lock symbol will appear at the left side of the contact.



Force	Description
On (X/Y)	Forcing the input contact or output contact selected ON
Off (X/Y)	Forcing the input contact or output contact selected OFF
Release (X/Y)	Releasing the contact from the locked state
Release All	Releasing all the contacts from the locked states

If an output contact in the program is forced ON or OFF, the output state of this contact will not be affected by the program execution result.

***. If the online monitoring function is disabled, the contacts will not be automatically released from the locked states. As a result, users have to check whether the contacts need to be released from the locked states after the test is complete.**

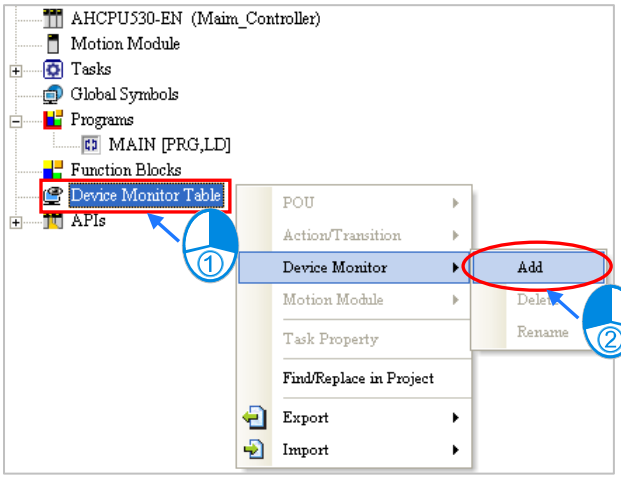
There are two ways to create a monitoring table. Users can create a monitoring table online or offline.

- **Method 1**

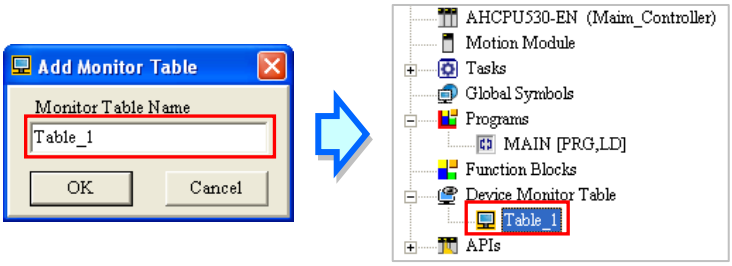
  Click **New Devices Table** on the **PLC** menu, or on the toolbar.

- **Method 2**

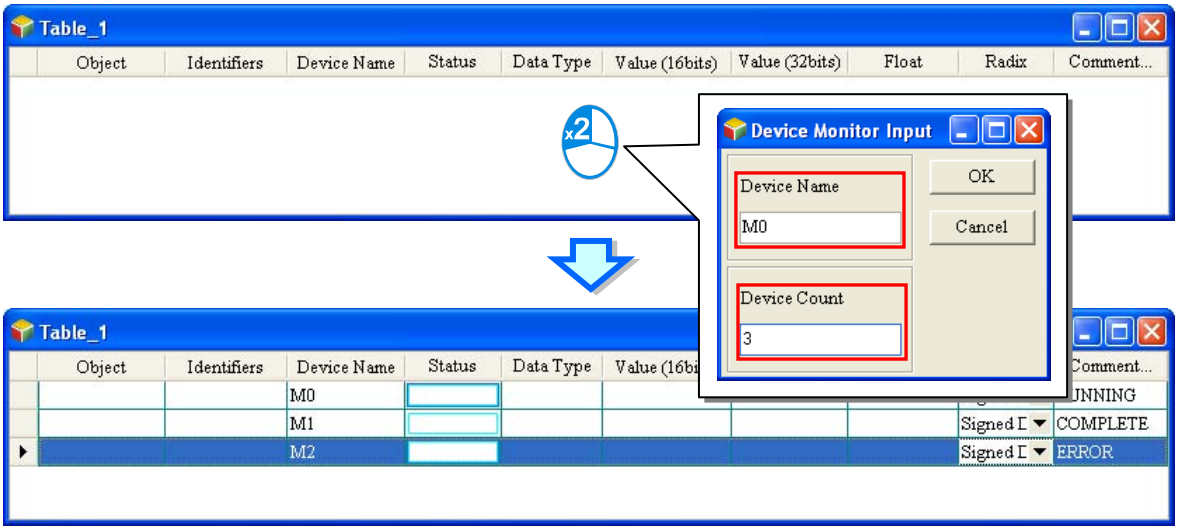
Right-click **Device Monitoring Table** in the project management area, point to **Device Monitor** on the context menu, and click **Add**.



Type a table name in the **Add Monitor Table** window, and then click **OK**. An item will be under **Device Monitor Table** in the project management area. If users want to open the monitoring table, they can double-click the item. Besides, the users can create several monitoring tables in the project, and the monitoring tables created will be saved with the project.



After the item is double-clicked, a window will appear. The users can add items which will be monitored to the window. If the users want to add an item to the window, they have to double-click the blank in the monitoring table, or type a device name directly, and type a start address and the number of devices which will be monitored in the **Device Monitor Input** window. Please notice that 100 items at most can be added to a monitoring table.

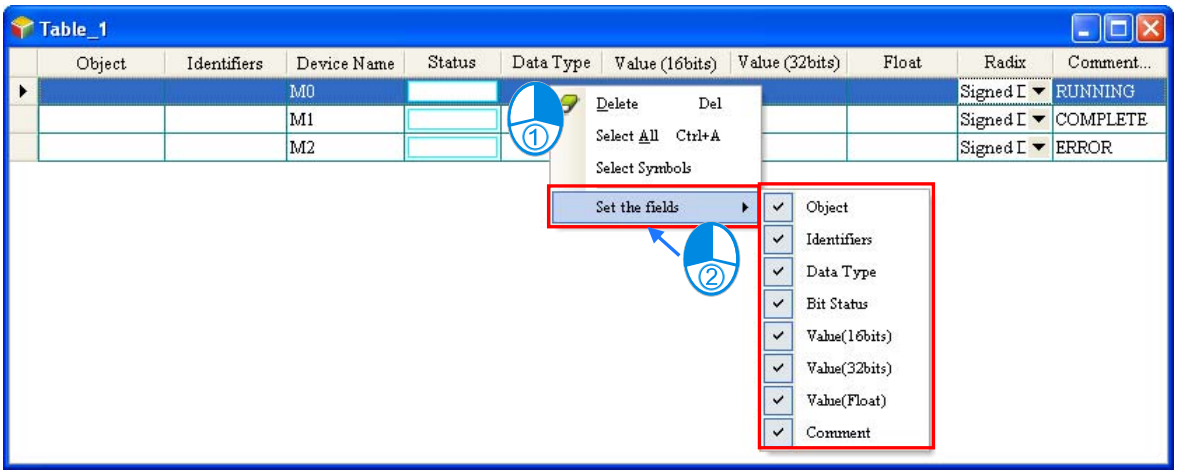


The users can press Insert on the keyboard to switch between inserting an item in the monitoring table and replacing an item in the monitoring table. The mode which is selected is displayed in the status bar in ISPSoft.

If the insertion mode is selected, the item added will be above the item selected in the monitoring

table. If the replacement mode is selected, the item added will overwrite the item selected in the monitoring table.

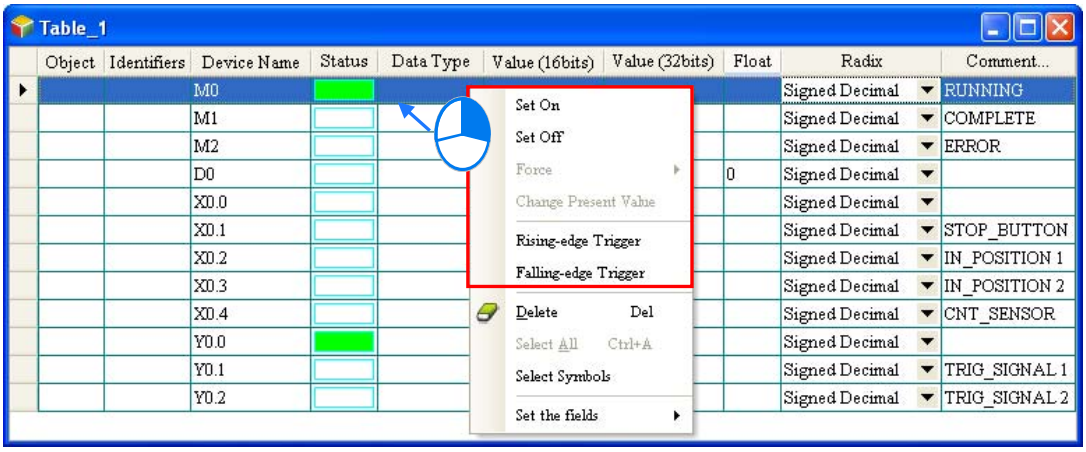
If the users want to hide certain columns in the monitoring table, they can right-click the monitoring table, point to **Set the Fields**, and unselect certain items. After an item is unselected, the corresponding column will disappear.



The description of the columns in the monitoring table is as follows.

Column	Description
Source	The source of a symbol
Identifier	The identifier of a symbol
Device name	The name of a device monitored
Status	If a bit device or a contact is monitored, the state will be ON or OFF.
Data type	If a symbol is monitored, the data type of the symbol will be displayed.
Value (16 bits)	In the online mode, a 16-bit value is displayed.
Value (32 bits)	In the online mode, a 32-bit value is displayed.
Float	In the online mode, a 32-bit floating-point number is displayed.
Radix	Users can select a format in which a value is represented.
Comment	The comments on a device or the comment on a symbol is displayed.

After the monitoring table is created, the users can monitor the items in the monitoring table in the online mode. Besides, after the users right-click an item in the monitoring table in the online mode, a context menu which is the same as the context menu which will after a device in the program editing window is clicked will appear. The users can change the state of the item or the value in the item by clicking an item on this context menu.



The program created in this chapter can be tested and debugged through the monitoring table created in this section. Please refer to chapter 14 for more information about testing and debugging a program.

MEMO

Chapter 5 Program Organization Units and Tasks

Table of Contents

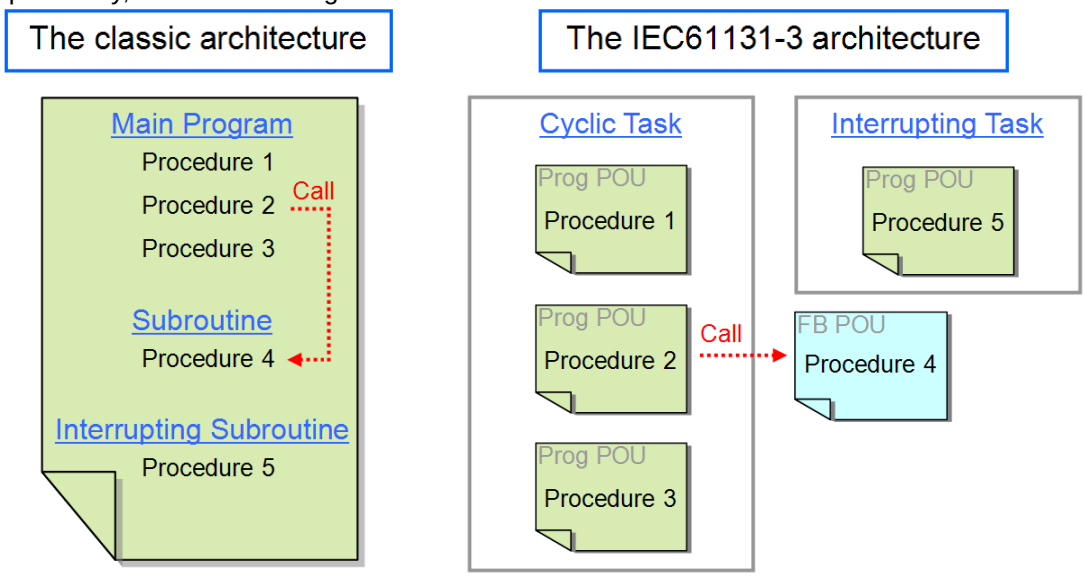
5.1 Program Organization Units and Tasks 5-2
5.2 Program Organization Units 5-2
5.2.1 Program Architecture and Types 5-2
5.2.2 POUs in ISPSoft 5-3
5.3 Tasks 5-4
5.3.1 Managing the Tasks in ISPSoft 5-4
5.3.2 Tasks in the Project Management Area 5-5
5.3.3 Executing the POUs Assigned to a Task 5-6
5.4 Managing a POU 5-8
5.4.1 Creating and Setting a POU 5-8
5.4.2 Enabling a POU 5-11
5.4.3 Deleting and Copying a POU 5-12
5.4.4 Setting and Unlocking a POU Password 5-13
5.4.5 Exporting a POU of the Program Type 5-14
5.4.6 Importing a POU of the Program Type 5-15
5.4.7 Exporting a POU of the Function Block Type 5-16
5.4.8 Importing a POU of the Function Block Type 5-17
5.5 Managing Tasks 5-18
5.5.1 Setting the Property of a Task and the Condition of an Interrupt 5-20
5.5.2 Assigning a POU to a Task 5-22
5.5.3 Arranging the POUs 5-23
5.6 Example 5-24
5.6.1 Writing a Program by Means of IEC 61131-3 5-24
5.6.2 Example of an Interrupt Subroutine in a DVP Series PLC 5-28
5.6.3 Example of an Interrupt Subroutine in an AH500 Series CPU Module 5-34

5.1 Program Organization Units and Tasks

Program organization units (POUs) and tasks are important programming concepts related to IEC 61131-3. They upgrade programming from traditional program writing to project management. The difference between traditional ladder diagrams and IEC 61131-3 is described below.

Traditional ladder diagram	IEC 61131-3
Main program	1. A single program is divided into several independent program organization units. 2. A program organization unit which has been created is assigned to a cyclic task.
General Subroutine	1. A subroutine is a function block. 2. A function block which has been created can be called by a program organization unit.
Interrupt subroutine	1. Different interrupt subroutines are independent program organization units. 2. A program organization unit which has been created is assigned to an interrupt task.

Traditional ladder diagrams and IEC 61131-3 are represented by the following architectures. In the classic architecture, users write a source code. In the IEC 61131-3 architecture, a program is divided into several program organization units. Every program organization unit can be developed independently, and can be assigned a task.



If a control system reaches a certain scale, the internal control procedures will be quite complex, and the relation between the procedures will be closer. If the classic architecture is adopted, a lot of time and manpower must be invaded in developing and maintaining a program. However, in the IEC 61131-3 architecture, the program organization units are reusable and modular. Besides, the program organization units can be integrated easily by means of the management of the tasks. The concepts related to program organization units and tasks will be introduced in the following section.

5.2 Program Organization Units

5.2.1 Program Architecture and Types

Program organization units are basic elements of a program in a PLC. They are different from a traditional program in a PLC. The characteristic of the program architecture introduced by IEC 61131-3 is that a program is divided into several units. These units are called program organization

units.

In the classic architecture, a source code for a PLC is composed of all procedures, including subroutines. If the size of a program becomes larger, the maintenance of the program and the debugging of the program will be a burden. In the IEC 61131-3 architecture, a program is divided into several units according to the functions or characteristics. It is convenient to develop and maintain a program. Besides, owing to the fact that program organization units are modular, different program organization units can be developed by different designers. It benefits the distribution of manpower and the execution of the project.

There are two types of POUs in ISPSoft. They are programs (PROGs) and function blocks (FBs).

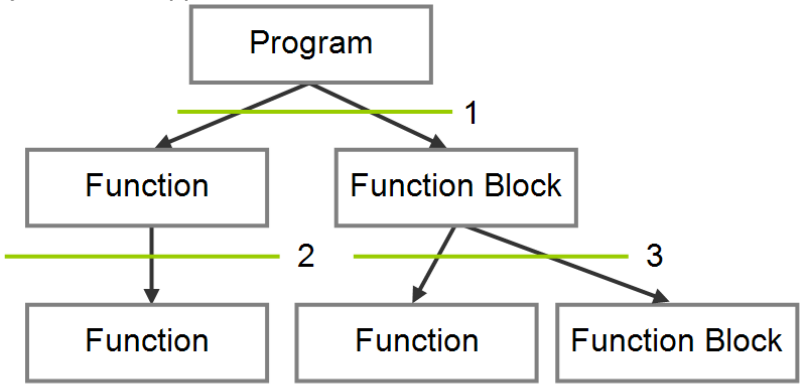
- **Program (PROG)**

  A POU of the program type plays a role according to the task to which it is assigned. If a program POU is assigned to a cyclic task, the POU of the program type acts as a main program. If the POU of the program type is assigned to an interrupt task, the POU of the program type acts as a subroutine. Besides, a POU of the program type can call a function block (FB).

- **Function block (FB)**

  A static symbol can be declared in a function block (FB). As a result, the value of the symbol after an operation can be retained. Owing to the fact that the operation is performed on the value memorized in the function block and an input value, the output values may be different even if the input values are the same. Besides, a function block can call another function block.

The relation among POUs is shown below. A function (FC) is a type of POU. Owing to the fact that ISPSoft presently does not support functions, functions will not be introduced.

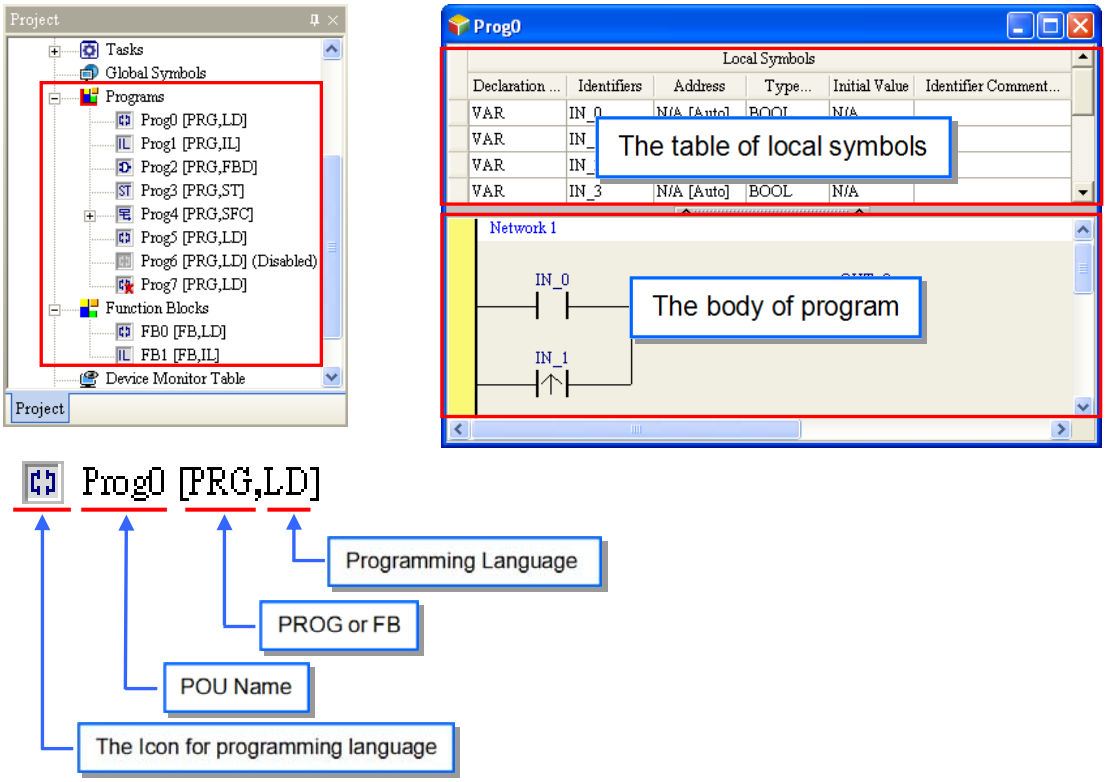


1. PROG calls FB or FC
2. FC calls FC
3. FB calls FC or FB

5.2.2 POUs in ISPSoft

All POUs created by users are listed in the program management area. The POUs of the program type and the POUs of the function block type are managed separately. Besides, the programming language in which a POU is written determines the icon representing the POU, and the information about the POU is put at the end of the POU name.

After a POU in the project management area is double-clicked, a POU editing window will appear. The POU editing window is composed of two parts. A table of local symbols is at the upper part of the window, and the body of a program is at the lower part of the window. Besides, the editing environment at the lower part of the window varies with the programming language used. Please refer to the following sections for more information about symbol tables and program editing.



The icons representing POUs are described below.

Icon	Description
	The POU is a ladder diagram.
	The POU is an instruction list.
	The POU is a function block diagram.
	The POU is a structured text.
	The POU is a sequential function chart.
	If the icon representing a POU is gray, the POU is disabled. A POU which is disabled is skipped when the program is compiled, and is not executed.
	If the icon representing a POU is marked with a red cross, the POU is not assigned to any task. A POU which is not assigned to any task is skipped when the program is compiled, and is not executed.

***. Please refer to section 5.4.2 for more information about enabling a POU. Please refer to the following section for more information about tasks.**

5.3 Tasks

Every POU of the program type is assigned to a specific task. The tasks determine the order in which the POUs of the program type are executed or enabled. In other words, every POU of the program type is like a basketball player, and the tasks are the roles that the POUs play.

5.3.1 Managing the Tasks in ISPSoft

In ISPSoft, not all the POUs of the program type in the project are executed. A POU of the program type will be executed after it is assigned to a task, and the task determined how the POU is executed. If a POU is not assigned to a task, the POU will be taken as a general source code, will be saved with the project, and will not be translated into an execution code. Besides, only POUs of the program type need to be assigned to tasks, and function blocks are called by POUs.

The characteristic of the IEC 61131-3 architecture is that a program is divided into several POUs which can be developed independently. When all POUs are compiled, they are rearranged and combined into an execution code which can be scanned step by step. The POUs are rearranged and combined according to the tasks to which the POUs are assigned.

In ISPSoft, a POU of the program type is assigned to a task. However, more one POU can be assigned to the same task, and the order in which these POUs are executed can be specified. Besides, after users add a POU of the program type, they must assign it to a default task.

There are three types of tasks. They are cyclic tasks, timed interrupt tasks, and conditional interrupt tasks.

- **Cyclic task**

  A POU assigned to a cyclic task is scanned back and forth. Owning to the fact that there is only one cyclic task in a project for a DVP series PLC, the POUs which will be scanned cyclically are assigned to this cyclic task. However, there are 32 cyclic tasks in a project for an AH500 series CPU module. These cyclic tasks are numbered from 0 to 31. The smaller the task number is, the earlier the task is executed. Besides, AH500 series CPU modules support the applied instructions (TKON and TKOFF) which control tasks. Users can enable or disable a task during the operation of an AH500 series CPU module.

  A POU is which a sequential function chart can only be assigned to a cyclic task. It can not be assigned to an interrupt task.

- **Timed interrupt task**

  A POU of the program type which is assigned to a timed interrupt task is like a timed interrupt subroutine. If the time when timed interrupts should occur is reached, the POUs assigned to the timed interrupt task will be executed in order. Besides, the number of timed interrupt tasks depends on a PLC selected. The number of interrupt sources that a PLC supports is the number of timed interrupt tasks that the system has.

- **Conditional interrupt task**

  There are several types of conditional interrupt tasks. For example, external interrupts, I/O interrupts, counting interrupts, and etc. Different PLCs provide different conditional interrupt tasks. Users have to make sure of the conditional interrupt tasks which are supported by the PLC before they assign the POUs of the program type to tasks. A POU of the program type which is assigned to a conditional interrupt task is like an interrupt subroutine. If the condition of an interrupt is met, e.g. the value of a counter reaches the setting value, the POUs assigned the interrupt task will be executed in order.

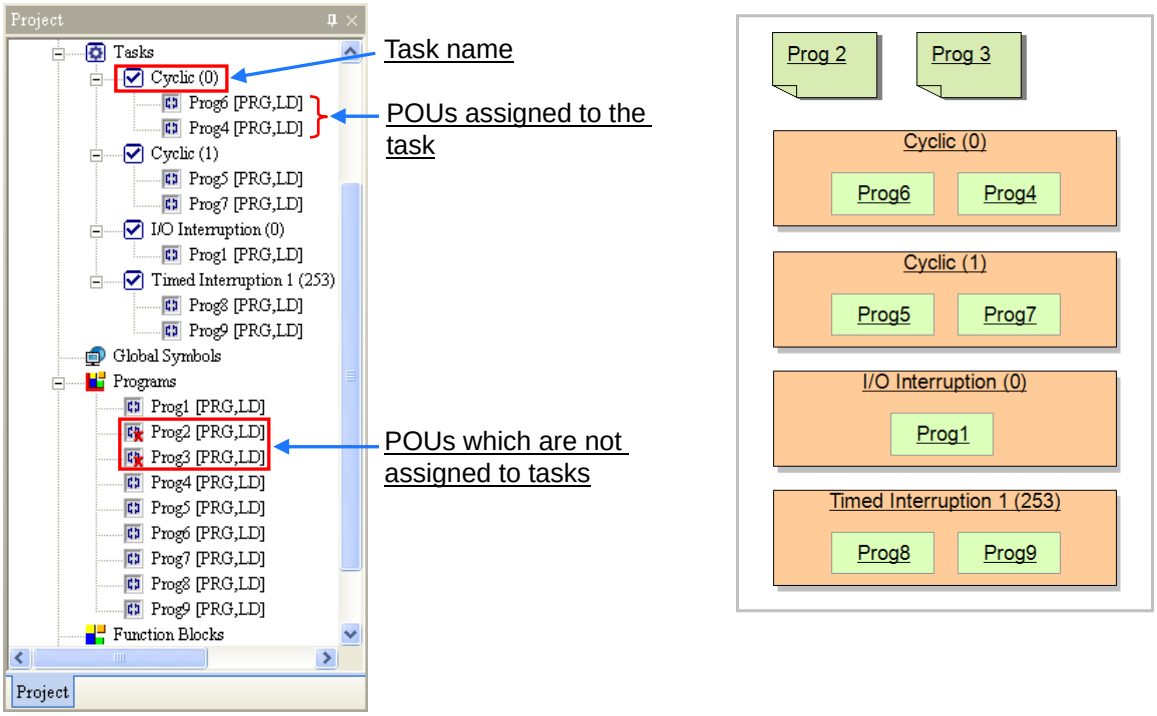
5.3.2 Tasks in the Project Management Area

The POUs which are assigned to tasks are listed in the **Tasks** section in the project management area, and the POUs which are not assigned to tasks are not listed in the **Tasks** section. The order in which the POUs listed in a task section are arranged is the order in which the POUs are executed.

Take the project for an AH500 series CPU module below for example. There are 9 POUs of the program type. The assignment of the POUs of the program type to tasks is as follows.

Task name	POU	Execution
Not assigned	Prog2 and Prog3	Owning to the fact that Prog2 and Prog3 are not assigned to tasks, Prog2 and Prog3 are not executed.
Cyclic (0)	Prog6 and Prog4	The smaller the task number is, the earlier the task is executed. As a result, the order in which Prog6, Prog4, Prog5, and Prog7 are scanned is Prog6→Prog4→Prog5→ Prog7.
Cyclic (1)	Prog5 and Prog7	
I/O interrupt (0)	Prog1	If the condition of the interrupt is met, Prog1 will be executed once.
Timed interrupt 1 (253)	Prog8 and Prog9	Timed interrupt 1 is triggered every specific period of time. The order in which Pog8 and Prog9 are executed is Prog8→Prog9.

***. The setting of the condition for an interrupt will be introduced in section 5.5. Please refer to instructions for PLCs for more information.**



Additional remark

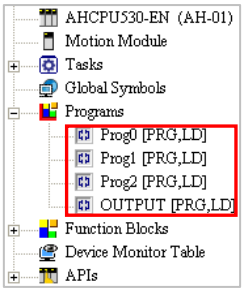
If users want to enable an interrupt function, they must insert the applied instruction EI in the cyclic program so that the interrupt subroutine can be executed normally.

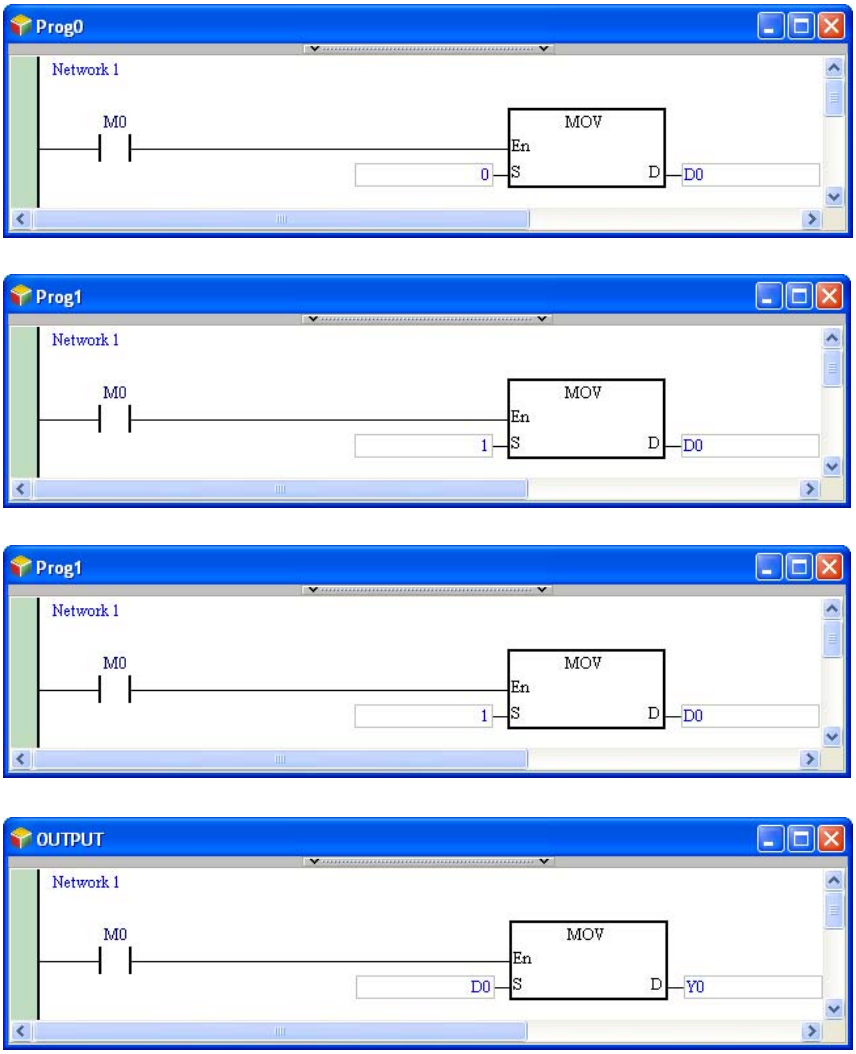
5.3.3 Executing the POUs Assigned to a Task

More one POU can be assigned to the same task, and the order in which the POUs listed in a task section in the project management area are arranged is the order in which the POUs are executed. Please refer to the following example. There are four POUs in the project. These POUs are described below.

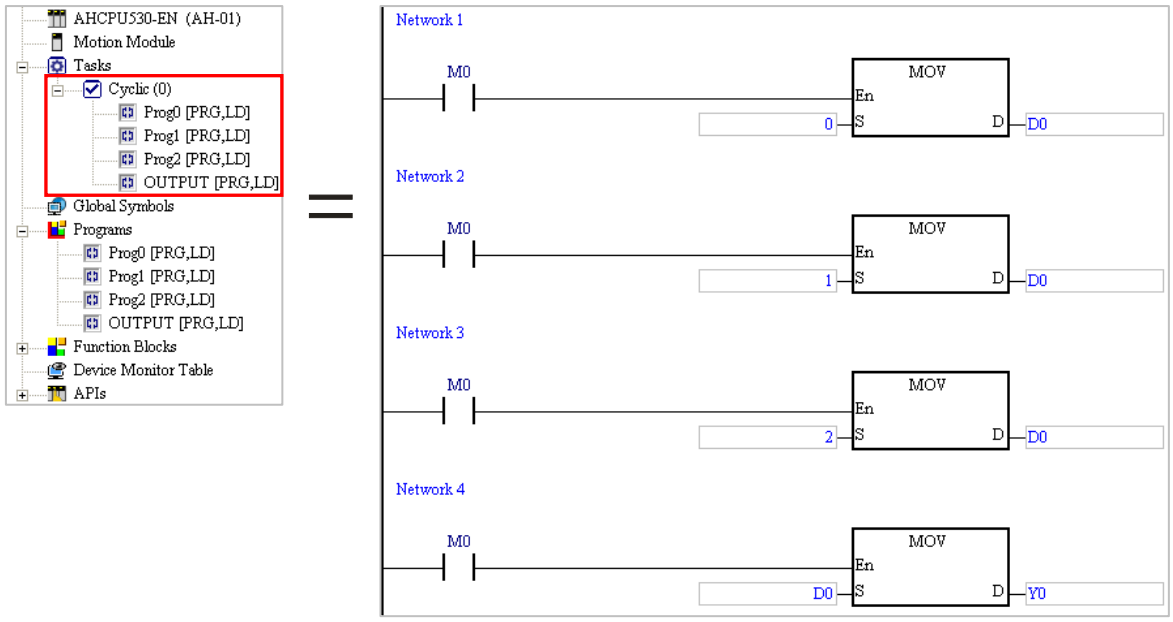
POU name	Description
Prog0	When M0 is ON, the value in D0 is 0.
Prog1	When M0 is ON, the value in D0 is 1.
Prog2	When M0 is ON, the value in D0 is 2.
OUTPUT	When M0 is ON, the value in Y0 is the value in D0.

The programs in the POUs are as follows.

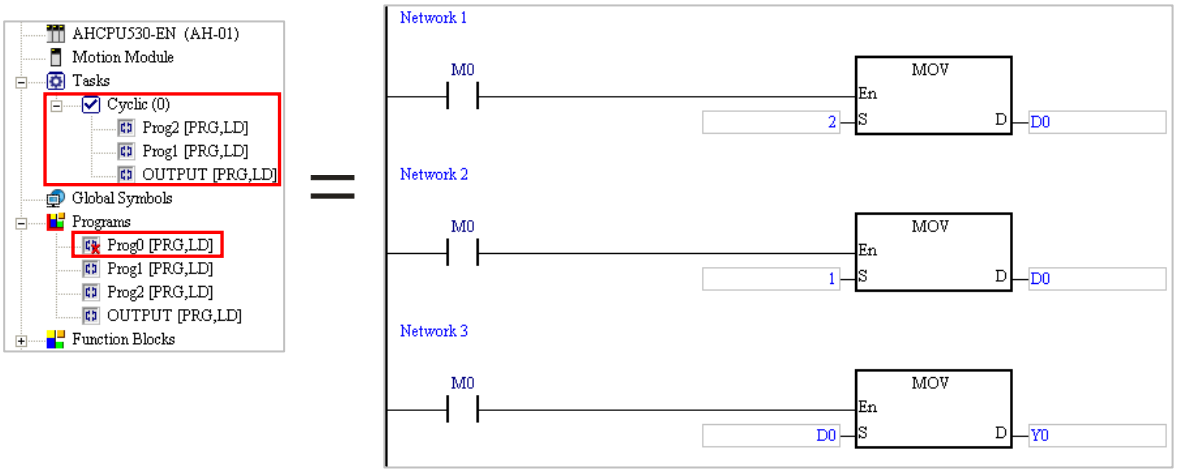




Suppose the four POUs of the program type are assigned to cyclic task (0). The arrangement of the POUs is shown below. The order in which these POUs are executed is Prog0→Prog1→Prog2→OUTPUT. The value in D0 in OUTPUT depends on the program in Prog2. When M0 is ON, the value in D0 in OUTPUT is 2. After the value in D0 is transferred to Y0, the value in Y0 will be 2. In other words, Y0.1 will be ON after a scan cycle is complete. The equivalent program is shown below.



If Prog2 is moved above Prog1, and Prog0 is removed from the **Cyclic (0)** section, the program in Prog0 will not be translated into an execution code, and the order in which the POUs are listed in the **Cyclic (0)** section will be Prog2→Prog1→OUTPUT. The value in D0 in OUTPUT depends on the program in Prog1. When M0 is ON, the value in Y0 is 1. In other words, Y0.0 will be ON after a scan cycle is complete.

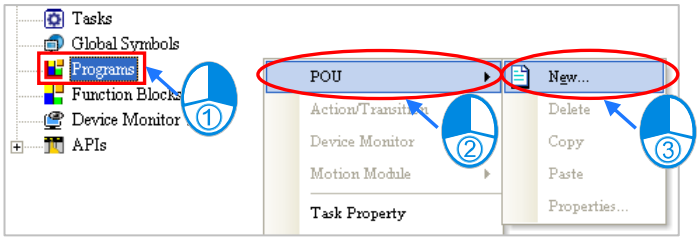


5.4 Managing a POU

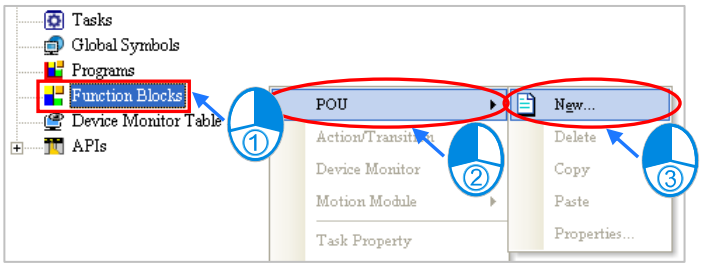
5.4.1 Creating and Setting a POU

(1) If users want to create a POU of the program type, they have to right-click **Programs** in the project management area, point to **POU** on the context menu, and click **New…**. If users want to create a POU of the function block type, they have to right-click **Function Blocks** in the project management area, point to **POU** on the context menu, and click **New…**.

● Adding a POU of the program type

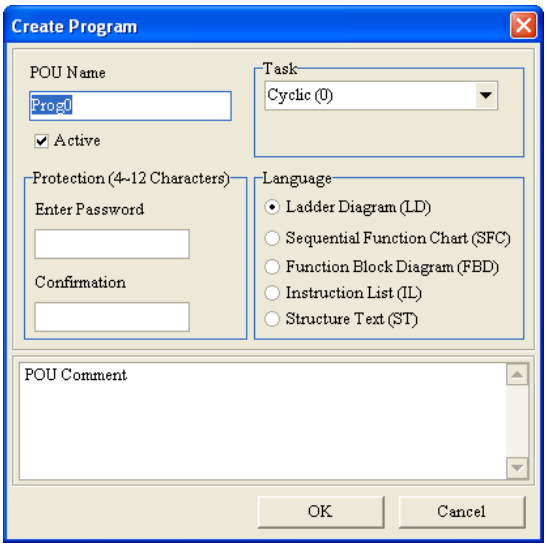


● Adding a POU of the function block type

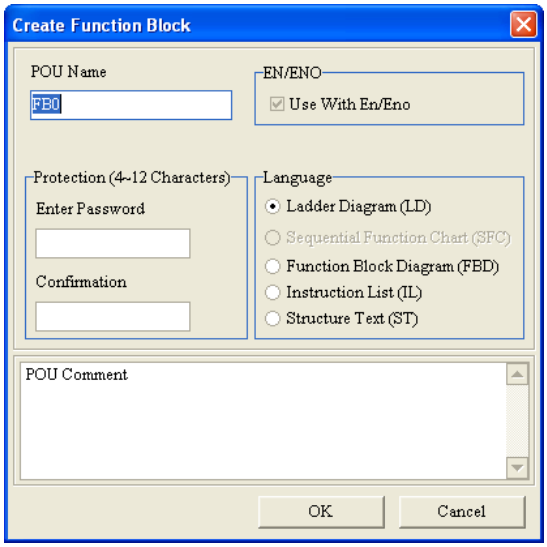


(2) The users have to set the attributes of the POU added in the window which appears. The attributes of the POU of the program type are slightly different from those of the POU of the function block type, and they are described below.

● Program (PROG)



● Function Block (FB)



● **POU Name**: Users can type a POU name in this box.

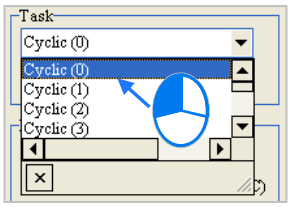
(a) Special marks and spaces can not be used. For example, *, #, ?, \, %, @, and etc. can not be used.

(b) A POU name is composed of thirty characters at most. Users have to notice that a Chinese character occupies two characters.

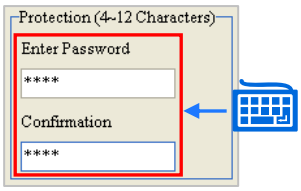
(c) Underlines can be used, but they can not be used continuously or put at the end of a POU name. For example, “POU_1” is a legal name, but “POU__1” and “POU_1_” are illegal names.

(d) If the POU created is a POU of the function block type, and the POU name begins with “P0_” or “P1_”, P0 or P1 will be assigned to the function block after the program is compiled.

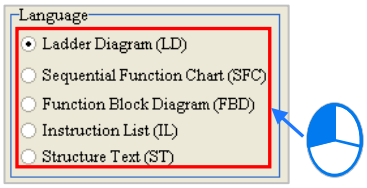
● **Task**: If the POU created is a POU of the program type, users have to assign the POU to a task.



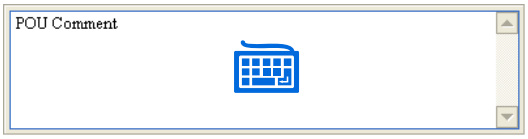
- **Password**: If users want to set a password, they have to type the same passwords in the two boxes. English letters, numerical digits, and special marks can be typed in the boxes.



- **Language**: Users have to select a programming language in which the POU is written.

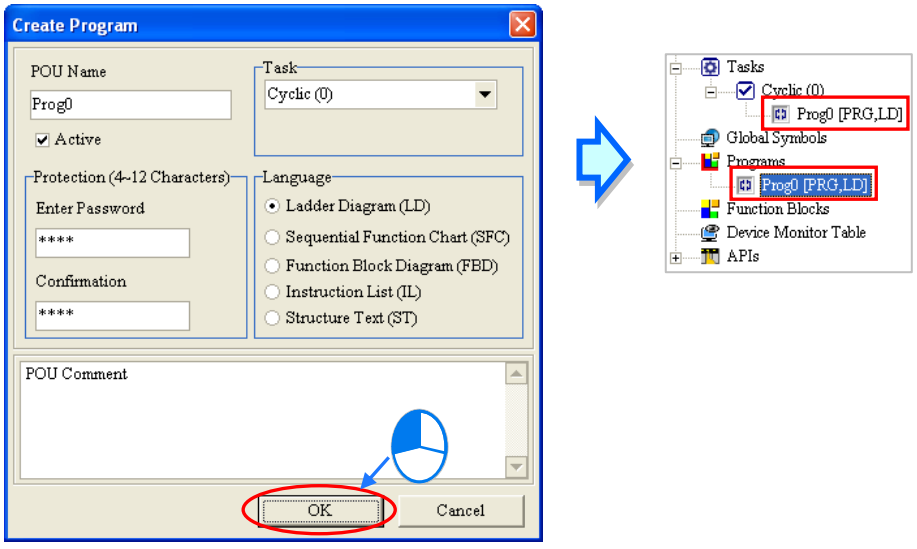


- **Comment**: Users can type a comment on the POU in this box.



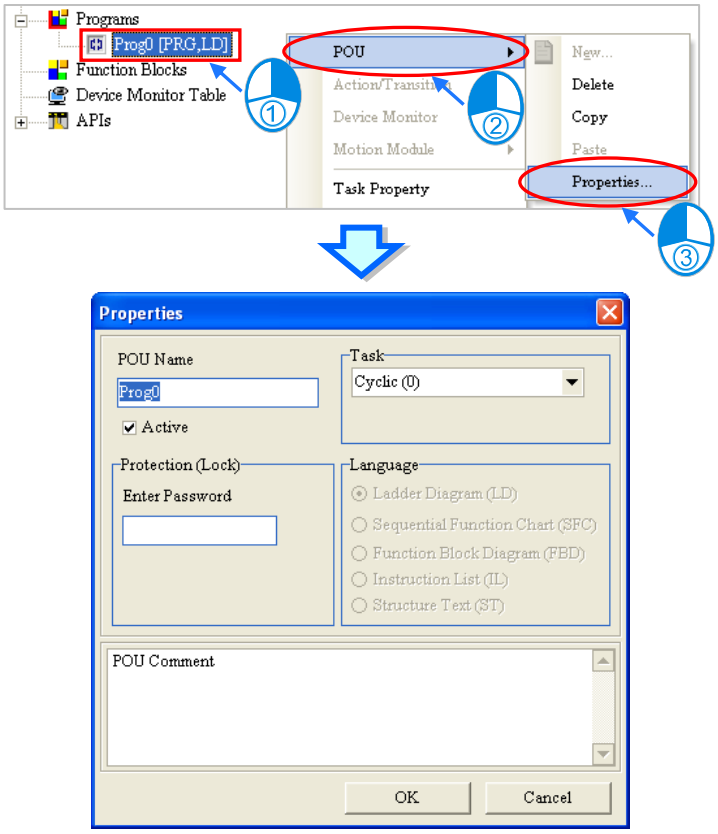
***. If the POU created is a POU of the program type, the Active checkbox will be in the window. The function of the Active checkbox will be introduced in section 5.4.2.**

(3) After the setting is complete, the users can click **OK**. If the POU created is a POU of the program type, the POU will be listed in the **Programs** section and the default task section. If the POU created is a POU of the function block type, the POU will be listed in the **Function Blocks** section.



(4) If the users want to modify the attributes of a POU which has been created, they have to right-click the POU, point to **POU** on the context menu, click **Properties…**, and set the

properties in the window. However, the users can not change the programming language in which the POU is written.

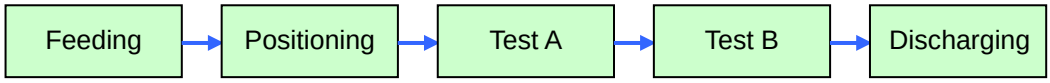


5.4.2 Enabling a POU

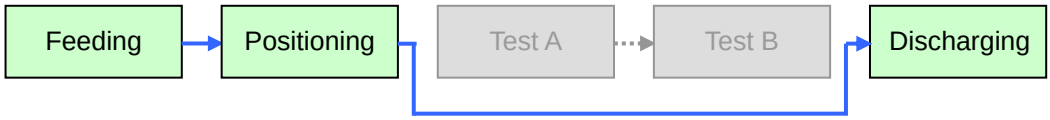
ISPSoft provides the function of disabling a POU temporarily. If a POU is disabled, it is skipped when the program is compiled, and the program in the POU is not executed. However, only POUs of the program type can be disabled.

This function can be applied to the test of a program or the debugging of a program. If users want to skip certain procedure during the test of a program or the debugging of a program, they can disable the POU. After the test of the program or the debugging of the program is complete, they can enable the POU.

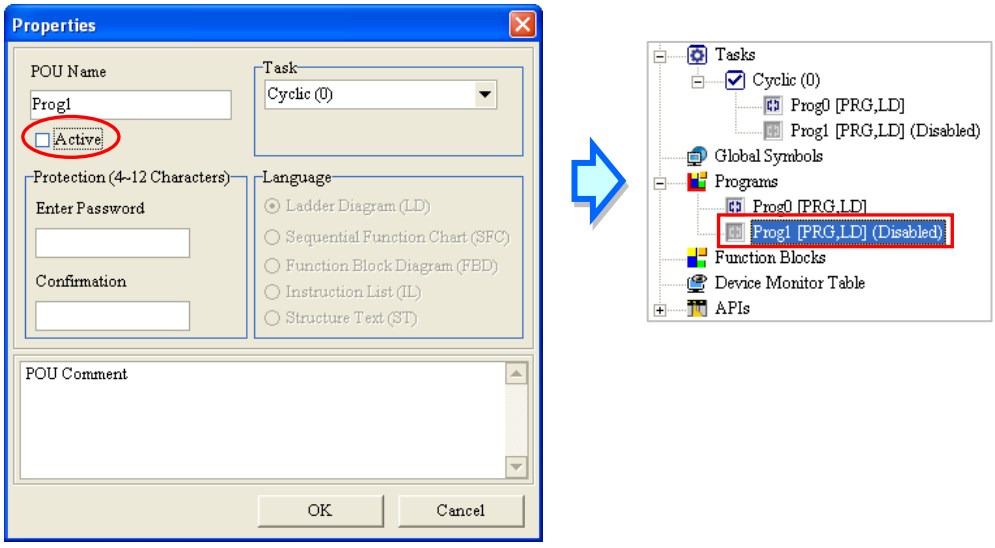
The equipment shown below is composed of five workstations, and the working practices are divided into five POUs of the program type.



If the assembling of the equipment is complete, but the assembling of the test instrument is not complete, users can disable the POUs corresponding to test A and test B. The two workstations are skipped when the other parts of the equipment are adjusted.



If users want to enable a POU, they can select the **Active** checkbox in the window. If the **Active** checkbox is unselected, the POU is disabled. A POU which is disabled is represented by a gray icon.

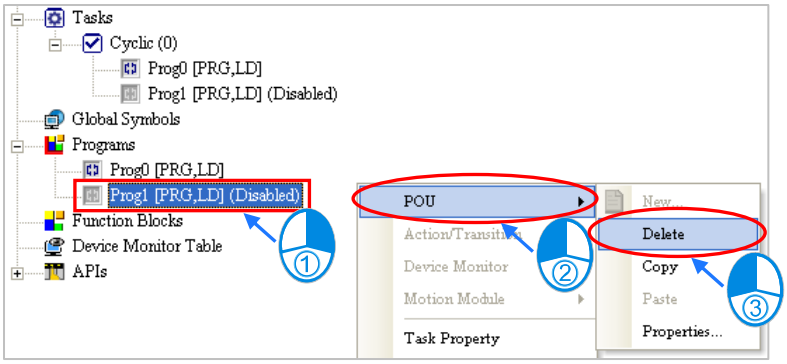


***1. Users can disable a POU by removing it from the original task section. However, if the users want to execute the POU later, they have to assign the POU to the original task again, and rearrange the POUs in the task section. As a result, it is more convenient for the users to disable the POU temporarily by unselecting the Active box in the window.**

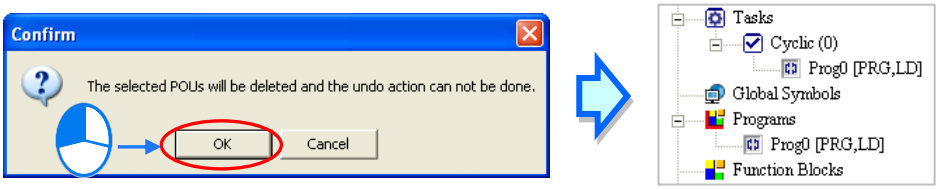
***2. A POU can be enabled not only by selecting the Active box in the window, but also by the task management tool. Please refer to section 5.5.3 for more information.**

5.4.3 Deleting and Copying a POU

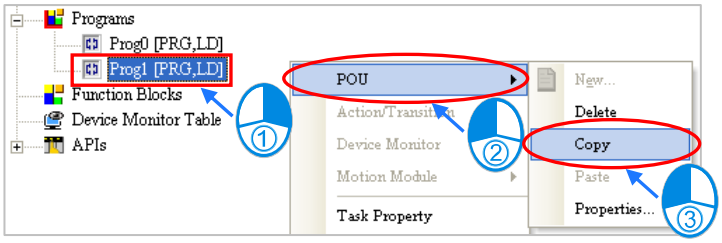
If users want to delete a POU, they have to right click the POU, point to **POU** on the context menu, and click **Delete**.



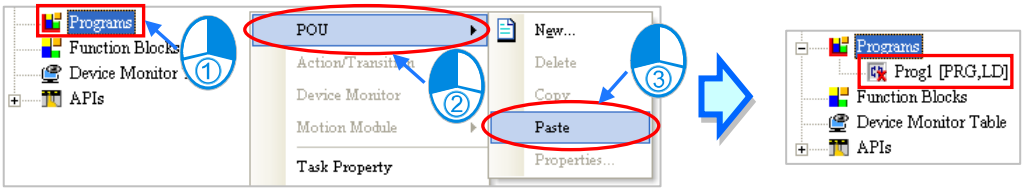
The POU can not be recovered after it is deleted. The POU will be deleted from the project after **OK** is clicked. If the POU deleted is a POU of the program type, they system will also delete it from the task section.



If users want to copy a POU, they have to right-click the POU, point to **POU**, and click **Copy**.



The POU can be copied into the same project or another project. If the POU copied is a POU of the program type, it can not be copied into the **Function Blocks** section. If the POU copied is POU of the function block type, it can not be copied into the **Programs** section. If the users want to paste the POU, they have to right-click **Programs** or **Function Blocks**, point to **POU** on the context menu, and click **Paste**. Besides, if the POU copied is a POU of the program type, the POU pasted will not be assigned to any task.

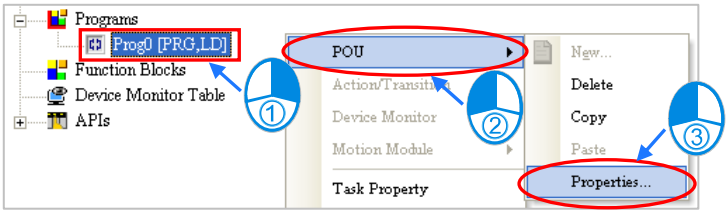


***. If a POU is pasted on a position where the same POU name exists, the system will put CopyOf_n at the end of the POU which is pasted. (n in CopyOf_n is a serial number.)**

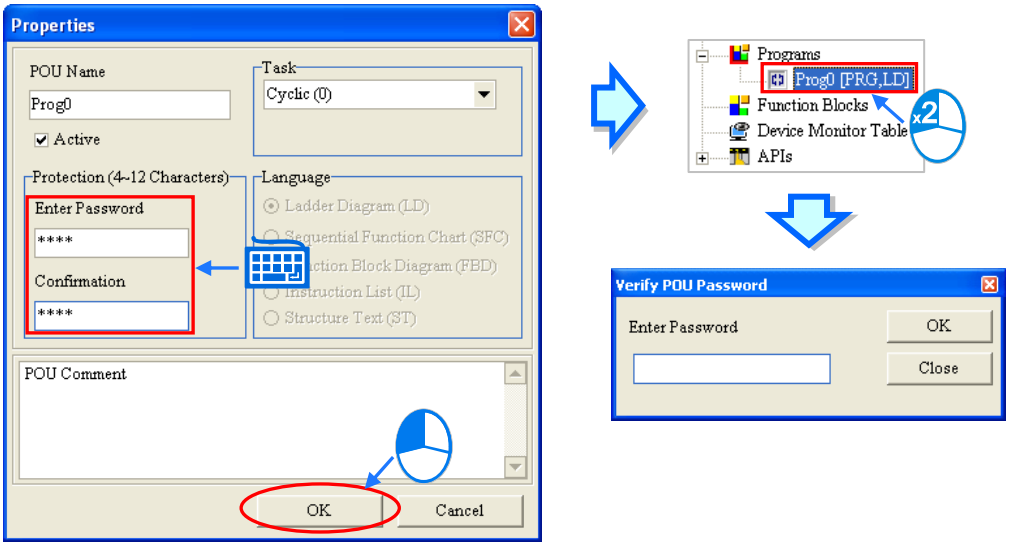
5.4.4 Setting and Unlocking a POU Password

Users can set a password for a POU in ISPSoft. The setting of a password, and the unlocking of the password are described below.

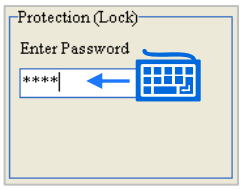
(1) When users create a POU, they can set a password for the POU. If users do not set a password a POU when the POU is created, but want to protect the POU with a password later, they have to right-click the POU in the project management area, point to **POU** on the context menu, and click **Properties…** to open the **Properties** window.



(2) Type the same passwords in the **Enter Password** box and the **Confirmation** box, and then click **OK**. English letters, numerical digits, and special marks can be typed in the boxes. Once the POU is protected with the password, the system asks the users to type the password whenever the window for the POU is opened.



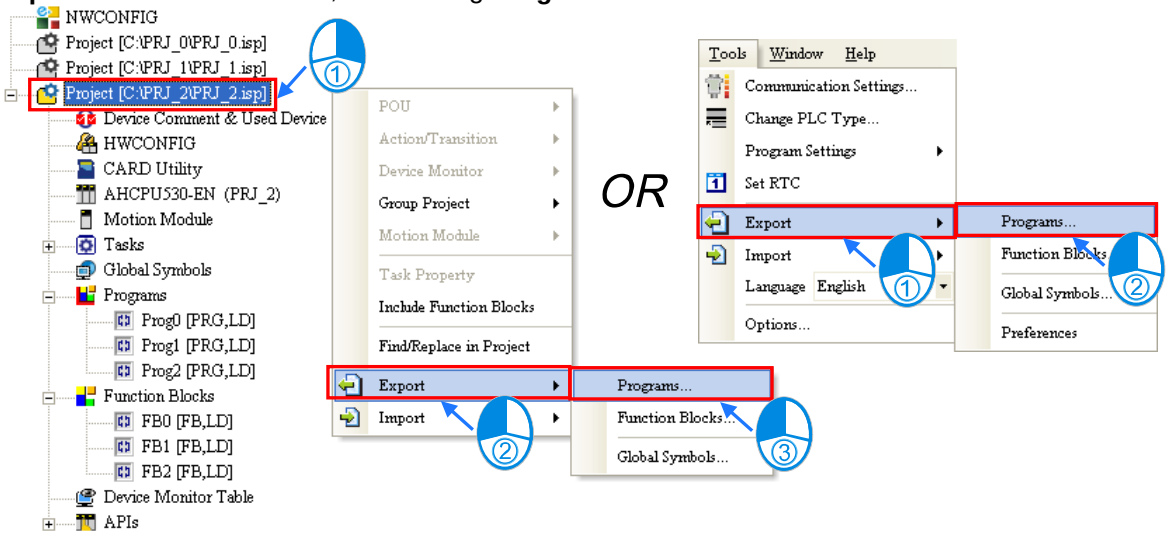
(3) If the users want to unlock the password, they have to open the **Properties** window again, type the password in the **Enter Password** box, and click **OK**.



5.4.5 Exporting a POU of the Program Type

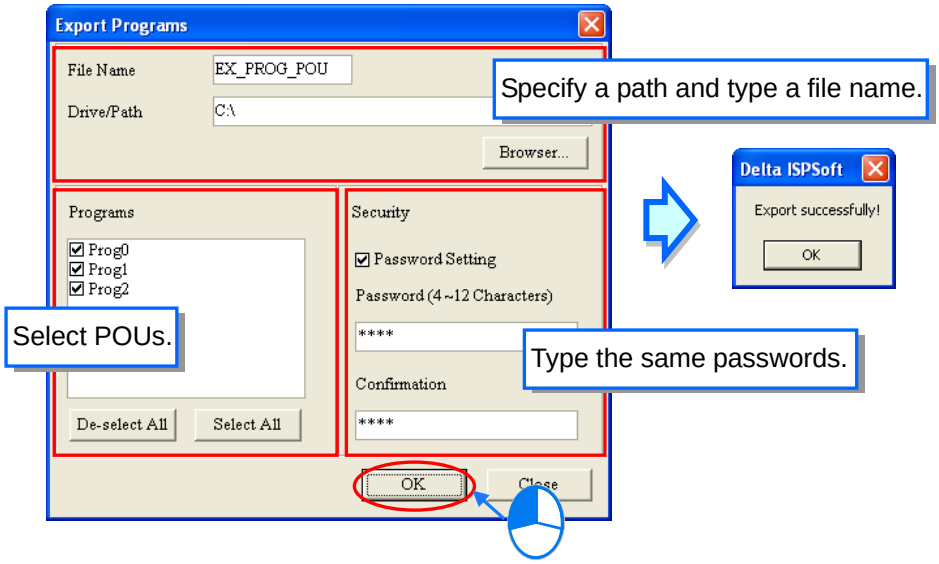
ISPSoft supports the function of exporting POUs, and the function of importing POUs. If there is a group of projects in ISPSoft, users can only export POUs in the project which is being edited presently.

Users can export POUs in the project which is being edited presently by clicking the **Tools** menu, pointing to **Export**, and clicking **Programs…**. They can also export POUs in the project which is being edited presently by right-clicking the project in the project management area, pointing to **Export** on the context menu, and clicking **Programs…**.



In the **Export Programs** window, specify a path in the **Drive/Path** box, type a file name in the **File Name** box, and select POUs of the program type which will be exported in the **Program** section. If the users want to protect the file which will be exported with a password, they have to select the **Password Setting** checkbox, type the same passwords in the **Password** box and the

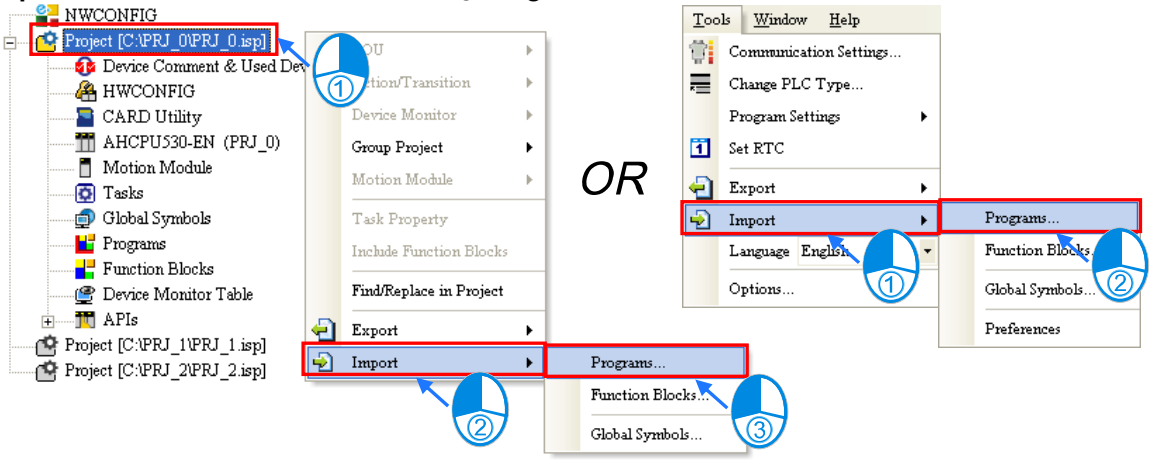
Confirmation box. English letters, numerical digits, and spaces can be typed in the boxes. Finally, click **OK**.



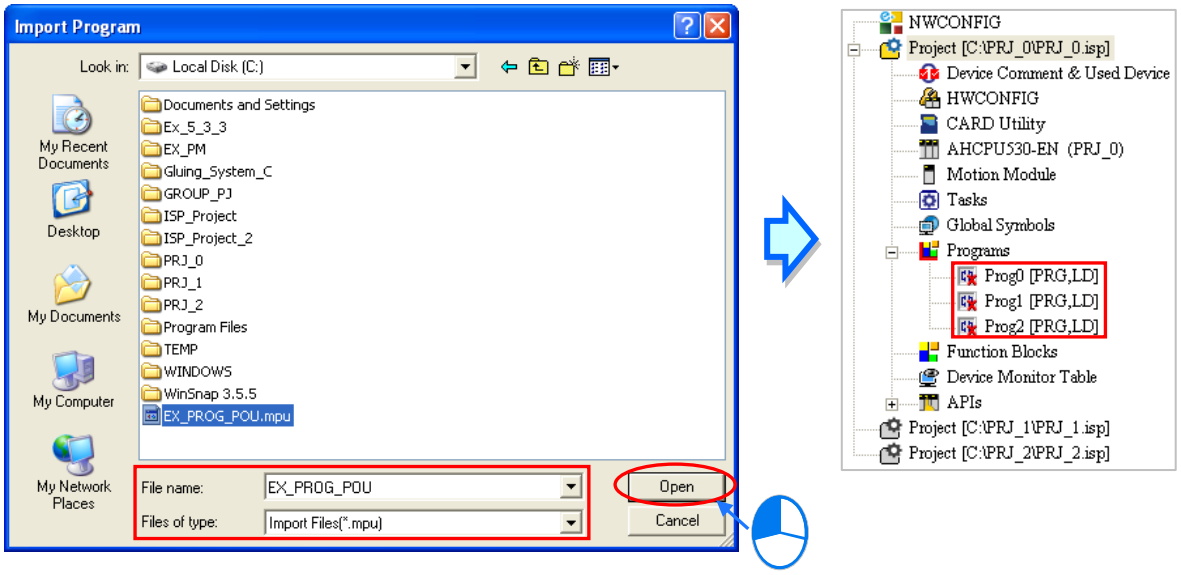
5.4.6 Importing a POU of the Program Type

Users can only import POUs into the project which is being edited presently. Besides, a POU of the function block type which has been exported can not be imported into the **Programs** section.

Users can import POUs into the project which is being edited presently by clicking the **Tools** menu, pointing to **Import**, and clicking **Programs...**. They can also import POUs into the project which is being edited presently by right-clicking the project in the project management area, pointing to **Import** on the context menu, and clicking **Programs...**.



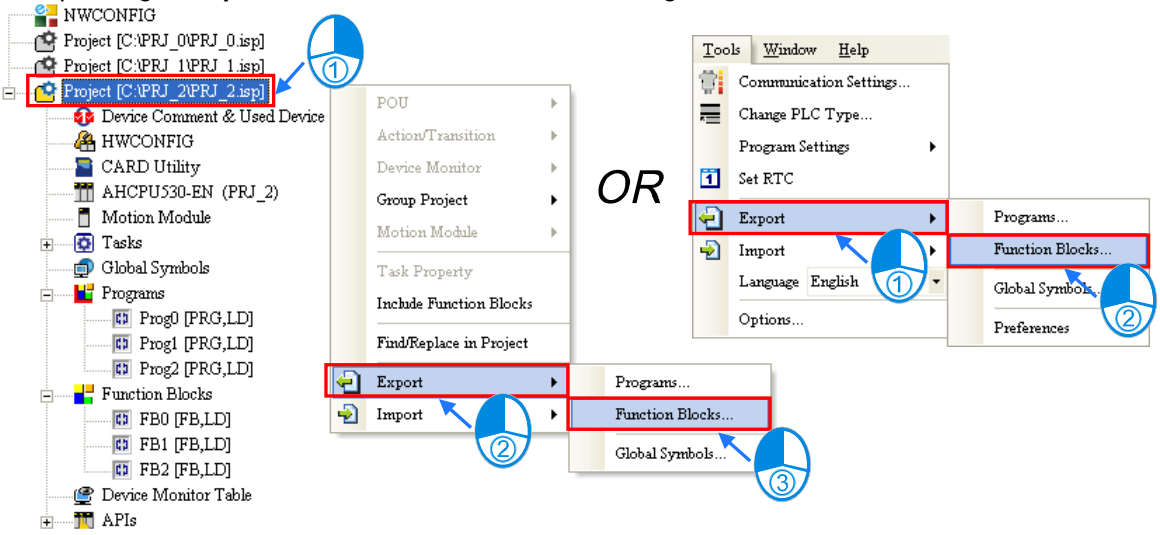
Select a file which will be imported, and then click **Open**. If the file is protected with a password, the users have to type the password in the **Password for Decrypting** window, and click **OK**. The POUs which are imported will not be assigned to any tasks. Besides, if the name of a POU which will be imported is the same as the name of a POU in the project, the system will not allow the POU to be imported, but the other POUs will be imported normally.



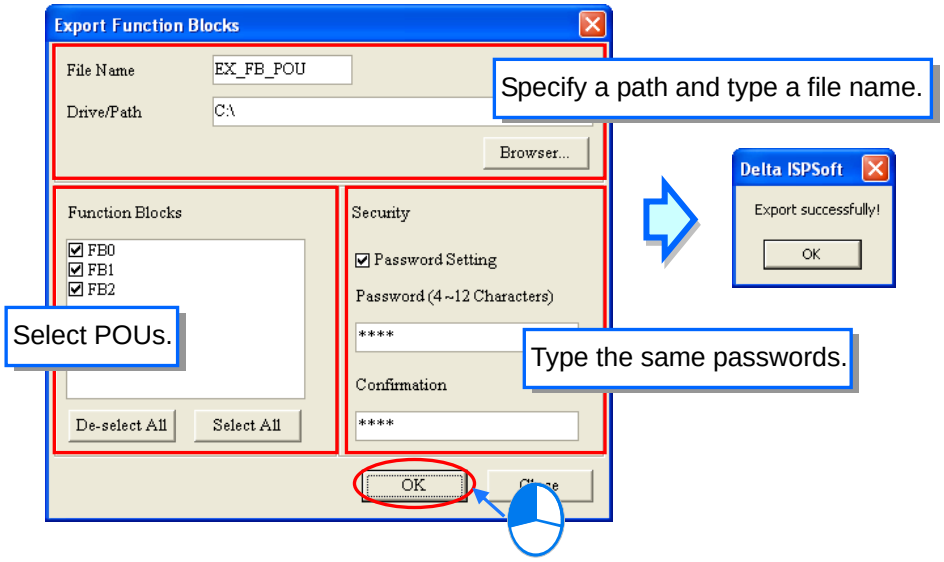
5.4.7 Exporting a POU of the Function Block Type

ISPSoft supports the function of exporting POUs, and the function of importing POUs. If there is a group of projects in ISPSoft, users can only export POUs in the project which is being edited presently.

Users can export function blocks in the project which is being edited presently by clicking the **Tools** menu, pointing to **Export**, and clicking **Function Blocks…**. They can also export function blocks in the project which is being edited presently by right-clicking the project in the project management area, pointing to **Export** on the context menu, and clicking **Function Blocks…**.



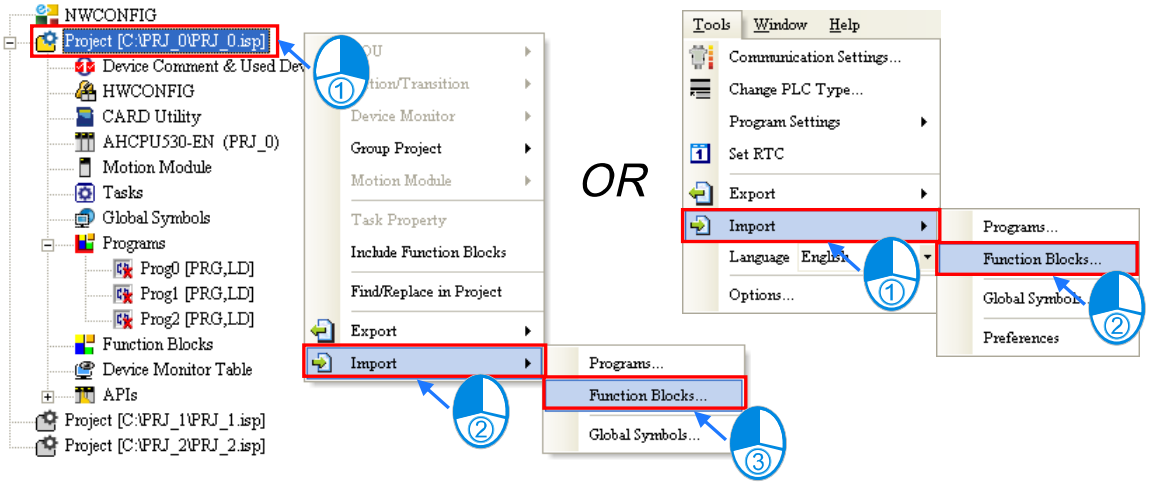
In the **Export Function blocks** window, specify a path in the **Drive/Path** box, type a file name in the **File Name** box, and select POUs of the function block type which will be exported in the **Function Blocks** section. If the users want to protect the file which will be exported with a password, they have to select the **Password Setting** checkbox, type the same passwords in the **Password** box and the **Confirmation** box. English letters, numerical digits, and spaces can be typed in the boxes. Finally, click **OK**.



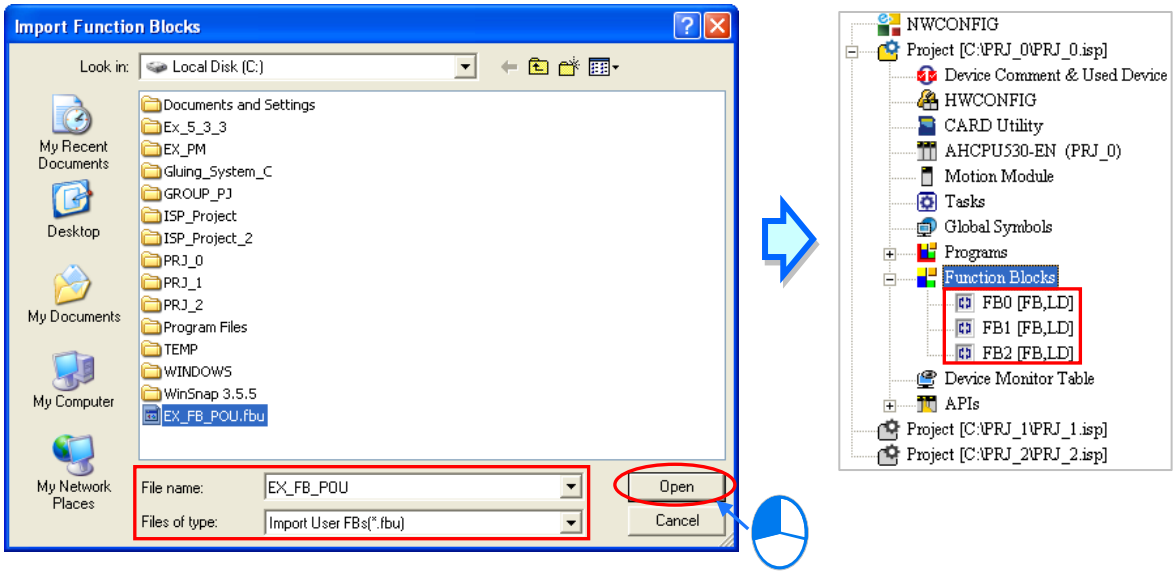
5.4.8 Importing a POU of the Function Block Type

Users can only import POUs into the project which is being edited presently. Besides, a POU of the program type which has been exported can not be imported into the **Function Blocks** section.

Users can import POUs into the project which is being edited presently by clicking the **Tools** menu, pointing to **Import**, and clicking **Function Blocks…**. They can also import POUs into the project which is being edited presently by right-clicking the project in the project management area, pointing to **Import** on the context menu, and clicking **Function Blocks…**.

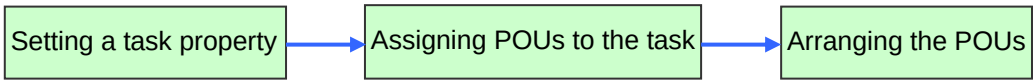


Select a file which will be imported, and then click **Open**. If the file is protected with a password, the users have to type the password in the **Password for Decrypting** window, and click **OK**. Besides, if the name of a POU which will be imported is the same as the name of a POU in the project, the system will not allow the POU to be imported, but the other POUs will be imported normally.



5.5 Managing Tasks

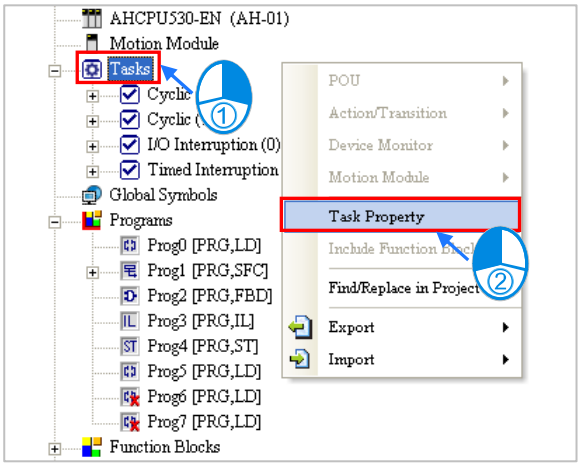
The steps of managing the tasks in ISPSoft are shown below. Please refer to the following sections for more information about the management of the tasks in ISPSoft.



Before managing the tasks in ISPSoft, users have to start the task management tool. There are three ways to start the tasks management tool.

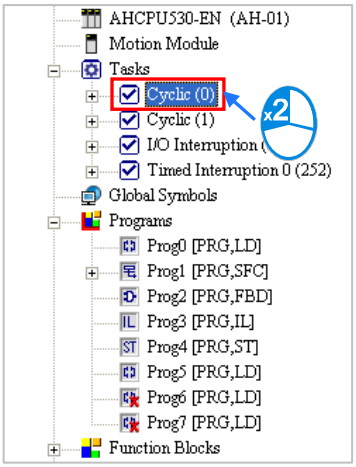
- **Method 1**

  Right-click **Tasks** in the project management area, and then click **Task Property** on the context menu to open the **Task Manager** window.



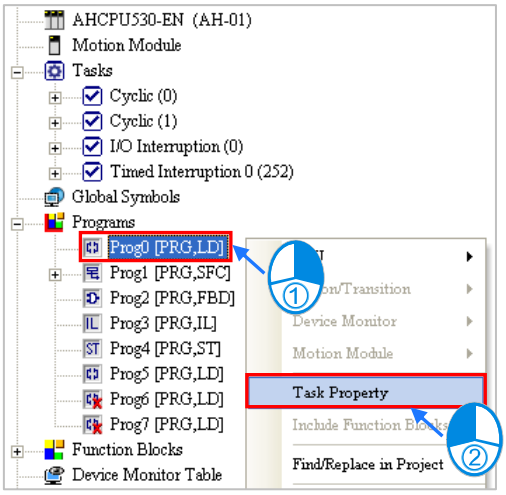
- **Method 2**

  Unfold the **Tasks** section in the project management area, and then double-click a task which will be set to open the **Task Manager** window.

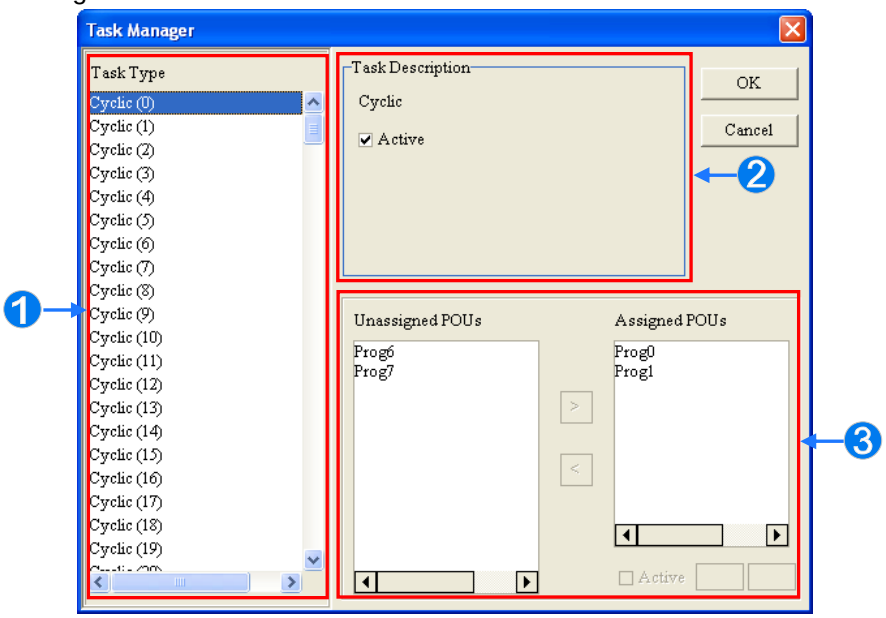


● **Method 3**

Right-click a POU which will be assigned to a task in the **Programs** section in the project management area, and then click **Task Property** on the context menu to open the **Task Manager** window.



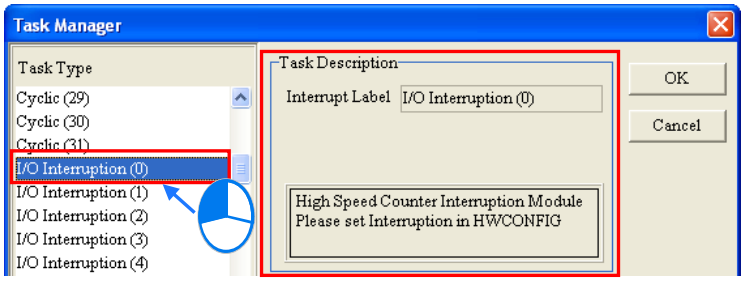
The Task Manager window is shown below.



1. Task type: The tasks available are listed here.
2. Task description: The description of a task selected is displayed here.
3. Task management area: Users can assign POUs to tasks, and arrange POUs.

5.5.1 Setting the Property of a Task and the Condition of an Interrupt

After a task in the **Task Type** section is selected, the description of the task will be displayed in the **Task Description** section. Users can set the condition of executing the task in the **Task Description** section. Besides, the tasks provided by a project for a PLC depend on the functions of the PLC selected, and different tasks have different properties.

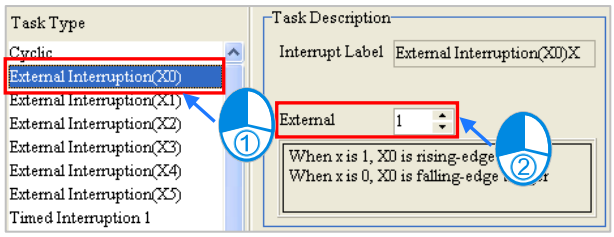


The setting of the properties of tasks can be illustrated with some simple examples. Users can refer to the examples below and the operation manual for a PLC for more information about setting the properties of tasks.

Example 1: If users want to set the property of an interrupt task for a DVP-SV series PLC, they have to make sure of the interrupts provided by the DVP-SV series PLC. The interrupts provided by a DVP-SV series PLC are listed in the table below. (The table is in DVP-PLC Application Manual–Programming.)

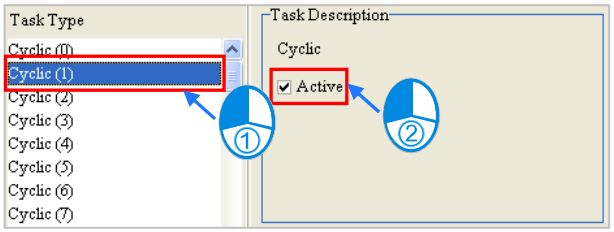
External interrupt	I00□(X0), I10□(X1), I20□(X2), I30□(X3), I40□(X4), and I50□(X5) (□=1: Rising edge-triggered ; □=0: Falling edge-triggered)
Timed interrupt	I6□□ and I7□□ (□□=1~99; Time unit=1 ms) I8□□ (□□=1~99; Time unit=0.1 ms)
High-speed counter interrupt	I010, I020, I030, I040, I050, and I060
Pulse interrupt	I110, I120, I130, and I140
Communication interrupt	I150, I160, and I170
Frequency measurement card interrupt	I180

The **Task Type** section for a DVP-SV series PLC is shown below. If users want to set a task corresponding to the interrupt I001, they have to select **External Interruption (X0)** in the **Task Type** section first. (I001 is triggered when the signal passing through X0 goes from low to high.) The description of **External Interruption (X0)** will be displayed in the **Task Description** section. According to the message in the **Task Description** section, I001 is rising edge-triggered if the value in the **External** box is 1. As a result, the users have to select 1 in the **External** box.

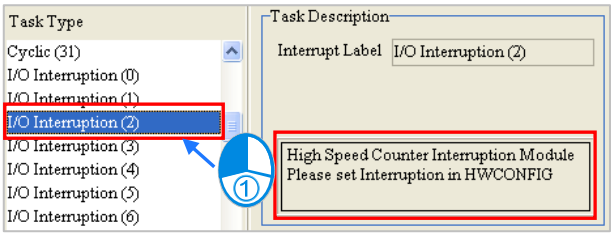


Example 2: The figure below is the **Task Manager** window for AHCPU530-EN. The task selected is **Cyclic (1)**. Owing to the fact that users can enable and disable a cyclic task in an

AH500 series CPU module through the applied instructions TKON and TKOFF, they can set the initial state of the cyclic task. If the **Active** checkbox is unselected, **Cyclic (1)** is disabled when the program is executed. **Cyclic (1)** is not enabled until a POU assigned to another task uses the instruction TKON to enable **Cyclic (1)**. Please refer to AH500 Programming Manual for more information about TKON and TKOFF.

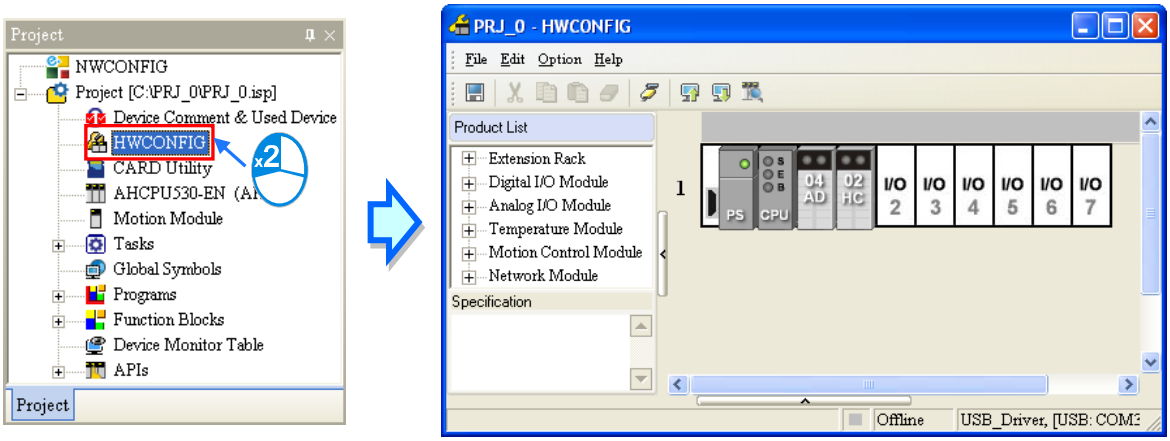


Example 3: The figure below is the **Task Manager** window for AHCPU530-EN. The task selected is **I/O Interruption (2)**. Owning to the fact that users can set the condition of an I/O interrupt by themselves, the message in the **Task Description** section indicates that the condition of **I/O Interruption (2)** can be set in HWCONFIG. The setting of the condition of **I/O Interruption (2)** in HWCONFIG is described below.

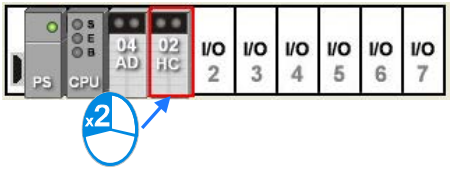


Suppose **I/O Interruption (2)** will be triggered after channel 1 on AH02HC-5A receives a certain number of pulses. The setting of **I/O Interruption (2)** in HWCONFIG is described below. Users can refer to chapter 3 for more information about setting the parameters in a module, and refer to AH500 Operation Manual for more information about the interrupts in an AH500 series CPU module.

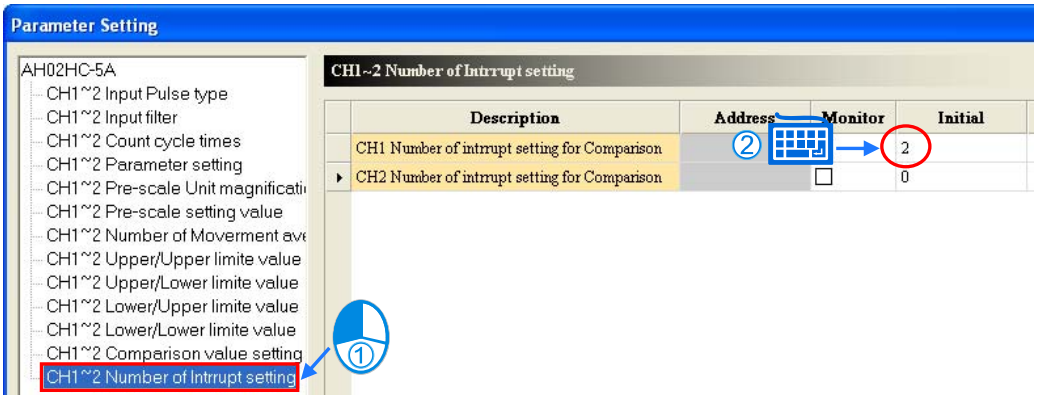
(1) Double-click **HWCONFIG** in the project management area in ISPSOft.



(2) Double-click AH02HC-5A to open the **Parameter Setting** window.

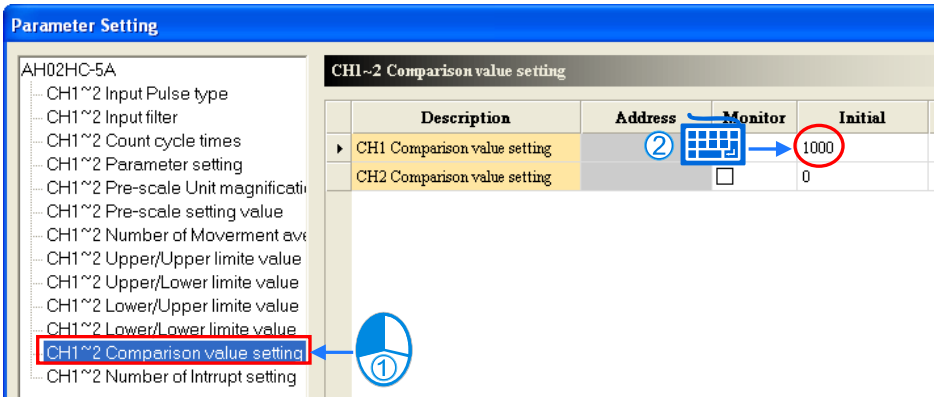


(3) Select **CH1~2 Number of Interrupt Setting**, and type 2 in the **Initial** cell for **CH1 Number of Interrupt Setting for Comparison**.

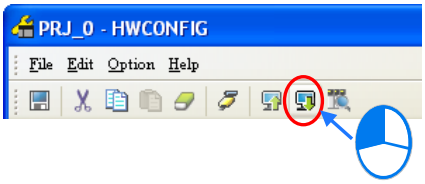


***. If the value in the Initial cell for CH1 Number of Interrupt Setting for Comparison is the same as the value in the Initial cell for CH2 Number of Interrupt Setting for Comparison, the same I/O interrupt will be triggered after channel 1 or channel 2 receives a certain number of pulses.**

(4) Select **CH1~2 Comparison Value Setting**, and type 1000 in the **Initial** cell for **CH1 Comparison Value Setting**.



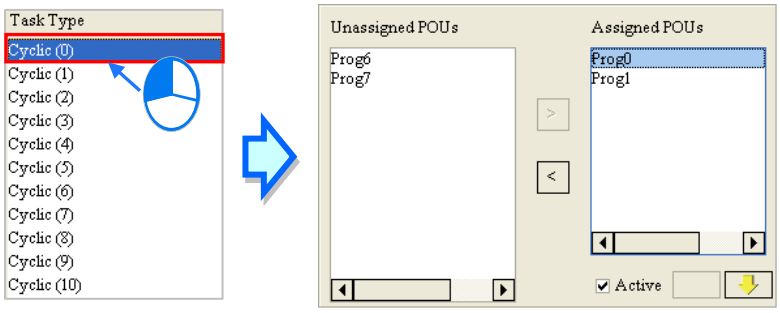
(5) Be sure to save the file after the setting is complete. The parameters set in HWCONFIG must be downloaded to the CPU module so that they can take effect.



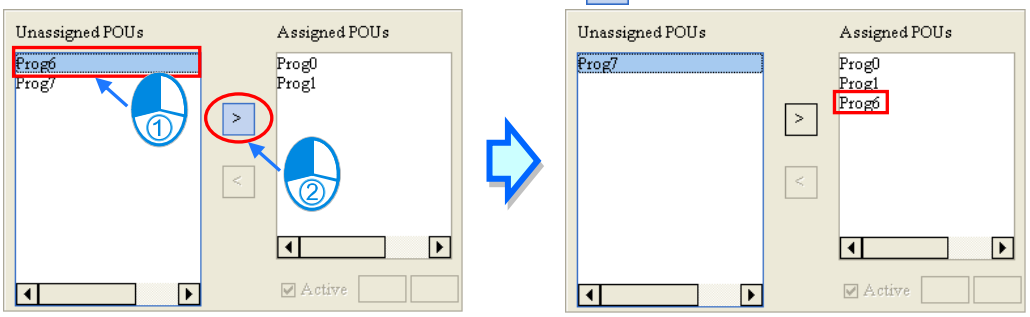
5.5.2 Assigning a POU to a Task

After the setting of the property of a task is complete, users can assign POUs to the task. POUs which are sequential function charts can only be assigned to cyclic tasks, and can not be assigned to interrupt tasks.

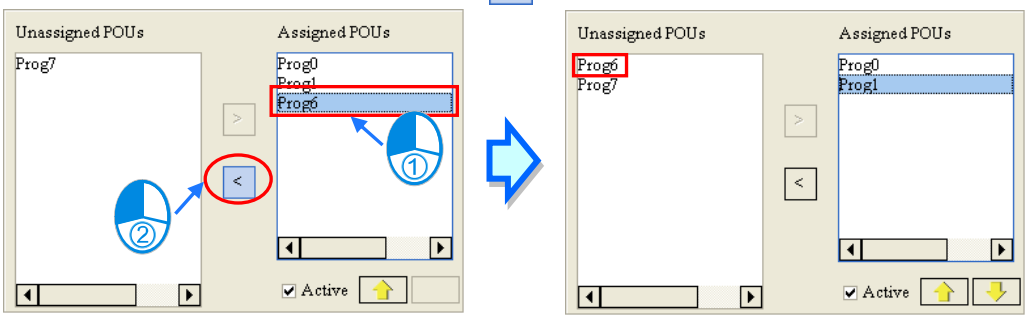
(1) After a task in the **Task Type** section is selected, the assignment of the POUs will be displayed in the task management area. The POUs which have not been assigned to the task are in the **Unassigned POUs** section, and the POUs which have been assigned to the task are in the **Assigned POUs** section.



(2) If users want to assign a POU in the **Unassigned POUs** section to the task, the can select the POU in the **Unassigned POUs** section, and click > .



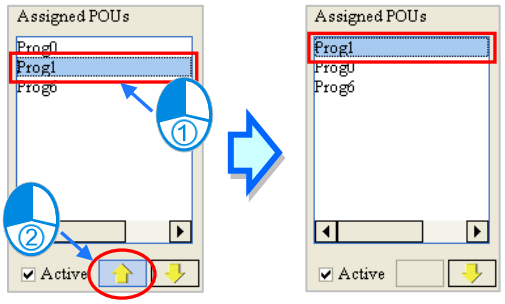
(3) If the users want to remove a POU from the **Assigned POUs** section, they can select the POU in the **Assigned POUs** section, and click < .



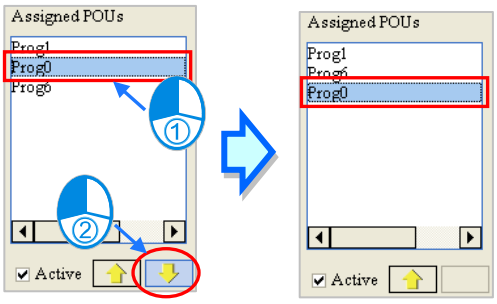
5.5.3 Arranging the POUs

After the assignment of POUs to a task is complete, users can arrange the POUs. The order in which the POUs are listed in the **Assigned POUs** section is the order in which the POUs are executed.

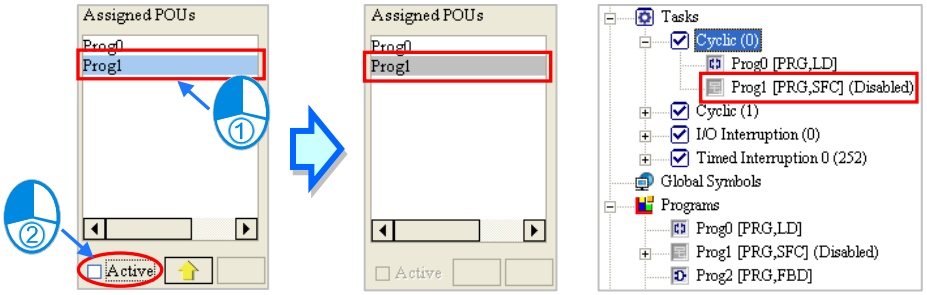
(1) If users want to execute a POU in the **Assigned POUs** section earlier, they can select the POU in the **Assigned POUs** section, and click to move the POU upward.



(2) If users want to execute a POU in the **Assigned POUs** section later, they can select the POU in the **Assigned POUs** section, and click to move the POU downward.



(3) After a POU in the **Assigned POUs** section is selected, users can select the **Active** checkbox if they want to enable the POU, or unselect the **Active** checkbox if they want to disable the POU. A POU which is disabled is represented by a gray icon in the project management area.



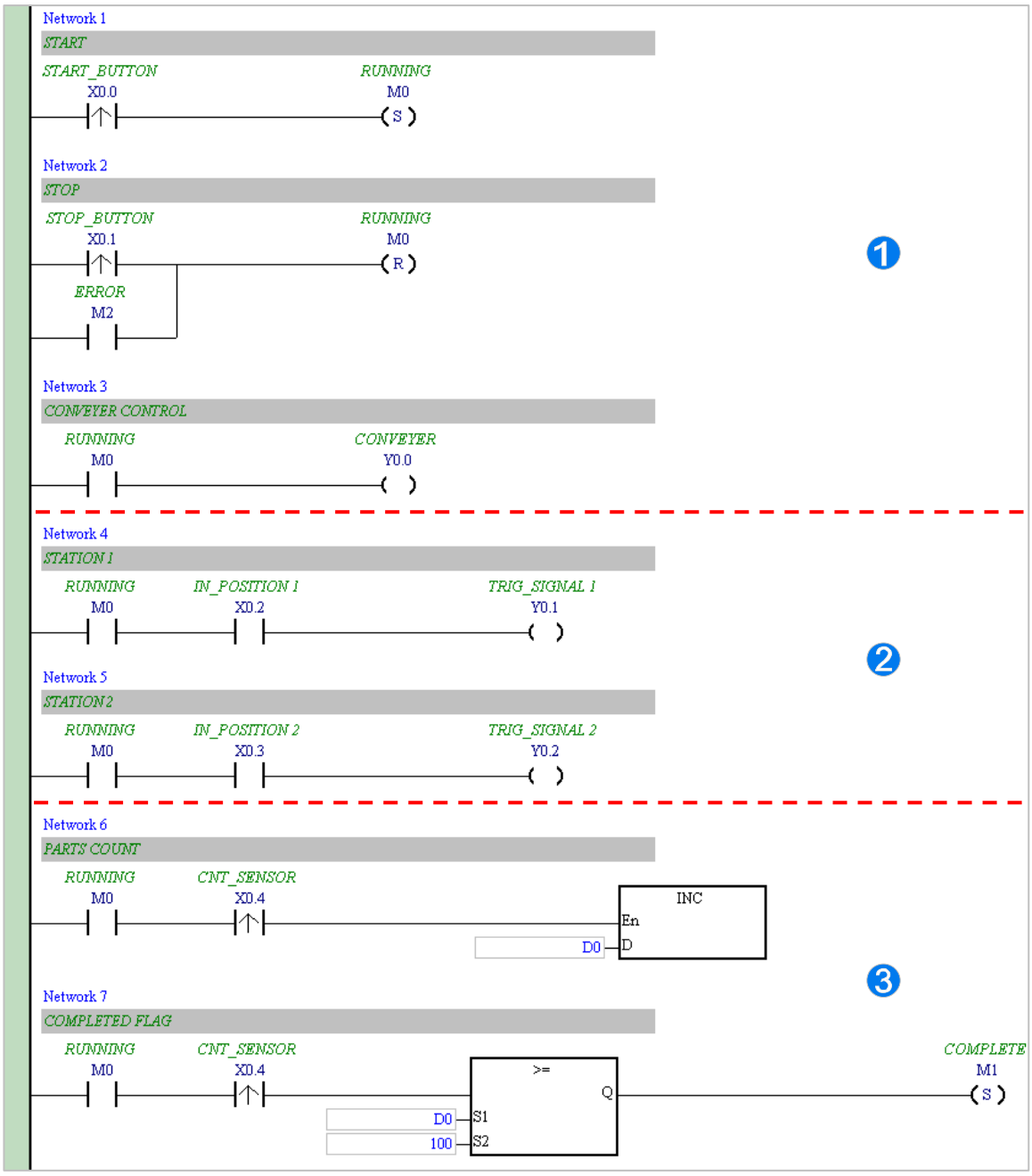
***.Please refer to section 5.4.2 for more information about enabling a POU.**

(4) Click **OK** in the **Task Manager** window after the setting of the task is complete.

5.6 Example

5.6.1 Writing a Program by Means of IEC 61131-3

The program created in chapter 4 will be rewritten by means of IEC 61131-3. The program created in chapter 4 is shown below. It will be divided into three parts according to the functions listed below.



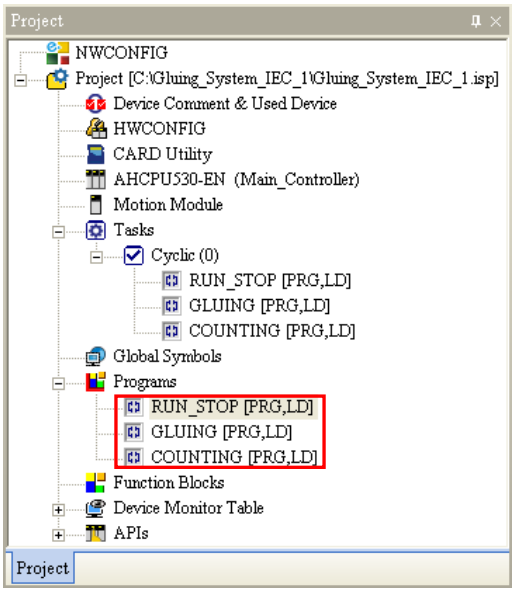
1 This part of the program functions to start or stop the system.

2 This part of the program functions to inject the glue.

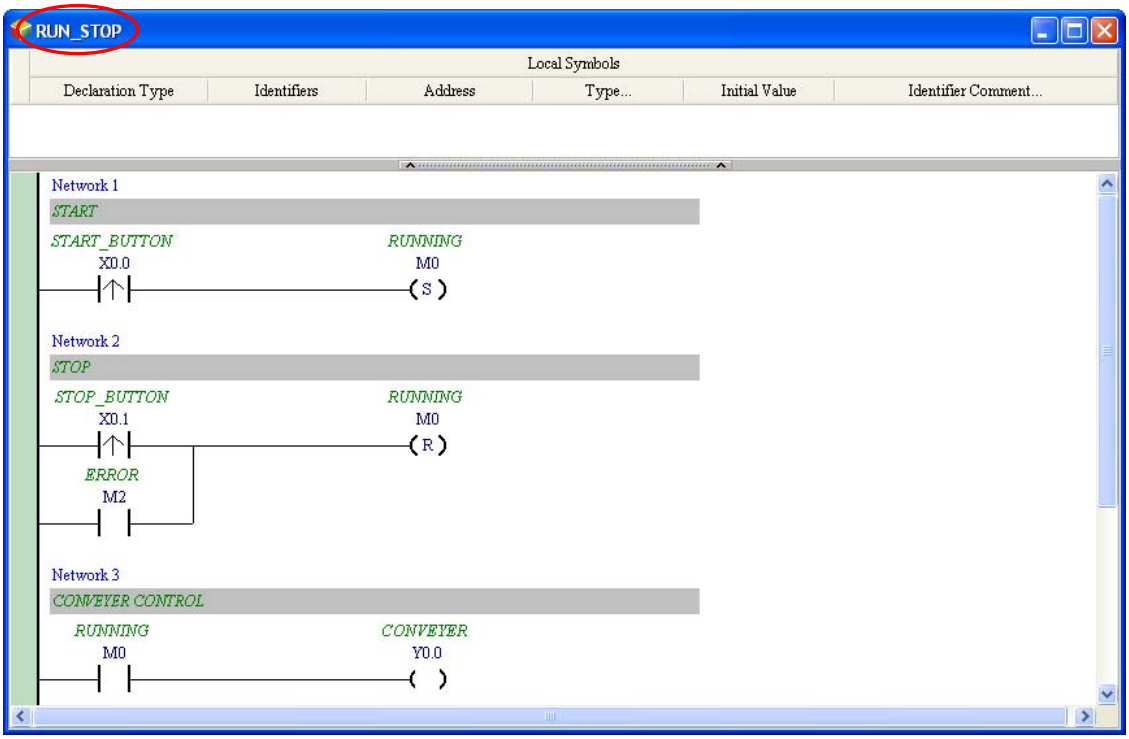
3 This part of the program functions to count the parts which are conveyed.

This section aims at the architecture of a program. The contents related to configuring hardware, setting parameters, testing a program, and debugging a program are the same as the contents of chapter 4. Users can try to complete the related setting by themselves.

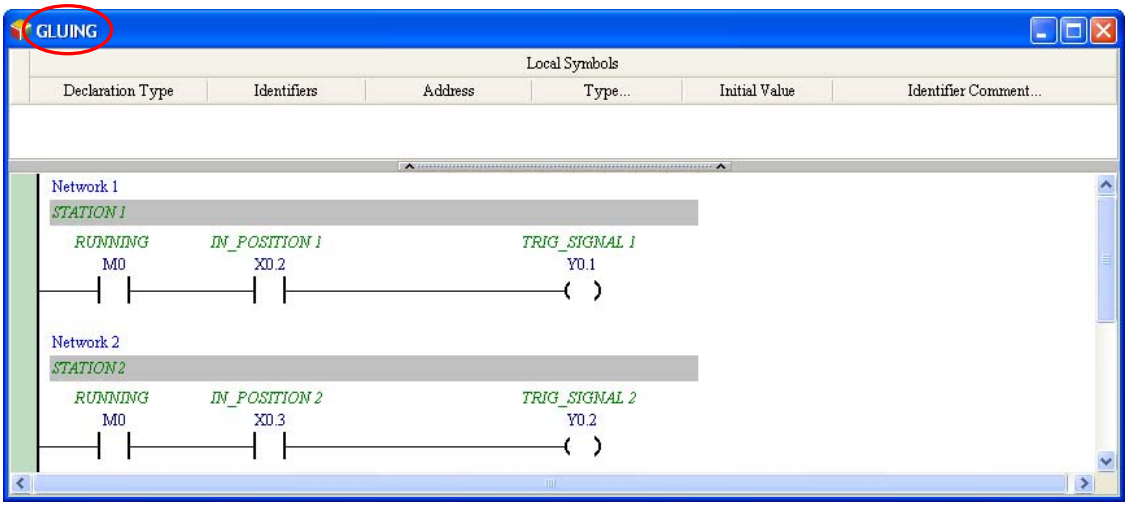
First, create a new project. Then, refer to chapter 4, and complete the hardware configuration. Finally, refer to section 5.4.1, and create three POUs in the **Programs** section in accordance with the three points listed above. These three POUs are assigned to **Cyclic (0)**.



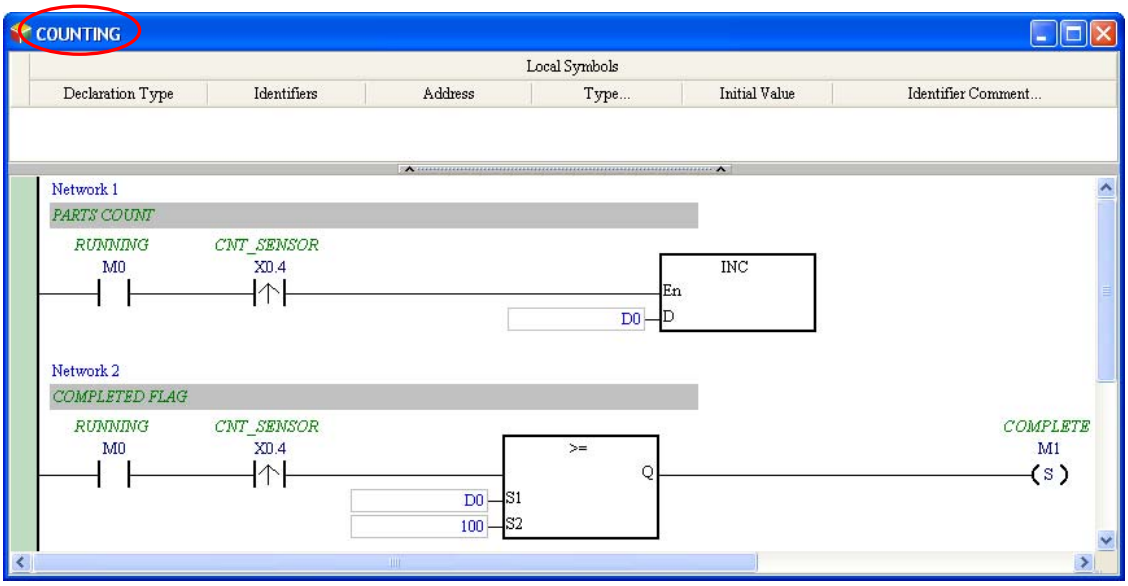
Please refer to chapter 4, and write the programs in the POUs shown below.
The program in **RUN_STOP** functions to start or stop the system.



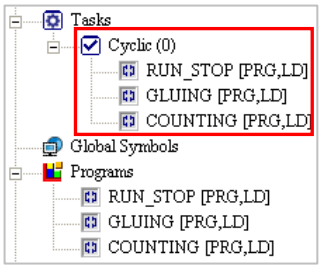
The program in **GLUING** functions to inject the glue.



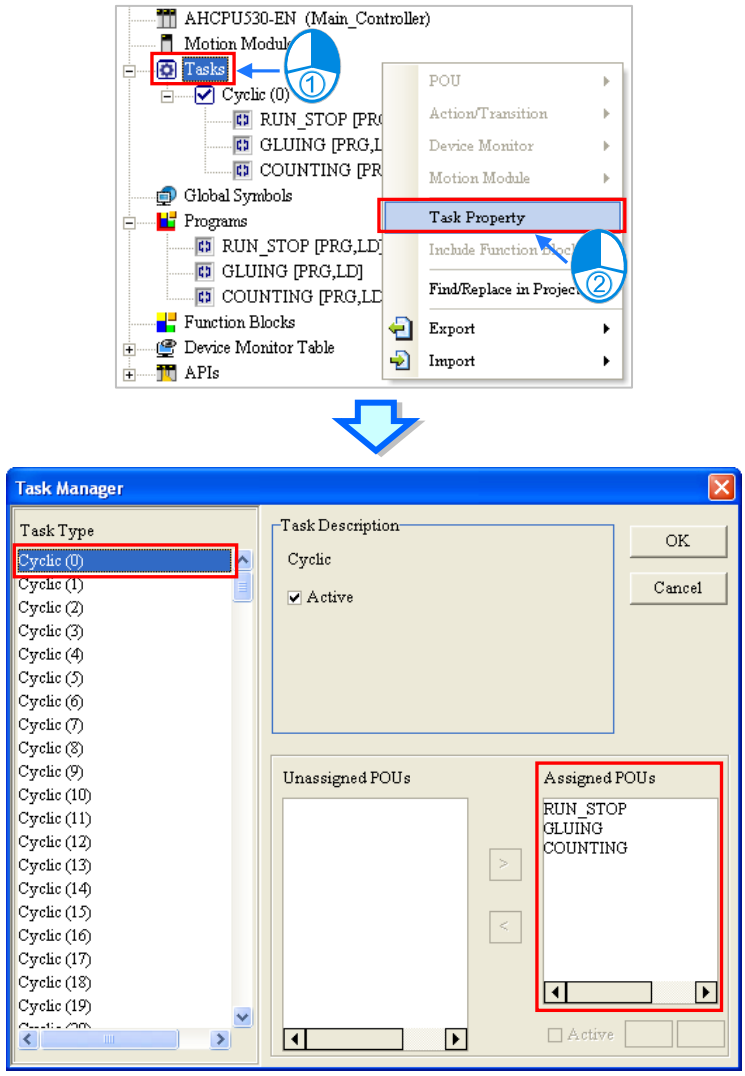
The program in **COUNTING** functions to count the parts which are conveyed.



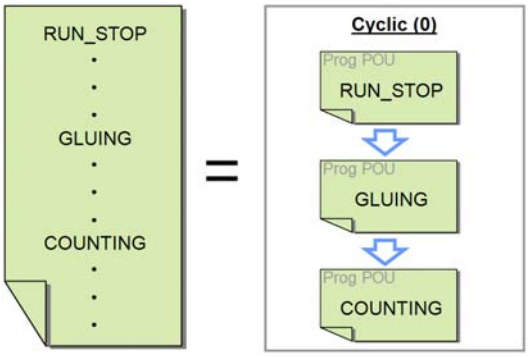
After the creation of the programs is complete, users have to make sure that the order in which the POUs in the **Tasks** section are arranged is the same as the order shown below. If the order in which the POUs in the **Tasks** section are arranged is different from the order shown below, the users have to rearrange the POUs.



If the users want to rearrange the POUs, they have to right-click **Tasks** in the project management area, and click **Task Property** on the context menu to open the **Task Manager** window. Please refer to section 5.5, and rearrange the POUs.



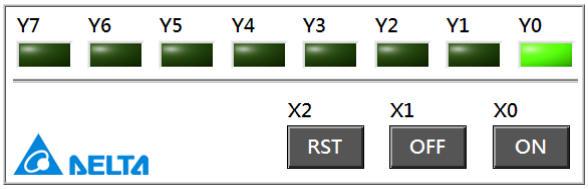
If the creation of the POUs is complete, the result of executing the program will be the same as the result of executing the ladder diagram in chapter 4. Please refer to chapter 4, and compile the program. The program can be tested after it is downloaded to the CPU module.



5.6.2 Example of an Interrupt Subroutine in a DVP Series PLC

- **Example**

  If the ON button is pressed (X0 is turned ON), Y0~Y7 will scanned in order. If the OFF button is pressed (X1 is turned ON), the output will stop, and the present output state will be retained. If the RST button is pressed (X2 is turned ON), the output will be reset, and Y0~Y7 will be scanned again.



- **Planning a program**

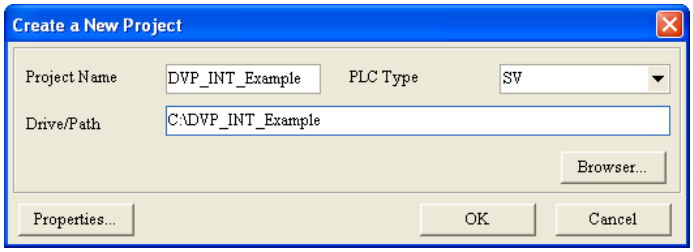
The model used in this example is a DVP-SV PLC, and four POUs are created. The functions of the POUs are described below.

POU name	Function
Run_Stop	1. Reset the operation flag M0, transfer the value 1 to the output data register D0, and enable the interrupt. 2. When X0 is turned from OFF to ON, the operation flag M0 is set to ON. 3. When X1 is turned from OFF to ON, the operation flag M0 is set to OFF.
LED	The states of Y0~Y7 are updated according to the value in the output data register D0.
Shift	The values of the bits in the output data register D0 is shifted by one bit to the left every 0.5 seconds. If the value in D0 is larger than 128, the value 1 will be transferred to D0.
Reset	Transfer the value 1 to the output data register D0, that is, set bit 0 in D0 to ON. The POU will be executed after the external interrupt I201 is triggered. (I201 is triggered when the signal passing through X2 goes from low to high.)

***. Please refer to DVP-PLC Application Manual—Programming for more information about the interrupts in a DVP-SV series PLC and the related applied instructions.**

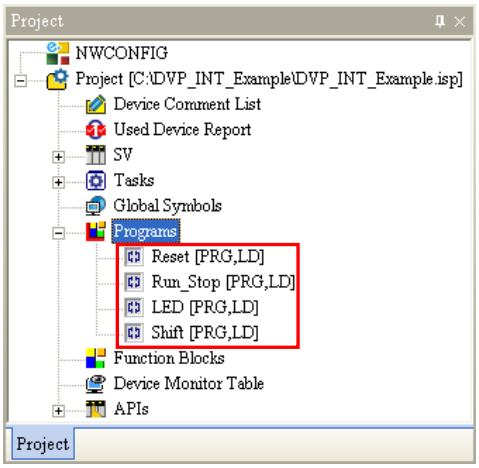
- **Creating the program**

Please create a project for a DVP-SV series PLC.

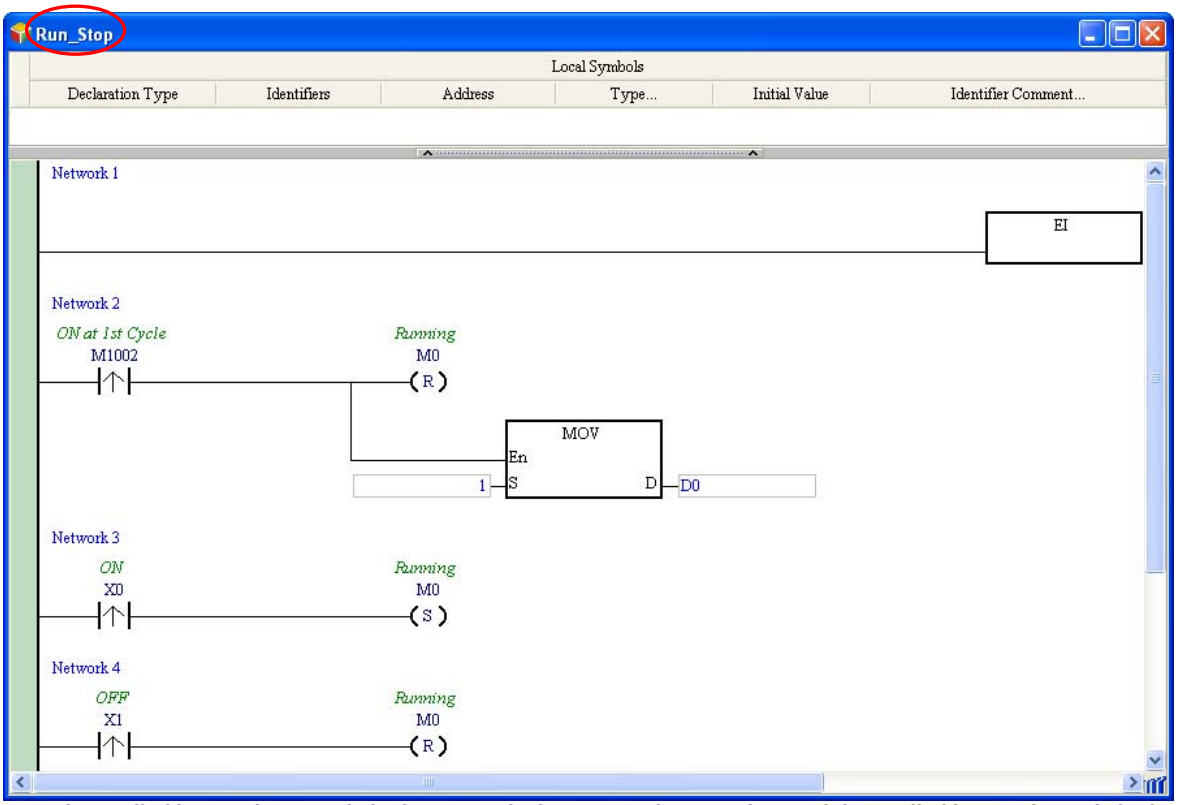


This section aims at the architecture of a program. The parts related to system setting and communication setting are skipped. Users can refer to the related sections, and complete the related setting by themselves. Please refer to section 3.1 for more information about system setting, and section 2.4 for more information about communication setting.

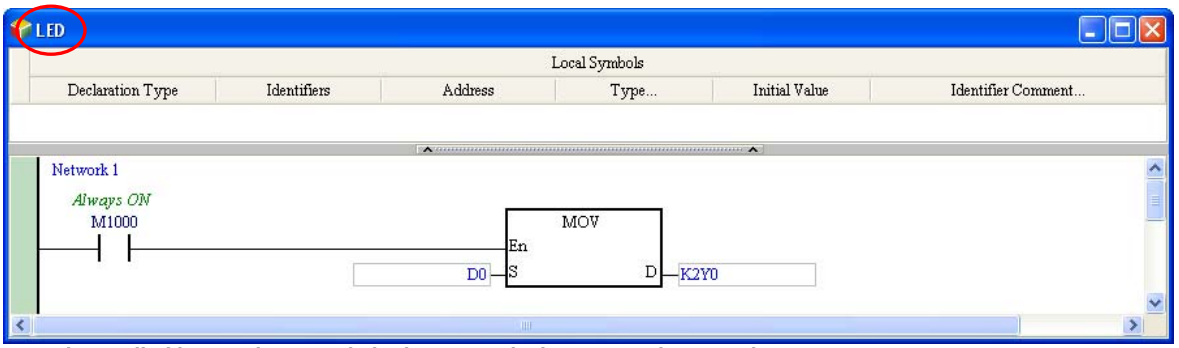
Create the four POUs in the **Programs** section in the project management area shown below. When these POU are created, they are assigned to **Cyclic**.



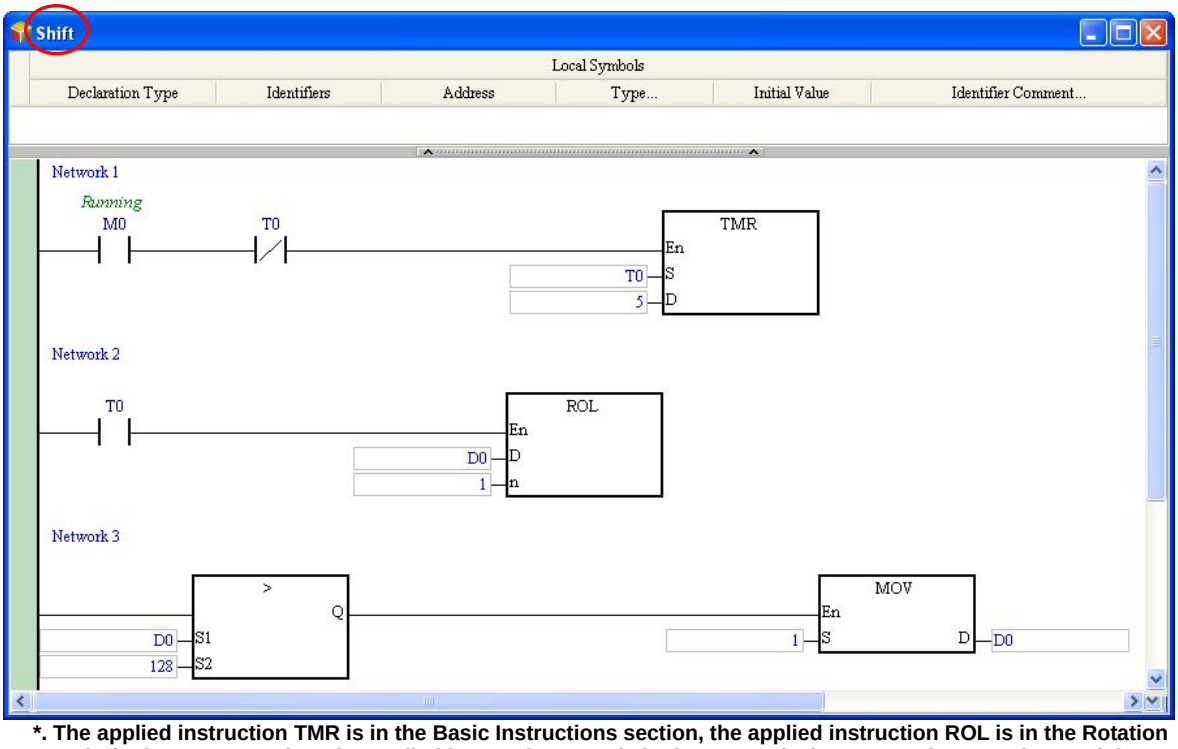
Write the programs in the POUs shown below.



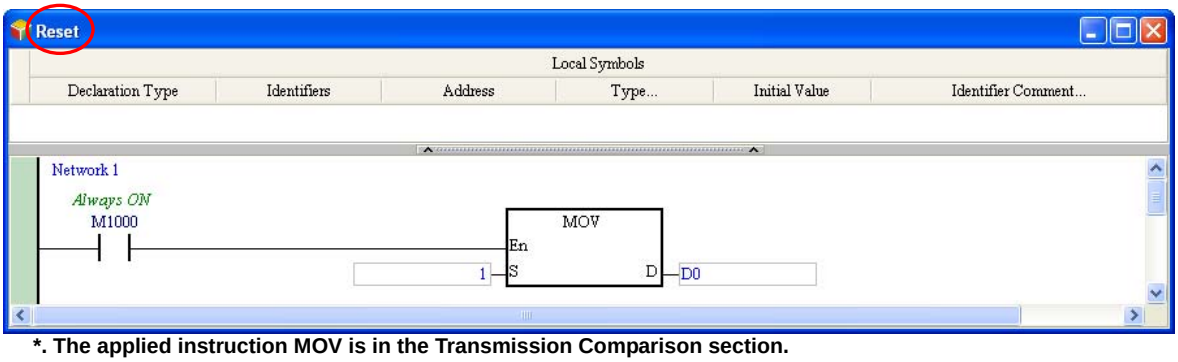
***. The applied instruction MOV is in the Transmission Comparison section, and the applied instruction EI is in the Loop Control section.**



***. The applied instruction MOV is in the Transmission Comparison section.**

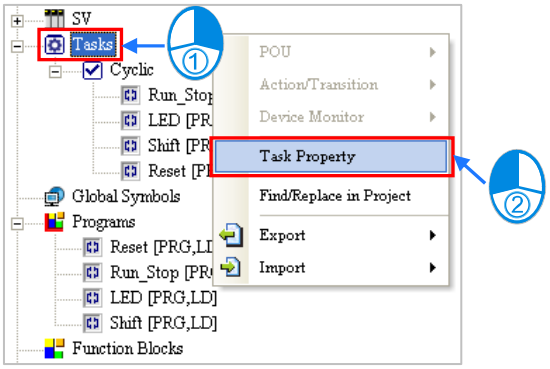


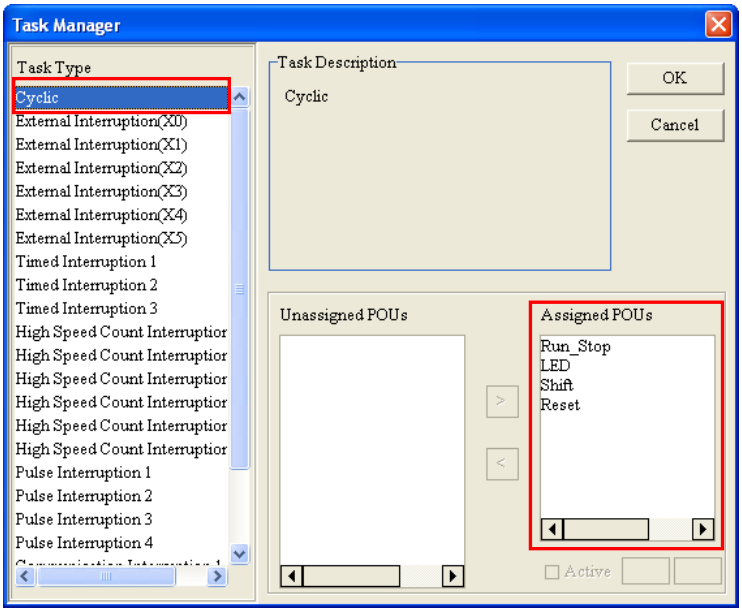
***. The applied instruction TMR is in the Basic Instructions section, the applied instruction ROL is in the Rotation and Displacement section, the applied instruction MOV is in the Transmission Comparison section, and the comparison contacts are in the Contact Type Logic Operation section.**



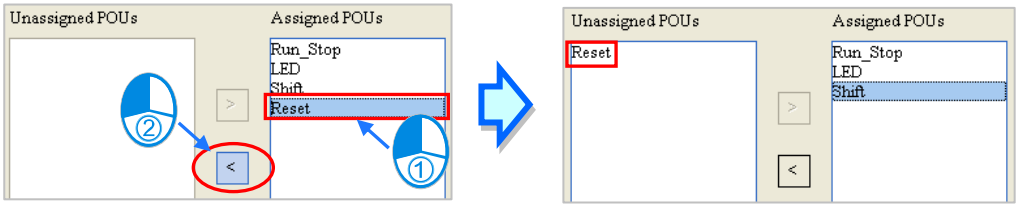
***. The applied instruction MOV is in the Transmission Comparison section.**

Open the **Task Manager** window after the writing of the program is complete. Owing to the fact that the four POUs are assigned to **Cyclic** when they are created, the four POUs are in the **Assigned POUs** section for **Cyclic**.

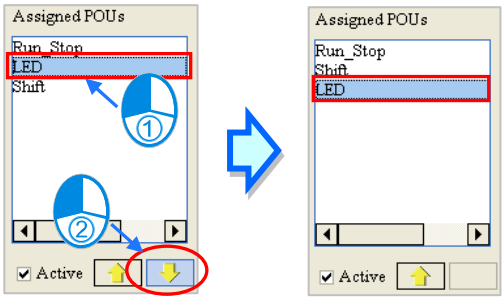




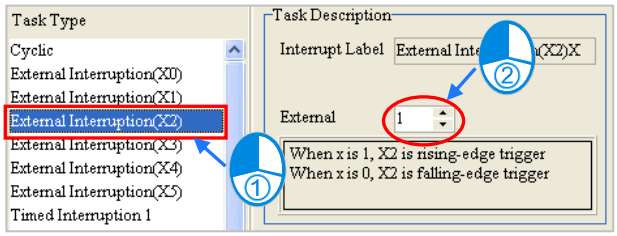
Owing to the fact that **Reset** will be executed after an external interrupt is triggered, **Reset** is removed from the **Assigned POUs** section for **Cyclic**.



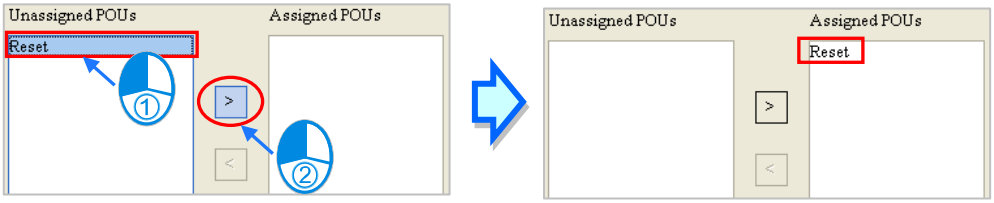
After the assignment of the POUs to **Cyclic** is complete, users can arrange the POUs in the **Assigned POUs** section. Owing to the fact that **Run_Stop** and **Shift** must be executed earlier than **LED**, the order in which **Run_Stop**, **Shift**, and **LED** are listed in the **Assigned POUs** section is **Run_Stop→Shift→LED**.



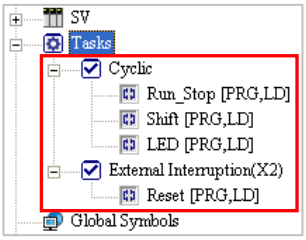
Select **External Interruption (X2)** in the **Task Type** section, and select 1 in the **External** box in the **Task Description** section. **External Interruption (X2)** is triggered when the signal passing through X2 goes from low to high.



Assign **Reset** to **External Interruption (X2)**. Owing to the fact that there is only one POU in the **Assigned POUs** section, the users do not need to arrange the POU.



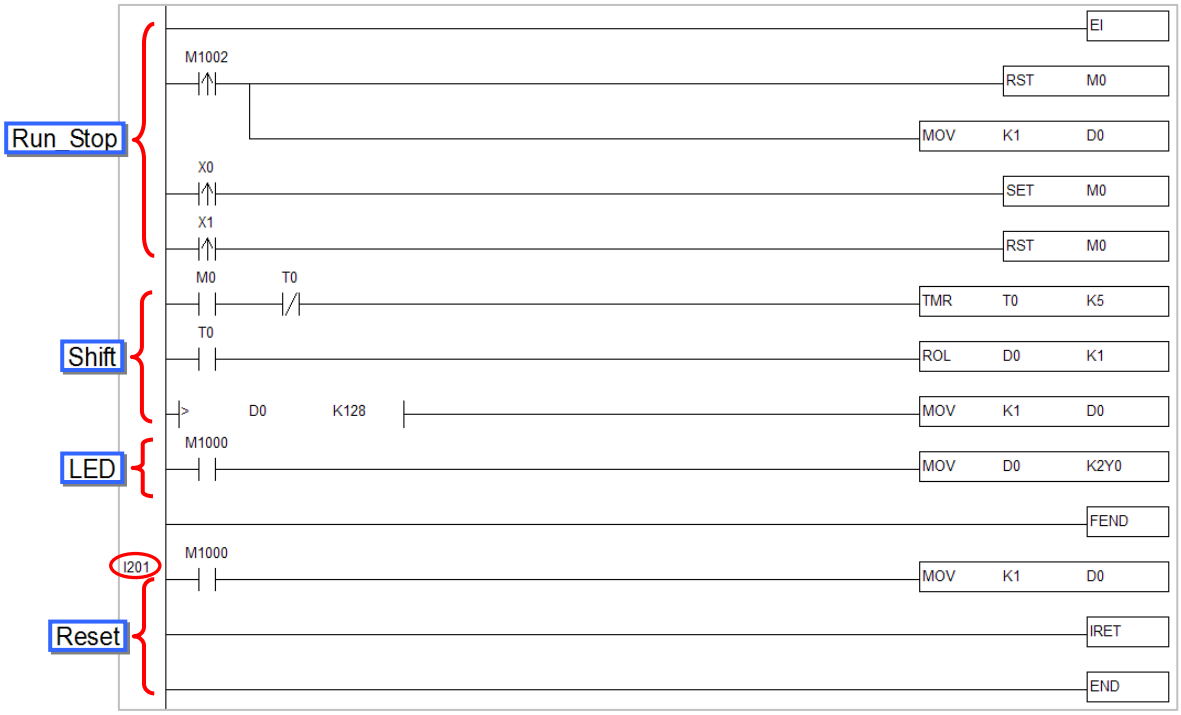
After **OK** is clicked, the assignment and the setting in the **Task Manager** window will be applied. The assignment is shown below. The writing of the program has been completed so far. The users can try to compile the program, download the program to the PLC, and test the program by themselves.



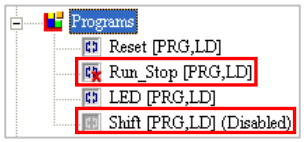
Additional remark

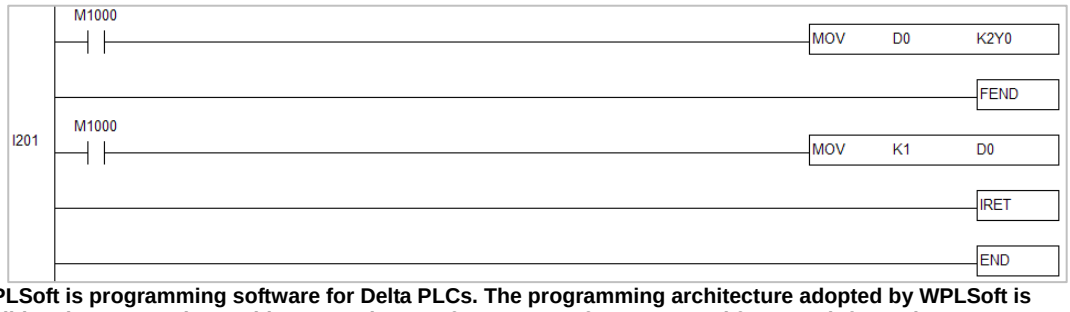
After the program above is compiled successfully and downloaded to the PLC, the users can upload the program in the PLC to WPLSoft. The program uploaded to WPSoft is shown below.

The main program is composed of the three POUs which are assigned to the cyclic task. The order in which the the POUs are executed is **Run_Stop→Shift→LED**. The contents of the interrupt subroutine I201 are the same as the contents of the POU **Reset**.



The users can remove **Run_Stop** from the **Tasks** section, and disable **Shift**. After the program is compiled and downloaded to the PLC, the users can upload the program in the PLC to WPLSoft. The main program will not include the two POUs **Run_Stop** and **Shift**.



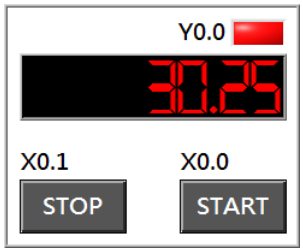


***. WPLSoft is programming software for Delta PLCs. The programming architecture adopted by WPLSoft is traditional programming architecture. Please refer to WPLSoft User Manual for more information.**

5.6.3 Example of an Interrupt Subroutine in an AH500 Series CPU Module

- **Example**

  If the START button is pressed (X0.0 is turned ON), the system will begin to count. The STOP button must be pressed (X0.1 must be turned ON) within 30 seconds. Otherwise, the alarm will ring (Y0.0 will be turned ON).



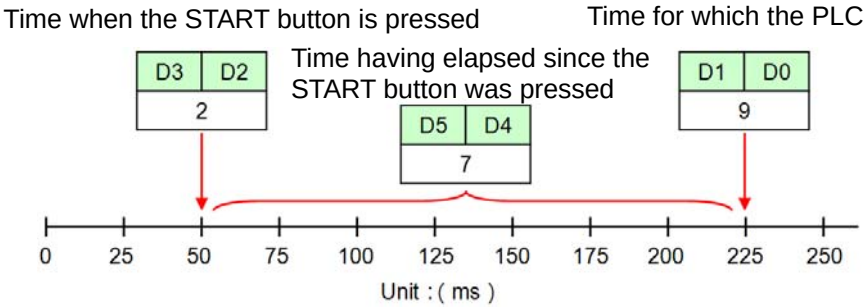
- **Hardware configuration**

  In this example, the CPU module AHCPU530-EN, the digital I/O module AH16AP11R-5A, and the four-slot main backplane AHBP04M1-5A are used. (The hardware configuration here is the same as the hardware configuration in chapter 4.)

- **Planning a program**

  In this example, a timed interrupt is triggered every 25 milliseconds, and time intervals are measured through a program. In the figure below, **Timed Interruption (0)** is triggered every 25 milliseconds, and the value in the 32-bit register (D1, D0) increases by one whenever **Timed Interruption (0)** is triggered. As long as the PLC keeps running, **Timed Interruption (0)** is triggered every 25 milliseconds, and the value in the 32-bit register (D1, D0) increases by one whenever **Timed Interruption (0)** is triggered. Users can calculate the time for which the PLC has run by reading the value in (D1, D0). For example, if the value in (D1, D0) is 10, the time for which the PLC has run is 250 milliseconds (10x25 ms).

  After the START button is pressed, the present value in (D1, D0) will be transferred to (D3, D2), and the value in (D1, D0) will increase. After the present value in (D1, D0) is transferred to (D3, D2), the value in (D3, D2) will remain unchanged. Users can calculate the time which has elapsed since the START button was pressed by subtracting the value in (D3, D2) from the value in (D1, D0). The value gotten is stored in (D5, D4).



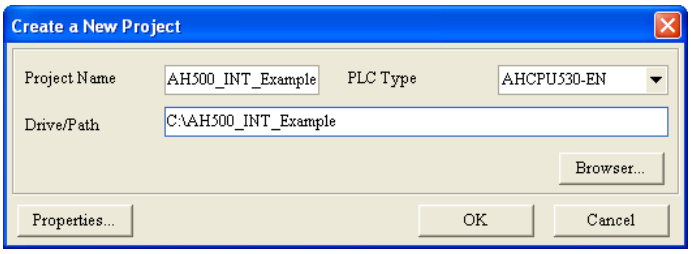
A timed interrupt in an AH500 CPU module is illustrated with the example in this section. The instructions TKON and TKOFF are also illustrated with the example in this section. Five POUs are created. The functions of the POUs are described below.

POU name	Function
INT_Timer	The value in the 32-bit register (D1, D0) increases by one every 25 milliseconds. (The POU is assigned to **Timed Interruption (0)**.)
Initialize	Reset the flags M0 and M1, transfer the value 0 to the 32-bit registers (D1, D0), (D3, D2), (D5, D4), and (D7, D6), and enable the interrupt. During the first scan cycle, **Cyclic (0)** to which the POUs are assigned is enabled, and **Cyclic (1)** to which **Initialize** is assigned is disabled. As a result, **Initialize** is only executed during the first cycle.
Control	When X0.0 is turned form OFF to ON, the operation flag M0 is set to ON, and the value in (D1, D0) is transferred to (D3, D2). When X0.1 is turned from OFF to ON, or when an overflow occurs, the operation flag M0 is reset.
Time_CHK	Users can calculate the time which has elapsed since the START button was pressed by subtracting the value in (D3, D2) from the value in (D1, D0). The value gotten is stored in (D5, D4). If the value gotten is less than 0, the overflow flag M1 is set to ON.
Signal	If the time which has elapsed is more than 30 seconds (25 ms x1200), or the overflow flag M1 is ON, the alarm Y0.0 will be set to ON.

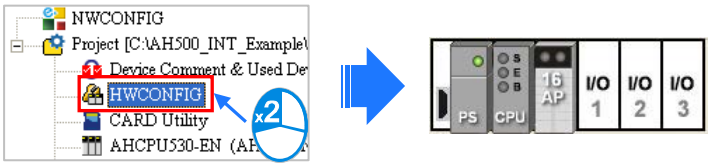
***. Please refer to AH500 Operation Manual and AH500 Programming Manual for more information about the interrupts in an AH500 series CPU module and the related applied instructions.**

- **Creating the program**

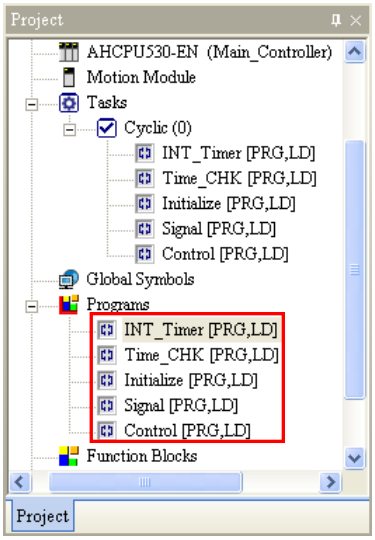
  Please create a project for AHCPU530-EN.



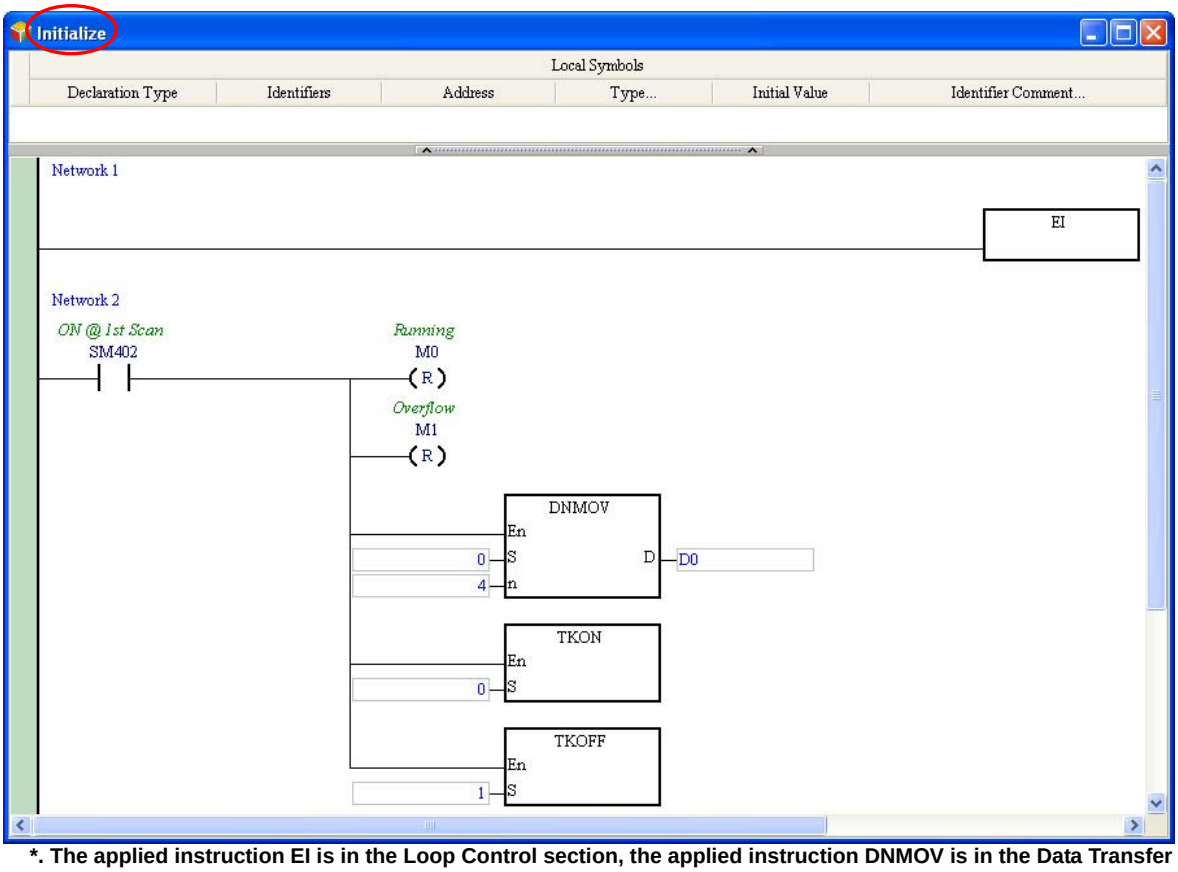
The hardware configuration here is the same as the hardware configuration in chapter 4. Please refer to section 4.4, and complete the hardware configuration and the parameter setting.



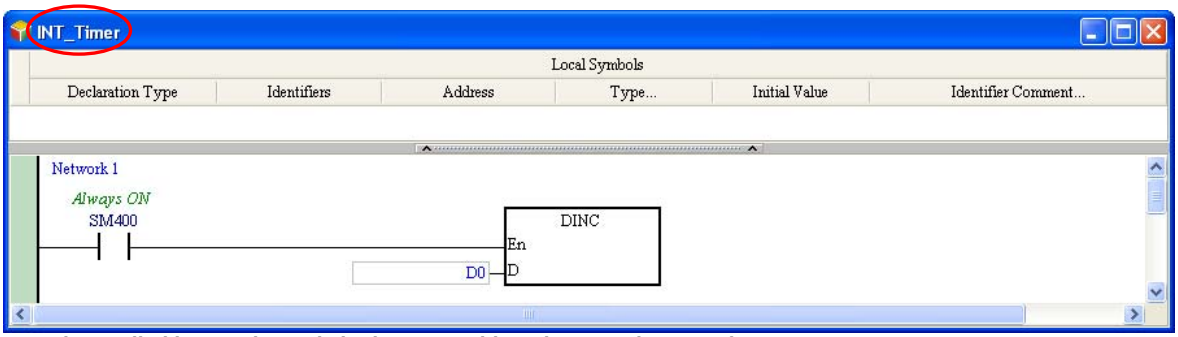
Create the five POUs in the **Programs** section in the project management area shown below. When these POU are created, they are assigned to **Cyclic (0)**.



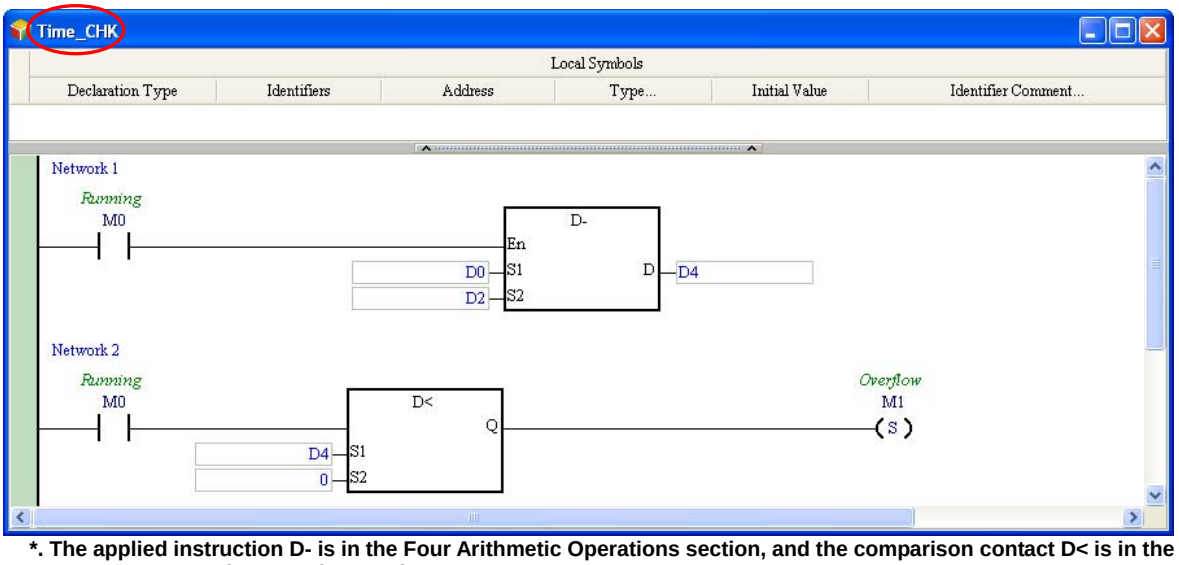
Write the programs in the POUs shown below.



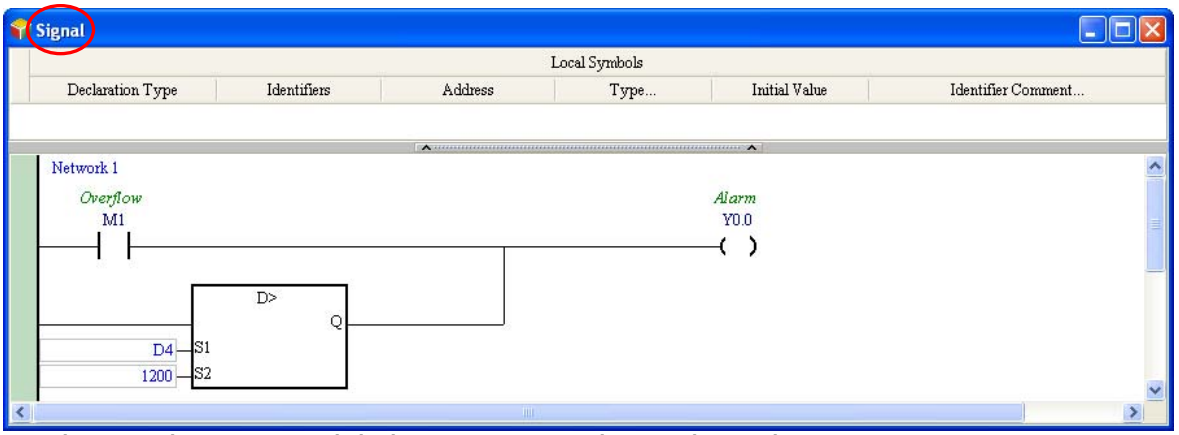
***. The applied instruction EI is in the Loop Control section, the applied instruction DNMOV is in the Data Transfer section, and the applied instructions TKON and TKOFF are in the Task Control section.**



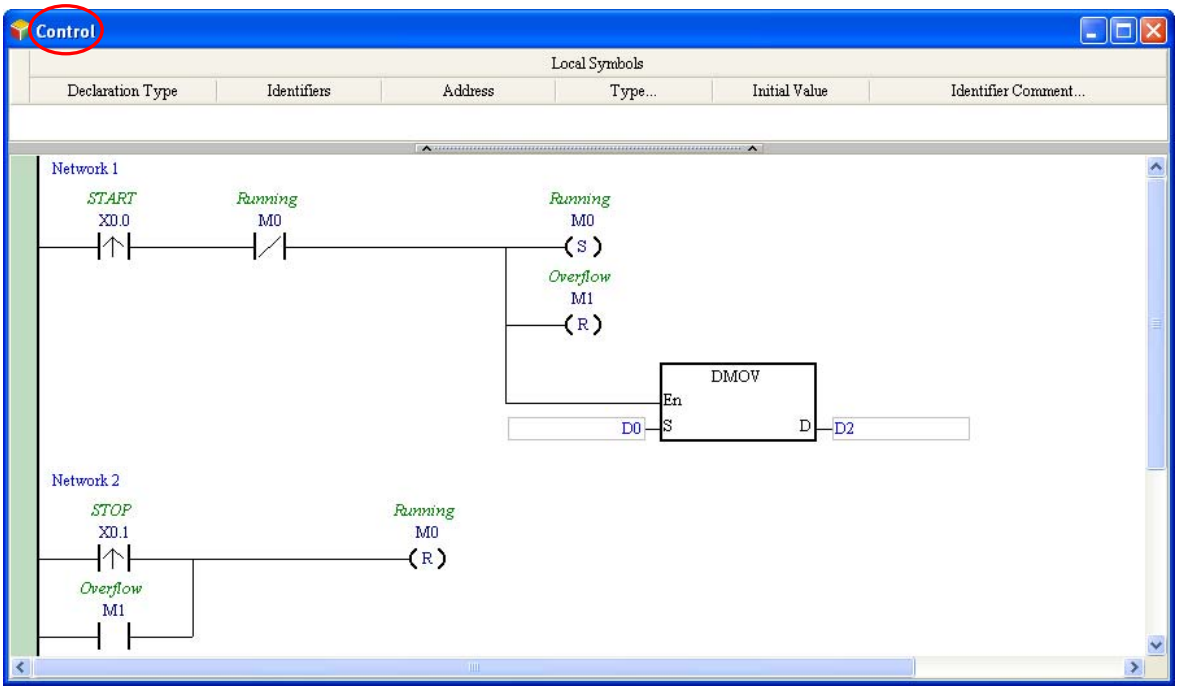
***. The applied instruction EI is in the Four Arithmetic Operations section.**



***. The applied instruction D- is in the Four Arithmetic Operations section, and the comparison contact D< is in the Contact Type Logic Operation section.**

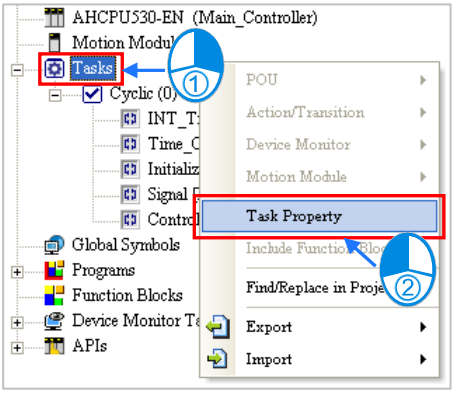


***. The comparison contact D> is in the Contact Type Logic Operation section.**

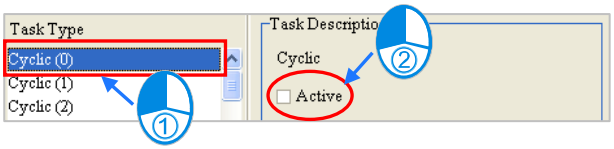


***. The applied instruction DMOV is in the Data Transfer section.**

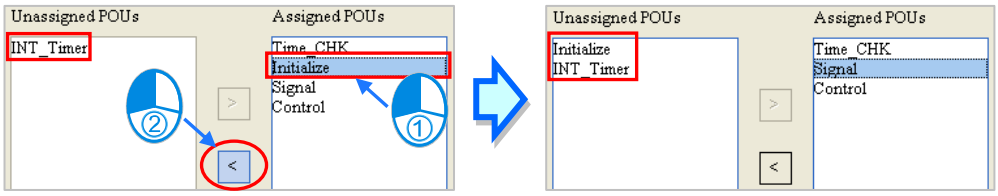
Open the **Task Manager** window after the writing of the program is complete. Owing to the fact that the five POUs are assigned to **Cyclic (0)** when they are created, the five POUs are in the **Assigned POUs** section for **Cyclic (0)**.



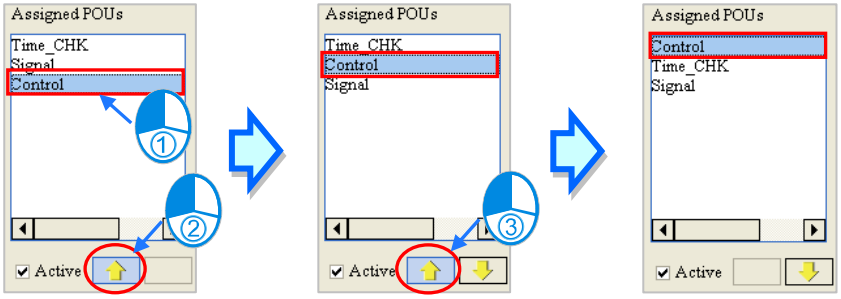
Set the property of **Cyclic (0)** first. Owing to the fact that **Cyclic (0)** is enabled by **Initialize**, the **Active** checkbox is unselected.



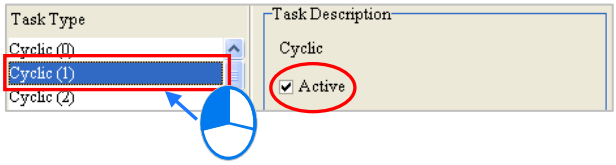
INT_Timer must be assigned to **Timed Interruption (0)**, and **Initialize** must be assigned to **Cyclic (1)**. As a result, **INT_Timer** and **Initialize** are removed from the **Assigned POUs** section for **Cyclic (0)**.



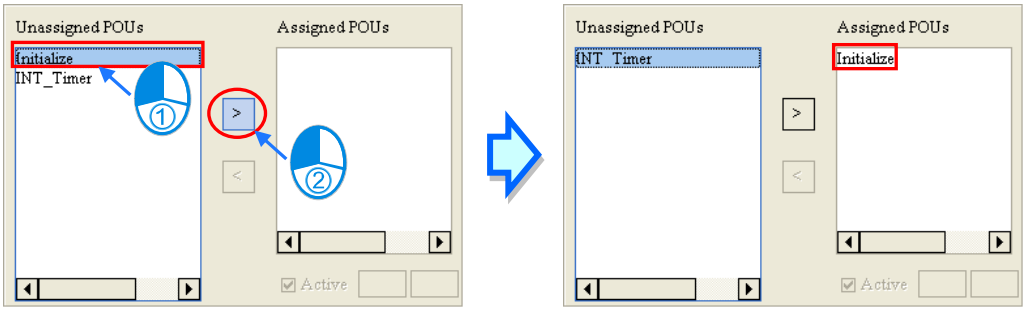
After the assignment of the POUs to **Cyclic (0)** is complete, users can arrange the POUs in the **Assigned POUs** section. Owing to the fact that **Control** and **Time_CHK** must be executed earlier than **Signal**, the order in which **Control**, **Time_CHK**, and **Signal** are listed in the **Assigned POUs** section is **Control→Time_CHK→Signal**.



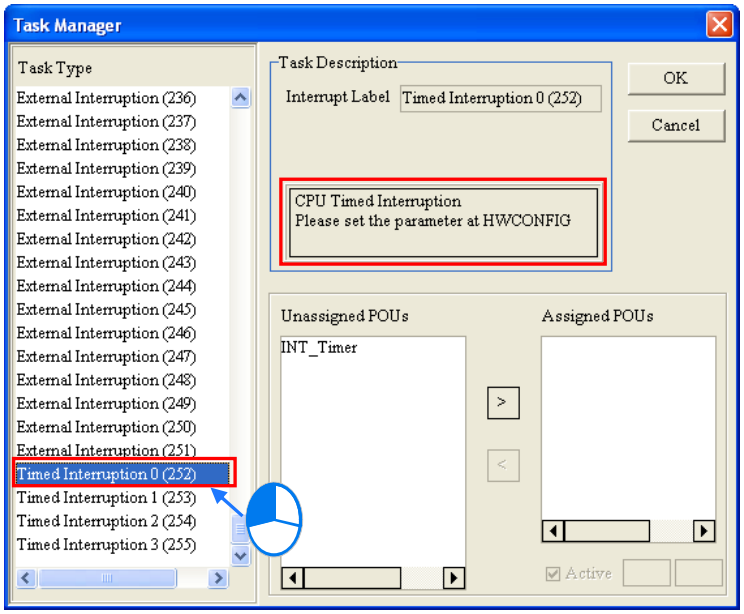
Select **Cyclic (1)** in the **Task Type** section. Owing to the fact that **Cyclic (1)** will be disabled after the first scan cycle is complete, the **Active** checkbox is selected.



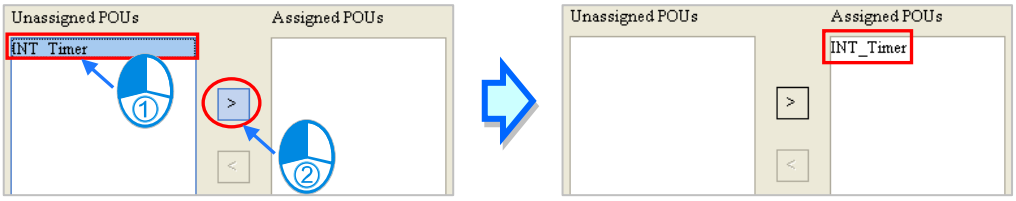
Assign **Initialize** to **Cyclic (1)**. Owing to the fact that there is only one POU in the **Assigned POUs** section, the users do not need to arrange the POU.



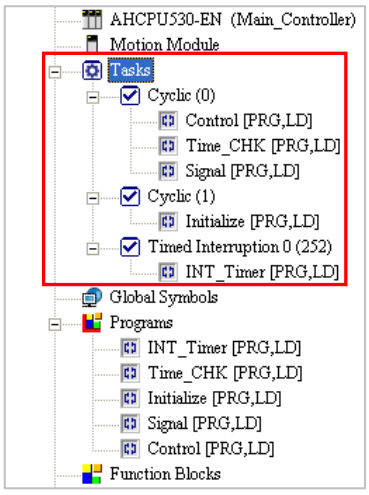
Select **Timed Interruption 0 (252)** in the **Task Type** section. The message in the **Task Description** section indicates that the condition of **Timed Interruption 0 (252)** can be set in HWCONFIG. The users have to assign a POU to this task, and set the property of this task.



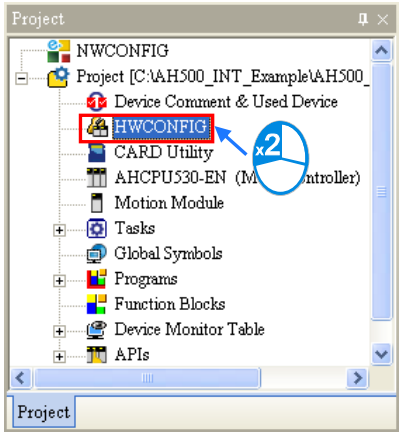
Assign **INT_Timer** to **Timed Interruption 0 (252)**.



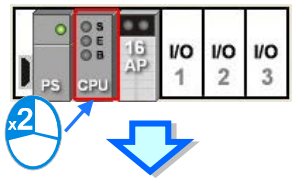
After **OK** is clicked, the assignment and the setting in the **Task Manager** window will be applied. The assignment is shown below.

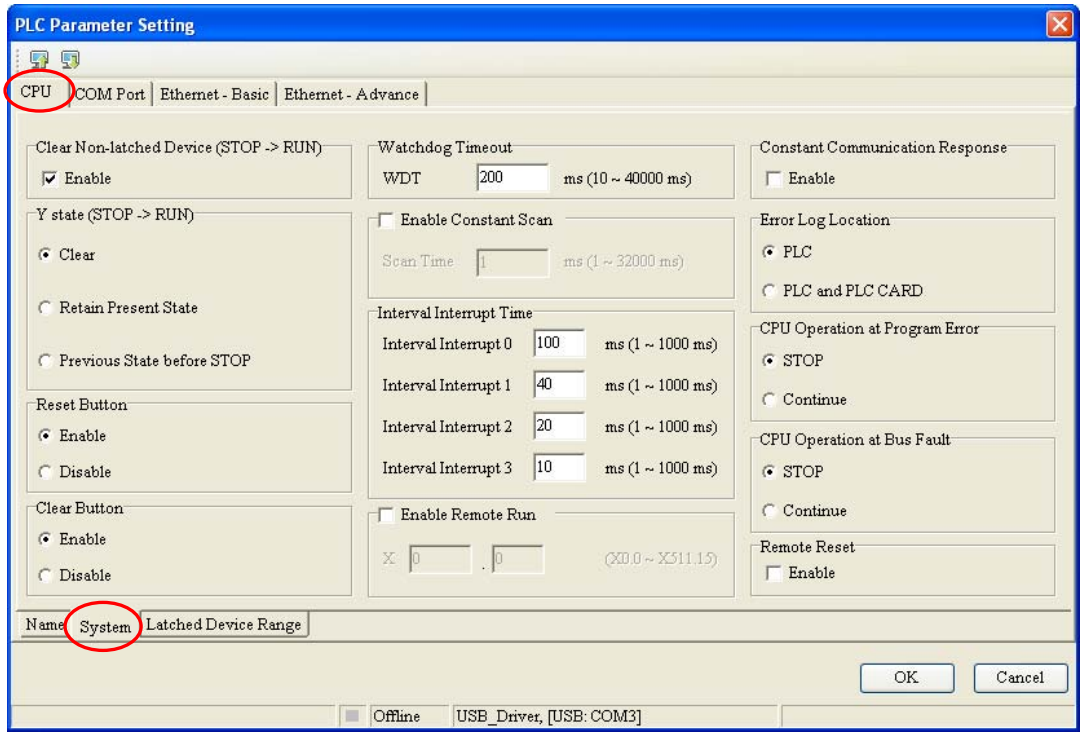


In order to set the property of **Timed Interruption 0 (252)**, HWCONFIG is started.

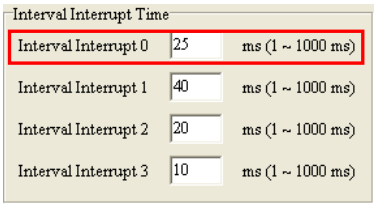


After HWCONFIG is started, the users can double-click the CPU module to open the **PLC Parameter Setting** window. Click the **CPU** tab at the top of the window, and the **System** tab at the bottom of the window.



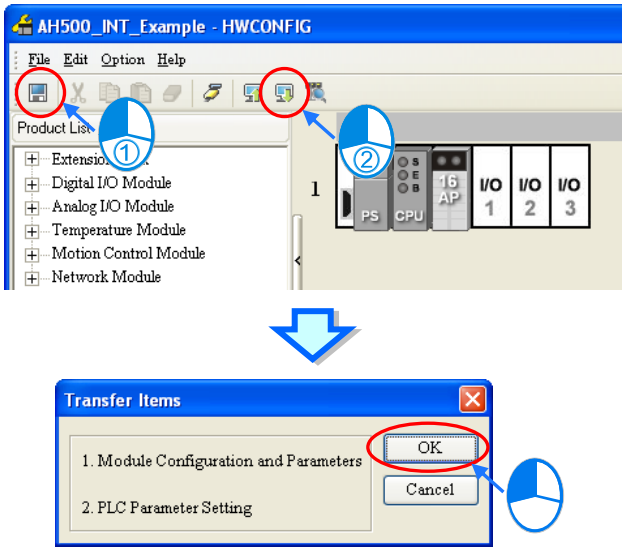


Find the **Interval Interrupt Time** section. Type 25 in the **Interval Interrupt 0** box, and then click **OK**.



Be sure to save the file after the setting is complete. The parameters set in HWCONFIG must be

downloaded to the CPU module so that they can take effect. Please refer to chapter 3 for more information about HWCONFIG.



The writing of the program has been completed so far. The users can try to compile the program, download the program to the CPU module, and test the program by themselves.

MEMO

Chapter 6 Symbols

Table of Contents

6.1 Introduction of Symbols 6-2
6.1.1 Application of Symbols and Creation of Identifiers 6-2
6.1.2 Symbol Classes 6-2
6.1.3 Data Types 6-3
6.1.4 Assigning a Device to a Symbol and Setting the Initial Value of a Symbol 6-5
6.1.5 Modifying a Symbol with an Index Register 6-6
6.1.6 Bit in a Device Represented by a Symbol (Only for AH500 Series CPU Modules) 6-7
6.2 Managing the Symbols in ISPSoft 6-8
6.2.1 Symbol Tables 6-8
6.2.2 Adding a Symbol 6-9
6.2.3 Using a Symbol Whose Data Type is ARRAY or STRING 6-13
6.2.4 Modifying a Symbol and Editing a Symbol Table 6-15
6.2.5 Clearing the Device Addresses Assigned to the Symbols 6-16
6.2.6 Downloading the Initial Values of the Symbols 6-16
6.2.7 Exporting a Symbol Table 6-17
6.2.8 Importing a Symbol Table 6-17
6.2.9 Arranging the symbols 6-20
6.2.10 Filtering the Symbols 6-21
6.2.11 Setting a Range of Devices (Only for DVP Series PLCs) 6-22
6.3 Example 6-23
6.3.1 Planning a Symbol Table 6-23
6.3.2 Writing a Program 6-24

6.1 Introduction of Symbols

During the process of developing a traditional program for a PLC, it generally takes much time to manage device addresses. Besides, managing or debugging the program in a big project is a burden on users. As a result, the concept of symbols in a high-level programming language is introduced into IEC 61131-3. A device in a PLC can be represented by a symbol, and a device can be automatically assigned to a symbol. The time of assigning devices is saved, a program is more readable, and the efficiency of developing a program increases.

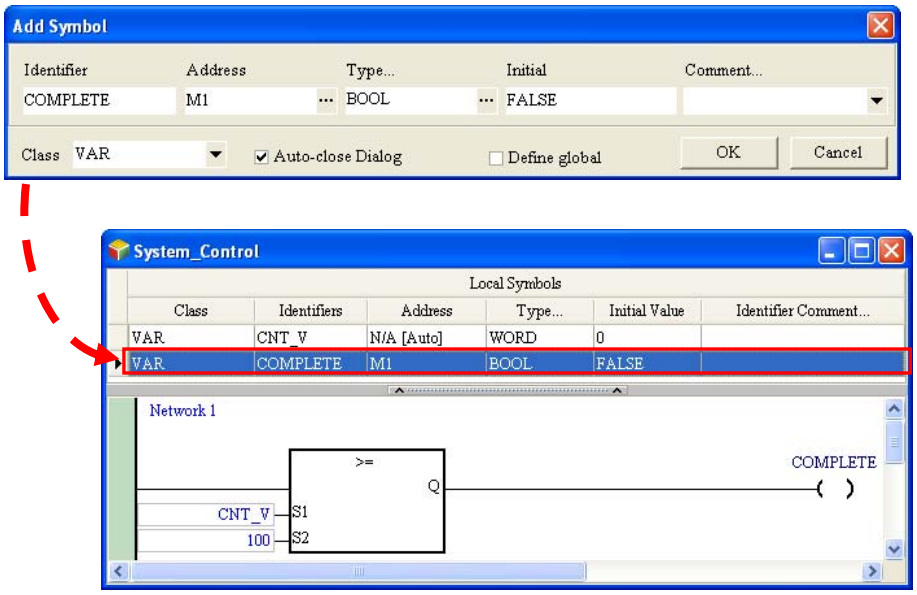
***. Variables in ISPSoft are called symbols. As a result, variables are the same as symbols in terms of meaning in this manual.**

6.1.1 Application of Symbols and Creation of Identifiers

A symbol has to be declared before it is used, as shown below. There are two types of symbols. They are global symbols and local symbols. The global symbols can be used in all the POUs in a project, and the local symbols can only be used in the POU in which the local symbols are declared. Besides, the identifier of a local symbol in a POU can be the same as the identifier of a local symbol in another POU. However, if the identifier of a local symbol declared is the same as the identifier of a global symbol, the system will automatically regard the local symbol declared in a POU as a local symbol.

The regulations of creating the identifier of a symbol are as follows.

- An identifier is composed of 30 characters at most, and a Chinese character occupies two characters.
- The identifier of a symbol can not be a name reserved by the system, e.g. an instruction code, a device name, or a name given a special significance. However, if a name reserved by the system is a part of the identifier of a symbol, the identifier is a legal name. For example, “M0” is an illegal name, but “_M0” is a legal name.
- Spaces can not be used. For example, “INPUT CH0” is an illegal name.
- Underlines can be used, but they can not be used continuously or put at the end of an identifier. For example, “INPUT_CH0” is a legal identifier, but “INPUT__CH0” and “INPUT_CH0_” are illegal identifiers.
- Special marks can not be used. For example, *, #, ?, \, %, @, and etc. can not be used.



6.1.2 Symbol Classes

In terms of functions, symbols can be classified into five classes. The characteristics of these five classes are described below.

- **VAR - General symbol**

  The symbols of this class are for general operations only. The significance of a symbol of this class depends on the data type of the symbol or the corresponding device.

- **Latched symbol**

  Only AH500 series modules support the symbols of this class. Users can not declare the symbols of this class in a function block. The function of the symbols of the VAR_RETAIN class is the same as the function of the symbols of the VAR class. The difference between the symbols of the VAR_RETAIN class and the symbols of the VAR class lies in the fact that the devices automatically assigned to the symbols of the VAR_RETAIN class are latched devices. As a result, the values of the symbols of this class will be retained after the PLC is disconnected.

  ***. Users can not assign a device to a symbol of this class. Besides, the data type can not be TIMER, COUNTER, or STEP.**

- **VAR_INPUT - Symbol used as an input pin of a function block**

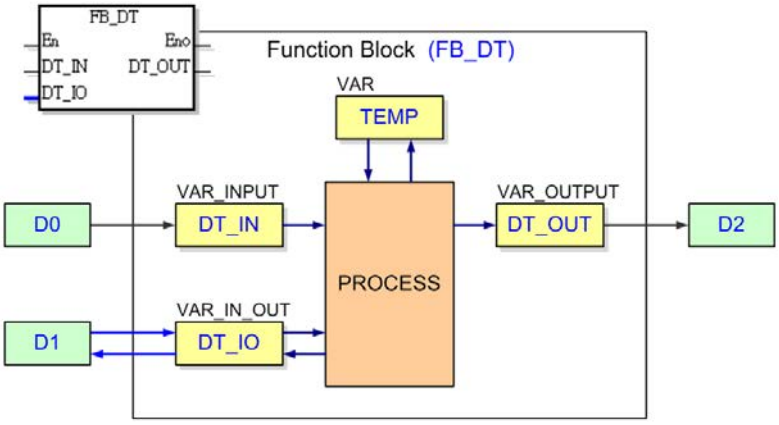
  A symbol of this class is used as an input pin of a function block, and is declared in the function block. If a function block is called, the symbol of this class can receive the input value from the caller. Besides, in a ladder diagram/function block diagram, the symbols of this class are put at the left sides of the function blocks, and the pins which receive the values from the caller are assigned to the symbols of this class.

- **VAR_OUTPUT - Symbol used as an output pin of a function block**

  A symbol of this class is used as an output pin of a function block, and is declared in the function block. After the execution of a function block is complete, the operation result will be sent to the caller through the symbol of this class. Besides, in a ladder diagram/function block diagram, the symbols of this class are put at the right sides of the function blocks, and the pins which sent the operation results to the caller are assigned to the symbols of this class.

- **VAR_IN_OUT - Symbol used as a feedback pin of a function block**

  A symbol of this class is used as a feedback pin of a function block, and is declared in the function block. Please refer to the following example. When the function block is called, the caller sends the value in D1 to DT_IO, which is a symbol of the VAR_IN_OUT class. After the operation comes to an end, the final value of DT_IO is sent to D1. Besides, in a ladder diagram/function block diagram, the symbols of this class are put at the right sides of the function blocks, and the blue pins which connect to the caller are assigned to the symbols of this class.



6.1.3 Data Types

The data type of a symbol determines the significance of the value of symbol. Suppose there are two symbols VAR_1 and VAR_2. The data type of VAR_1 is BOOL, and the data type of VAR_2 is WORD. When VAR_1 and VAR_2 are used in a program, VAR_1 represents a contact, and VAR_1 represents a 16-bit device which can be involved in arithmetic or data transfer.

The data types supported by ISPSoft are listed below. Different models support different data types, and the symbols which can be declared in a POU of the program type are different from the symbols which can be declared in a POU of the function block type.

Data type	Description	AH500		DVP	
		Program	FB	Program	FB
BOOL	Boolean data type A Boolean value represents a state of a contact	✓	✓	✓	✓
WORD	16-bit value 16-bit data can be stored.	✓	✓	✓	✓
DWORD	32-bit value 32-bit data can be stored.	✓	✓	✓	✓
LWORD	64-bit value 64-bit data can be stored.	✓	✓	✓	✓
INT	16-bit signed integer The highest bit represents a positive/negative sign.	✓	✓		
DINT	32-bit signed integer The highest bit represents a positive/negative sign.	✓	✓		
LINT	64-bit signed integer The highest bit represents a positive/negative sign.	✓	✓		
REAL	32-bit floating-point value A decimal is represented by a 32-bit value.	✓	✓	✓	✓
LREAL	64-bit floating-point value A decimal is represented by a 64-bit value.	✓	✓		
ARRAY	Array data type When a symbol is declared, the size of the array and the array type which will be stored are specified. (An array is composed of 2048 elements at most.)	✓	✓	✓	✓
STRING	String data type An ASCII code is represented by an 8-bit value. When a symbol is declared, the length of the string is specified. (A string is composed of 128 ASCII codes at most.) (*3)	✓	✓		
STEP	It is used as a recognition flag for a step. (*1) (*3)	✓		✓	✓
Function Block	It represents the name of a function block. (*2)	✓	✓	✓	
COUNTER	Counter data type It represents a counter. (*3)	✓		✓	✓
TIMER	Timer It represents a timer. (*3)	✓		✓	✓
POINTER	It is a pointer for a word device. (*4)		✓		
T_POINTER	It is a pointer for a timer. (*4)		✓		
C_POINTER	It is a pointer for a counter.		✓		
HC_POINTER	It is a pointer for a high-speed counter. (*4)		✓		

***1. Please refer to chapter 12 for more information about steps.**

***2. A symbol in a function block has a special significance. Please refer to chapter 7 for more information.**

***3. The data type of a symbol of the VAR_INPUT class, the VAR_OUTPUT class, or the VAR_IN_OUT class can not be STRING, STEP, COUNTER, or TIMER.**

***4. Please refer to chapter 7 for more information about pointers.**

6.1.4 Assigning a Device to a Symbol and Setting the Initial Value of a Symbol

A device is assigned to a symbol according to the data type of the symbol. Users can set the initial value of a symbol. If a project is downloaded, the initial values of the symbols can be written into the devices assigned to the symbols.

The principle of assigning devices to symbols is as follows. Different models have different assignment principles.

Model	Assignment principle
AH500	● Users can assign devices to the global symbols and the local symbols declared in the POUs of the program type. The system can also automatically assign devices to the global symbols and the local symbols declared in the POUs of the program type. ● The system automatically assigns devices to the local symbols declared in the function blocks. Users can not assign devices to the local symbols declared in the function blocks by themselves. ● The devices assigned by the system are memories reserved by the CPU module, and are not data registers or auxiliary relays. If the data type of a symbol is STEP, TIMER, or COUNTER, the system will automatically assign a stepping relay, a timer, or a counter to the symbol.
DVP	● Users can assign devices to all the global symbols and the local symbols. The system can also automatically assign devices to all the global symbols and the local symbols. ● The devices assigned by the system are usable devices. (Users can set a range of devices which can be assigned automatically.)

The relation between the data types and the device types which can be assigned is described below.

Data type	AH500		DVP	
	Device assigned by users	Device assigned by the system	Device assigned by users	Device assigned by the system
BOOL	Contact M/SM or bit in the device D/L/X/Y	Internal memory	Contact M/X/Y	Contact M
WORD	D/L/X/Y/E/SR	Internal memory	D/E/F	D
DWORD	D/L/X/Y/E/SR	Internal memory	D/E/F	D
LWORD	D/L/X/Y/E/SR	Internal memory	D/E/F	D
INT	D/L/X/Y	Internal memory	None	
DINT	D/L/X/Y	Internal memory	None	
LINT	D/L/X/Y	Internal memory	None	
REAL	D/L/X/Y	Internal memory	D	D
LREAL	D/L/X/Y	Internal memory	None	
STRING	D/L/X/Y	Internal memory	None	
STEP	S	S	S	S
COUNTER	C/HC	C	C	C
TIMER	T	T	T	T
ARRAY	The devices assigned to a symbol whose data type is ARRAY depend on the array type specified. The array is composed of the devices starting from the device assigned by users or the system, and the number of devices in the array conforms to the size of the array. The devices assigned to a symbol whose data type is ARRAY can not be SR/SM/E/F devices.			

***1. If the device assigned to a symbol is an internal memory in an AH500 series CPU module, users can not know the actual device address.**

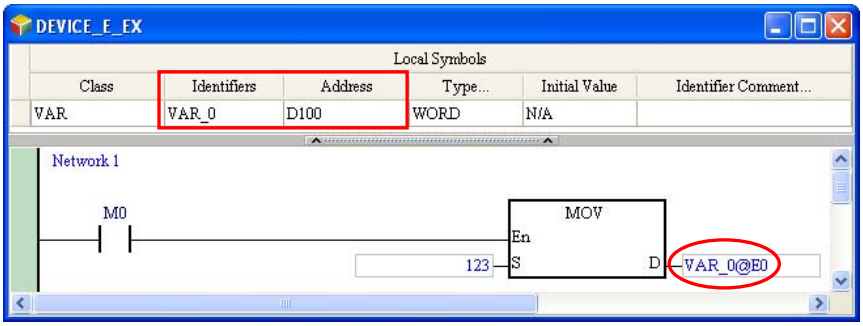
***2. Please refer to section 6.2.3 for more information about ARRAY and STRING.**

***3. Please refer to section 6.2.11 for more information about setting a range of devices which can be assigned automatically in a DVP series PLC.**

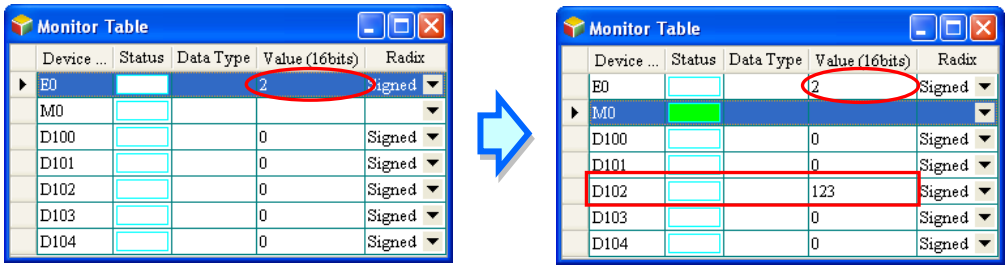
***4. A symbol in a function block has a special significance. Please refer to chapter 7 for more information.**
***5. For example, X0.0 and Y0.1 are bits in the word devices X and Y. Please refer to appendix B for more information.**

6.1.5 Modifying a Symbol with an Index Register

In ISPSoft, a symbol can be modified by an index register. The format is **Identifier@Index register**. An index register can be an E device, an F device, or a symbol to which an E device or an F device is assigned.

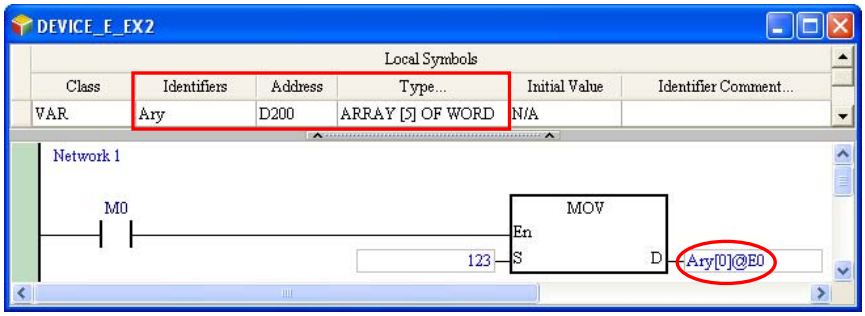


The data stored in an index register indicates the offset for an object modified. Take the figure above for example. If the value in the index register E0 is 2, VAR_0@E0 indicates that 2 is added to the device address (D100) assigned to VAR_0, that is, VAR_0@E0 represents D102. When M0 is ON, the value 123 is transferred to the data register D102.



The same rule applies to a symbol whose data type is ARRAY. The format is **Identifier[Index]@Index register**. The index must be a constant. If the index is a symbol, the symbol whose data type is ARRAY can not be modified by the index register.

In the figure below, Ary is an array composed of five elements, and the start device address is D200. When the system compiles the program, D200~D204 are assigned Ary. If the value in the index register E0 is 2, Ary[0]@E0 indicates that 2 is added to the device address (D200) assigned to Ary[0], that is, Ary[0]@E0 represents D202, the device address assigned to Ary[2]. If the value in the index register E0 is 6, Ary[0]@E0 represents D206. D206 is not within the range of devices assigned to the array, but this usage is legal. Owing to the fact that the data stored in an index register indicates the offset for an object modified, users have to be more careful when they modify a symbol whose data type is ARRAY with an index register. Please refer to section 6.2.3 for more information about the usage of arrays.



An AH500 series CPU module differs from a DVP series PLC in the use of an index register to modify an operand. Please refer to the following table.

Application	Example	AH500	DVP
Modifying an operand in an instruction	MOV 100 VWD@E0	Depending on the rules of the instructions	Depending on the rules of the instructions
Modifying a contact or coil	LD VBL_0@E0 OUT VBL_1@E1	Supporting the modification	Depending on the models
Modifying a symbol whose data type is ARRAY	ARY[0]@E0	Supporting the modification	Supporting the modification

***1. In an AH500 series CPU module, only the E devices are index registers. In a DVP series PLC, the E devices and the F devices are index registers. Please refer to the operation manual for a model for more information.**

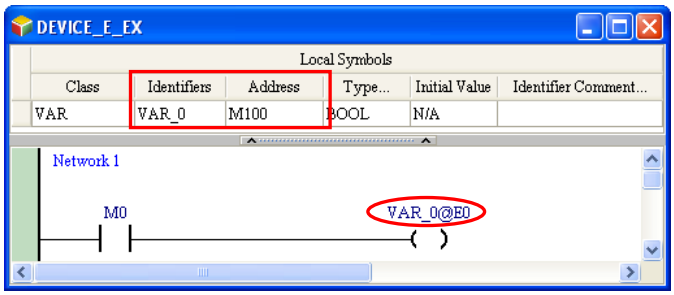
***2. An index register can not be modified by another index register. For example, E0@E1 is not allowed. As a result, before a symbol is modified by an index register, users have to make sure that the device assigned to the symbol is not an E device or an F device.**

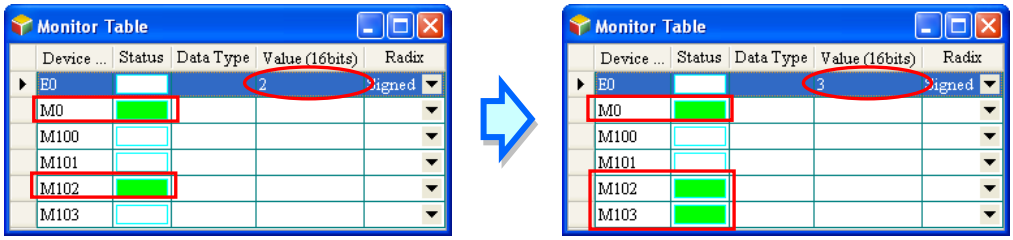
Additional remark

(a) The data stored in an index register indicates the offset for a device modified. If the system automatically assigns a device to a symbol modified, the use of an index register will cause the program to be executed incorrectly because users do not know which device is assigned to the symbol.

(b) If the addition of the value in an index register to a device address produces a device address which is not within the range, the PLC will operate incorrectly. As a result, users have to be more careful when they modify a symbol with an index register. Take AHCPU530-EN for example. If the value in E0 is larger than 35, the operand D65500@E0 represents a device which is not within the range of data registers, and an error occurs in the CPU module.

(c) If the value in an index register is changed, the device which actually operates differs from the original device. As a result, if the original device is not used in the program, the final value in the original device is retained. Take AHCPU530-EN for example. In the figure below, if the value in E0 is 2. M102 is ON when M0 is ON. If the value in E0 is changed from 2 to 3, the device which actually operates becomes M103. M103 is ON when M0 is ON. Owing to the fact that M102 is not used in the program, M102 remains ON.



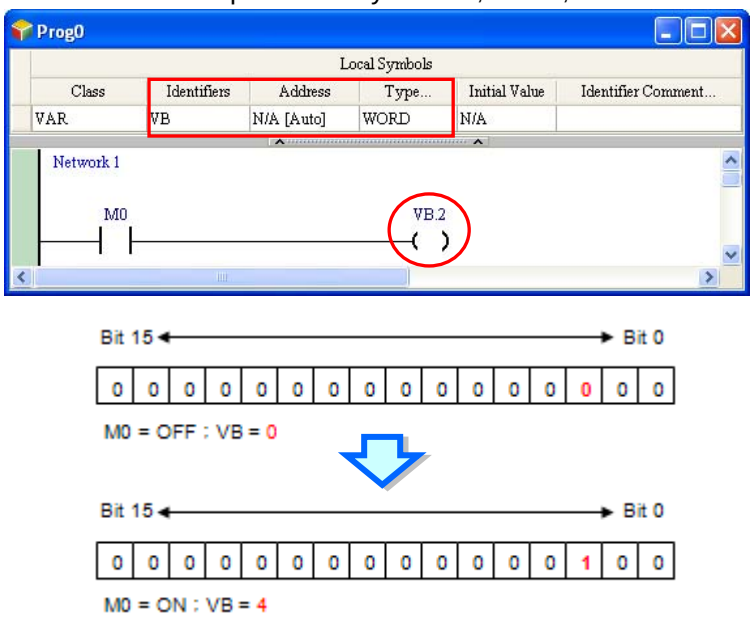


6.1.6 Bit in a Device Represented by a Symbol (Only for AH500 Series CPU Modules)

An AH500 series CPU module supports the manipulation of the bits in a word device. The format is **Word device.Bit number**. For example, D0.2 indicates that bit 2 in D0 is manipulated. The same rule applies to the symbols. The format is **Identifier.Bit number**, e.g. VB.2. The rule also applies to the symbols whose data type is ARRAY. The format is **Identifier[Index].Bit number**, e.g. Ary[0].1. The bit number must be a decimal constant. No matter what the data type of a symbol modified is,

the bit number must be within the range between 0 and 15. Besides, the bits in a device represented by a symbol whose data type is BOOL, STEP, COUNTER, TIMER, a function block, HC_POINTER, C_POINTER, or T_POINTER can not be manipulated.

In the figure below, the data type of VB is WORD. Suppose the present value of VB is 0. When M0 is ON, the value of bit 2 in the device represented by VB is 1, that is, the value of VB is 4.



In an AH500 series CPU module, an index register can be used with the manipulation of the bits in a device. If an index register is used with the manipulation of the bits in a device, the mark @ has high priority. Please refer to the following example. The data type of VB is WORD.

Example 1: VB.1@E0 (VB represents D100. The value in E0 is 3.)

1@E0 is interpreted first. As a result, VB.1@E0 is equivalent to VB.4. The bit which is actually manipulated is bit 4 in D100.

Example 2: VB@E0.1 (VB represents D100. The value in E0 is 3.)

VB@E0 is interpreted first. As a result, VB@E0.1 represents D103.1. The bit which is actually manipulated is bit 1 in D103.

Example 3: VB@E0.1@E1 (VB represents D100. The value in E0 is 3, and the value in E1 is 2.)

VB@E0 and 1@E1 are interpreted separately. As a result, VB@E0 represents D103, and 1@E1 represents 3. VB@E0.1@E1 represents D103.3. The bit which is actually manipulated is bit 3 in D103.

***1. If a bit in a device is manipulated, the system will faithfully get the state of the bit. Users have to judge whether the state of the bit is significant by themselves.**

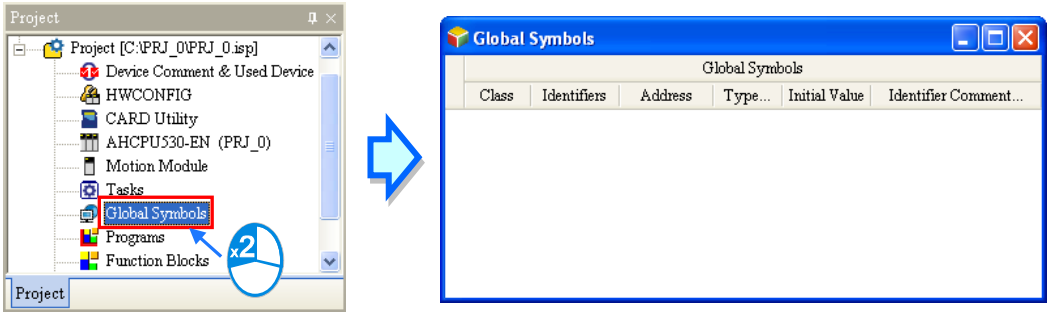
***2. In the present version of ISPSoft, a bit number must be within the range between 0 and 15. However, in the future version of ISPSoft, the data length of a symbol will allow the bit number to be within the range between 0 and 31, or between 0 and 63, and a symbol will represent a bit, e.g. VB.VC.**

6.2 Managing the Symbols in ISPSoft

6.2.1 Symbol Tables

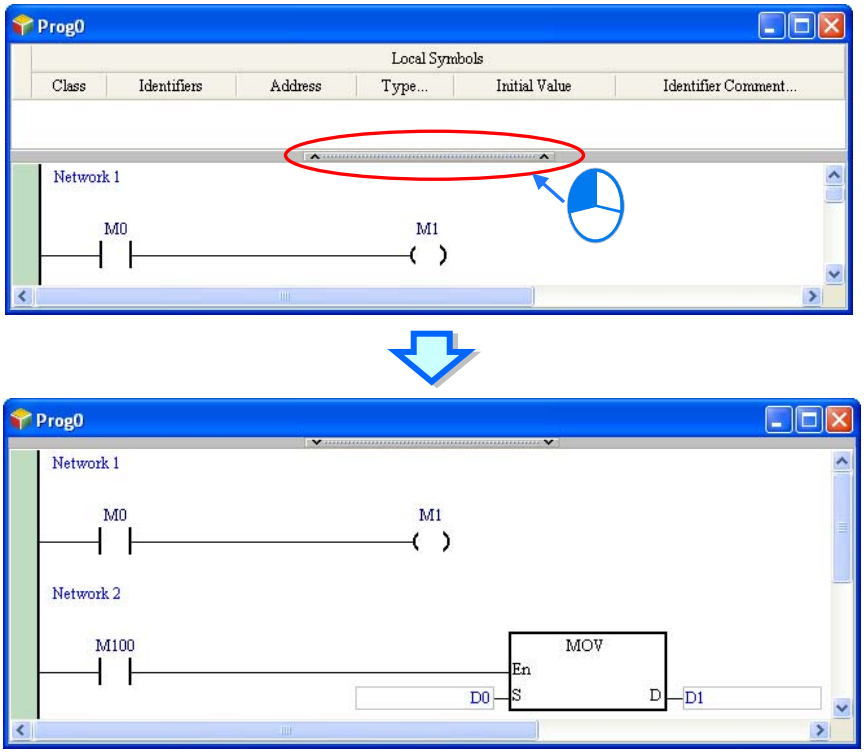
- **Global symbol table**

  Double-click **Global Symbols** in the project management area to open the **Global Symbols** window.



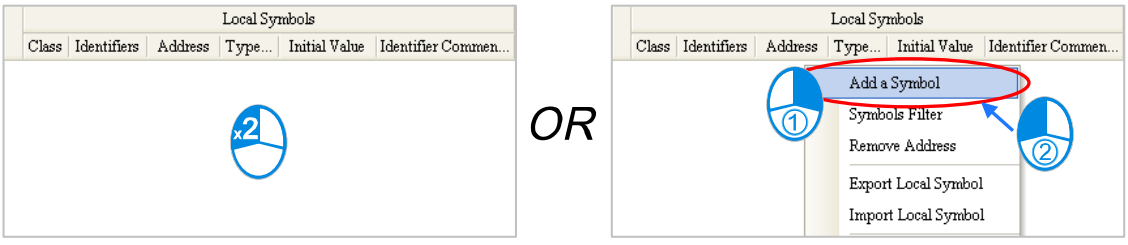
- **Local symbol table**

  In ISPSoft, the local symbol table in a POU is at the top of the window for the POU. If users click the button under a local symbol table, the local symbol table will be hidden. After the users click the button again, the local symbol table will be displayed.



6.2.2 Adding a Symbol

(1) After users double-click or right-click the blank in a symbol table, they can click **Add a Symbol** on the context menu.

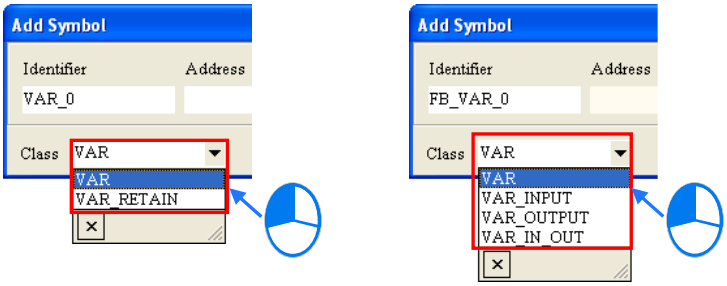


(2) After **Add a Symbol** on the context menu is clicked, the **Add Symbol** window will appear. Type an identifier in the **Identifier** box. Please refer to section 6.1.1 for more information about the regulations of creating the identifier of a symbol.

***1. The window which appears varies with the model selected, and the symbol table created (the global/local symbol table). However, the setting does not vary.**

***2. There is a way to open a declaration window in every programming environment. Please refer to chapter 8~chapter 12 for more information.**

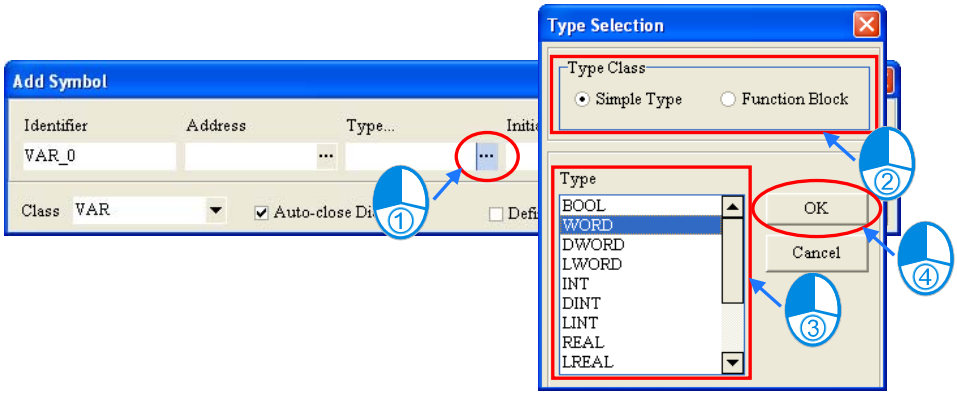
(3) Select a class in the **Class** drop-down list box. The items in the **Class** drop-down list box vary with the symbol table created (the global/local symbol table). Please refer to section 6.1.2 for more information.



***1. The items in the Class drop-down list box for a global symbol or a local symbol in a POU are VAR and VAR_RETAIN. However, in a project for a DVP series PLC, VAR_RETAIN is not in the Class drop-down list box for a global symbol or a local symbol in a POU.**

***2. The items in the Class drop-down list box for a local symbol in a POU of the function block type are VAR, VAR_INPUT, VAR_OUTPUT, and VAR_IN_OUT.**

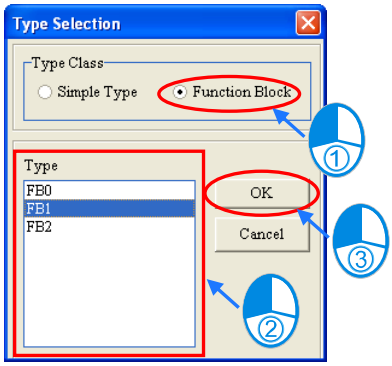
(4) Click the button at the right side of the **Type…** box, set the data type of the symbol in the **Type Selection** window, and click **OK**. Please refer to section 6.1.3 for more information about the data types.



The setting of **Function Block**, **STRING**, and **ARRAY** are described below.

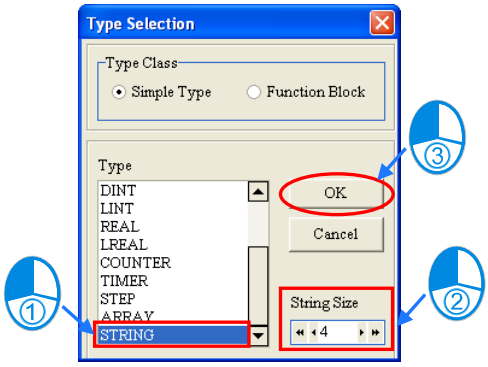
- **Function Block**

  Select the **Function Block** option button in the **Type Class** section, and then select a function block definition in the **Type** box. If the data type of a symbol is a function block, a function block instance is declared. Please refer to chapter 7 for more information.



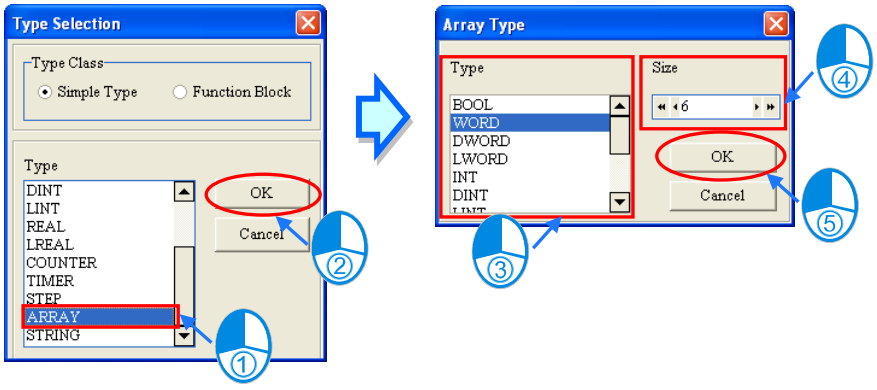
- **STRING**

  If the data type of a symbol is STRING, users have to set the size of the string. The size of a string is within the range between 1 character and 128 characters. The size of a string that users set is the maximum number of characters in the string. Users can click ◂ or ▸ to decrease or increase the digit in the ones place of the setting value, and they can click ◂◂ or ▸▸ to decrease or increase the digit in the tens place of the setting value.

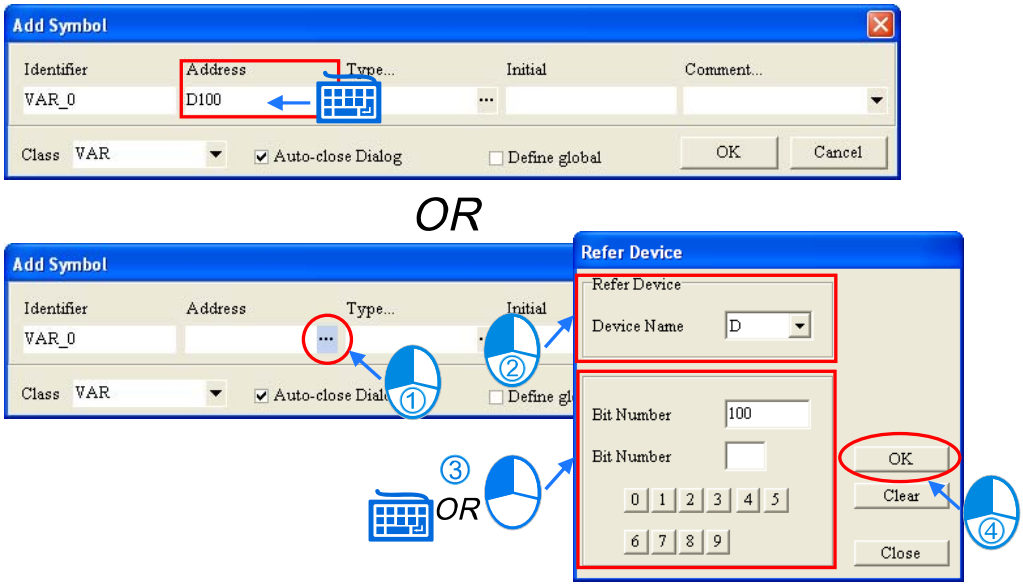


- **ARRAY**

  If the data type of a symbol is ARRAY, the **Array Type** window will appear after **OK** in the **Type Selection** window is clicked. Users have to select an array type and set the size of the array in the **Array Type** window. The size of an array is within the range between 1 element and 2048 elements. Users can click ◂ or ▸ to decrease or increase the digit in the ones place of the setting value, and they can click ◂◂ or ▸▸ to decrease or increase the digit in the tens place of the setting value.



(5) If the **Address** box is blank, the system will automatically assign a device address to the symbol. If users want to specify a device address, they can type the device address in the **Address** box, or specify the device address in the **Refer Device** window after they click the button at the right side of the **Address** box. Please refer to section 6.1.4 for more information.



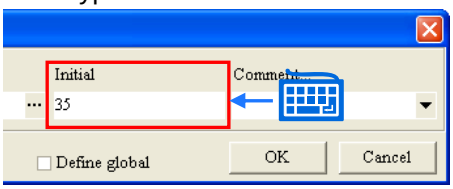
***1. If the data type of a symbol is a function block, users can not assign a device address to the symbol.**

***2. In a project for an AH500 series CPU module, users can not assign device addresses to the local symbols in the function blocks.**

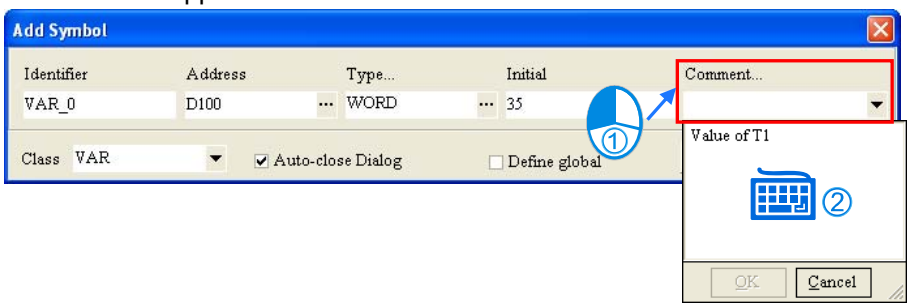
***3. In an AH500 series CPU module, the bits in some devices can be manipulated, e.g. X0.0 and D0.1. If one of these devices is selected in the Refer Device window, the Bit Number box will appear.**

(6) If users want to set the initial value of the symbol, they can type the initial value in the **Initial** box. If the data type of the symbol is ARRAY, the **Array Initial Values** window will appear after the **Initial** box is clicked. In the **Array Initial Values** window, users can set the initial values of the elements in the array.

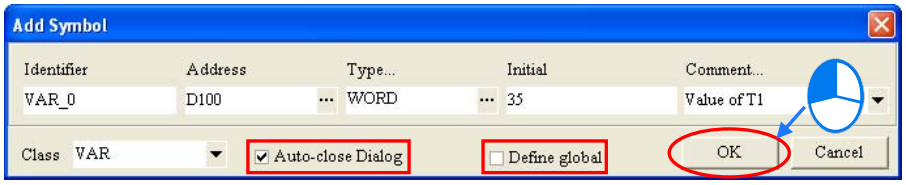
If the data type of the symbol is BOOL, the initial value is either TRUE (abbreviated to T) or FALSE (abbreviated to F). If users type 1 or 0 in the **Initial** box, the system will automatically convert it into TRUE or FALSE. If the data type of the symbol is STRING, the number of characters in the **Initial** box can not be larger than the maximum number of characters in the string which has been set. The characters can not be put in double quotes. Otherwise, the double quotes will be regarded as a part of the string. Besides, if the data type of the symbol is a function block, users can not type an initial value in the **Initial** box.



(7) If After users click the **Comment** box, they can type a comment about the symbol in the drop-down box that appears.



(8) There are two checkboxes in the **Add Symbol** window. Please refer to the descriptions below for more information. After users make sure of the setting, they can click **OK** to add the symbol.



- **Auto-close Dialog**

  If users unselected the **Auto-close Dialog** checkbox, the **Add Symbol** window will not be closed automatically after OK is clicked. Please refer to section 2.3.1 for more information.

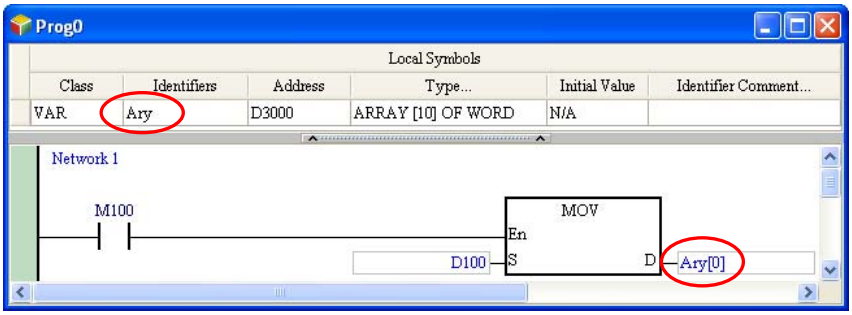
- **Define Global**

  If users select the **Define Global** checkbox, the symbol declared here will appear in the global symbol table. However, the **Define Global** checkbox is not in the **Add Symbol** window for a global symbol.

6.2.3 Using a Symbol Whose Data Type is ARRAY or STRING

- **Symbol whose data type is ARRAY**

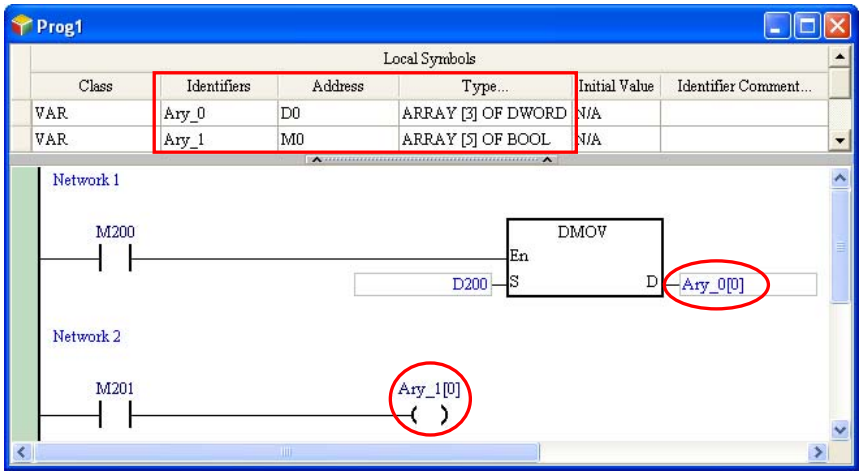
  If users want to use a symbol whose data type is ARRAY in a program, the expression format is **Identifier[Index]**. Generally speaking, the index is a decimal constant. If the programming language in a project for an AH500 series CPU module is a structured text, the index can be a symbol. The minimum index value must be 0, whether the index is a constant or a symbol. The maximum index value can not be larger than or equal to the number of elements in the array. For example, if the number of elements in an array is 10, the index value must be within the range between 0 and 9. If the index is a symbol, the value of the symbol can not be larger than the number of elements in the array. Otherwise, an error will occur during the operation.



If users declare a symbol whose data type is ARRAY, they have to select an array type, and set the size of the array. The number of elements in an array must be within the range between 1 and 2048. A start device address is assigned to a symbol whose data type is ARRAY according to the array type. Please notice that the start device address is assigned to a symbol whose data type is ARRAY can not be an SR device, an SM device, an E device, or an F device. Please refer to section 6.1.4 for more information. Besides, an array is composed of the devices starting from the device assigned by users or the system, and the number of devices in the array conforms to the size of the array.

In the figure below, **ARRAY[3] OF DWOED** in the **Type…** cell for **Ary_0** indicates that the array is composed of 3 elements, and the array type is DWORD. The device address in the **Address** cell for **Ary_0** indicates that the start device address is D0. As a result, the array is composed of D0~D5 (6 word devices).

ARRAY[5] OF BOOL in the **Type…**cell for **Ary_1** indicates that the array is composed of 5 elements, and the array type is BOOL. The device address in the **Address** cell for **Ary_1** indicates that the start device address is M0. As a result, the array is composed of M0~M4.



When users declare a symbol whose data type is ARRAY, they can set the initial value of the symbol. Please refer to the following example.

[1,2,3,4,5] in the **Initial Value** cell for **A_Ary** indicates that the initial value of A_Ary[0] is 1, the initial value of A_Ary[1] is 2, the initial value of A_Ary[2] is 3, the initial value of A_Ary[3] is 4, and the initial value of A_Ary[4] is 5. **[1,3(0),5]** in the **Initial Value** cell for **B_Ary** indicates that the initial value of B_Ary[0] is 1, the initial value of B_Ary[1] is 0, the initial value of B_Ary[2] is 0, the initial value of B_Ary[3] is 0, and the initial value of B_Ary[4] is 5.

Local Symbols

Class	Identifiers	Address	Type...	Initial Value	Identifier Comment...
VAR	A_Ary	D7000	ARRAY [5] OF WORD	[1,2,3,4,5]	
VAR	B_Ary	D7005	ARRAY [5] OF WORD	[1,3(0),5]	

A_Ary		B_Ary	
Element	**Initial value**	**Element**	**Initial value**
A_Ary[0]	1	B_Ary[0]	1
A_Ary[1]	2	B_Ary[1]	0
A_Ary[2]	3	B_Ary[2]	0
A_Ary[3]	4	B_Ary[3]	0
A_Ary[4]	5	B_Ary[4]	5

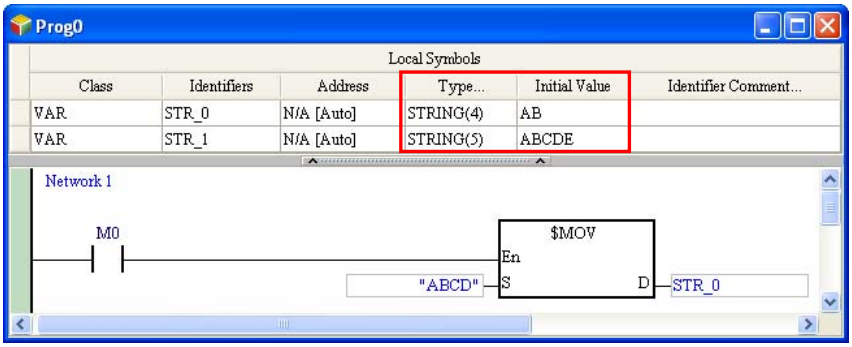
- **Symbol whose data type is STRING**

In ISPSoft, a string is composed of ASCII codes, and the ASCII codes are surrounded by double quotes, e.g. “ABCD”. A symbol whose data type is STRING is generally used with an applied instruction. Please refer to the programming manual for a model for more information. Besides, only AH500 series CPU modules support the symbols whose data types are STRING, and the data type of a symbol of the VAR_INPUT class, the VAR_OUTPUT class, or the VAR_IN_OUT class in a function block can not be STRING. As a result, if users want to process a string in a function block, they can create the string as a global symbol. Please refer to chapter 7 for more information about the usage of function blocks.

When users declare a symbol whose data type is STRING, they can specify the maximum size of the string. The number of characters in a string is within the range between 1 and 128, and one character occupies 1 byte. The number of devices assigned to the characters in a string must conform to the size of the string, and one extra byte must be assigned to the ending character in the string. If the last character and the ending character in a string does not occupy the two bytes in a word device, the ending character will be assigned another device. For example, two word devices (four bytes) will be assigned to a string which is composed of three characters, and three word devices (six bytes) will be assigned to a string which is composed of four or five characters.

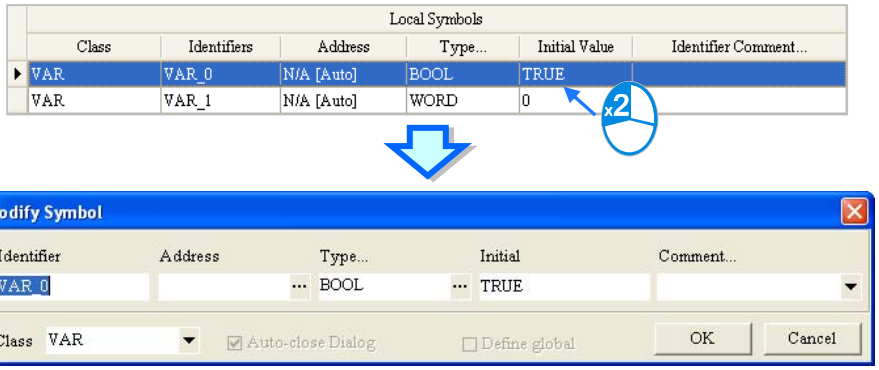
Please refer to the following figure. The number in the parentheses in the **Type…** cell for a symbol whose data type is STRING indicates the maximum size of the string. Besides, the characters in the **Initial Value** cell for a symbol whose data type is STRING can not be put in

double quotes, and the number of characters in the **Initial Value** cell for the symbol whose data type is STRING must be less than or equal to the maximum number of characters in the string.

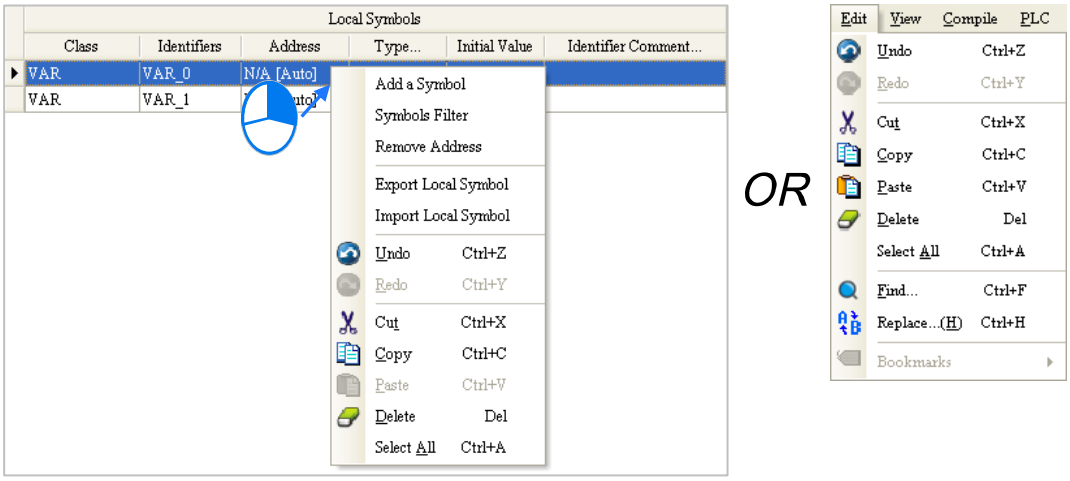


6.2.4 Modifying a Symbol and Editing a Symbol Table

If users want to modify the attributes of a symbol, they can double-click the symbol to open the declaration window, and modify the setting values in the boxes according to the description in section 6.2.2.



Besides, if users want to edit an item in a symbol table, they can click an item on the **Edit** menu, or an item on the context menu. Please refer to the following table for more information about the basic items on the **Edit** menu or the context menu.



Item	Function
Undo	Undoing the last action (*The number of previous actions that can be undone is 20.)
Redo	Redoing an action which has been undone
Cut	Cutting the symbol selected
Copy	Copying the symbol selected

Item	Function
Paste	Pasting an object which has been copied or cut on the present symbol table
Delete	Deleting the symbol selected
Select All	Selecting all the symbols in the symbol table

***1. After a symbol copied or cut from a project for a model or a POU is pasted on a project for another model or another POU, users have check whether the attributes of the symbol pasted are correct.**

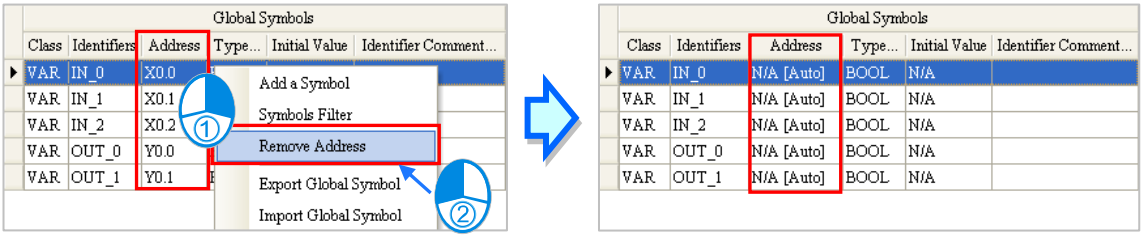
***2. A symbol cut or copied from a project for an AH500 series CPU module can not be pasted on a project for a DVP series PLC, and vice versa.**

***3. If a symbol is pasted on a symbol table where the same identifier exists, the system will put CopyOf_n at the end of the symbol which is pasted. (n in CopyOf_n is a serial number.)**

6.2.5 Clearing the Device Addresses Assigned to the Symbols

Users can clear all the devices addresses assigned to the symbols in a symbol table, whether the devices are assigned by the users or by the system. After the program is compiled again, the system will automatically assign device addresses to these symbols. Besides, only the device addresses in the symbol table selected are cleared, the device addresses in the other symbol tables are not cleared.

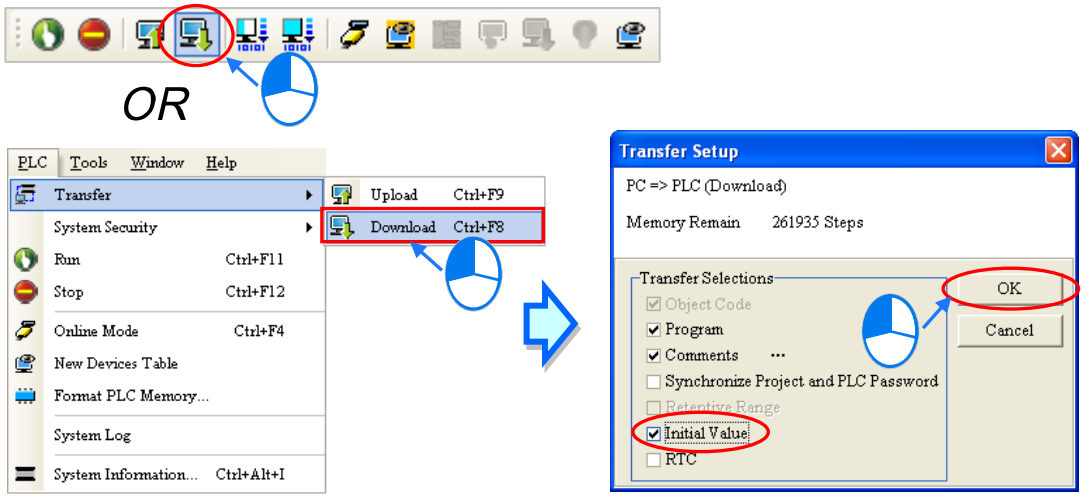
Right-click a symbol table, and click **Remove Address** on the context menu.



6.2.6 Downloading the Initial Values of the Symbols

Users can set the initial value of a symbol when they declare the symbol. If a project is downloaded, the initial values of the symbols can be written into the devices assigned to the symbols. However, the initial values of the symbols in a project are written into the devices assigned to the symbol only when the project is downloaded. If a PLC is disconnected or stops running, and runs again, the values of the symbols in the program in the PLC will not be the initial values. In order to ensure that the values of the symbols in the program in a PLC are the initial values whenever the PLC begins to run, it is suggested that users download the initial values of the symbols. Besides, when the online update is executed, the system does not download the initial values of the symbols.

Users can click the **PLC** menu, point to **Transfer**, and click **Download** to open the **Transfer Setup** window. The can also click on the toolbar to open the **Transfer Setup** window. After the users select the **Initial Value** checkbox in the **Transfer Selections** section, and click **OK**, the initial values will be written into the PLC.

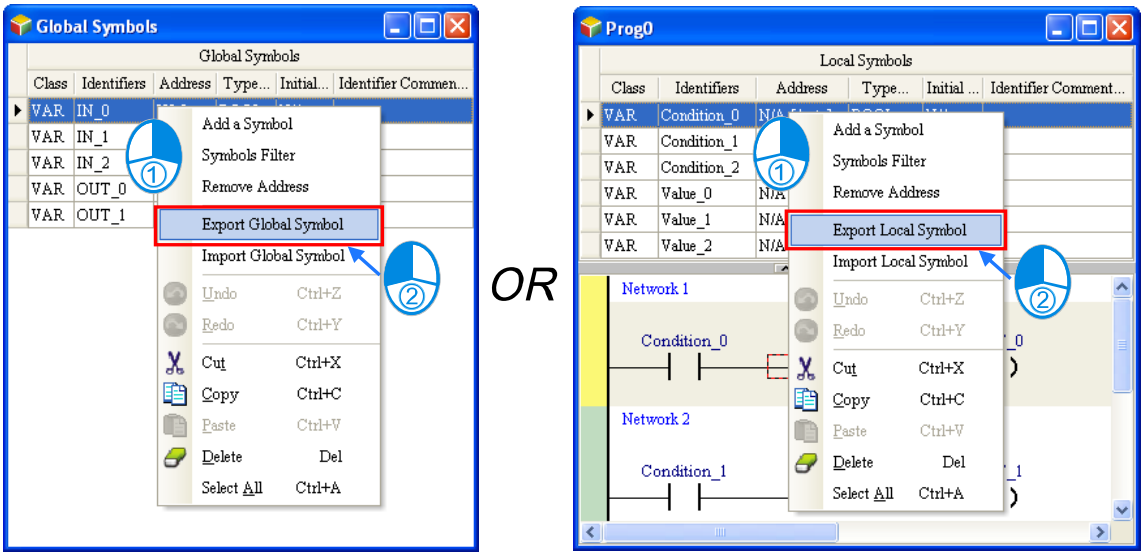


***. Before the initial values are downloaded, users have to make sure that ISPSoft is connected to the PLC normally. Please refer to section 2.4 for more information about communication setting.**

6.2.7 Exporting a Symbol Table

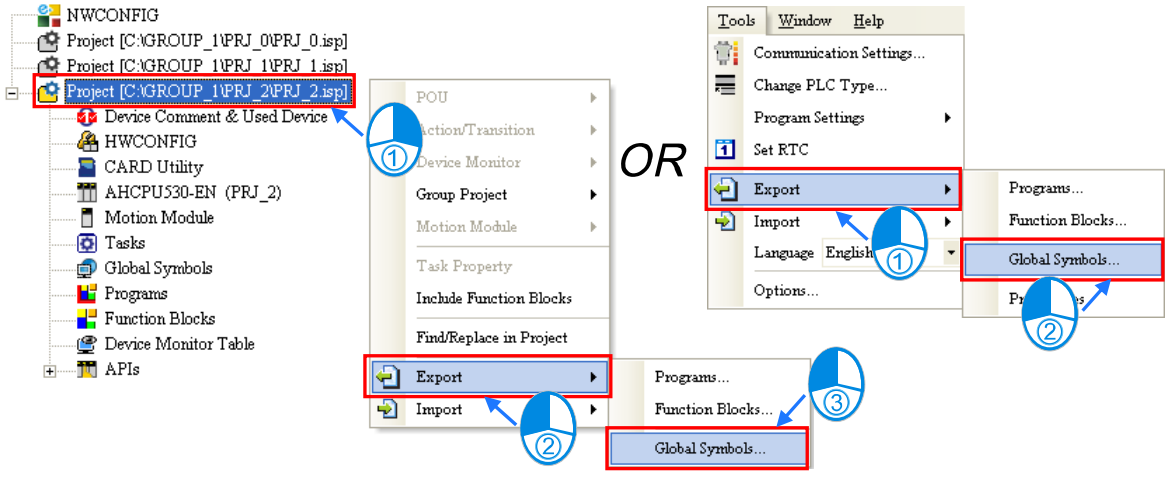
The symbols created in a project can be exported as a CSV file. The data which is exported can be edited and managed through Microsoft Excel.

If users want to export a symbol table, they can right-click the symbol table, and click **Export Global Symbol** or **Export Local Symbol** on the context menu.



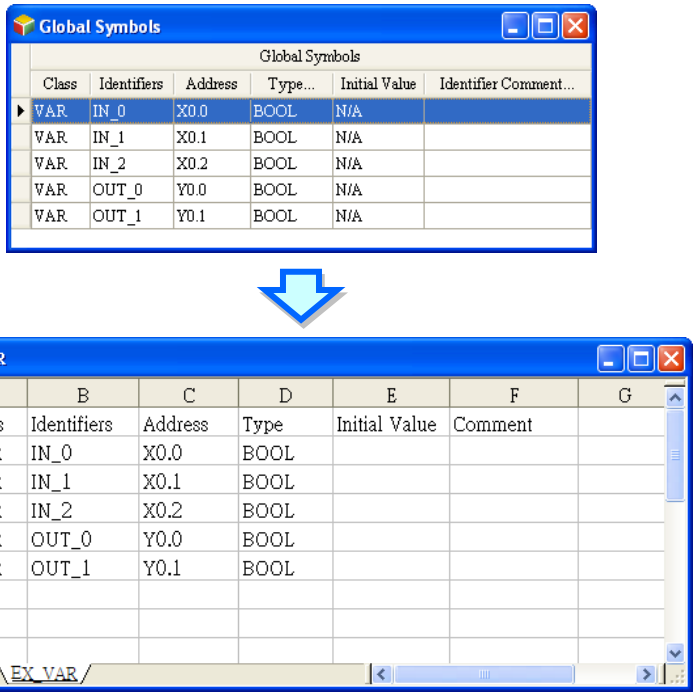
The users can also export the global symbol table in a project through the toolbar or the project management area. However, if there is a group of projects in ISPSoft, the users can only export the global symbol table in the project which is being edited presently.

After users click the **Tools** menu, point to **Export**, and click **Global Symbols…**, the global symbol table in the project which being edited presently will be exported. The users can also export the global symbol table in the project which is being edited presently by right-clicking the project in the project management area, pointing to **Export**, and clicking **Global Symbols…**.

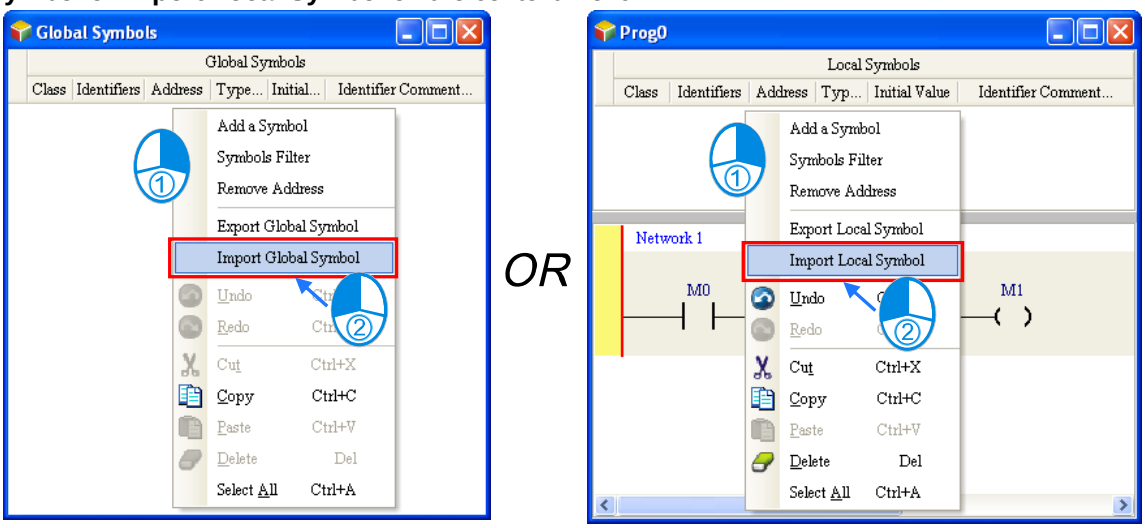


6.2.8 Importing a Symbol Table

If users want to create symbols, they can add the symbols to a symbol table. If the users want to create a large number of symbols easily, they can export a symbol table as a CSV file, edit the CSV file through Microsoft Excel, and import the new CSV file into ISPSoft.

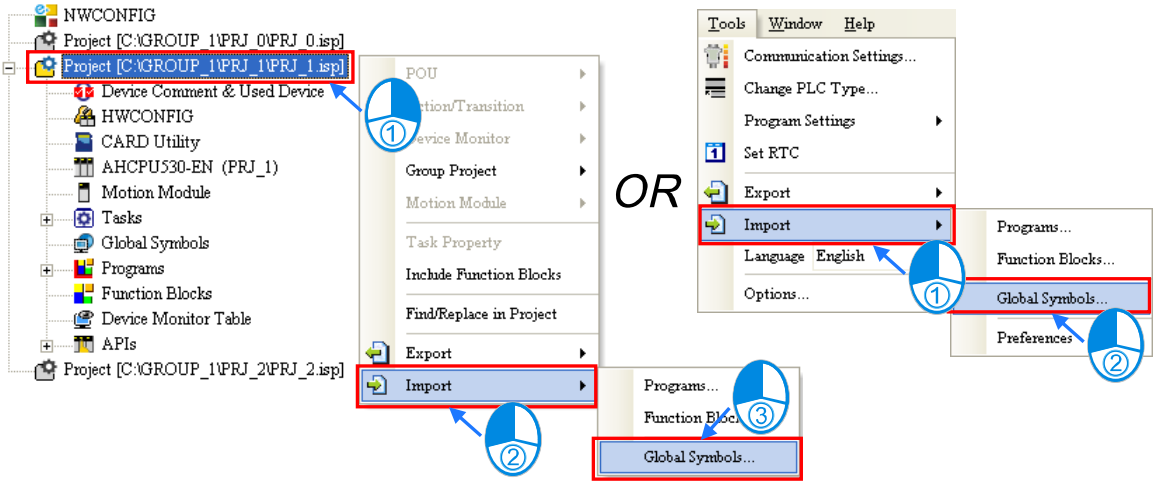


If users want to import a symbol table, they can right-click a symbol table, and click **Import Global Symbol** or **Import Local Symbol** on the context menu.

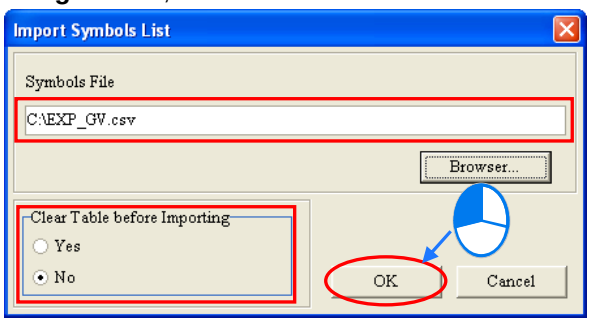


The users can also import the global symbol table in a project through the toolbar or the project management area. However, if there is a group of projects in ISPSoft, the users can only import the global symbol table in the project which is being edited presently.

After users click the **Tools** menu, point to **Import**, and click **Global Symbols…**, the global symbol table in the project which being edited presently will be exported. The users can also import the global symbol table in the project which is being edited presently by right-clicking the project in the project management area, pointing to **Import**, and clicking **Global Symbols…**.



Select a file which will be imported in the **Import Symbol List** window, select an option button in the **Clear Table before Importing** section, and click **OK**.



Additional remark

A symbol table exported from a project for a model or a POU can be imported into a project for another model or another POU. However, if a symbol in a file which will be imported is in one of the situations listed below, the symbol will be automatically eliminated or converted. Besides, if an error occurs when a symbol table is imported, the error log will be saved as a file whose primary filename is "**Project name_ErrorLog**" in a folder in which the project (*.isp) is saved. Users can know the error occurring when the symbol is imported by viewing the file.

(a) If a symbol in a file is imported into a symbol table where the same identifier exists, the symbol will be automatically eliminated from the file. If the contents of a symbol table are cleared before a symbol table is imported, no such problem will occur.

(b) If the data type of a symbol in a file is not supported by a project for a model, the symbol is eliminated from the file when the file is imported into the project.

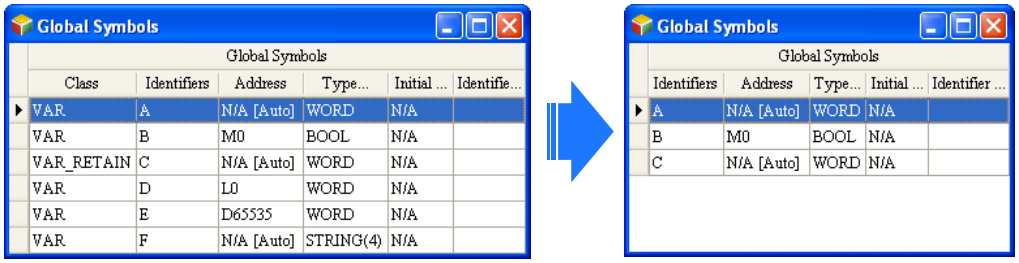
(c) If the device assigned to a symbol in a file is not supported by a project for a model, or the device is not within the range of devices in the model, the symbol is eliminated from the file when the file is imported into the project.

(d) If the class of a symbol in a file is incompatible with the symbol table in a POU or a global symbol table, the symbol is eliminated from the file when the file is imported into the symbol table in the POU or the global table.

(e) If a symbol in a project for an AH500 series CPU module is a symbol of the VAR_RETAIN class, the symbol will become a symbol of the VAR class after the symbol is imported into a project for a DVP series PLC.

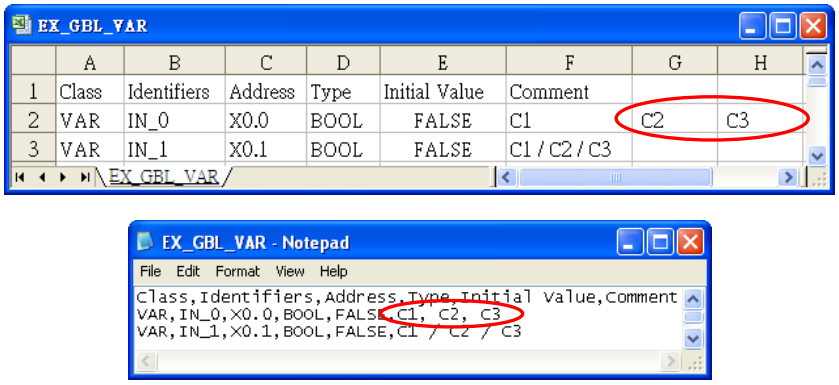
An example of importing the global symbol in a project for AHCPU530-EN into a project for a DVP-SV series PLC is shown below. A, B, and C in the project for AHCPU530-EN can be imported into the project for a DVP-SV series PLC. C in the project for AHCPU530-EN is a symbol of the VAR_RETAIN class. After C in the project for AHCPU530-EN is imported into the project for the DVP-SV series PLC, C becomes a symbol of the VAR class. Owing to the fact L0 is not supported by the DVP-SV series PLC, D is eliminated. D65535 is not within the range of devices in the DVP-SV series PLC, and therefore E is eliminated. F is eliminated because the DVP-SV series PLC

does not support the symbols whose data types are STRING.

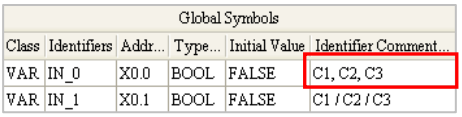


Users not only have to pay attention to the data formats in symbol tables, but also have to pay attention to the comments on the symbols in a symbol table.

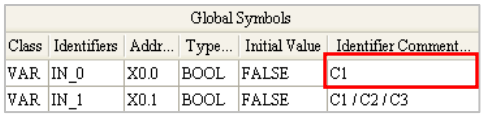
A CSV file can be opened in Microsoft Excel and Notepad, as shown below. In a CSV file, the cells are divided by the commas. As a result, if there are commas in the **Identifier Comment…** cell for a symbol in a symbol table, the system will automatically regard the contents after the first comma as the contents of other cells, as shown by C2 and C3 which are in the red circle below. After the symbol table is imported again, the contents after the first comma will be lost.



- Before the symbol table is exported

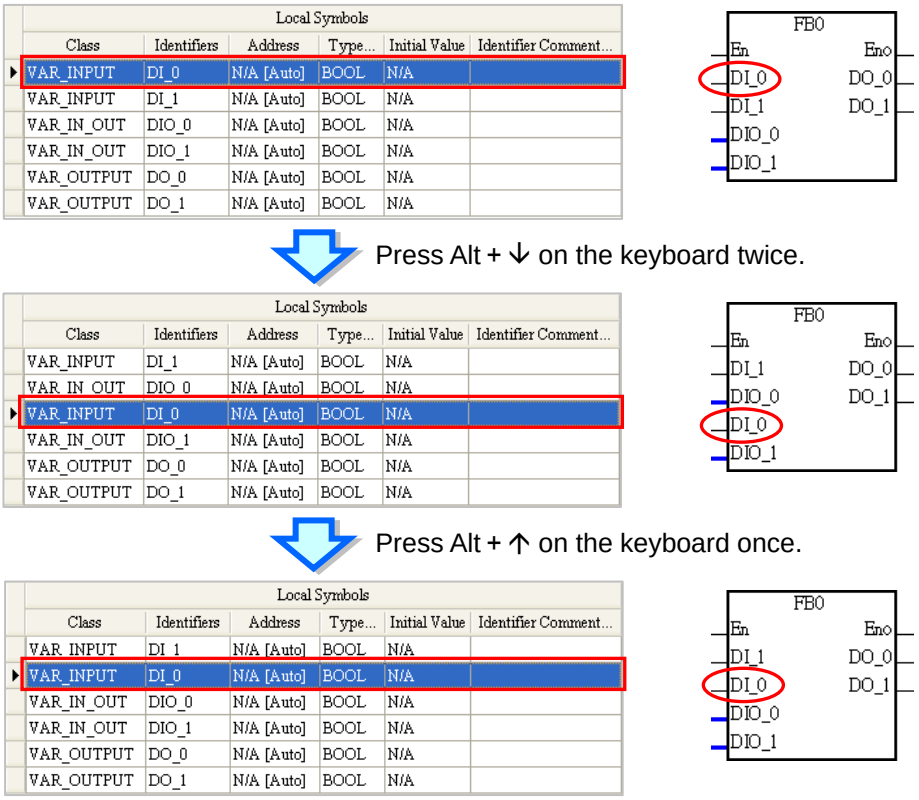


- After the symbol table is imported

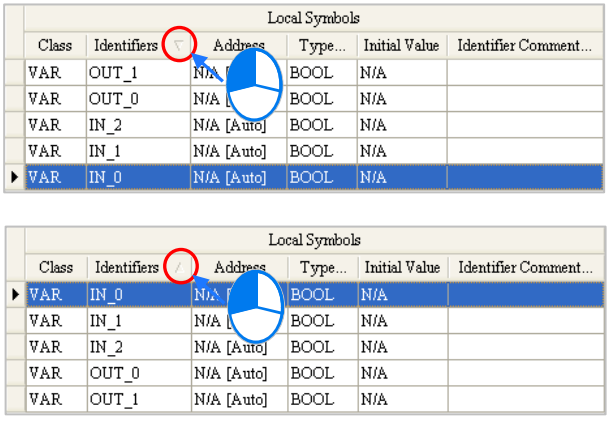


6.2.9 Arranging the symbols

In the local symbol table in a function block, the order in which the symbols of the VAR_INPUT class, the symbols of the VAR_IN_OUT class, and the symbols of the VAR_OUTPUT class are arranged affects the order in which the operands in the function block are arranged. If users want to arrange the symbols in the local symbol table in a function block, they can click a symbol which will be moved upward or moved downward, and press Alt + ↑ or Alt + ↓ on the keyboard.



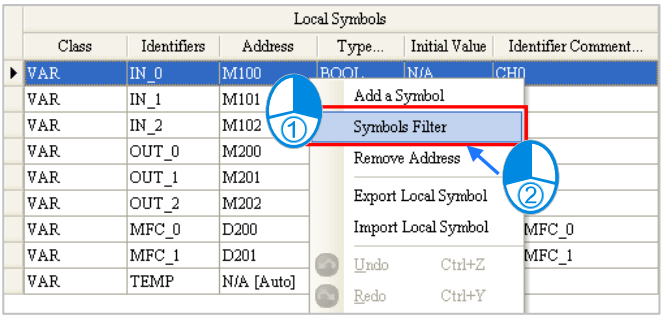
Besides, if users click the heading **Identifier** or **Address** in the global symbol table in a project or the local symbol table in a POU of the program type, the data in the table will be sorted according to the contents of the **Identifier** column or the **Address** column, and a mark indicating the sort direction will appear in at the right side of the heading **Identifier** or **Address**. Owing to the fact that the order in which the symbols in the local symbol table in a function block are arranged affects the order in which the operands in the function block are arranged, the data in the local symbol table in a function block can not be sorted in this way.



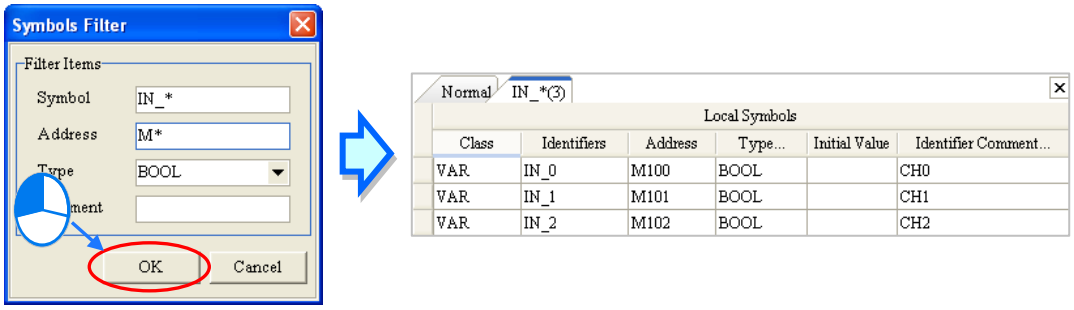
6.2.10 Filtering the Symbols

Users can view the symbols which meet the same conditions.

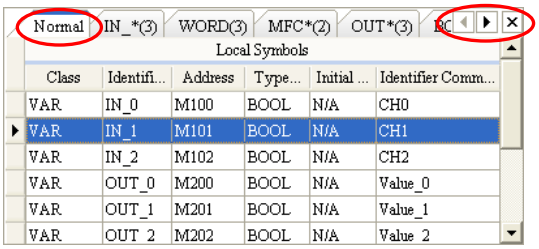
(1) Right-click a symbol table, and click **Symbols Filter** on the context menu.



(2) Users can set the conditions in the **Symbols Filter** window. The wildcard character " * " can be used to substitute for any character or characters in a string. The symbols which satisfy the conditions set are sieved out. If the users want to ignore a certain condition, they can leave the box for the condition blank. Click **OK** after the setting is complete.



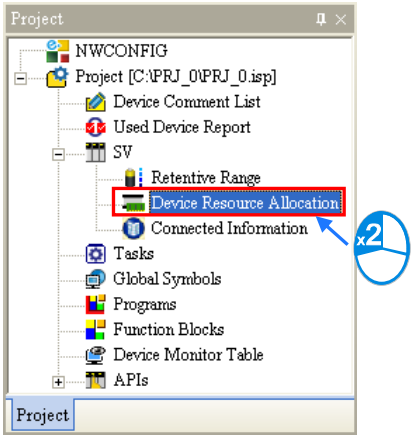
(3) The symbols which are sieved out are displayed in the additional page. The users can switch between the pages by clicking the tabs at the top of the table. If several groups of symbols are sieved out, the users can switch among the tabs by clicking ◀ or ▶ in the upper right corner of the window. If users click ✕ in the upper right corner of a page, the page will be closed. Besides, these additional pages are for view only. Users can not edit the tables in the pages. After the users click the **Normal** tab, they can edit the original symbol table.



6.2.11 Setting a Range of Devices (Only for DVP Series PLCs)

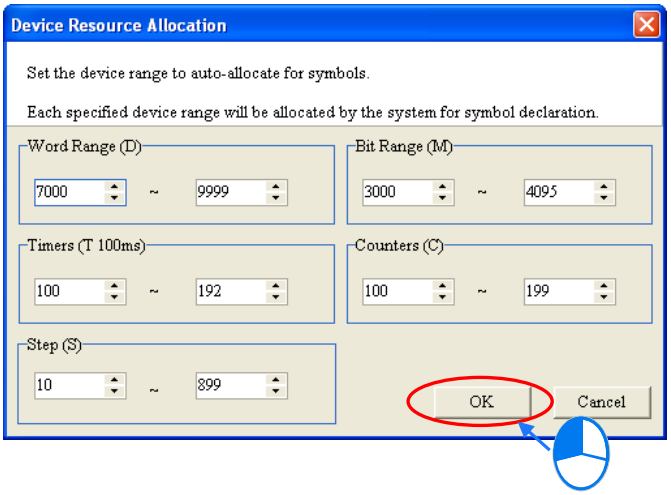
The devices automatically assigned to the symbols in a project for a DVP series PLC are devices available to users. Users set a range of devices which can be automatically assigned to the symbols in a project so that they can manage the devices conveniently. They can set a range of devices for a project created. When the program in a project is compiled, devices within the range which is set are assigned to the symbols. If the number of symbols is larger than the devices within the range, an error message showing that the devices are insufficient will appear after the program in a project is compiled. Please refer to section 6.1.4 for more information about assigning devices to the symbols in a project.

Unfold the PLC section in the project management area in a project for a DVP series PLC, double-click **Device Resource Allocation**.



Users can set a device range in the **Device Resource Allocation** window. Every type of device has its limitations. For example, the data registers in a DVP-SV series PLC start from D2000, the auxiliary relays start from M2000, the counters are 16-bit counters, and the timers take 100 milliseconds as the timing unit. If users select a device address which is not allowed, the system will modify it automatically.

Click **OK** after the setting is complete. The setting here will be saved with the project.



6.3 Example

An example of creating a ladder diagram is shown in chapter 4. In section 5.6.1, the program created in chapter 4 is rewritten by means of IEC 61131-3. In the following sections, the device names in the program created in section 5.6.1 will be replaced by symbols.

6.3.1 Planning a Symbol Table

The symbols which correspond to the device names in the program created in section 5.6.1 are listed in the following table. Please refer to the descriptions below for more information.

Original device name	Identifier	Address specified	Declaration position	Function
X0.0	START_BT	X0.0	Global symbol table	START button
X0.1	STOP_BT	X0.1	Global symbol table	STOP button
X0.2	InP_SNR_1	X0.2	Global symbol table	In position sensor 1
X0.3	InP_SNR_2	X0.3	Global symbol table	In position sensor 2
X0.4	CNT_SNR	X0.4	Global symbol table	Counting sensor
Y0.0	CONVEYER	Y0.0	Global symbol table	Conveyer

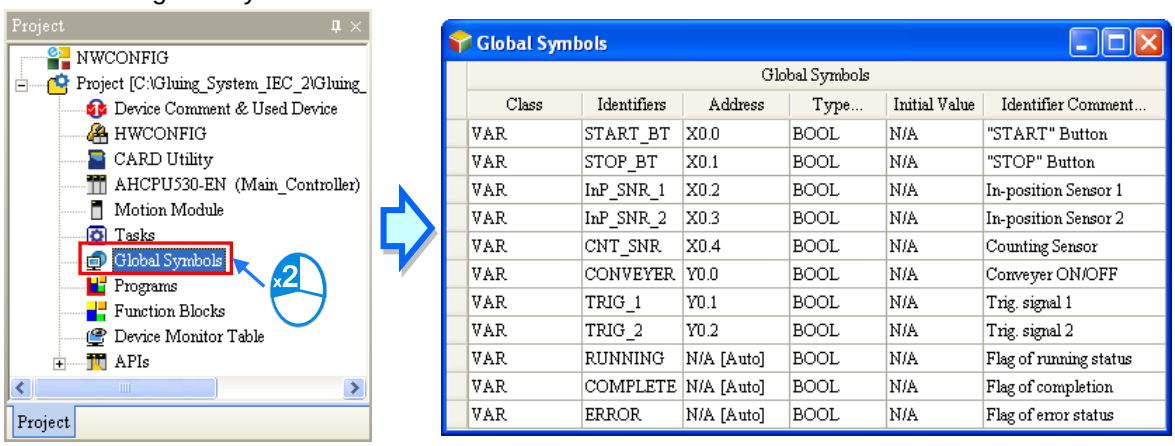
Original device name	Identifier	Address specified	Declaration position	Function
Y0.1	TRIG_1	Y0.1	Global symbol table	Trigger signal for injector 1
Y0.2	TRIG_2	Y0.2	Global symbol table	Trigger signal for injector 2
M0	RUNNING	A device address is assigned automatically.	Global symbol table	Operation flag
M1	COMPLETE	A device address is assigned automatically.	Global symbol table	Completion flag
M2	ERROR	A device address is assigned automatically.	Global symbol table	Error flag
D0	CNT_DT	A device address is assigned automatically.	Local symbol table	Number of parts which are conveyed

- The X devices and the Y devices in the PLC are contacts connected to the equipment. The X device addresses and the Y device addresses are related to the wiring of the equipment. Users have to specify the device addresses by themselves. Besides, it is suggested that the X devices and the Y devices should be created in the global symbol table although some X devices and Y devices are only used in a certain POU. After the X devices and the Y device are created in the global symbol table, the subsequent maintenance such as changing the wiring and accessing the contacts in the additional program will be more convenient to users. However, it is not necessary for users to create the X devices and the Y devices in the global symbol table. The users can decide by themselves.
- Owing to the fact that M0, M1, and M3 are used in the different POUs, the symbols correspond to the three flags are declared in the global symbol table. Users do not need to assign device addresses to the symbols, and therefore the system assigns device addresses to them.
- The number of parts which are conveyed is stored in D0, and D0 is only used in the **COUNTING**. As a result, the symbol corresponding to D0 is declared in the local symbol table in the **COUNTING**, and the system assigns a device address to D0.

6.3.2 Writing a Program

Users can directly modify the program created in section 5.6.1, or write the program shown in section 5.6.1 again. If the users want to write the program again, they have to create a new project, and refer to chapter 4 and section 5.6.1 for more information.

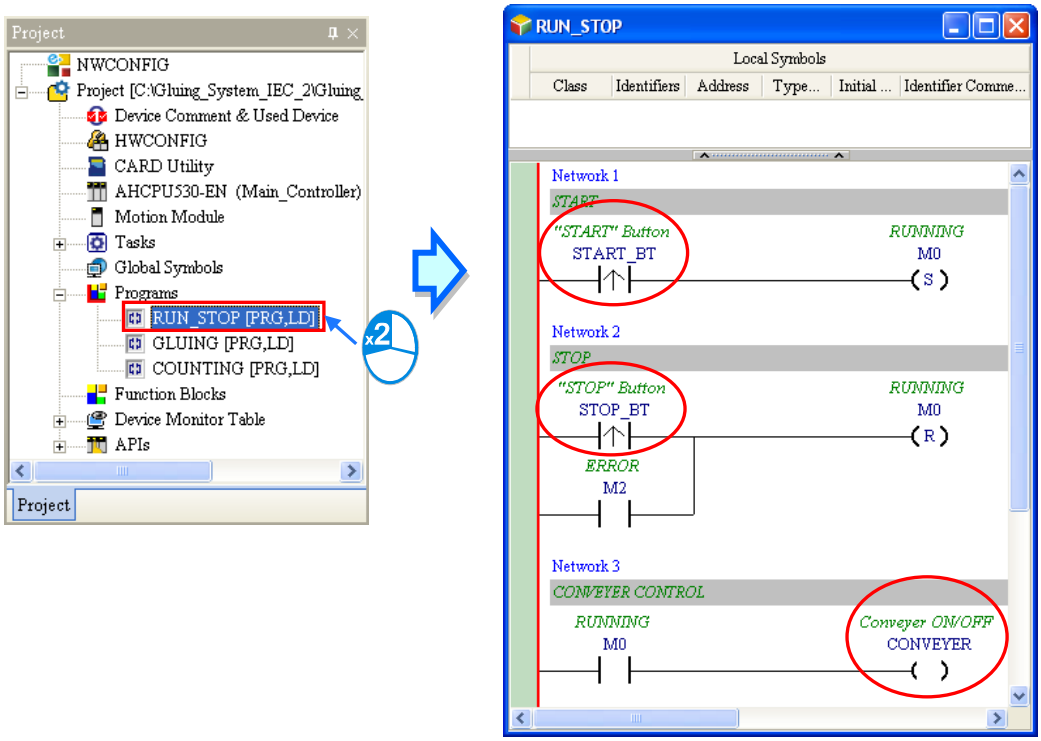
First, open the **Global Symbols** window. Then, refer to section 6.2, and create the symbols shown below in the global symbol table.



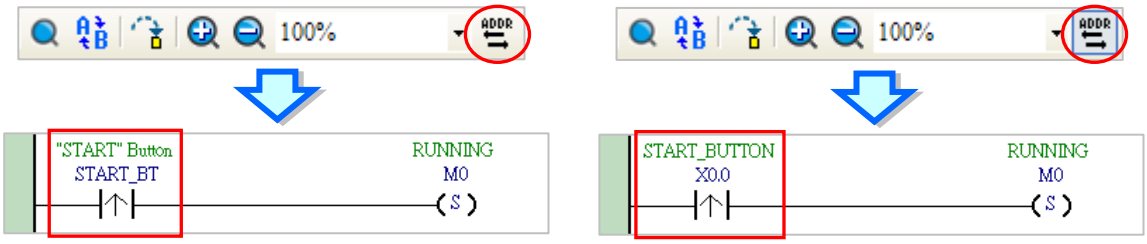
Global Symbols

Class	Identifiers	Address	Type...	Initial Value	Identifier Comment...
VAR	START_BT	X0.0	BOOL	N/A	"START" Button
VAR	STOP_BT	X0.1	BOOL	N/A	"STOP" Button
VAR	InP_SNR_1	X0.2	BOOL	N/A	In-position Sensor 1
VAR	InP_SNR_2	X0.3	BOOL	N/A	In-position Sensor 2
VAR	CNT_SNR	X0.4	BOOL	N/A	Counting Sensor
VAR	CONVEYER	Y0.0	BOOL	N/A	Conveyer ON/OFF
VAR	TRIG_1	Y0.1	BOOL	N/A	Trig. signal 1
VAR	TRIG_2	Y0.2	BOOL	N/A	Trig. signal 2
VAR	RUNNING	N/A [Auto]	BOOL	N/A	Flag of running status
VAR	COMPLETE	N/A [Auto]	BOOL	N/A	Flag of completion
VAR	ERROR	N/A [Auto]	BOOL	N/A	Flag of error status

Double-click **RUN_STOP** in the project management area after the creation of the symbols in the global symbol table is complete. If users directly modify the program created in section 5.6.1, they will find that the symbols to which they assign the device addresses replace the corresponding device names, and the comments on the symbols replace the original comments on the devices. Please refer to the contents below if the program displayed in **RUN_STOP** is not the same as that shown in the figure below.

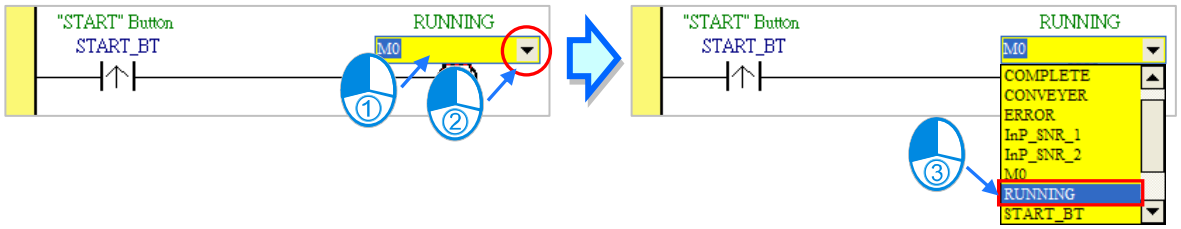


The users can choose whether to display the symbols to which they assign devices in the ladder diagram. If ADDR on the toolbar is pressed, the device names and the comments on the devices will be displayed in the program. If ADDR is not pressed, the identifiers and the comments on the symbols will be displayed in the program.

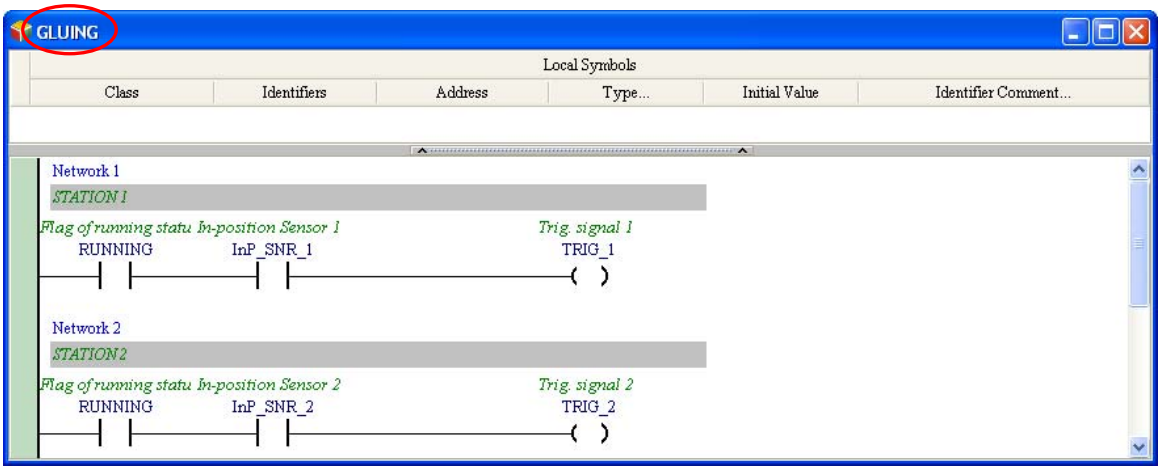


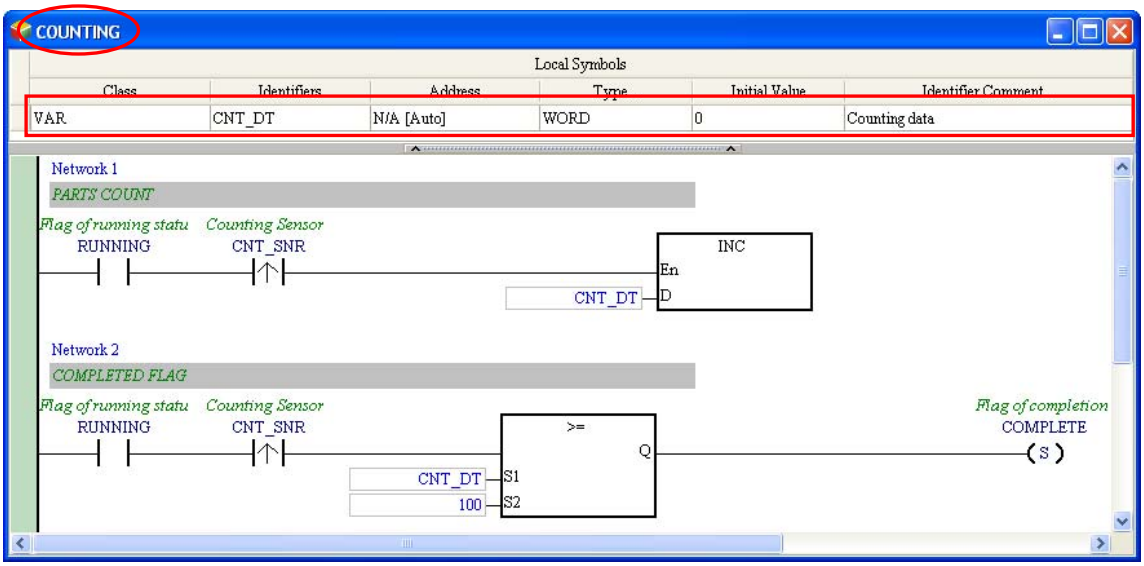
***. M0 is not assigned to any symbol, and therefore it remains unchanged if ADDR is not pressed. Likewise, if users do not assign a device address to a symbol, or the system automatically assigns an internal memory to a symbol, the symbol will remain unchanged after ADDR is pressed.**

The operation flag M0 can be replaced by the identifier RUNNING. The users can refer to chapter 4, and type RUNNING directly. They can also click M0, click ▼ in the box, and select an identifier on the drop-down list. Besides, the users can press Page Down on the keyboard when they edit the box. The users can select an identifier on the drop-down list by ↑ or ↓ on the keyboard. Press Enter on the keyboard after the selection is complete.

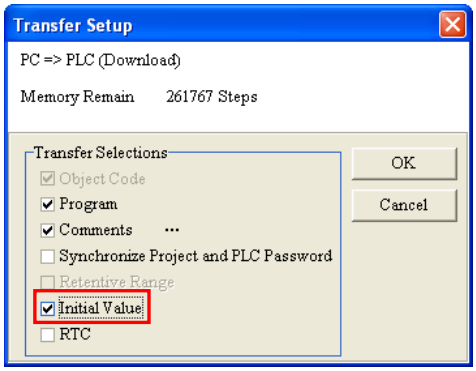


Refer to the description above, and write the program in the POU shown below.





If users want to write the program shown in section 5.6.1 again, they have to refer to chapter 4 and section 5.6.1, and complete the hardware configuration, the parameter setting, and the task management. After the writing of the program is complete, the users have to compile the program, and download the parameters and the program to the CPU module. Before the program is downloaded to the CPU module, the users have to select the **Initial Value** checkbox in the **Transfer Setup** window. Once the **Initial Value** checkbox in the **Transfer Setup** window is selected, the initial value of CNT_DT in **COUNTING** will be written into the CPU module.



MEMO

Chapter 7 Function Block

Table of Contents

7.1 Knowing Function Blocks 7-2
7.1.1 Introduction of Function Blocks 7-2
7.1.2 Characteristics and Advantages of Function Blocks 7-3
7.2 Structure of the Function Blocks in ISPSoft 7-3
7.2.1 En Pin of a Function Block 7-4
7.2.2 Symbols in a Function Block 7-4
7.2.3 Input/Output Pins of a Function Block 7-5
7.2.4 Symbol Whose Data Type is POINTER, T_POINTRER, C_POINTER, or HC_POINTER 7-8
7.2.5 Function Block Definition and Function Block Instance 7-11
7.2.6 Calling Relation Between Function Blocks 7-14
7.2.7 Assigning a Memory Block to a Function Block 7-16
7.3 Using a Function Block 7-20
7.3.1 Basic Specifications for the Function Blocks 7-20
7.3.2 Special Pulse Instructions Used in Function Blocks (Only for AH500 Series CPU modules) 7-21
7.3.3 Monitoring the Program in a Function Block 7-22
7.3.4 Modifying the Program in a Function Block 7-22
7.4 Example 7-23
7.4.1 Planning a Program 7-23
7.4.2 Creating the Program 7-24

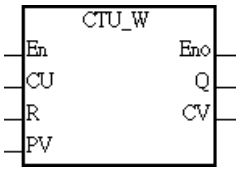
7.1 Knowing Function Blocks

Function blocks (abbreviated as FB) play an important role in the writing of the program in a PLC. They are also supported by IEC 61131-3. Owing to the characteristics and advantages of function blocks, ISPSoft provides much support and many functions for function blocks.

7.1.1 Introduction of Function Blocks

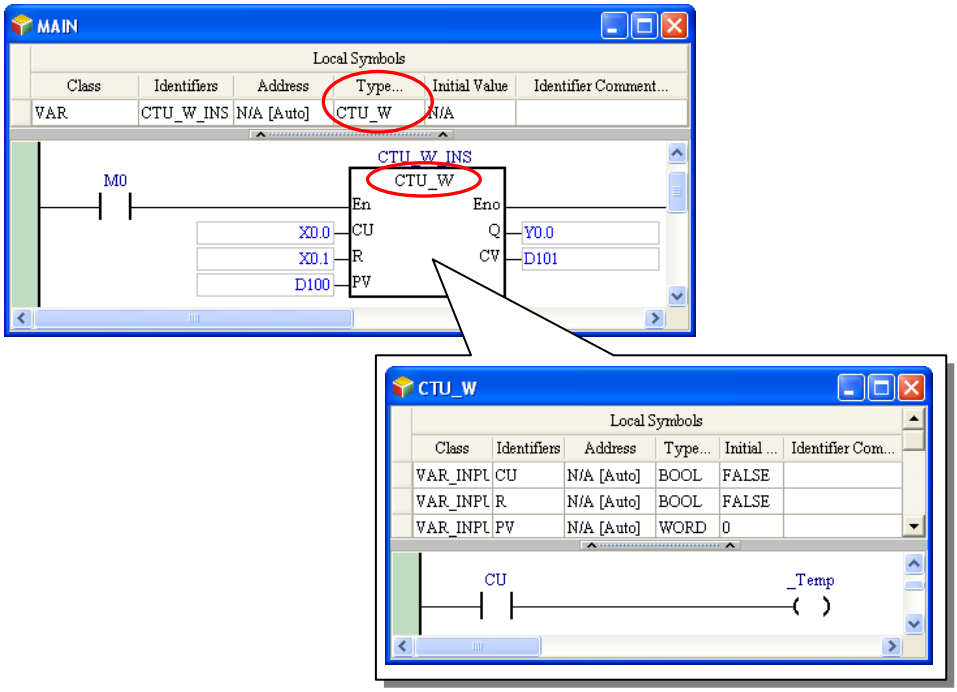
A function block is a component in a program which performs an operation. It is a type of POU, but it can not operate by itself. After a POU of the program type calls a function block, and sends the related parameters, the function of the function block can be executed. After the execution of the function block is complete, the internal operation result will be sent to the device or symbol specified by the superior POU (caller).

A function block is shown below. The appearance of a function block is similar to that of an applied instruction in that there are input pins and output pins. The usage of a function block is also similar to that of an applied instruction. A function block is different from an applied instruction in that users have to declare a symbol whose data type is a function block in the superior POU (caller) which calls the function block. Besides, a function block can call another function block.



A function block is a memory unit. Every function block is assigned a substantial memory block where the values of the internal symbols are stored. Owing to the fact that the value of the symbol in a function block will be retained after the function block is executed, every execution result is affected by the last execution result. In other words, the output values may be different even if the input values are the same.

An application example of a function block is shown in the figure below. There is a symbol whose data type is a function block in the local symbol table in the superior POU (caller). The name in the **Type…** cell is the same as the name of the function block. Please refer to the following sections for more information about the usage of a function block.



7.1.2 Characteristics and Advantages of Function Blocks

Owing to the fact that function blocks present characteristics and advantages that traditional programming of a PLC does not have, function blocks are supported by IEC 61131-3, and are widely used by users.

- **Modular design**

  A large program is divided into several subroutines, and the subroutines are created as function blocks. The function blocks are arranged and called by POUs of the program type.

- **Highly independent**

  Different function blocks can be created by means of different programming languages such as a ladder diagram, an instruction list, a structured text, and a function block diagram according to the characteristics of the function blocks or a programmer's habit. Besides, when a POU calls a function block, the programming language by which the program in the function block is written does not need to be considered. For example, if a complex operation has to be performed in a part of a ladder diagram, users can create the part of the ladder diagram as a function block by means of a structured text, and the function block can be called by the original ladder diagram.

- **Reusable**

  Once a function block is created, it can be used repeatedly as long as users conform to the rule of using the function block.

- **Highly portable**

  After a function block is created, it can be used in the original project. Besides, after the function block is exported, it can be imported into another project. As a result, users can gradually create their own function blocks.

- **Function blocks can be maintained conveniently.**

  The program in a function block is an independent module. If an error occurs in a function block, or the function of a function block does not meet the actual requirement, users can just modify the program in the function block, and do not need to debug or modify the whole program,

- **The readability of a program is increased.**

  Users can write a complex program or a program which will be used repeatedly in a function block, and the function block can be called by the original program. The structure of the original program becomes more simplified, and the program becomes more readable.

- **Highly confidential**

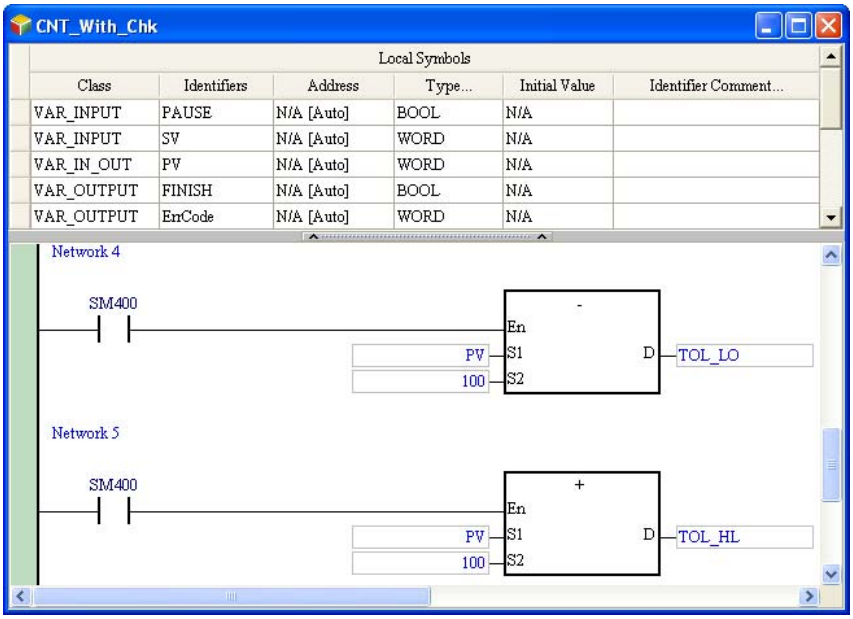
  After users set a password for a function block created, they can create their own core technology in the function block without care, and provide the function block for other users. Besides, system suppliers or software developers can provide function blocks which have specific functions for customers.

- **Highly efficient**

  The application of function blocks provides a highly efficient development environment in terms of project development. Owing to the fact that function block are modular, developers and manufacturers can take part in the development of a project together, and the use of human resources are more flexible.

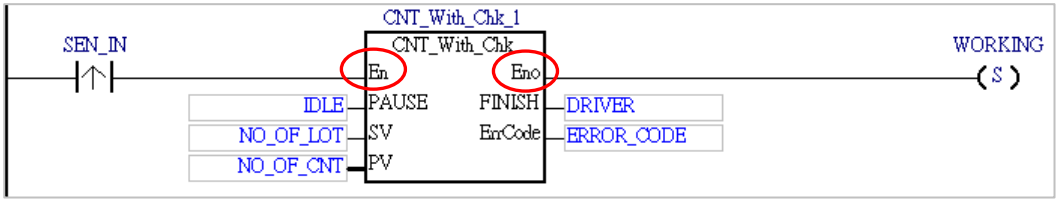
7.2 Structure of the Function Blocks in ISPSoft

Please refer to the following figure. The internal structure of a function block is similar to that of a POU of the program type. It is composed of a local symbol table and a program. A function block is edited in a way in which a POU of the program type is edited. The programming languages that users can make use of to create the function blocks in ISPSoft are ladder diagrams, instruction lists, structured texts, and function block diagrams, but are not sequential function charts. Besides, a function block can call another function block. When a function block is called, the programming language by which program in the function block is written does not need to be considered.



7.2.1 En Pin of a Function Block

A superior POU (caller) calls a function block in a way similar to the way in which an applied instruction is executed. Whether a function block is executed depends on the logic state sent to the En pin of the function block. If the logic state sent to the En pin of a function block is ON, the function block will be executed. If the logic state sent to the En pin of a function block is OFF, the function block will not be executed. The Eno pin of a function block functions to send the logic state sent to the En pin of the function block. Besides, if a function block is used in a structured text, the function block does not have an En pin. Please refer to chapter 11 for more information.



7.2.2 Symbols in a Function Block

Function blocks are similar to POUs of the program type in that users can create local symbols in the function blocks. Local symbols can also be used in a function block. However, if local symbols are used in a function block in a project, the function block may not portable. If the function block is imported into another project, the project may not have the same global symbols.

The symbol classes related to function blocks are described below. Please refer to chapter 6 for more information.

- Symbol class

Class	Description
VAR	The symbols of this class are for the operations in function blocks only. After a function block is executed, the value of the symbol in the function block will be retained.
VAR_INPUT	A symbol of this class is used as an input pin of a function block. It receives the value of an operand specified by the superior POU (caller). When the function block is executed, the value of the operand is copied into the function block.
VAR_OUTPUT	A symbol of this class is used as an output pin of a function block. After the execution of a function block is complete, the operation result will be sent to the operand specified by the superior POU (caller) through the symbol of this class.

Class	Description
VAR_IN_OUT	A symbol of this class is used as a feedback pin of a function block. When the function block is called, the value of an operand specified by the superior POU (caller) is sent to the function block through the symbol of this type. After the function is executed, the operation result is sent to the operand specified by the superior POU (caller).

- Data types

Model	Description
AH500	BOOL, WORD, DWORD, LWORD, INT, DINT, LINT, REAL, LREAL, ARRAY, STRING, POINTER, T_POINTER, C_POINTER, HC_POINTER, and a function block
DVP	BOOL, WORD, DWORD, LWORD, REAL, ARRAY, STEP, TIMER, and COUNTER

***1. The data type of a symbol of the VAR_INPUT class, the VAR_OUTPUT class, or the VAR_IN_OUT class can not be STRING, STEP, COUNTER, or TIMER.**

***2. Please refer to section 7.2.4 for more information about POINTER, T_POINTER, C_POINTER, and HC_POINTER.**

- Assigning memories to the symbols in a function block

Model	Description
AH500	When the symbols whose data type is a function block, and the local symbols created in a function block are declared, users can not assign device addresses to them. The system will automatically assign memories reserved by the CPU module.
DVP	Every symbol whose data type is a function block is assigned a P device. Besides, the system automatically assigns devices in the PLC to the local symbols in a function block instance according to the data types of the local symbols.

***. Please refer to sections 7.2.5~7.2.7 for more information about function block instances and the assignment of memories.**

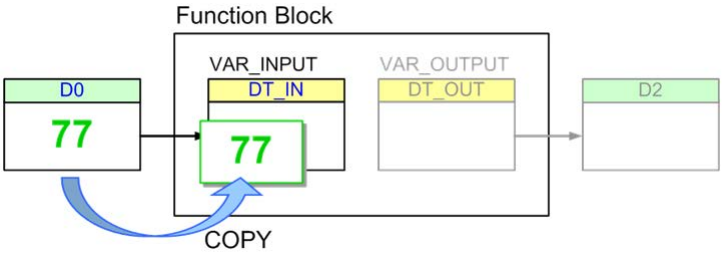
7.2.3 Input/Output Pins of a Function Block

When a POU calls a function block, it sends the value in a device to the input pin of the function block. After the function block is executed, the operation result is sent to the device in the POU through the output pin of the function block.

Please refer to the following figures.

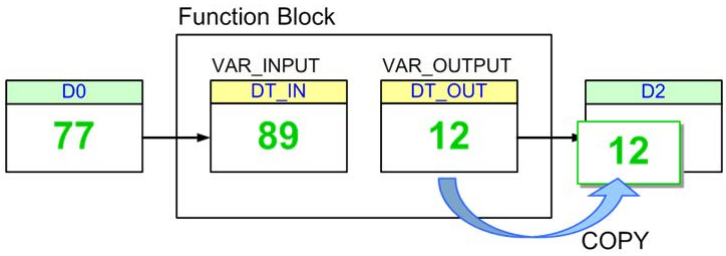
- **Before a function block is executed**

  The caller assigns D0 to DT_IN, the input pin of the function block. When the function block is called, the system sends the present value in D0 to DT_IN. (DT_IN is a symbol of the VAR_INPUT type.)

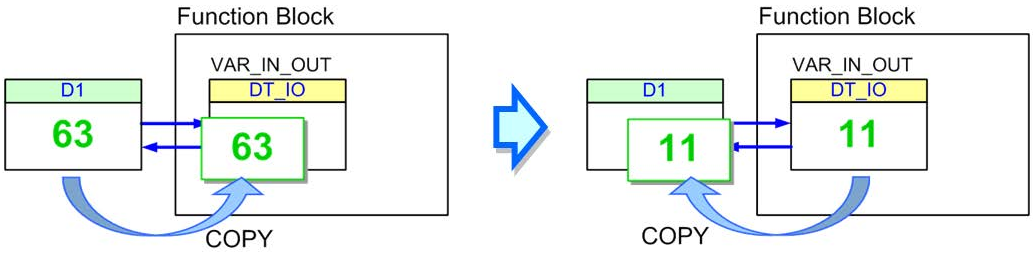


- **After a function block is executed**

  After the function block is executed, the value of DT_IN becomes 89. The value in D0 is unchanged. Besides, DT_OUT sends the final operation result to D2, the device specified by the caller. Even if the value in D2 is overwritten during the execution of the program, the value of DT_OUT will remain unchanged. (DT_ OUT is a symbol of the VAR_OUTPUT class.)

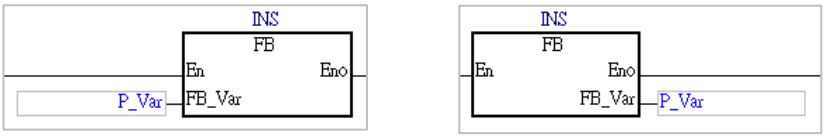


Although the value of the symbol of the VAR_IN_OUT class will be sent to the device that the caller assigns to the symbol after the function block is executed, the present value in the device specified by the caller is sent to the symbol before the symbol sends the final operation result to the device.



***. The data type of the symbol of the VAR_IN_OUT class in a function block in a project for an AH500 series PUC module can be POINTER. The caller can send the device address to the function block through the symbol whose data type is POINTER. Please refer to the following section for more information.**

Please refer to the function blocks below. FB_Var is the pin of the function block, and P_Var is the operand that the superior POU (caller) assigns to FB_Var. The data type of P_Var and that of FB_Var must conform to the basic principle described below. However, in order to make the writing of a program more convenient, ISPSoft provides some special principles. As long as the data type of P_Var and that of FB_Var conform to one of the special principles, the system will regard the data types as legal data types even if the data types does not conform to the basic principle.



7

- **Basic principle for the general data types**
  The data type of P_Var must be the same as the data type of FB_Var.
- **Special principles for the general data types** (Please refer to the following table.)
  1. If either the data type of P_Var or the data type of FB_Var is WORD, DWORD, or LWORD, the data type of the other symbol can be WORD, DWORD, LWORD, INT, DINT, LINT, REAL, or LREAL. If this condition is met, the data length of P_Var can be larger than or equal to the data length of FB_VAR.
  2. If the data type of P_Var is STEP, the data type of FB_Var can be BOOL. The state of the step in the caller will be used in the function block.

Data type		FB_Var (function block)								
		WORD	DWORD	LWORD	INT	DINT	LINT	REAL	LREAL	BOOL
P_VAR (caller)	WORD	✓			✓					
	DWORD	✓	✓		✓	✓		✓		
	LWORD	✓	✓	✓	✓	✓	✓	✓	✓	
	INT	✓			✓					
	DINT	✓	✓			✓				
	LINT	✓	✓	✓			✓			

Data type		FB_Var (function block)								
		WORD	DWORD	LWORD	INT	DINT	LINT	REAL	LREAL	BOOL
P_VAR (caller)	REAL	✓	✓					✓		
	LREAL	✓	✓	✓					✓	
	BOOL									✓
	STEP									✓

In addition to the principles for the general data types described above, there are principles for the symbols whose data type is ARRAY.

- **Basic principles for the symbols whose data types are ARRAY**
  1. The array type of P_Var must be the same as the array type of FB_Var.
  2. If the array type of P_Var is the same as the array type of FB_Var, the size of the array for P_Var can be larger than the size of the array for FB_VAR.
- **Special principles for the symbols whose data types are ARRAY**
  1. When the array types are compared, the first special principle for the general data types applies.
  2. If either the data type of P_Var or the data type of FB_Var is ARRAY, the system will regard the other symbol whose data type is not AARAY as a symbol whose data type is ARRAY, and the size of the array for the symbol will be one element.
  3. If either the array type of P_Var or the array type of FB_Var is WORD, DWORD, or LWORD, the data type of the other symbol can be WORD, DWORD, LWORD, INT, DINT, LINT, REAL, or LREAL. Besides, users can compare the data lengths by means of summation. For example, a symbol whose array type is DWORD is compatible with a symbol whose array type is INT if the size of the array for the latter symbol is two elements, and a symbol whose array type WORD is compatible with a symbol whose array type is LINT if the size of the array for the former symbol is four elements. However, a symbol whose array type is INT will not compatible with a symbol whose array type is LINT even if the size of the array for the former symbol is four elements. Neither the data type of the former symbol nor the data type of the latter symbol is WORD. The principle is not satisfied.

The following are examples.

✓ Example 1: The data type of P_Var (in the caller) is INT, and the data type of FB_Var (in the function block) is INT.

➔ The basic principle for the general data types is satisfied. As a result, the data types are legal.

✓ Example 2: The data type of P_Var (in the caller) is WORD, and the data type of FB_Var (in the function block) is INT.

➔ The first special principle for the general data types is satisfied. As a result, the data types are legal.

✓ Example 3: The data type of P_Var (in the caller) is LWORD, and the data type of FB_Var (in the function block) is DINT。

➔ The first special principle for the general data types is satisfied. As a result, the data types are legal.

✗ Example 4: The data type of P_Var (in the caller) is WORD, and the data type of FB_Var (in the function block) is LINT.

➔ Although the data type of P_Var is WORD, the data length of P_Var is less than that of FB_Var. As a result, the data types are illegal.

✗ Example 5: The data type of P_Var (in the caller) is LINT, and the data type of FB_Var (in the function block) is DINT.

➔ The data type of P_Var is not the same as the data type of FB_Var. As a result, the data types are illegal.

✓ Example 6: The array type of P_Var (in the caller) is INT, and the size of the array for P_Var (in

the caller) is four elements. The array type of FB_Var (in the function block) is INT, and the size of the array for FB_Var (in the function block) is three elements.

➔ The basic principles for the symbols whose data types are ARRAY are satisfied. As a result, the array types are legal.

✓ Example 7: The array type of P_Var is WORD, and the size of the array for P_Var is two elements. The data type of FB_Var is WORD (not ARRAY).

➔ The second special principle for the symbols whose data types are ARRAY is satisfied. As a result, the array type and the data type are legal.

✓ Example 8: The array type of P_Var is WORD, and the size of the array for P_Var is four elements. The data type of FB_Var is DWORD (not ARRAY).

➔ The third special principle for the symbols whose data types are ARRAY is satisfied. As a result, the array type and the data type are legal.

✗ Example 9: The array type of P_Var is DINT, and the size of the array for P_Var is four elements. The array type of FB_Var is INT, and the size of the array for FB_Var is two elements.

➔ The array type of P_Var is not the same as the array type of FB_Var. As a result, the array types are illegal.

✓ Example 10: The data type of P_Var is DWORD (not ARRAY). The array type of FB_Var is WORD, and the size of the array for FB_Var is two elements.

➔ The third special principle for the symbols whose data types are ARRAY is satisfied. As a result, the data type and the array type are legal.

Additional remark

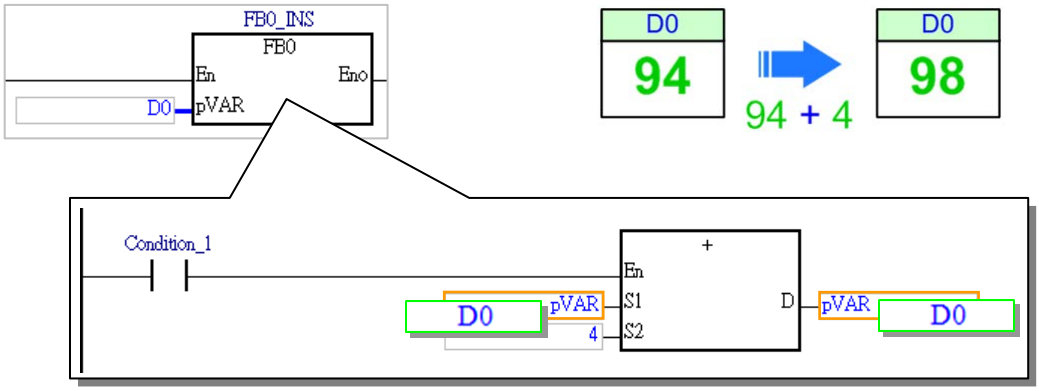
(a) If the principles above are satisfied, the data length of P_Var (in the caller) is larger than that of FB_Var (in function block), and the value of P_Var is sent to FB_Var, the values of the bits starting from the lowest bit in P_Var are retrieved, and the number of values retrieved conforms to the data length of FB_Var. If principles above are satisfied, the data length of P_Var (in the caller) is larger than that of FB_Var (in function block), and the value of FB_Var is sent to P_Var, the values of the bits starting from the lowest bit in P_Var are overwritten, the number of values overwritten conforms to the data length of FB_Var, and the values which are not overwritten remains unchanged. Users have to be more careful about the correctness of the data when they use P_Var and FB_Var.

(b) The data type of a symbol in a function block in a project for an AH500 series CPU module can be POINTER. Please refer to section 7.2.4 for more information.

7.2.4 Symbol Whose Data Type is POINTER, T_POINTRER, C_POINTER, or HC_POINTER

A symbol whose data type is POINTER, T_POINTER, C_POINTER, or HC_POINTER functions to send the symbol or the device address specified by the caller to the function block. The use of a symbol whose data type is POINTER, T_POINTER, C_POINTER, or HC_POINTER is quite different from the use of a symbol whose data type is a general data type. A symbol whose data type is a general data type in a function block either receives the value from the device specified by the caller or sends the operation result to the device specified by the caller. If the data type of a pin of a function block is POINTER, T_POINTER, C_POINTER, or HC_POINTER, the data sent to the function block will be the symbol or the device address assigned to the pin. As a result, the object processed in a function block is the object which is specified.

The caller assigns D0 to pVAR in the function block, and the data type of pVAR in the function block is POINTER, T_POINTER, C_POINTER, or HC_POINTER, as shown in the figure below. If the function block is called and executed, pVAR in the function block will be regarded as D0.



The principles of using a symbol whose data type is POINTER, T_POINTER, C_POINTER, or HC_POINTER are as follows.

1. Only AH500 series CPU modules support a symbol whose data type is POINTER, T_POINTER, C_POINTER, or HC_POINTER.
2. The symbols whose data types are POINTER can only be declared in the function blocks in a project, and they must be symbols of VAR_IN_OUT class.
3. The pointers and the restrictions are described below.

Pointer	Restriction	
General pointer POINTER	**Maximum number**	16 symbols whose data types are POINTER can be declared in a function block.
	Reference	Symbol whose data type is WORD/DWORD/LWORD/INT/DINT/LINT, data register, link register, or X/Y device (e.g. X0/Y0)
Pointer for a timer T_POINTER	**Maximum number**	8 symbols whose data types are T_POINTER can be declared in a function block.
	Reference	Timer or symbol whose data type is TIMER
Pointer for a counter C_POINTER	**Maximum number**	8 symbols whose data types are C_POINTER can be declared in a function block.
	Reference	Counter or symbol whose data type is COUNTER
Pointer for a high-speed counter HC_POINTER	**Maximum number**	8 symbols whose data types are HC_POINTER can be declared in a function block.
	Reference	32-bit counter or symbol whose data type is COUNTER and to which a 32-bit counter is assigned

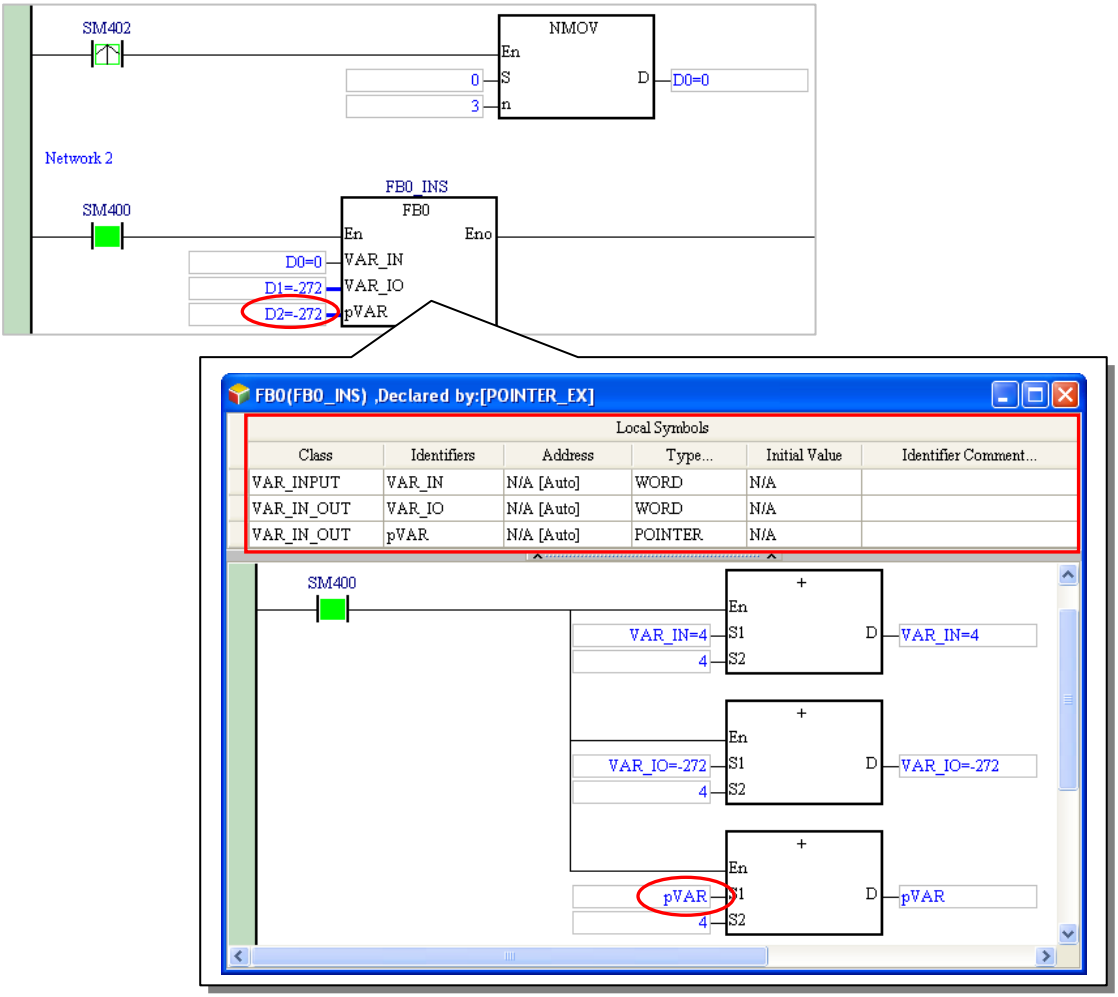
- **Example of using a symbol whose data type is POINTER**

  The function block FB0 has three input pins. The data types and the classes are in the red frame in the local symbol table below. D0, D1, and D2 in the POU (caller) are assigned to the three pins. (The initial values in D0, D1, and D2 are 0.) The execution of the program is as follows.

  - D0 is assigned to VAR_IN. VAR_IN is a symbol of the VAR_INPUT class, and the data type of VAR_IN is WORD. When the function block is called, the present value in D0 is copied into VAR_IN. After the operation inside the function block is performed, 4 will be added to the value of VAE_IN. The operation result does not affect D0, and therefore the value in D0 is always 0.
  - D1 is assigned to VAR_IO. VAR_IO is a symbol of the VAR_IN_OUT class, and the data type of VAR_IO is WORD. When the function block is called, the present value in D1 is copied into VAR_IO. After the operation inside the function block is performed, 4 is added to the value of VAR_IO. Owing to the fact that VAR_IO is a symbol of the VAR_IN_OUT class, the value of VAR_IO is copied into D1 after the function block is executed. The value sent from D1 to VAR_IO is an accumulated value. The value in D1 accumulates during the execution of the function block.
  - D2 is assigned to pVAR. Owing to the fact that the data type of pVAR is POINTER, the sum

will be written into D2 after 4 is added to D2. The value in D2 accumulates during the execution of the function block.

Although D1 and D2 get the same execution results, they have different execution processes. Besides, owing to the fact that pVAR is regarded as D2, the present value of pVAR is not displayed in a way in which the present value of VAR_IO is displayed when the online monitoring function is enabled.



***. SM400 is a special auxiliary relay in an AH500 series CPU module. It is ON when the CPU module is powered. SM402 is ON during the first scan cycle.**

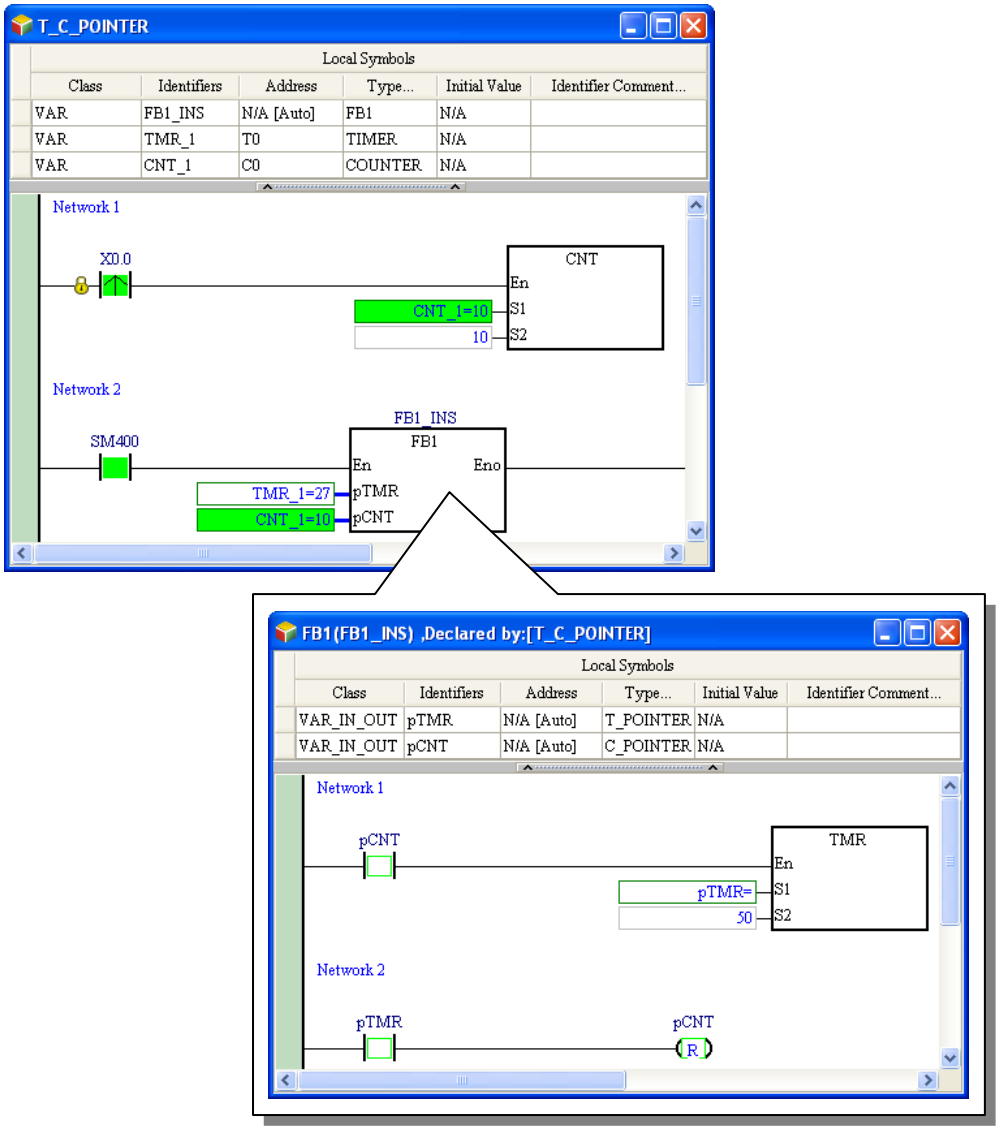
- **Example of using a symbol whose data type is T_POINTER and a symbol whose data type is C_POINTER**

  Owing to the fact that users can not declare a symbol whose data type is TIMER and a symbol whose data type is COUNTER in a function block in a project for an AH500 series CPU module, they can send a timer, and a counter to a function block by means of declaring a symbol whose data type is T_POINTER, and a symbol whose data type is C_POINTER in the function block.

  The function block FB1 has two input pins. pTMR is a symbol whose data type is T_POINTER, and pCNT is a symbol whose data type is C_POINTER. The caller assigns TMR_1 and CNT_1 to the two pins. (The data type of TMR_1 is TIMER, and the data type of CNT_1 is COUNTER.)

  In the superior POU, the value of CNT_1 increases by one when X0.0 is turned from OFF to ON. CNT_1 will not be ON until the value of CNT_1 is 10. Owing to the fact that TMR_1 and CNT_1 are assigned to the function block, the state of CNT_1 (pCNT) is judged when the function block is executed. When CNT_1 (pCNT) is ON, TMR_1 (pTMR) measures the time interval. When the value of pTMR is 50, CNT_1 (pCNT) is reset.

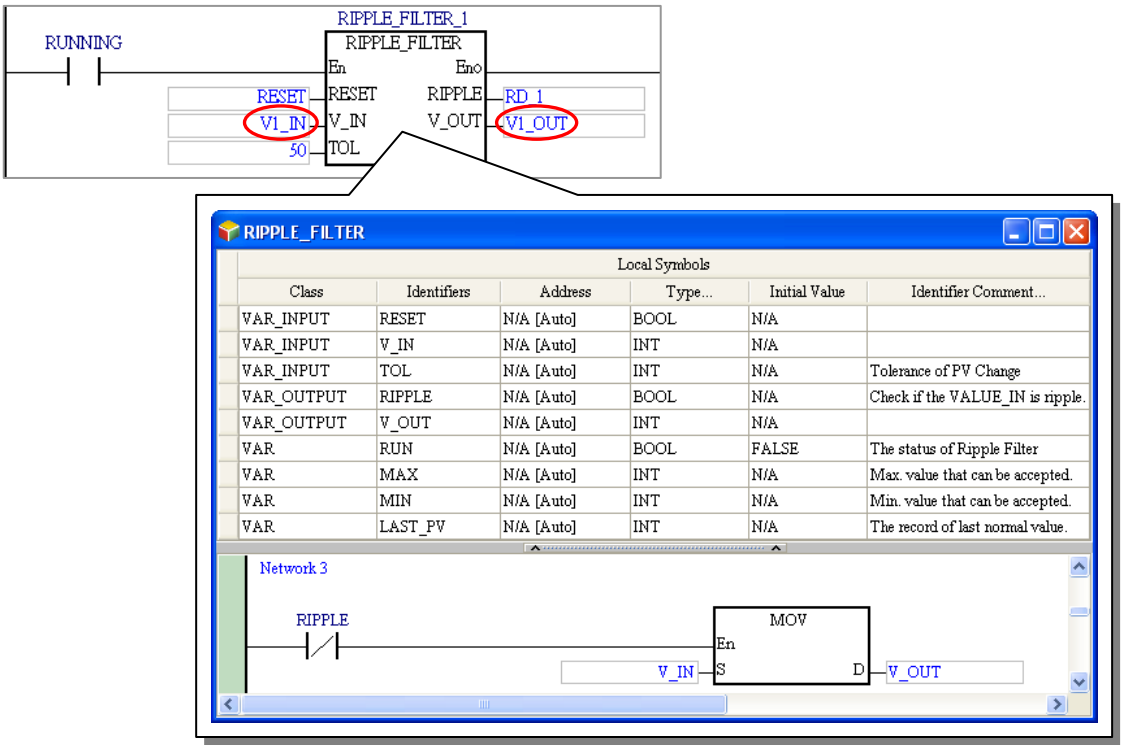
  Likewise, the present value of pTMR and that of pCNT are not displayed when the online monitoring function is enabled.



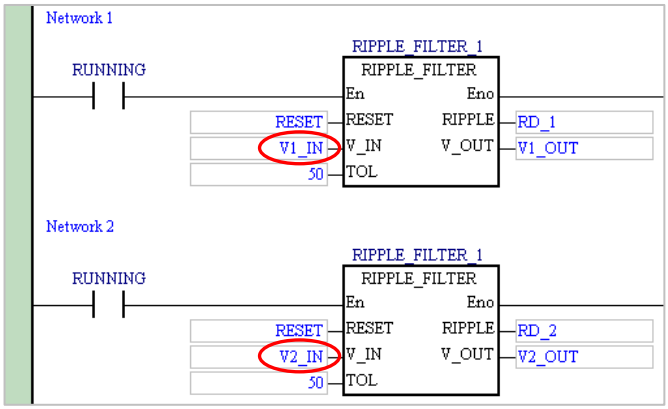
***. SM400 is a special auxiliary relay in an AH500 series CPU module. It is ON when the CPU module is powered.**

7.2.5 Function Block Definition and Function Block Instance

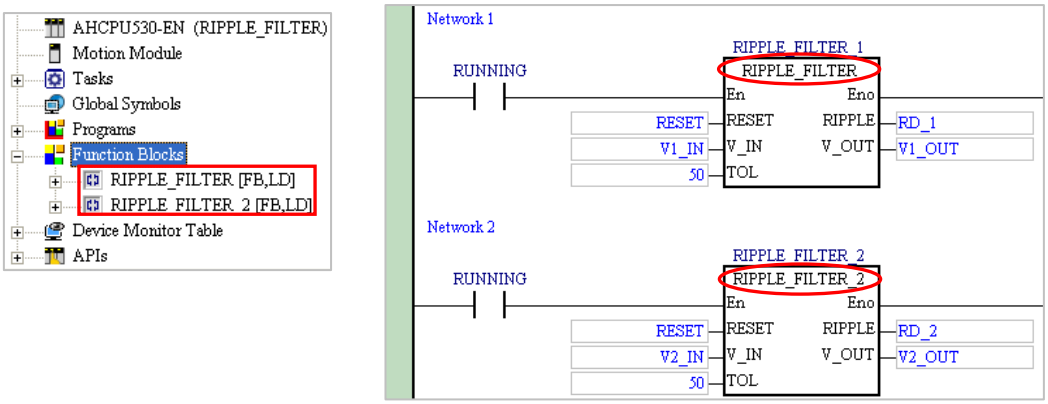
The definition of a function block and an example of a function block are important concepts. Please refer to the following example. Whenever the superior POU (caller) is executed, the value of V1_IN is sent to the function block RIPPLE_FILTER. After the value sent to the function block this time is compared with the value recorded last time, the final operation result will be sent to V1_OUT in the superior POU. In order to record the operation result, a local symbol in which the operation result can be stored is declared in the function block. When the program is compiled, the system assigns a substantial memory block to this local symbol.



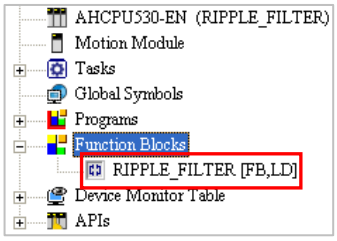
If several pieces of data in the superior POU are processed in the same way, and theses pieces of data are independent of one another, the execution result may be incorrect, as shown in the figure below. After the value of V2_IN is sent to the function block, it will be compared with the value gotten from the operation performed on V1_IN last time. As a result, V1_IN and V2_IN interfere with each other.



If the same POU of the function block type is created a gain, the execution result is correct although different memory blocks are assigned to the local symbols in the two POUs of the function block type when the program is compiled. However, the same work is repeated when users write the program.

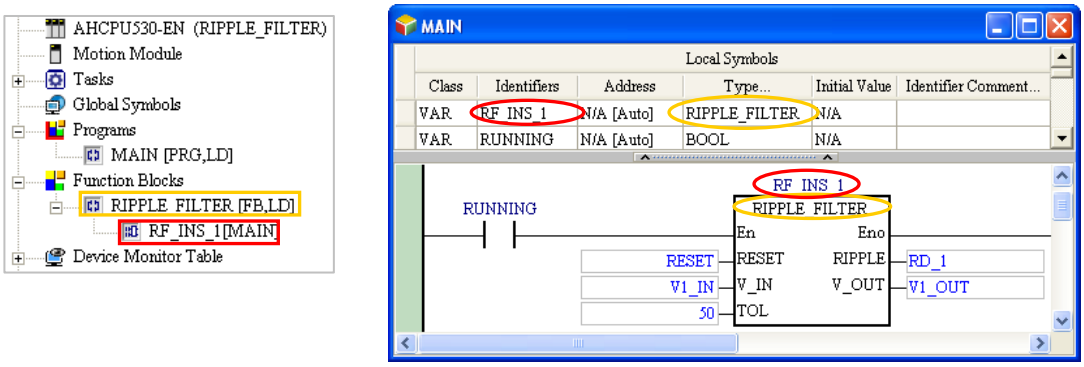


In view of this, the concept related to function block instances is introduced into IEC 61131-3. After users add a POU of the function block type, declare local symbols in the POU, and write a program in the POU, they will get a function block definition. The function block definition is like a document. It does not participate in any operation, and does not appropriate any resources that the PLC uses.

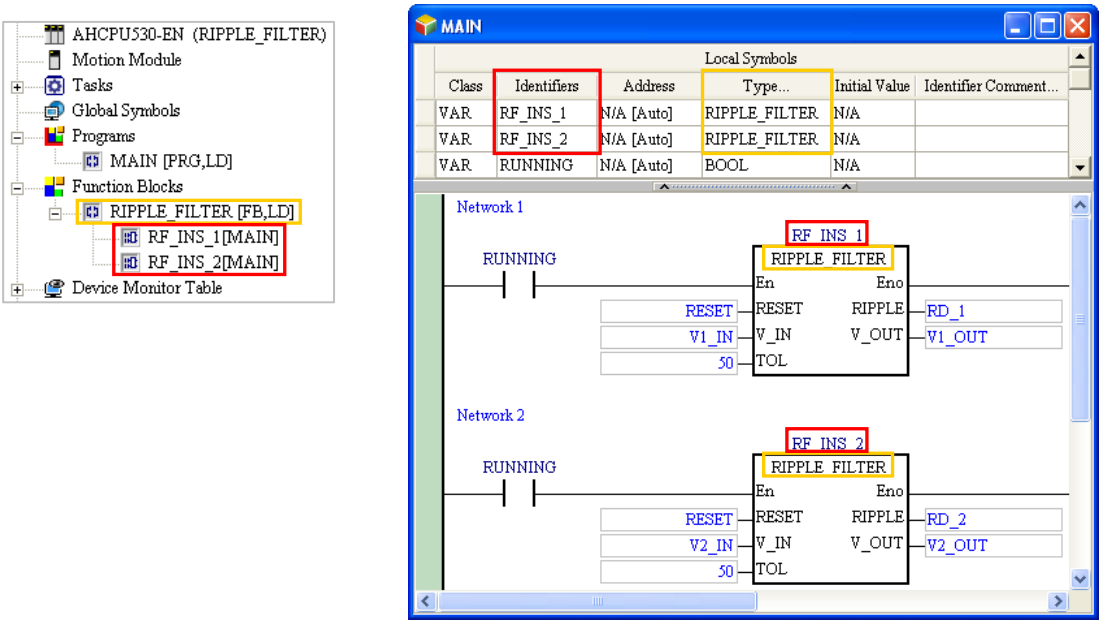


If a POU of the program type wants to call this function block, the users must declare a symbol whose data type is a function block in advance. In other words, the users have to produce an object which takes part in an operation according to the function block definition. The object produced is a function block instance.

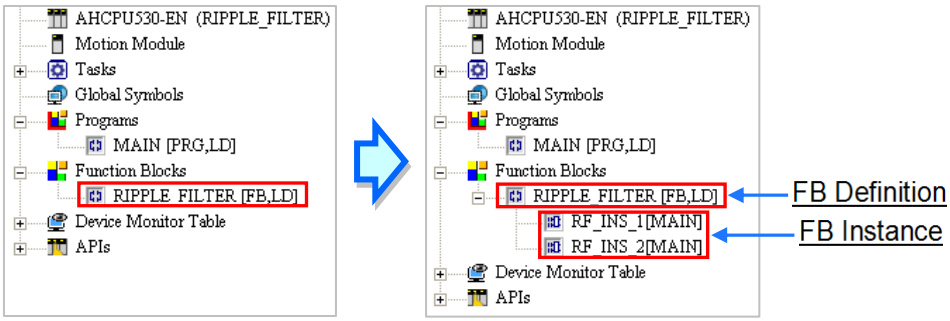
When the program is compiled, the system assigns a substantial memory block to the function block instance and the local symbols in the function block instance according to the definition.



If the same operation is required, and a different memory block is also required, users can make use of the characteristic of a function block instance. After a different symbol is declared for a function block which requires a different memory block, the system will regard the different symbol as a different function block instance. When the program is compiled, a different memory block is assigned to the different function block instance.



As shown in the figure below, only the function block definition is under **Function Blocks** in the project management area before the program in the project is compiled. After the program is compiled, the function block instances will be under the function block definition. The format of a function block instance is **Instance name[POU name]**.



If a symbol whose data type is general data type is declared, a device is assigned to the symbol when the program is compiled. Likewise, if a symbol whose data type is a function block is declared, a corresponding function block instance is produced when the program is compiled. Besides, the number of function block definitions which produce the instances in a project for an AH500 series CPU module can not be larger than 1024. (The function block definitions which do not produce the instances are not included.)

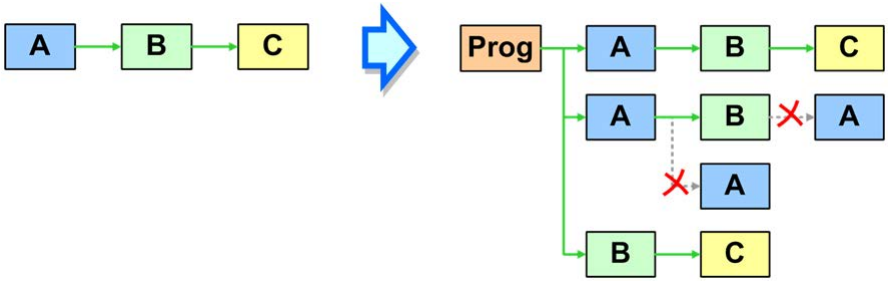
In sum, the operation of a function block involves a definition and an instance. Function block definitions are related to creation, and function block instances are related to execution. Take products for example. A function block definition corresponds to the specifications for a product developed by engineers, and a function block instance corresponds to the product produced according to the specifications.

7.2.6 Calling Relation Between Function Blocks

In ISPSoft, a function block can calls another function block, and 32 layers of function blocks at most can be called. The function block instance called by a POU of the program type is the first layer.

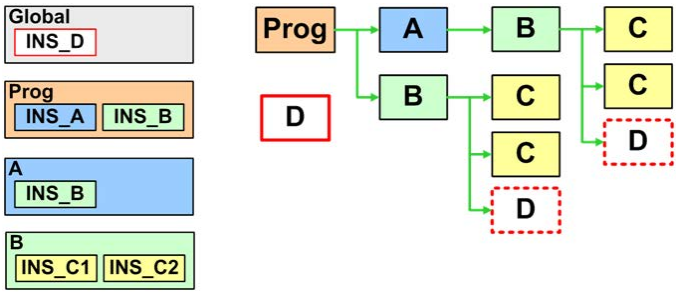
The program in function block A calls function block B, and the program in function block B calls function block C. In other words, function block A is superior to function block B and function block C, and function block B is superior to function block C. In ISPSoft, a function block is not allowed to call

itself. As a result, if function block A is superior to function block B, function block B and function block C can not call function block A, but the POU of the program type can call function block A, function block B, or function block C.



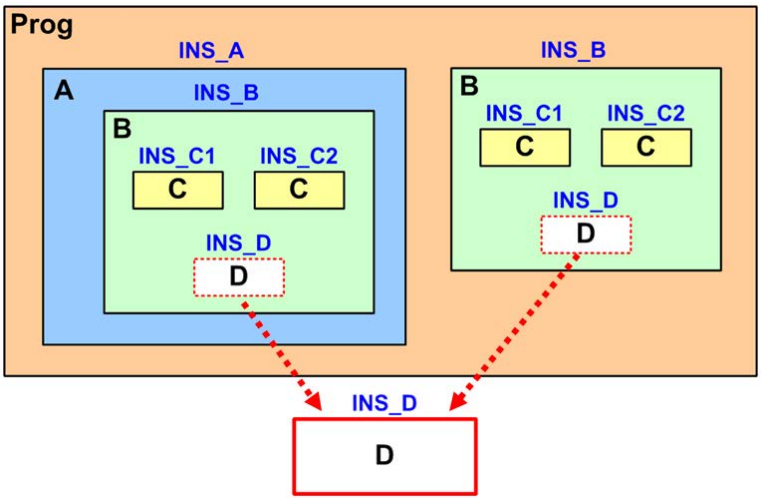
If function block A, function block C, and function B are not called by any POUs of the program type, they do not participate in the execution of the program, and the corresponding function block instances are not produced when the program is compiled. After function block A is declared in a certain POU of the program type or in a the global symbol table in a project, the corresponding function block instances will be produced according to the relation among the three function blocks during the compiling of the program.

Please refer to the declaration of the function block instances and the relation among the function blocks below. Function block D is declared in the global symbol table while the other function block instances are declared in the local symbol tables in the superior POUs (callers).

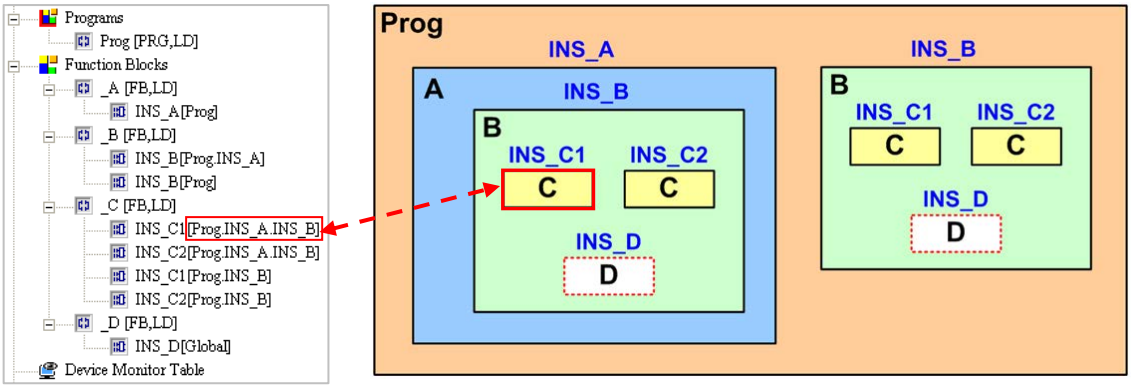


Declaration position		Function block which is declared
Global symbol table		One instance of function block D: INS_D
Local symbol table	POU of the program type	One instance of function block A: INS_A One instance of function block B: INS_B
	Function block A	One instance of function block B: INS_B
	Function block B	Two instances of function block C: INS_C1 and INS_C2

The function block instances produced when the program is compiled according to the relation described above are as follows. Owing to the fact two instances of function block C are declared in the local symbol table in function block B, two instances of function block C are produced when another instance of function block B is produced. Besides, owing to the fact that function block D is declared in the global symbol table, the function of the function block instance and the data stored in the function block instance can be used by all the POUs. In other words, the system only produces one instance of function block D. Although function block D is called by the two POUs, the same instance of function block D is executed in the two POUs.



The items which will appear in the project management area after the program is compiled are shown in the figure below. The superior POU of the program type and the inferior POUs are in the brackets at the end of a function block instance.



Users can not declare function block instances in the local symbol table in a function block in a project for a DVP series PLC. If a function block wants to call another function block, the inferior function block instance must be declared in the global symbol table. As a result, users have to be more careful about the correctness of the data when a function block calls another function block. If an independent memory block is required, users have to declare another instance of a function block in the global symbol table.

Users are allowed to declare function block instances in the local symbol table in a function block in a project for an AH500 series CPU module. If an inferior function block instance which is called by a function block is not called by another POU, the inferior block instance can be declared in the local symbol table in the function block. Besides, in a project for an AH500 series CPU module, the memory block the system assigns to a function block instance depends on the position where the function block instance is declared. Please refer to section 7.2.7 for more information.

7.2.7 Assigning a Memory Block to a Function Block

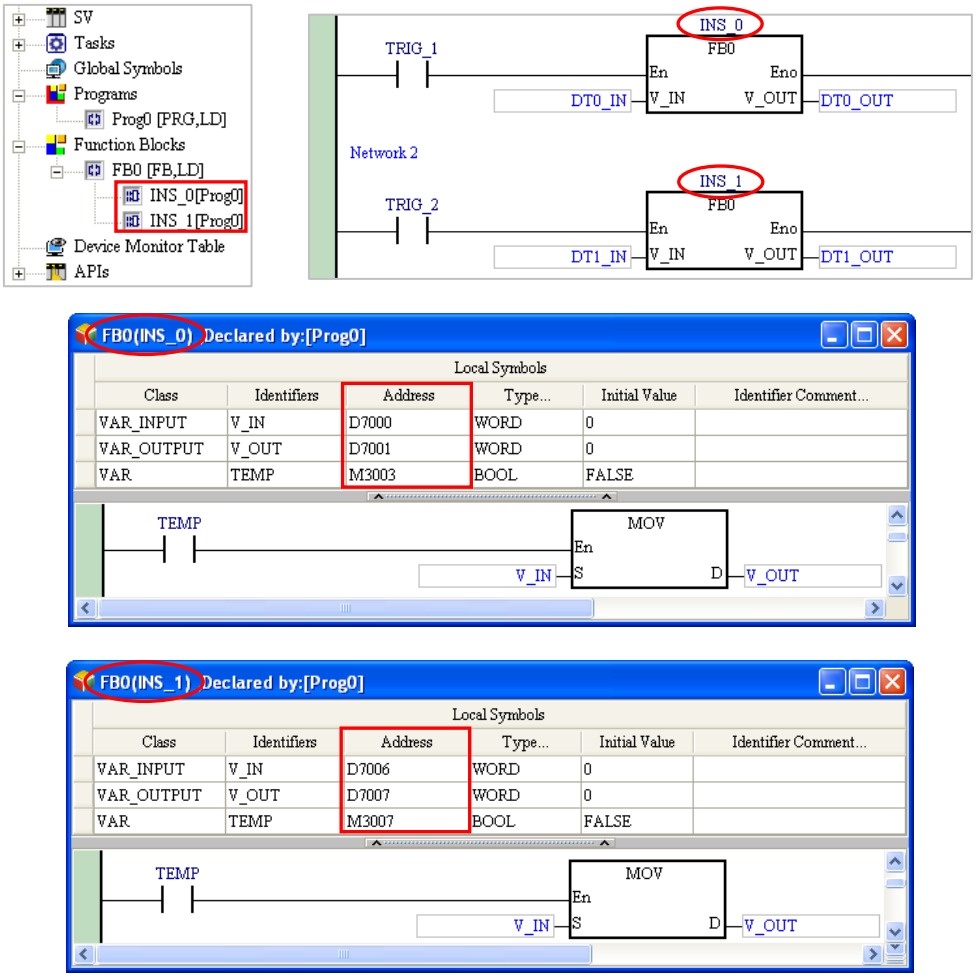
If users declare a local symbol to which the system automatically assigns a device address in a function block, the system assigns an independent memory block to the local symbol when the program is compiled and the instance is produced. If users declare a local symbol to which they assign a device address in a function block, the local symbol will be regarded as an absolute address. When the program is compiled, the system does not assign a memory block to this local symbol. Even in a different instance of the function block, the device that the users assign to the local symbol participate in the operation

A DVP series PLC and an AH500 series CPU module have different mechanisms for assigning memory blocks.

- **DVP series PLC**

  When a DVP series PLC assigns memory blocks, every function block instance is automatically assigned a P device. If the POU name of a function block begins with “P0_” or “P1_”, P0 or P1 will be assigned to the function block instance after the program is compiled. If more than one instance of the function block is declared, an error occurs when the program is compiled. The error occurs because the same P device is used repeatedly.

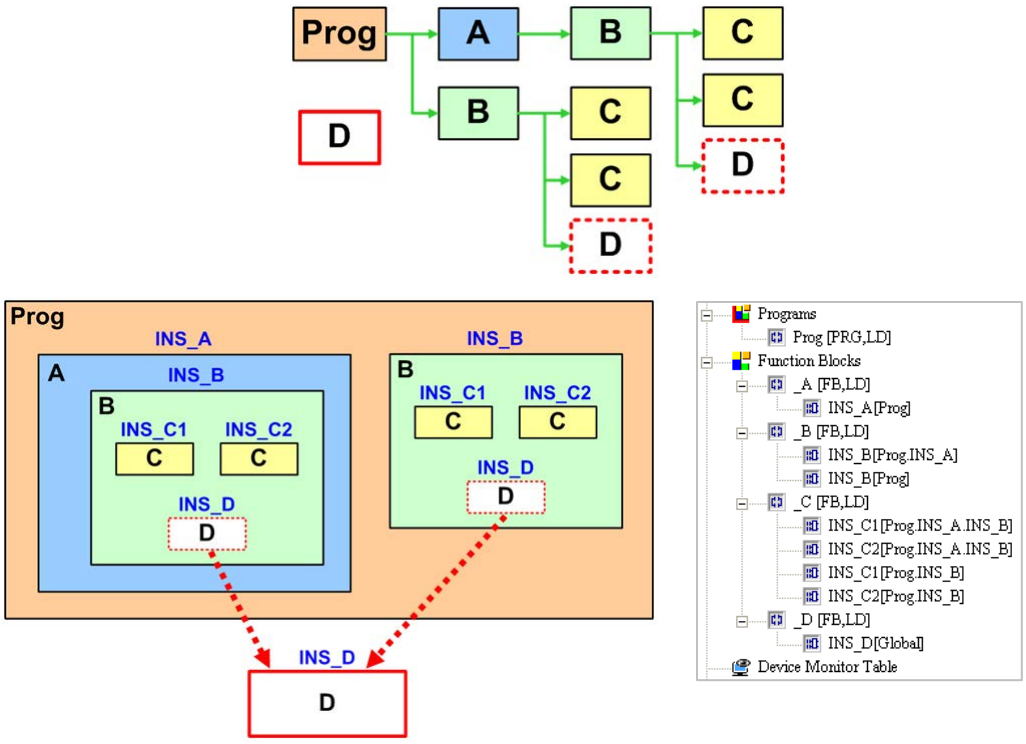
  The system assigns corresponding devices to the local symbols declared in a function block according to the data types of the local symbols. For example, the system assigns a data register to a local symbol whose data type is WORD, and an auxiliary relay to a local symbol whose data type is BOOL. As a result, the size of function block instance in a DVP series PLC depends on the local symbols declared, and different instances occupy different devices. Two function block instances produced in a DVP series PLC are shown in the figure below. After the windows for the two function block instances are opened, users can see in the local symbol tables that the system assigns different devices to the two instances.



- **AH500 series CPU module**

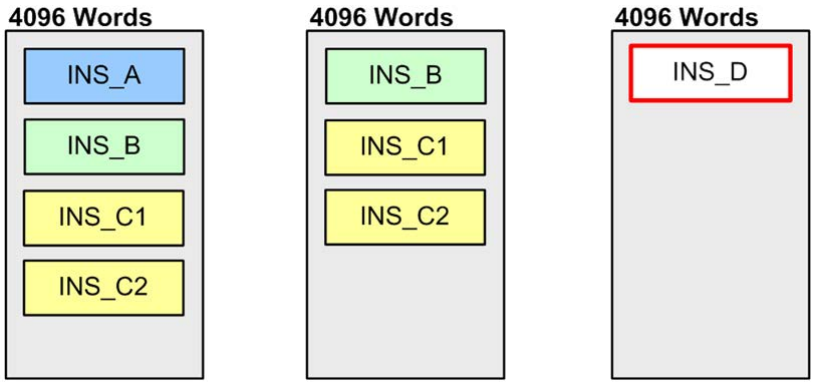
  The system in an AH500 series CPU module a reserves a fixed number of memory blocks for function block instances. The number of memory blocks varies with the model selected. Please refer to appendix B for more information about the specifications. The size of every memory block is 4096 words. Owing to the fact that the memory blocks assigned to the function blocks in a project for an AH500 series CPU module are memories reserved by the CPU module, users can not view the assignment result in the windows for the function blocks. The assignment of memory blocks to the function blocks in a project for an AH500 series CPU module is illustrated with the example in section 7.2.6.

  The relation among the function blocks in the example in section 7.2.6 is as follows. Please refer to section 7.2.6 for more information.



The system assigns a 4096-word memory block to the function block instance declared in a POU of the program type in a project for an AH500 series CPU module. The function block instance declared in the POU of the program type and the inferior function blocks are assigned the same memory block. Besides, the system assigns another 4096-word memory block to a function block instance declared in the global symbol table.

In the example above, the system assigns three 4096-word memory blocks.

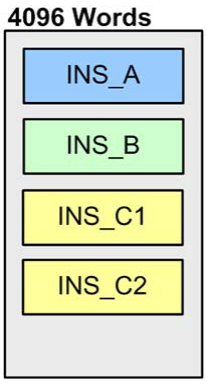


The memory blocks that an AH500 series CPU module reserves for function block instances are stored in the internal memory. The memory blocks assigned to the function blocks in a project for an AH500 series CPU module are not devices available to users. No matter what the size of the memory that a function block instance occupies is, the size of the memory block assigned to the function block instance is 4096 words. As a result, users have to be careful when they program a CPU module. The calculation of the size of a function block instance is as follows.

(1) The size of the memory that a function block instance occupies is 2 words.

(2) The size of the memory that a local symbol in a function block instance occupies is calculated by the word. The size which is less than one word is calculated as one word. The size of the memory that a symbol whose data type is a function block (inferior function block instance) occupies is calculated by means of (1).

Please refer to the following example. The local symbols declared in the function block definitions are listed in the table below.



Function block definition	Number of local symbols
Function block A	Two symbols whose data types are DWORD. One symbol whose data type is LREAL. One inferior function block instance.
Function block B	One symbol whose data type is WORD. Two symbols whose data type is REAL. Two inferior function block instances.
Function block C	Two symbols whose data type is BOOL. One symbol whose array type is BOOL and whose array size is 22 elements.

The calculation for the example is as follows.

(1) The size of the memory that a function block instance occupies is 2 words. There are four function block instances in the example, and therefore the size is 8 words.

(2) The size of the memory that a local symbol whose data type is not a function block in a function block instance occupies is calculated as follows.

- **Function block instance INS_A**
  - ➔ Two words (DWROD x 1)+Four words (LREAL x 1)
    The sum is 2+4=6 (words).
- **Function block instance INS_B**
  - ➔ One word (WORD x 1)+Four words (REAL x 2)
    The sum is 1+4=5 (words).
- **Function block instance INS_C1/INS_C2**
  The size of the memory that a local symbol occupies is calculated by the word. The size which is less than one word is calculated as one word. As a result, the size of the memory that a local symbol whose data type is BOOL occupies is calculated as one word. The size of the memory that a local symbol whose array type is BOOL and whose array size is 22 elements occupies is larger than one word, but is less than two words. The size is calculated as two words. The calculation is as follows.
  - ➔ Two words (BOOL x 2)+Two words (Boolean array which is composed of twenty-two elements x 1)
    The sum is 2+2=4. There are two instances of function block C, and therefore the size of the memory that the two function block instances occupy is 2x4=8 (words).

  The size of the memory that the function block instances and the local symbols occupy is 8+6+5+8=27 (words).

7.3 Using a Function Block

7.3.1 Basic Specifications for the Function Blocks

The specifications for the function blocks in ISPSoft are shown below. When users use function blocks, they have to make sure that the operation conforms to the specifications in the table. Otherwise, an error occurs when the program is compiled or executed.

- **Specifications for the function blocks**

Number of function block definitions	AH500: The number of function block definitions which produce the instances can not be larger than 1024. DVP: The number of function block definitions depends on the number of P devices available. (One function block instance occupies one P device.)
Number of function block instances	AH500: The maximum number of memory blocks depends on the model selected. (Please refer to appendix B.) DVP: The number of function block instances depends on the number of P devices available. (One function block instance occupies one P device.)
Size of a function block	AH500: The size of a memory block is 4096 words. DVP: The size of a function block depends on the devices assigned to the function block
Number of layers	A function block can calls another function block, and 32 layers of function blocks at most can be called.

- **Note**
  - The pins of a function block must be assigned operands. The operands can be device addresses or symbols. If the class of a symbol is VAR_INPUT, the operand assigned to the symbol can be a constant.
  - An inferior function block can not call the superior function block, and a function block can not call itself.
  - If a function block is exported, the definition of the function block is exported, but the definitions of the inferior function blocks and the global symbols used are not exported.
  - The En pins of the function blocks in ladder diagrams, sequential function charts, function block diagrams, and instruction lists must be connected to contacts. However, the En pins of the function blocks in structured texts do not have to be connected to contacts.
  - The programming languages which support the creation of function blocks include ladder diagrams, instruction lists, structured texts, and function block diagrams. Sequential function charts do not support the creation of function blocks.
  - The programming languages which support the calling of function blocks include ladder diagrams, instruction lists, structured texts, and function block diagrams. Users can also use function blocks in the actions in sequential function charts.
  - Users can modify function block definitions, but they can not modify function block instances. (Please refer to section 7.3.4.)
  - If the modification of a function block definition in the online mode results in the abnormal representation of the function block in the superior POU, users have to disable the online editing function, delete the abnormal representation of the function block, and add the normal representation of the function block. After the normal representation of the function block is added, the program can be compiled and downloaded. (Please refer to section 7.3.4 for more information.)
  - If a jump instruction is used in a function block, the jump destination must be in the same function block, and can not be outside the function block or in the inferior function block.
  - Users can not declare symbols whose data types are TIMER and symbols whose data types are COUNTER in the function blocks in a project for an AH500 series CPU module, but they can declare symbols whose data types are POINTER, symbols whose data types are T_POINTER, symbols whose data types are C_POINTER, and symbols whose data types are

HC_POINTER.

- ➢ Pulse instructions such as MOVP and CML can not be used in the function blocks in a project for an AH500 series CPU module. LDP/ANDP/ORP, LDF/ANDF/ORF, PLS, PLF, NP, PN, MC, MCR, and GOEND can not be used.
- ➢ AH500 series CPU modules provide special pulse instructions which can be used in function blocks. Please refer to the following section for more information.

***1. Please refer to section 5.4.1 for more information about creating a POU of the function block type, and section 6.2.2 for more information about declaring a function block instance.**

***2. Please refer to chapter 8~chapter 12 for more information about calling function blocks.**

7.3.2 Special Pulse Instructions Used in Function Blocks (Only for AH500 Series CPU modules)

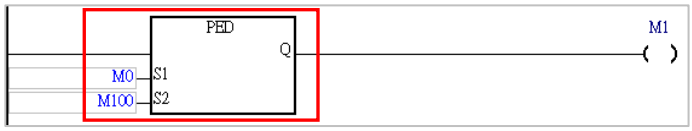
Pulse instructions such as LDP, LDF, NP, and PN can not be used in the function blocks in a project for an AH500 series CPU module, and therefore users can consider using the following instructions. Please refer to AH500 Programming Manual for more information about the instructions.

- PED and NED

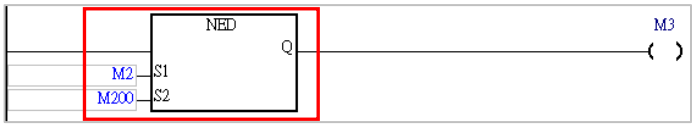
  PED and NED are similar to LDP and LDF in functions. PED and NED differ from LDP and LDF in that users have to specify Boolean devices. The Boolean devices that users specify when they use PED and NED in a project can not be used in other places in the project. Otherwise, an error will occur.

  - ➢ LDP (rising-edge detection) and LDF (falling-edge detection) can be used in a POU of the program type.

  - ➢ Users can use PED in place of LDP in a function block, but they have to specify a device, e.g. M100 in the figure below.



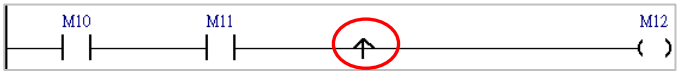
  - ➢ Users can use NED in place of LDF in a function block, but they have to specify a device, e.g. M200 in the figure below.



- FB_NP and FB_PN

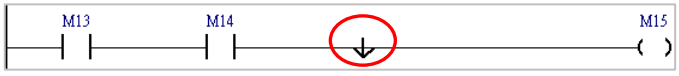
  FB_NP and FB_PN are similar to NP and PN in functions. FB_NP and FB_PN differ from NP and PN in that users have to specify Boolean devices. The Boolean devices that users specify when they use NED and PED in a project can not be used in other places in the project. Otherwise, an error will occur.

  - ➢ NP can be used in a POU of the program type.

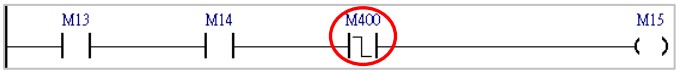


  - ➢ Users can use FB_NP in place of NP in a function block, but they have to specify a device, e.g. M300 in the figure below.

  - ➢ PN can be used in a POU of the program type.



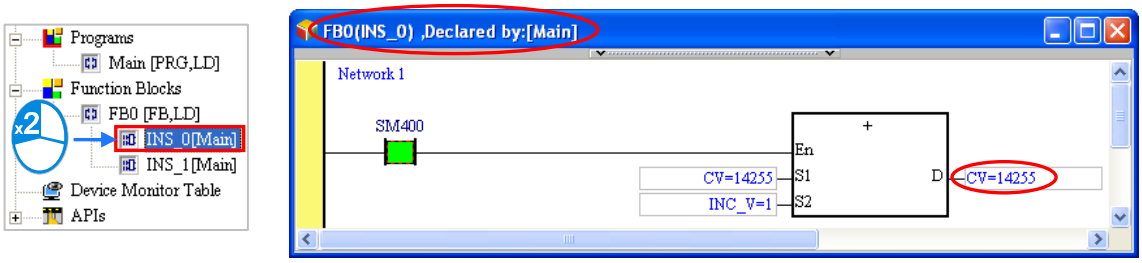
➢ Users can use FB_PN in place of PN in a function block, but they have to specify a device, e.g. M400 in the figure below.

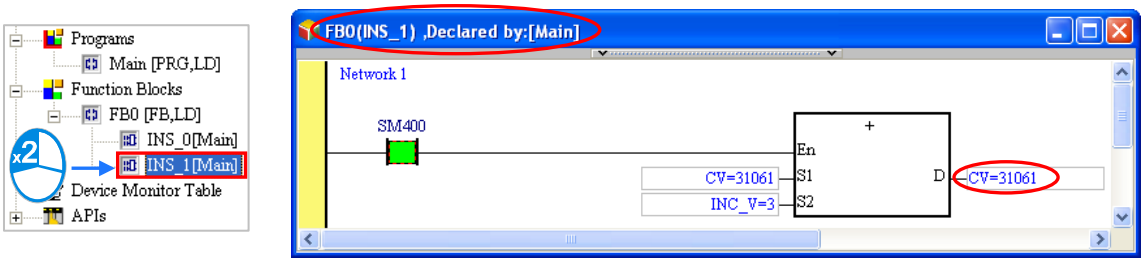


7.3.3 Monitoring the Program in a Function Block

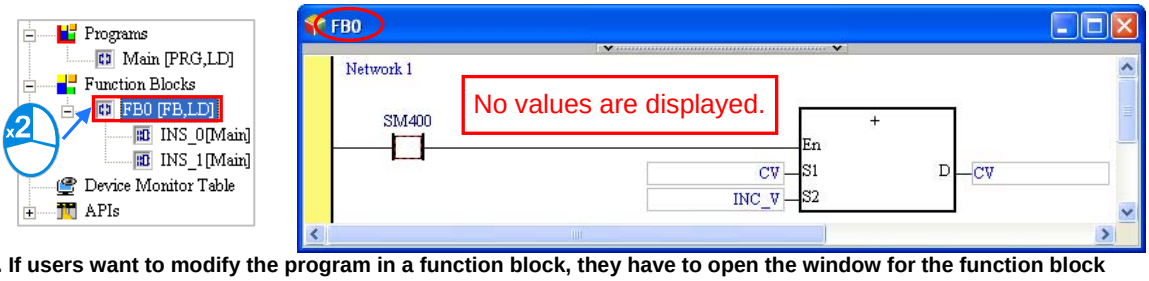
It is the functions block instances in a project that participate in the execution of the program in the project. If users want to monitor the program in a function block online, they have to open the window for the function block instance.

If users want to monitor the program in a function block online, they have to open the window for the function block instance, as shown below. The function block instances below operate independently.





If users open the window for a function block definition, no values will be displayed in the window, and users can not monitor the program in the function block.



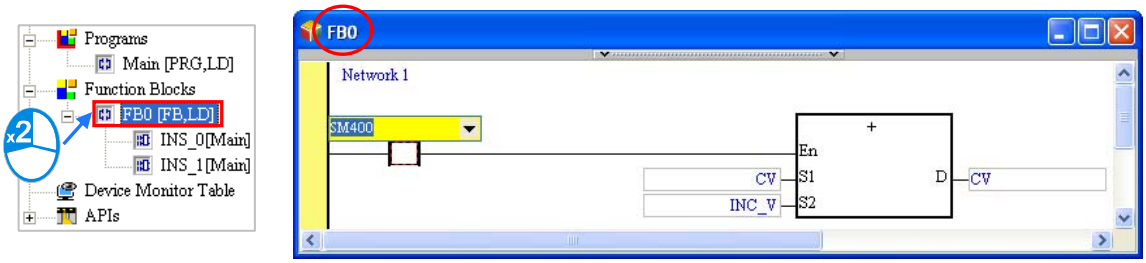
***. If users want to modify the program in a function block, they have to open the window for the function block definition. Please refer to section 7.3.4 for more information.**

7.3.4 Modifying the Program in a Function Block

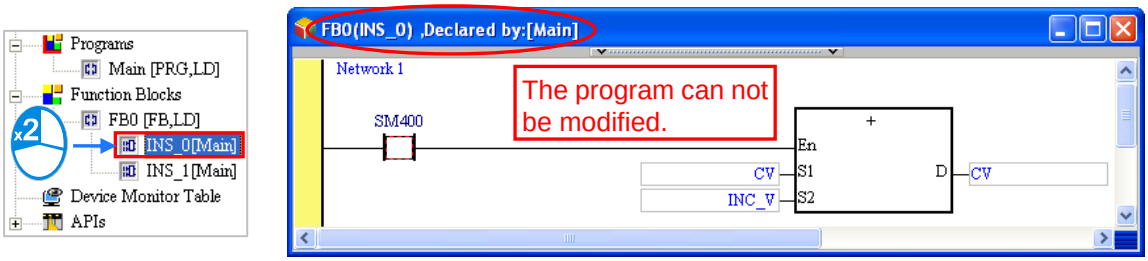
The object produced from a function block definition is a function block instance. The program in a function block instance can not be modified, but the program in a function block definition can be modified. After the program in a function block definition is modified, the modification result will be applied to the new instances produced from the function block definition. The new function block instances will not be produced until the program is compiled. Likewise, if users want to modify the program in a function block online, they have to open the window for the function block.

After users double-click a function block definition in the project management, they can modify the

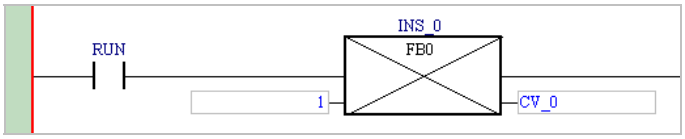
program in the window which appears, as shown below.



If users open the window for a function block instance, they can not modify the program in the window.



If a function block is called by another POU, the modification of the function block will result in the abnormal representation of the function block in the superior POU, as shown below. Users have to delete the abnormal representation of the function block, and add the normal representation of the function block.



7.4 Example

An example of creating a ladder diagram is shown in chapter 4. In section 5.6.1, the program created in chapter 4 is rewritten by means of IEC 61131-3. In section 6.3, the device names in the program created in section 5.6.1 are replaced by symbols. In the following sections, function blocks will be used in the program created in section 6.3.

7.4.1 Planning a Program

The POUs created in section 6.3 are described below.

POU name	Function
RUN_STOP	If the START button is pressed, the equipment begins to operate. If the STOP button is pressed, or an error is detected, the equipment stops operating.
GLUING	There are two injectors above the conveyer, and the two injectors inject glue in the same way. When in position sensor 1 is ON, the trigger signal for injector 1 is set to ON. When in position sensor 1 is OFF, the trigger signal for injector 1 is reset to OFF. When in position sensor 2 is ON, the trigger signal for injector 2 is set to ON. When in position sensor 2 is OFF, the trigger signal for injector 2 is reset to OFF.
COUNTING	The parts which are conveyed are counted. If the number of parts which are conveyed is 100, the completion flag is set to ON.

Two function blocks will be created according to the characteristics of the POUs.

- The two networks in the **GLUING** are the same, and therefore they can be created in a function block which can be used repeatedly. The function block will be called by the **GLUING**. Owing to the fact the two injectors operate independently, two independent function block instances will be

declared for them.

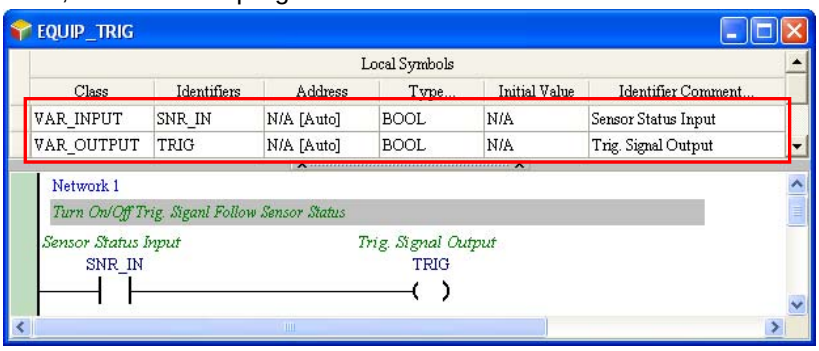
- The program in the **COUNTING** is not related to the other programs except that the operation flag and the completion flag are related to the other programs. In other words, the **COUNTING** is quite independent. In order to make the program in the **COUNTING** more readable, the program is created in a function block.

7.4.2 Creating the Program

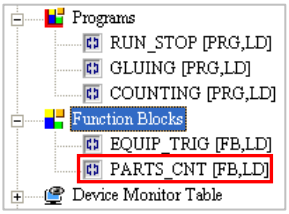
Users can directly modify the program created in section 6.3, or write the program shown in section 6.3 again. If the users want to write the program again, they have to create a new project, and refer to chapter 4~chapter 6 for more information.

Please refer to section 5.4.1, and create **EQUIP_TRIG** in the **Function Blocks** section in the project management area.

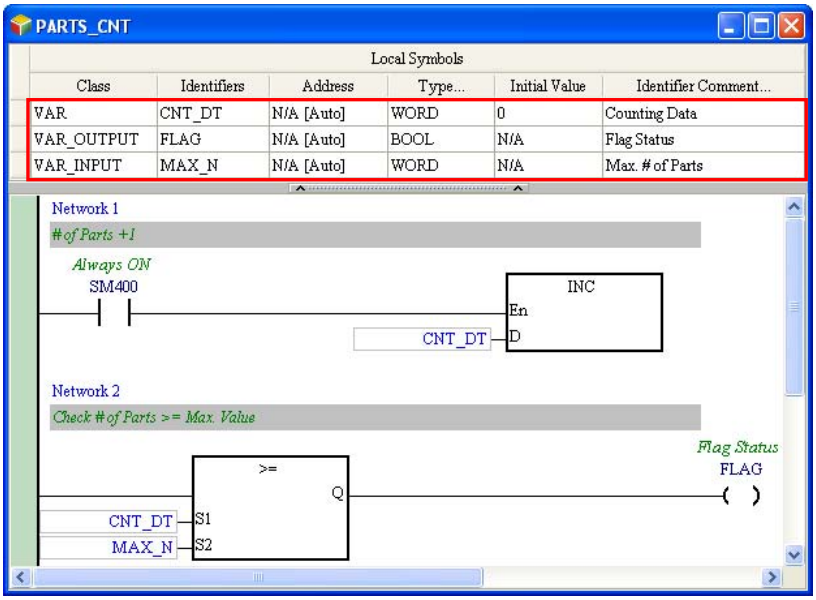
The program in **EQUIP_TRIG** functions to inject the glue. Please refer to section 6.2.2, and create the two symbols shown below in the local symbol table in the **EQUIP_TRIG** window. Please refer to chapter 4~chapter 6, and write the program shown below.



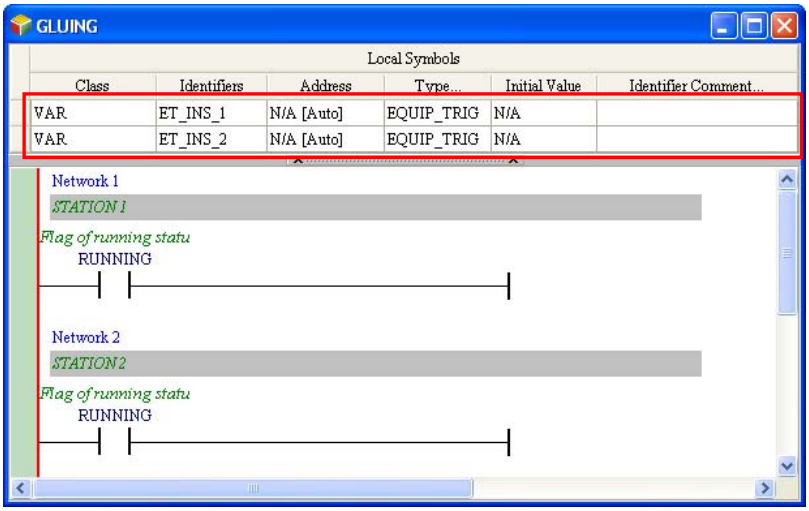
Create **PARTS_CNT** in the **Function Blocks** section in the project management area.



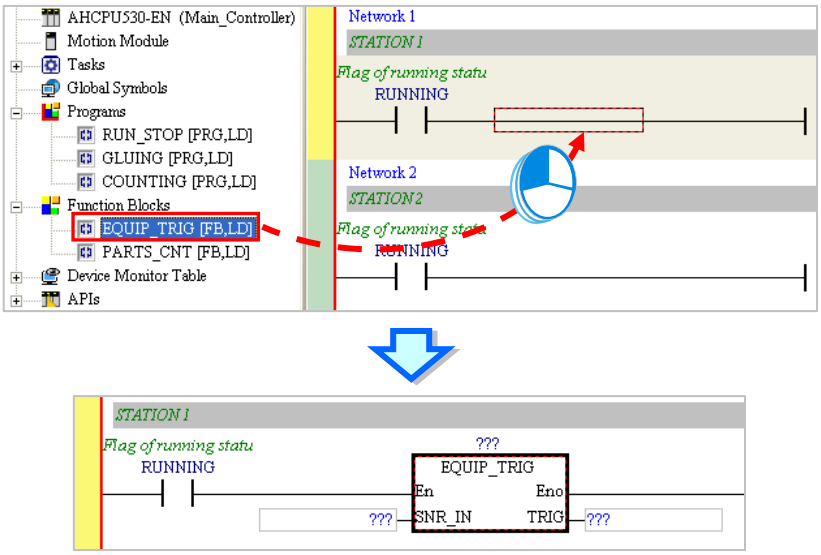
The program in **PARTS_CNT** functions to count the parts which are conveyed. Please refer to section 6.2.2, and create the three symbols shown below in the local symbol table in the **PARTS_CNT** window. The initial value of CNT_DT is 0. Please refer to chapter 4~chapter 6, and write the program shown below.



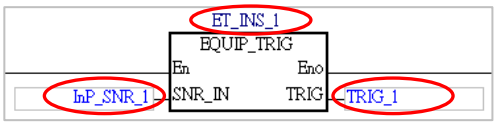
The function block **EQUIP_TRIG** is called by the POU **GLUING**. Please refer to section 6.2.2, and declare the two function block instances shown below in the local symbol table in the **GLUING** window. Write the program shown below.



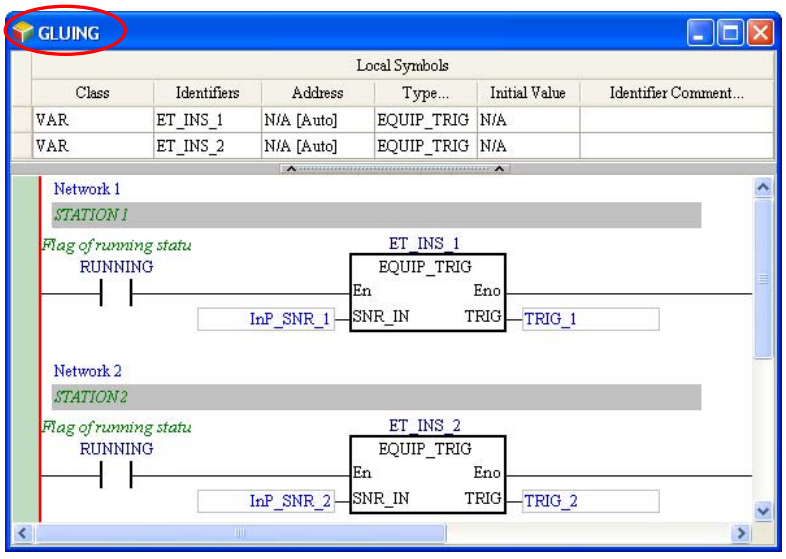
A function block is inserted into a program in a way in which an applied instruction is inserted into a program. Drag the function block **EQUIP_TRIG** in the project management area to the position where it will be inserted. If there is a function block instance under **EQUIP_TRIG** in the porject management area, users have to drag the function block definition rather than the function block instance.



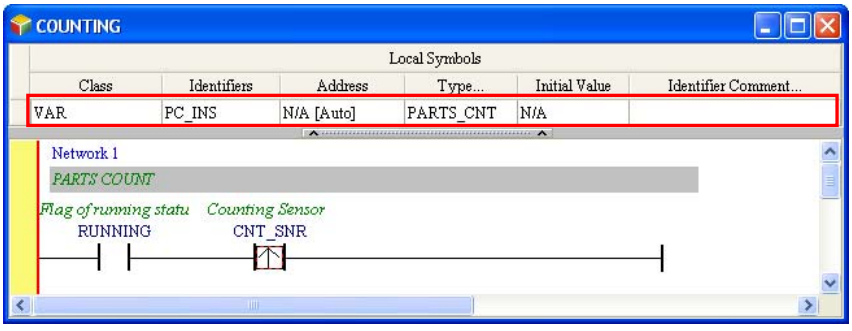
Specify a function block instance and two operands after the function block is inserted.



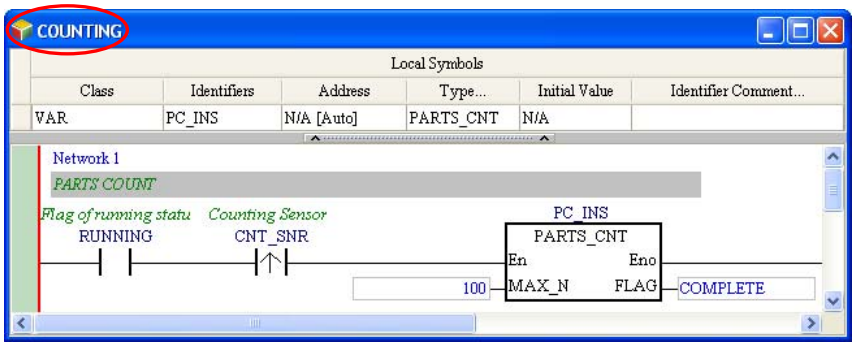
Write the program shown below.



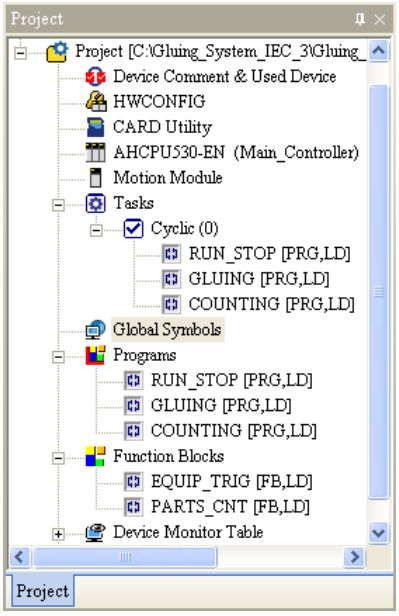
The function block **PARTS_CNT** is called by the POU **COUNTING**. Please declare the function block instance shown below in the local symbol table in the **COUNTING** window. The local symbol **CNT_DT** does not have any effect, and therefore it is deleted. Write the program shown below.

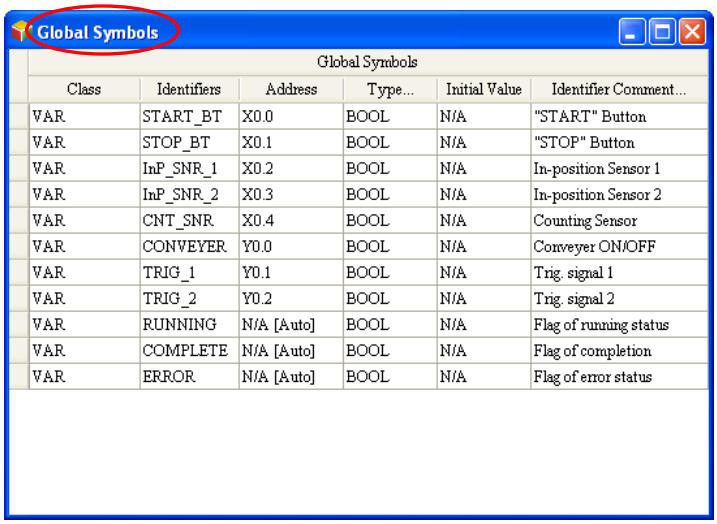


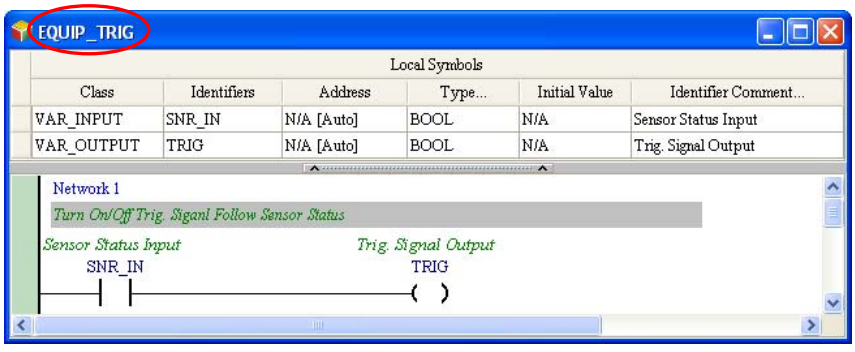
Please refer to the description above, and insert the function block **PARTS_CNT**. Write the program shown below.

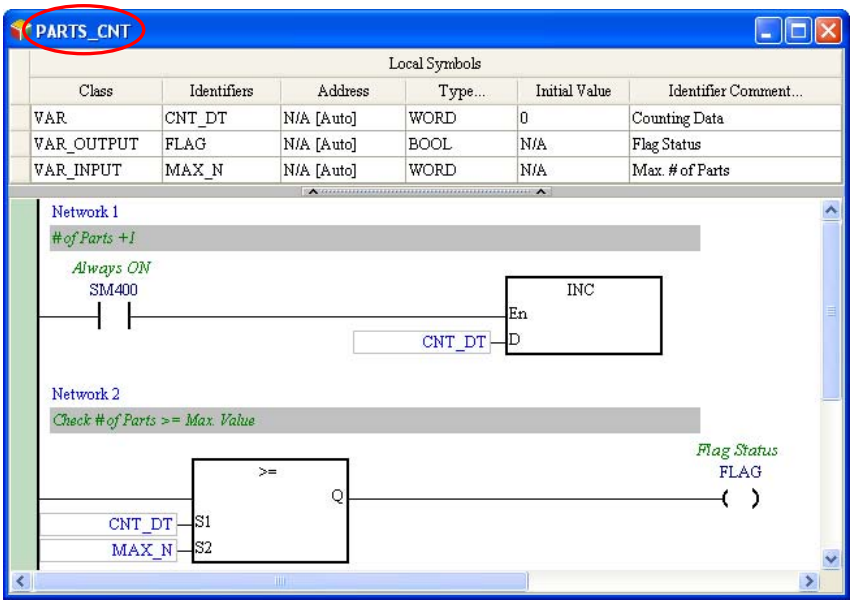


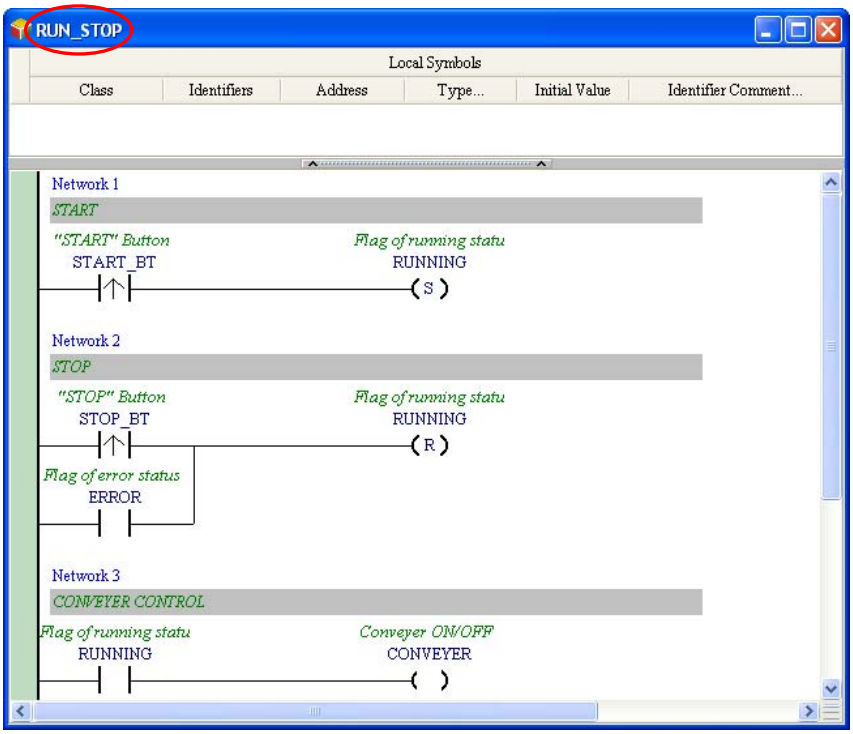
The editing of the program in this section has been completed so far. Users have to make sure that the program they write is the same as the program shown below.

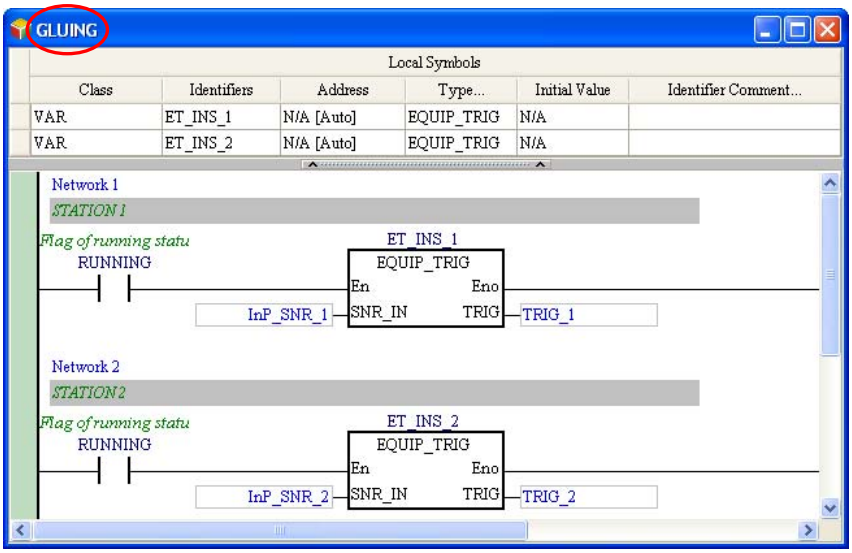


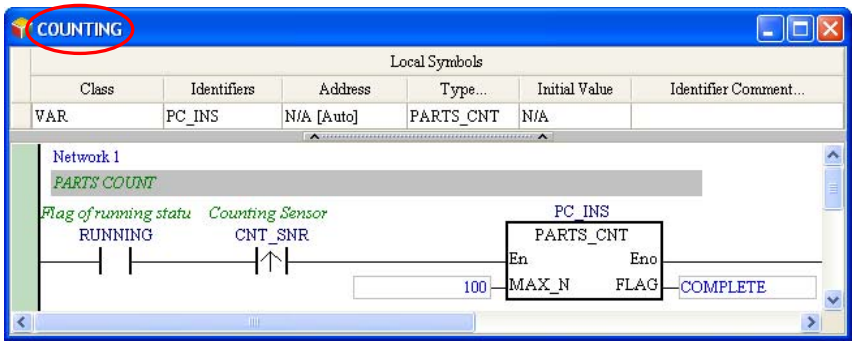




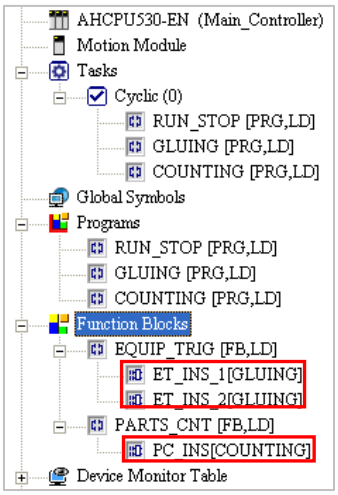

PARTS_CNT
Local Symbols
Class
Identifiers
Address
Type...
Initial Value
Identifier Comment...
VAR
CNT_DT
N/A [Auto]
WORD
0
Counting Data
VAR_OUTPUT
FLAG
N/A [Auto]
BOOL
N/A
Flag Status
VAR_INPUT
MAX_N
N/A [Auto]
WORD
N/A
Max. # of Parts
Network 1
# of Parts +1
Always ON
SM400
INC
En
CNT_DT
D
Network 2
Check # of Parts >= Max. Value
Flag Status
FLAG
>=
Q
CNT_DT
S1
MAX_N
S2


RUN_STOP
Local Symbols
Class
Identifiers
Address
Type...
Initial Value
Identifier Comment...
Network 1
START
"START" Button
START_BT
Flag of running statu
RUNNING
S
Network 2
STOP
"STOP" Button
STOP_BT
Flag of running statu
RUNNING
R
Flag of error status
ERROR
Network 3
CONVEYER CONTROL
Flag of running statu
RUNNING
Conveyer ON/OFF
CONVEYER



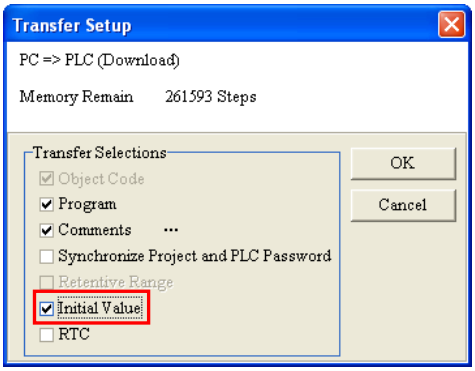


After the users make sure that the program they write is the same as program shown above, they can try to compile the program. After the program is compiled, the function block instances declared will appear in the project management area.

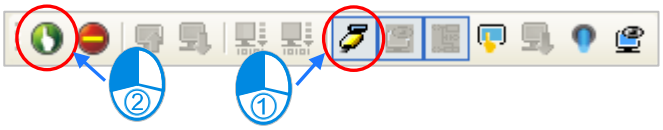


Please make sure that the hardware configuration, the parameter setting, and the task management are completed. If the hardware configuration, the parameter setting, and the task management are not completed, the users have to refer to chapter 4~chapter 6, and complete the hardware configuration, the parameter setting, and the task management. After the hardware configuration, the parameter setting, and the task management are completed, the users can compile the program again, and download the parameters and the program to the CPU module. Before the program is downloaded to the CPU module, the users have to select the **Initial Value** checkbox in the **Transfer Setup** window. Once the **Initial Value** checkbox in the **Transfer Setup** window is selected, the initial

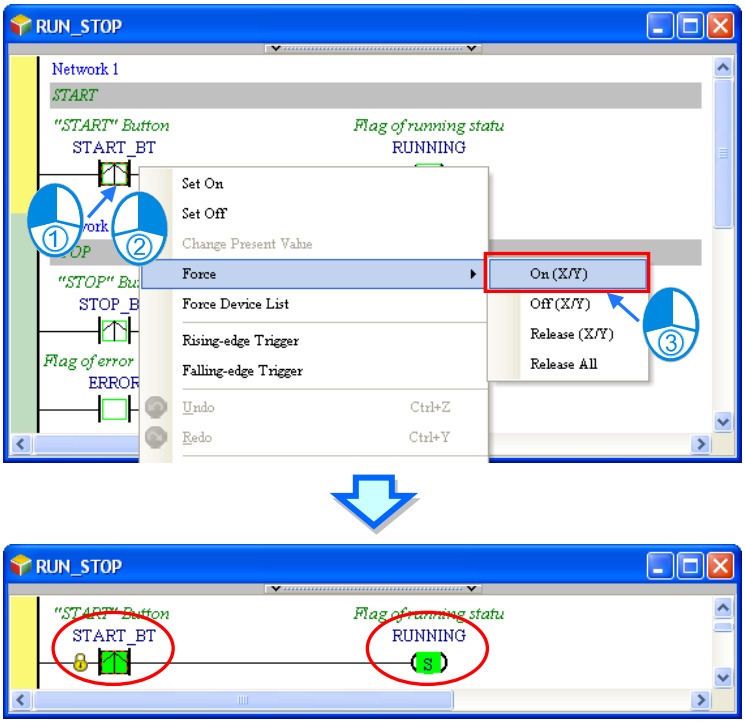
value of CNT_DT in **PARTS_CNT** will be written into the CPU module.



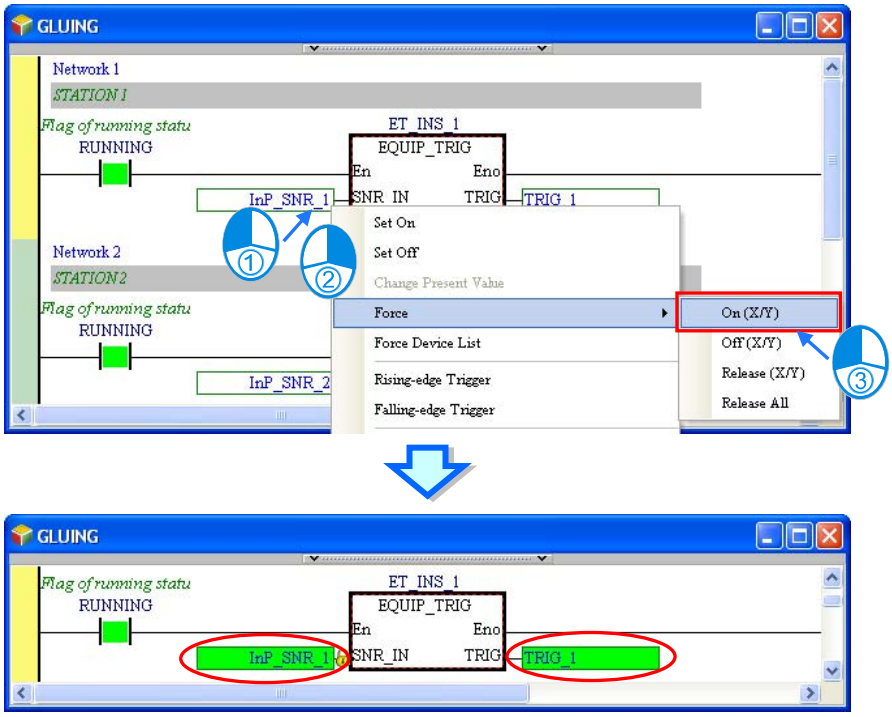
After the program is downloaded, the users can click and on the toolbar. Please follow the steps below, and test the characteristics of the function blocks.



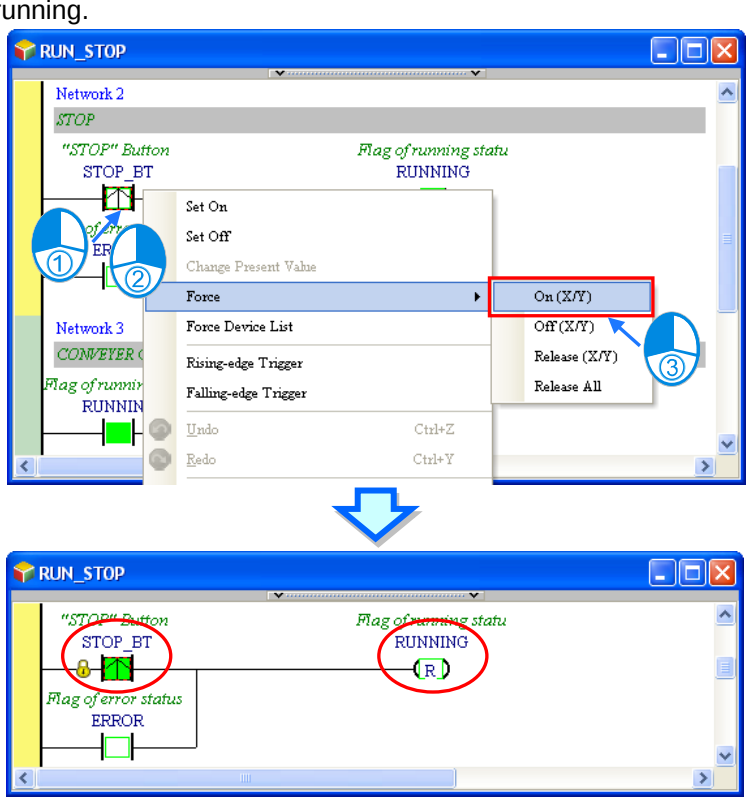
Open the window for **RUN_STOP**, and force **START_BT** ON. **RUNNING** is ON. The equipment begins to run.



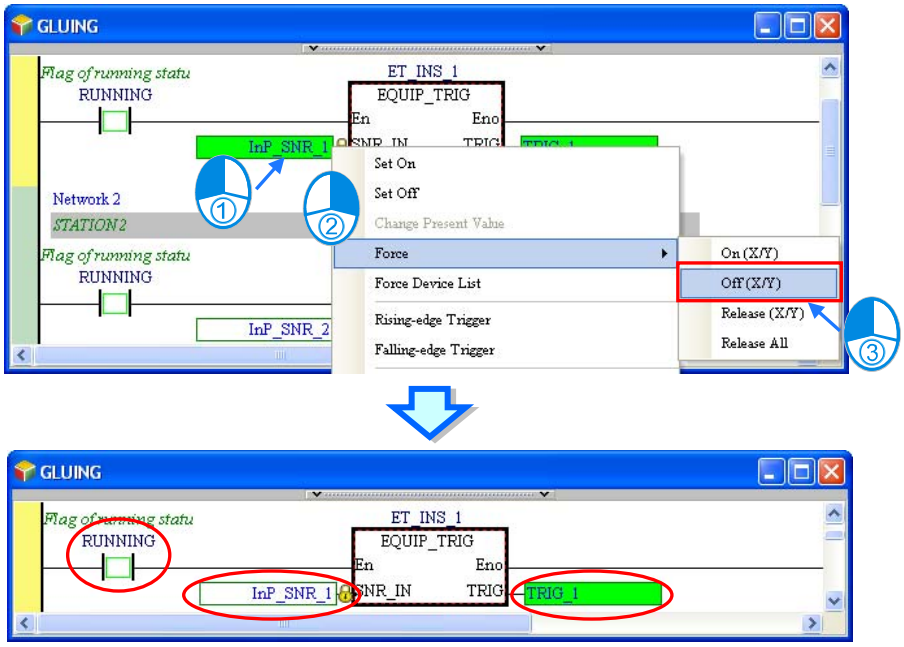
Open the window for **GLUING**, and force **InP_SNR_1** ON. **TRIG_1** is ON. The sensor senses a part, and a trigger signal is sent.



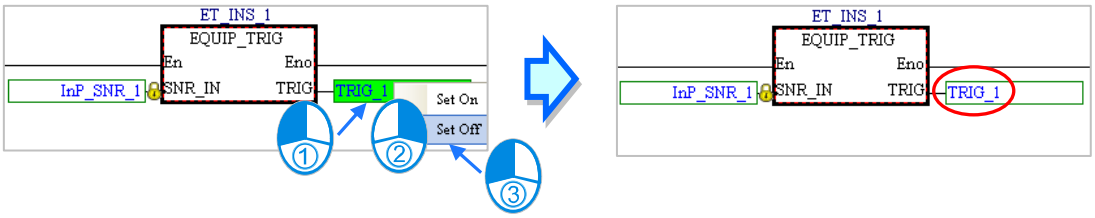
Open the window for **RUN_STOP**, and force **STOP_BT** ON. **RUNNING** is reset to OFF. The equipment stops running.



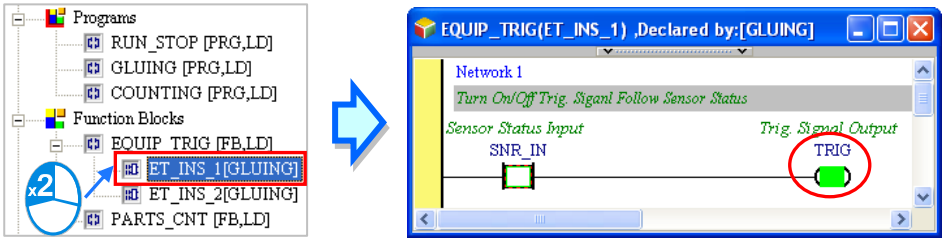
Open the window for **GLUING**, and force **InP_SNR_1** OFF. **TRIG_1** is still ON. The situation is correct. **RUNNING** is a flag connected to the En pin of the function block. **RUNNING** is OFF, and therefore the function block is not executed. The value of **TRIG**, a pin of the function block, is not sent to the symbol **TRIG_1**, and therefore the state of **TRIG_1** remains unchanged.



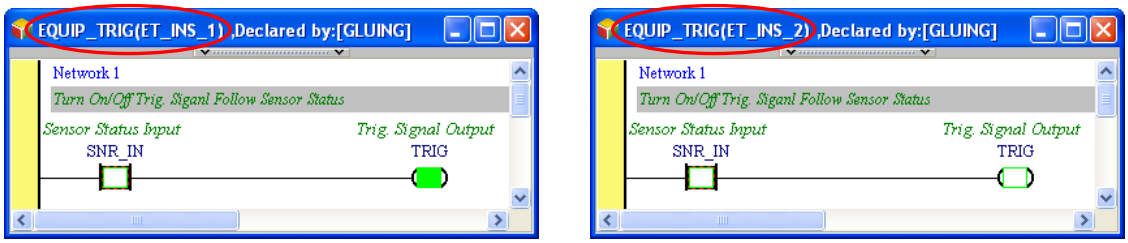
Set **TRIG_1** to OFF.



Open the window for **ET_INS_1**. **TRIG** in the window remains ON. This indicates that the states of the symbols in a function block will be retained after the function block is executed.



Open the window for **ET_INS_2**. **TRIG** in the window is OFF. The state of **TRIG** in the window for **ET_INS_2** is different the state of **TRIG** in the window for **ET_INS_1**. This indicates that the function block instances operate independently.



After the users follow the steps above, they can draw the following conclusions.

1. If the flag connected to the En pin of a function block is OFF, the program in the function block will not be executed.
2. The states of the symbols in a function block will be retained after the function block is executed.
3. The function block instances in a project operate independently.

Obviously, the execution of the program in this section is different from the execution of the program in section 6.3. In the program in section 6.3, **TRIG_1** is OFF when the flag **RUNNING** is OFF. In the program in this section, **TRIG_1** is ON when the flag **RUNNING** is OFF. This is an expectable and acceptable situation. In the plan for the equipment, **TRIG_1** is set to OFF so that it can be set to ON next time. As a result, the function of the equipment will not be affected even if the state of **TRIG_1** remains unchanged when the equipment stops running. Please refer to section 4.1 for more information about the plan for the equipment.

MEMO

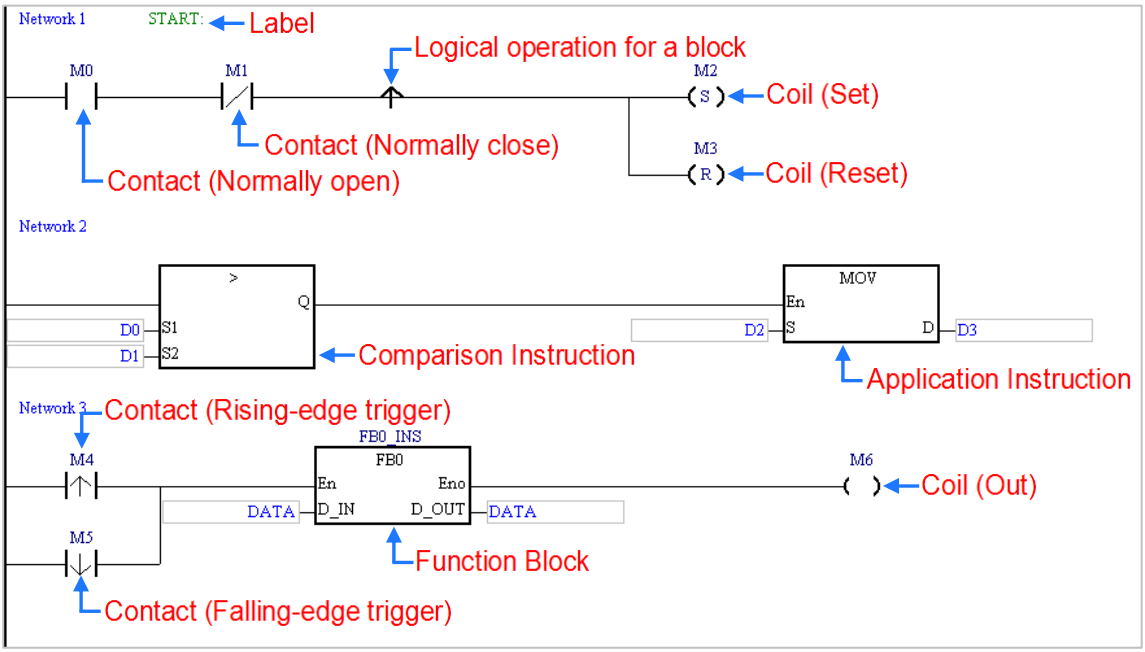
Chapter 8 Ladder Diagram

Table of Contents

8.1 Introduction of a Ladder Diagram 8-2
8.1.1 Editing Environment 8-2
8.1.2 Networks in a Ladder Diagram 8-3
8.1.3 Selecting Objects 8-4
8.2 Creating a Ladder Diagram in ISPSoft 8-5
8.2.1 Creating a Contact and a Coil 8-5
8.2.1.1 Inserting a Contact and Changing a Contact Type 8-6
8.2.1.2 Inserting a Coil and Changing a Coil Type 8-7
8.2.2 Using Devices, Symbols, and Constants in a Ladder Diagram 8-8
8.2.3 Typing Instructions 8-9
8.2.4 Inserting Applied Instructions and Function Blocks 8-9
8.2.5 Creating a Comparison Contact 8-12
8.2.6 Inserting a Block Logic Instruction 8-14
8.2.7 Creating Multiple Outputs 8-15
8.2.8 Putting a Label 8-16
8.2.9 Editing a Comment 8-17
8.2.10 Symbol Mode and Address Mode 8-19
8.2.11 Bookmark 8-19
8.2.12 Activating/Inactivating a Network 8-20

8.1 Introduction of a Ladder Diagram

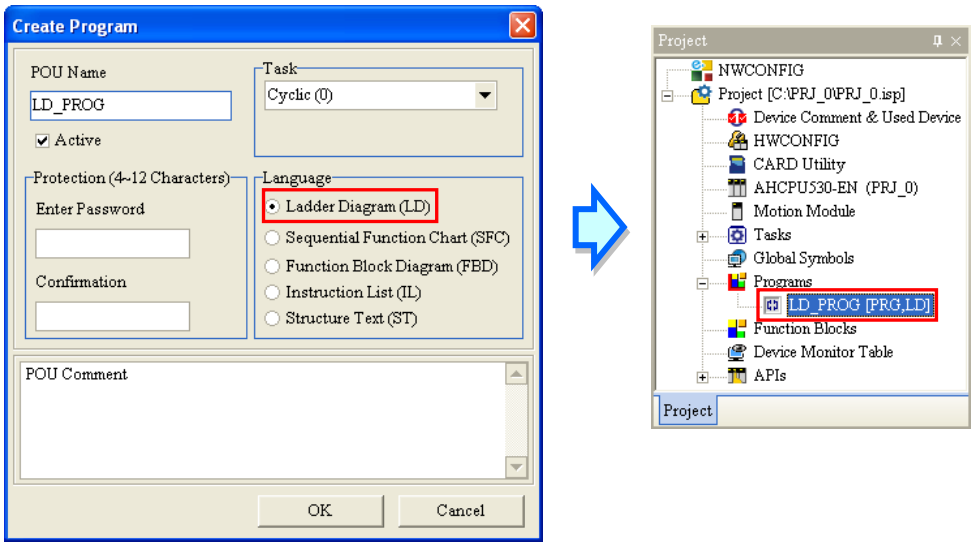
A ladder diagram is one of the programming languages defined by IEC 61131-3, and is widely used to create a PLC program. A ladder diagram is a programming language that represents a program by a graphical diagram based on the circuit diagrams of relay logic hardware. A ladder diagram in ISPSoft is shown below.



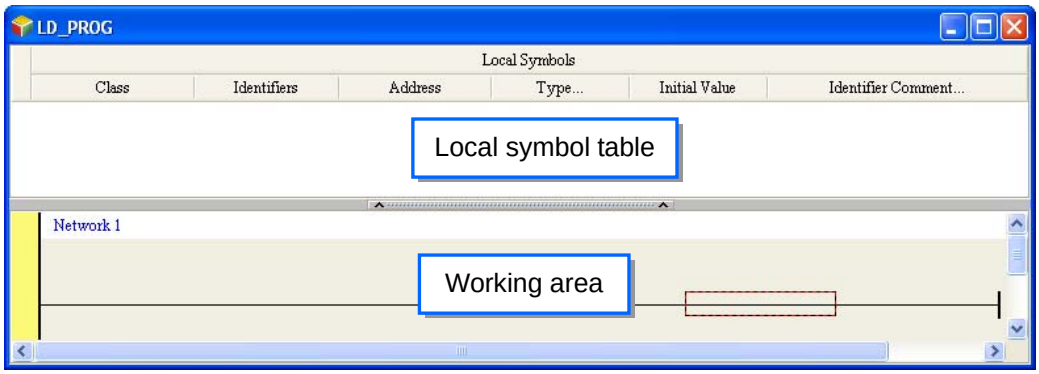
***. Please refer to DVP-PLC Application Manual–Programming for more information about ladder diagrams.**

8.1.1 Editing Environment

Select the **Ladder Diagram (LD)** option button in the **Language** section in the **Create Program** window. Please refer to section 5.4.1 for more information.



The environment in which a ladder diagram can be edited is shown below. The table at the upper part of the window is a local symbol table, and the area at the lower part of the window is a working area.



After a program editing window in which a ladder diagram can be created is opened, the corresponding toolbar will appear in the ISPSoft window. The functions are described below.

Icon	Keyboard shortcut	Function
	Shift+Ctrl+C	Displaying/Hiding the comments on the networks
	None	Displaying/Hiding the commands on the devices
	Shift+Ctrl+A	Activating/Inactivating the network selected
	Shift+Ctrl+B	Adding a bookmark to the network selected or deleting the bookmark from the network selected
	Shift+Ctrl+P	Going to the previous bookmarked position
	Shift+Ctrl+N	Going to the next bookmarked position
	Ctrl+I	Inserting a network above the network selected
	Shift+Ctrl+I	Inserting a network under the network selected
	ESC	Selection
	Typing an instruction	Inserting a contact
	Typing an instruction	Inserting a coil
	Typing an instruction	Inserting a comparison contact
&	Typing an instruction	Selecting a type of comparison contact
	Typing an instruction	Inserting a block logic instruction (NP/PN/INV/FB_NP/FB_PN)
	Typing an instruction	Selecting a type of block logic instruction
	Shift+Ctrl+U	Inserting an instruction or a function block

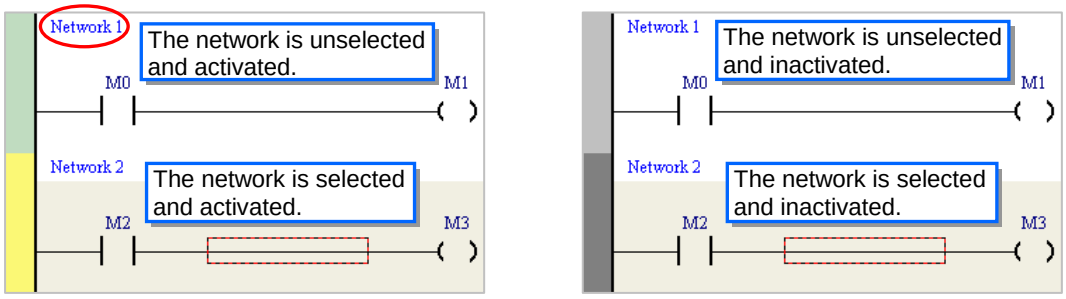
***. Please refer to section 8.2.3 for more information about typing an instruction.**

8.1.2 Networks in a Ladder Diagram

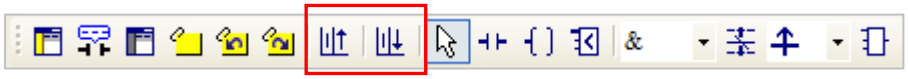
A ladder diagram consists of networks. Every network is an independent program. Besides, there is

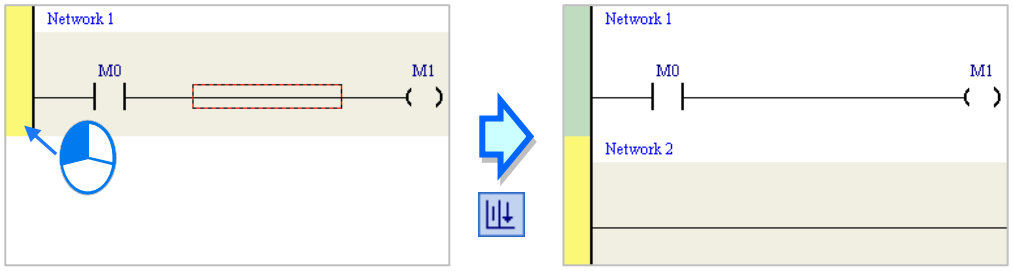
no limit on the number of objects which can be inserted in a network in ISPSoft, and therefore there is no mark which is used to connect two networks.

Network numbers are marked on the networks in a ladder diagram. The color at the left side of a network indicates the state of the network. A network can be activated or inactivated, and a network can be selected or unselected. Users can select colors which will be shown at the left sides of the networks in a ladder diagram in the **Options** window in ISPSoft. Please refer to section 2.3.1 for more information.

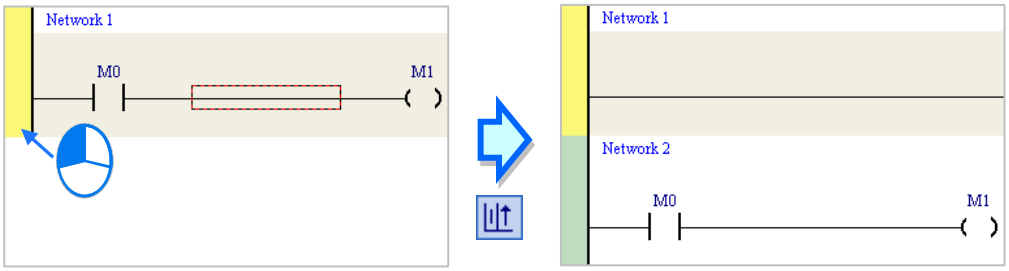


After a new window in which a ladder diagram will be created is opened, the system will insert a blank network in the window. If users want to add a network, they can select a network, and click on the toolbar. The network added will be under the network selected.



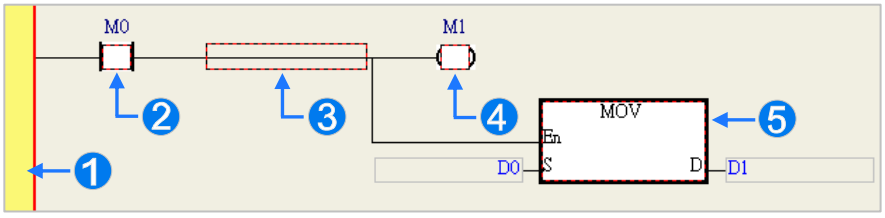


If users click on the toolbar, a blank network will be put above the network selected.



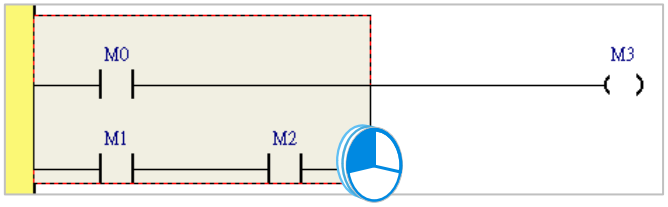
8.1.3 Selecting Objects

Before an object in a network in a ladder diagram is selected, users have to press Esc on the keyboard, or click on the toolbar. After the cursor appears as a small arrow, the users can click the object in the network. The basic selection is shown below.

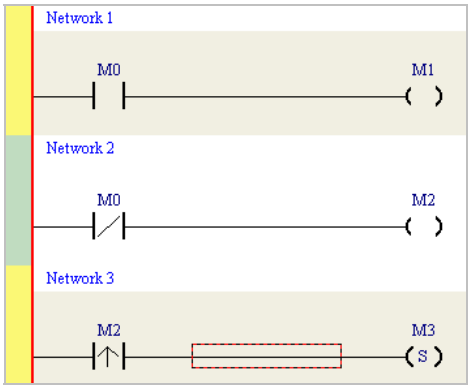


1. Selecting the network
2. Selecting the input contact
3. Selecting the network
4. Selecting the output coil
5. Selecting the block

If users want to select a group of devices, they can click a device, and drag it to draw a frame round the group of devices. The users can also select the group of devices by clicking the first device, pressing Ctrl+B on the keyboard, clicking the last device, and pressing Ctrl+B on the keyboard. Users must draw a frame round devices which are in the same network, and the devices must be adjacent to one another. Besides, input devices and output devices can not be in the same frame.



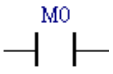
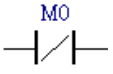
If users want to select several networks, they can hold down Ctrl on the keyboard while they click the networks. The users can also select a range of networks by pressing Shift on the keyboard, clicking the first network within the range, and the last network within the range.



8.2 Creating a Ladder Diagram in ISPSoft

8.2.1 Creating a Contact and a Coil

The representations of contacts and the representations of coils in ISPSoft are listed below.

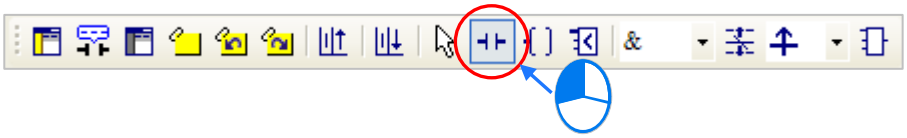
Contact	Normally-open contact	
	Normally-closed contact	
	Rising edge-triggered contact	
	Falling edge-triggered contact	
Coil	OUT	
	SET	

Coil	**RESET**	M1 (R)

8.2.1.1 Inserting a Contact and Changing a Contact Type

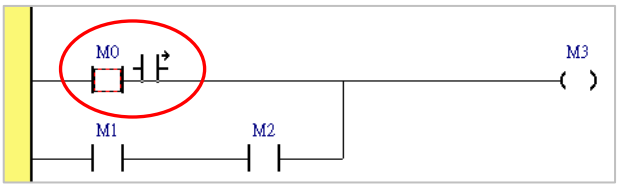
(1) Select a line, a contact, or a group of devices. Before users connect a contact to a group of devices in parallel or in series, they have to draw a frame round the group of devices.

After users make sure of the position in which a contact will be inserted, they can click on the toolbar.

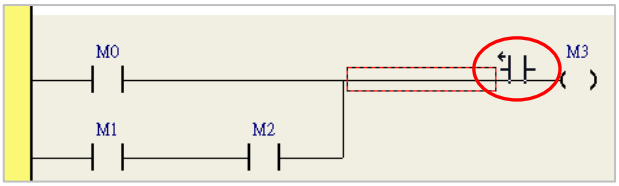


(2) The mouse cursor appears as a contact when the mouse cursor is near the position in which a contact will be inserted.

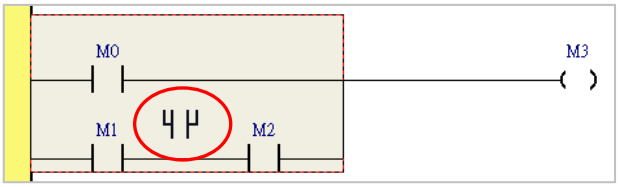
- If the mouse cursor is near a contact, a contact will be connected to the contact in parallel or in series.



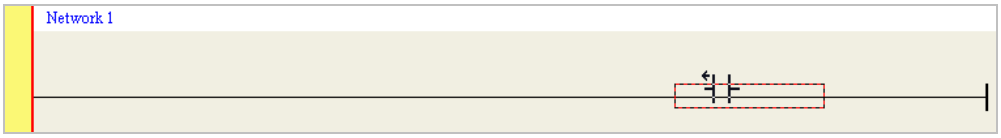
- If the mouse cursor is near a line, a contact will be inserted in the line.



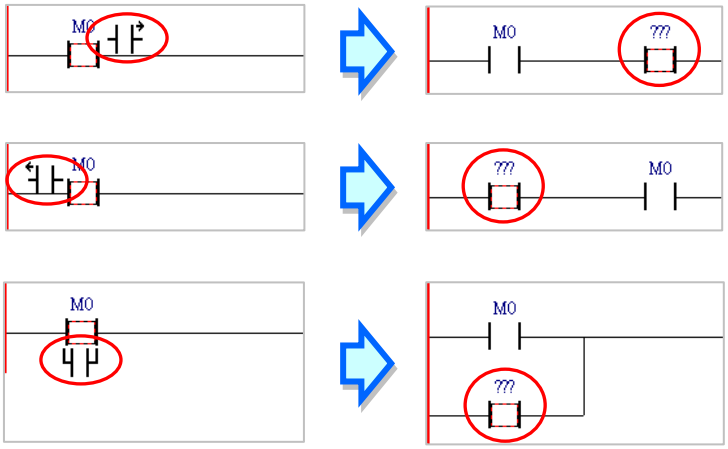
- If the mouse cursor is near a group of devices, a contact will be connected to the group of devices in parallel or in series.



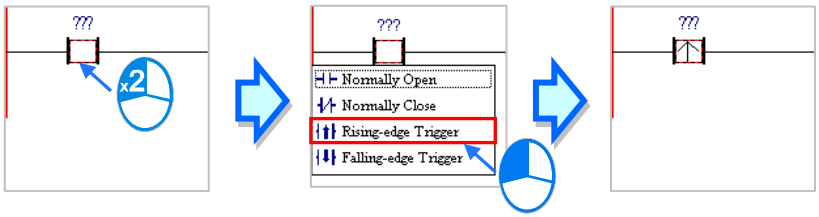
- If users want to insert a contact in a new network, they can move the mouse cursor to the red frame in the new network.



(3) The mouse cursor appears as a contact when the mouse cursor is at the right side of a contact, at the left side of a contact, or at the bottom of a contact. Users can decide where to insert a contact. After users click the position in which a contact will be inserted, the system will insert the contact.



(4) After a contact is inserted, users can click on the toolbar or press Esc on the keyboard. After the contact is double-clicked, users can select a contact type on the drop-down list.

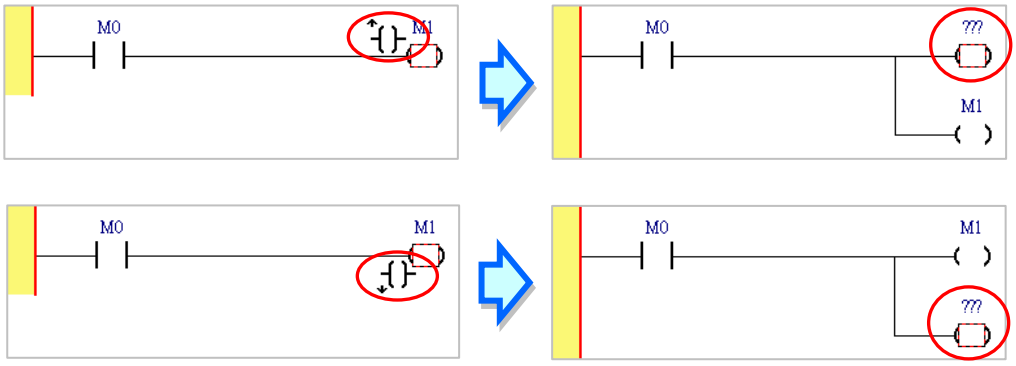


8.2.1.2 Inserting a Coil and Changing a Coil Type

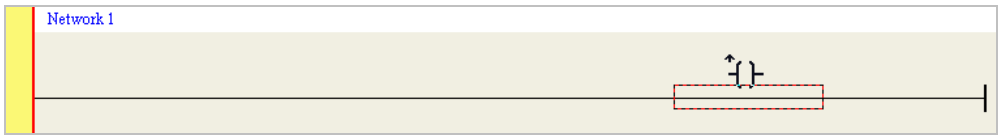
(1) Select a line, a contact, a coil, or an applied instruction. After users make sure of the position in which a coil will be inserted, they can click on the toolbar.

(2) The mouse cursor appears as a coil when the mouse cursor is near the position in which a coil will be inserted. If the mouse cursor is at the top of a coil or at the top of an applied instruction, a coil will be put above the coil or the applied instruction. If the mouse cursor is at the bottom of a coil or at the bottom of an applied instruction, a coil will be put under the coil or the applied instruction. If the mouse cursor is near the line in a network, a coil will put above all the output devices in the network.

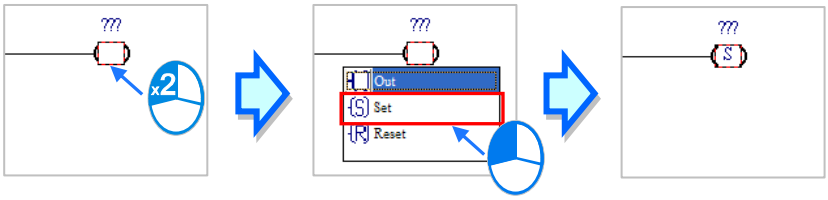
After users click the position in which a coil will be inserted, the system will insert the coil.



If users want to insert a coil in a network where there is no output device, the mouse cursor must be near the line selected.

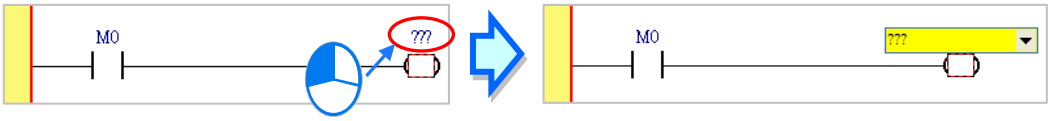


(3) After a coil is inserted, users can click on the toolbar or press Esc on the keyboard. After the coil is double-clicked, users can select a coil type on the drop-down list.

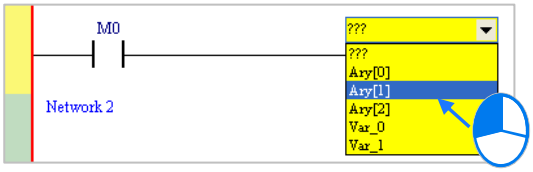


8.2.2 Using Devices, Symbols, and Constants in a Ladder Diagram

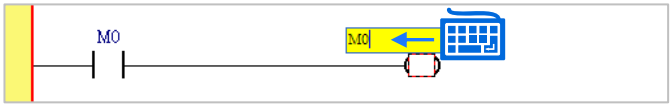
After users click ??? above an object inserted in a network, they can type an operand.



The users can click in the box, and select a symbol on the drop-down list. If a symbol whose data type is ARRAY is declared, the elements in the array will be on the drop-down list. Ary[0], Ary[1], and Ary[2] in the figure below are elements.



The users can type a device name or a symbol in the box. If they type a symbol which is not declared, the **Add Symbol** window will appear after they press Enter on the keyboard. (Please refer to section 2.3.1 for more information about declaring a symbol.)



After users click a network and press Enter on the keyboard, they can edit a box. The users can edit the next box in the network after they press Enter on the keyboard. Besides, the next network will be selected after the users press Tab on the keyboard. The users can edit a box with the keyboard. If the users have declared symbols, they can press Page Down on the keyboard when they edit a box. The symbols on the drop-down list are symbols which can be assigned to the operands. The users can select a symbol by the mouse or the up/down key on the keyboard. After the editing is complete, the users can press Enter on the keyboard to jump to the next box. If the users want to end the editing, they can press Esc on the keyboard.

Additional remark

(a) If users want to use constants in a ladder diagram in ISPSoft, the constants must be represented in the following ways.

- Decimal value: 23456 (A value which is not preceded by any mark will be regarded as a decimal value.)
- Hexadecimal value: 16#5BA0 (A value is preceded by 16#.)
- Octal value: 8#55640 (A value is preceded by 8#.)
- Binary value: 2#101101110100000 (A value is preceded by 2#.)
- String: “XYZ” (Characters are put in double quotes.)

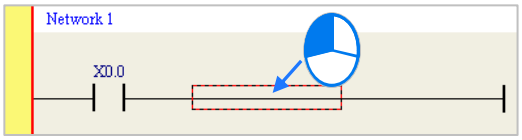
➢ Boolean value:
AH500: SM400 (normally-open contact), or SM401 (normally-closed contact) is used.
DVP: M1000 (normally-open contact), or M1001 (normally-closed contact) is used.

(b) If users want to use a symbol whose data type is ARRAY, the expression format is **Identifier[Index]**. Generally speaking, an index is a decimal constant. The indexes in a ladder diagram can not be symbols. The minimum index value for a symbol whose data type is ARRAY must be 0, and the maximum index value for a symbol whose data type is ARRAY can not be larger than the number of elements in the array.

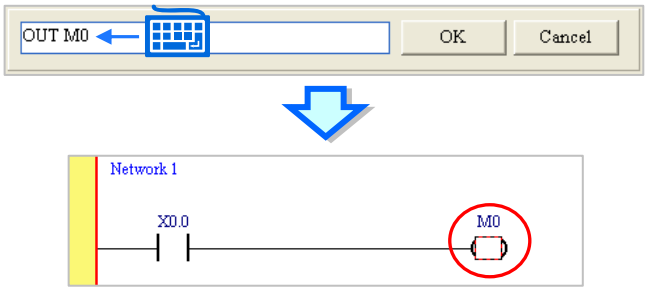
8.2.3 Typing Instructions

Users can add devices, applied instructions and function blocks to a ladder diagram in ISPSoft by typing instructions.

(1) Select a position in which an object will be inserted in a working area.

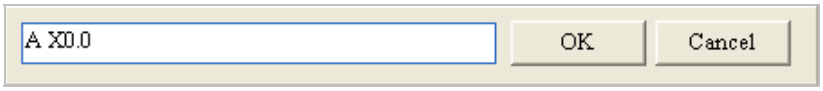


(2) Type an instruction. As soon as the instruction is typed, a box which can be edited appears. After the typing of the instruction is complete, users can press Enter on the keyboard or click **OK** at the right side of the box. (The instruction is case-insensitive.)

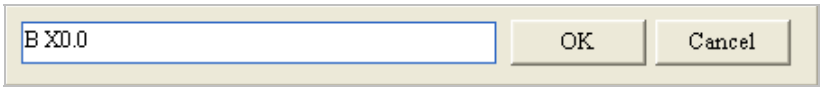


Besides, contacts and coils can be created by typing simple instructions. Please refer to the following figures. (The instructions typed are case-insensitive.)

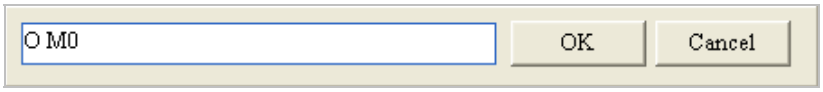
- Inserting a normally-open contact (contact A): “A Device address”



- Inserting a normally-closed contact (contact B): “B Device address”



- Inserting an output coil (OUT): “O Device address”



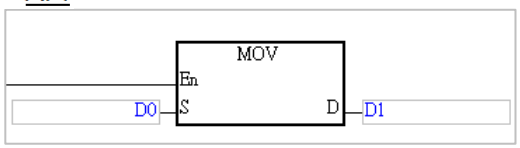
8.2.4 Inserting Applied Instructions and Function Blocks

The applied instructions and function blocks in a ladder diagram in ISPSoft are represented by blocks. The pins of a block representing an applied instruction include the pins corresponding to the operands specified in the applied instruction and an En pin. The pins of a block representing a function block include the pins corresponding to the symbols declared in the function block and an En pin. The En pin of a block representing an applied instruction or a function block is connected to the logic state preceding the En pin. If the logic state connected to the En pin of a block representing an applied instruction or a function block is ON, the applied instruction or the function block will be

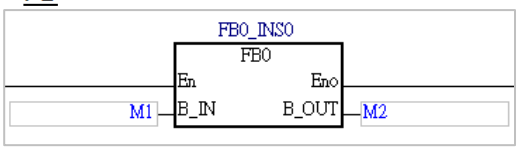
executed.

The pins of a block representing a function block include an Eno pin whereas the pins of a block representing an applied instruction do not include an Eno pin. An applied instruction can only be put at the end of a network in a ladder diagram, and can not be followed by any object. Besides, users can type a function block instance in the box above a function block. Please refer to chapter 7 for more information.

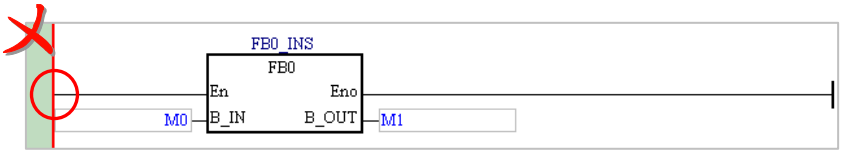
- API



- FB



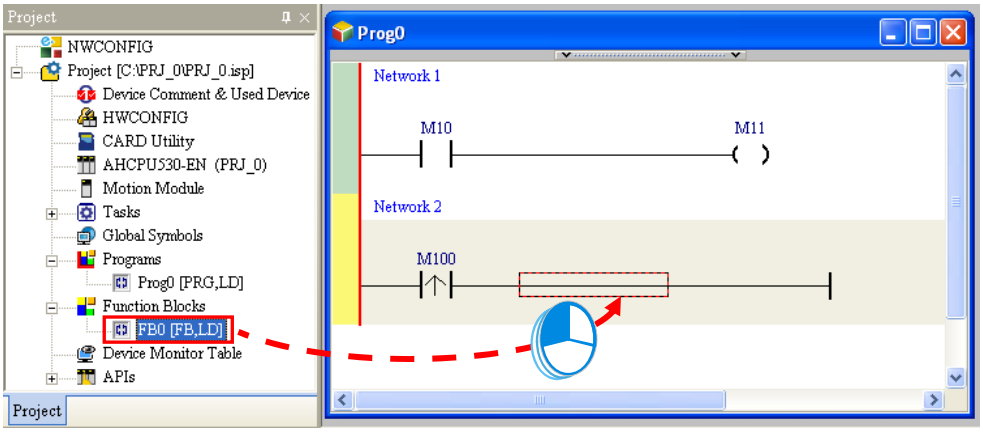
If users want to insert an applied instruction or a function block in a ladder diagram, the En pin of the block representing the applied instruction or the function block must be connected to a device or a group of devices, and must not be connected a mother line directly. The function block in the figure below is used incorrectly.



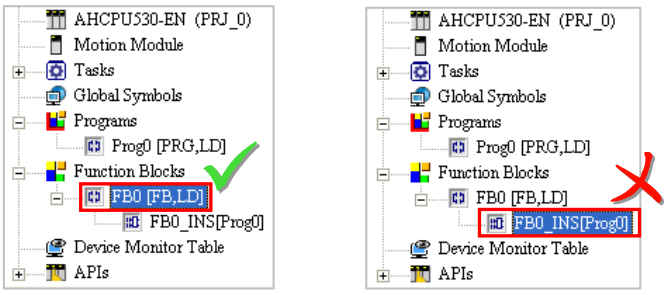
Users can insert an applied instruction or a function block in a ladder diagram in ISPSoft in one of the three ways described below.

- **Method 1**

  Unfold the **APIs** section or the **Function Blocks** section in the project management area, and find the item which will be inserted. Select the item, and drag it to the position in which it will be inserted.

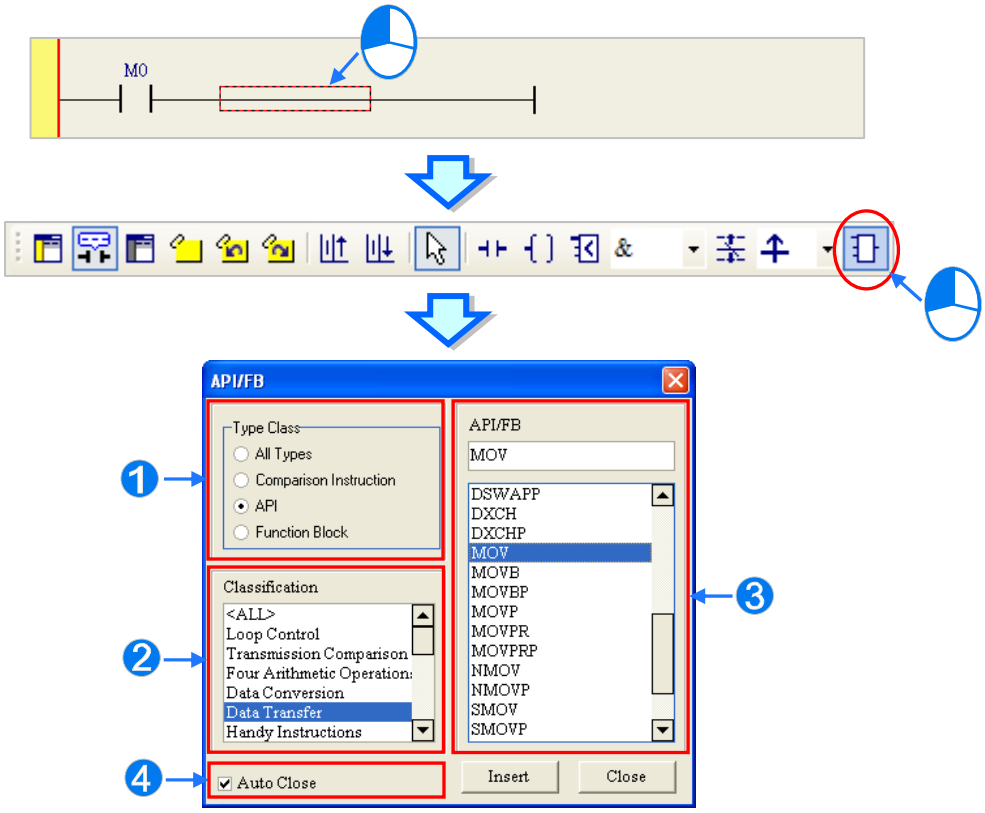


Function block definitions can be dragged, but function block instances can not be dragged.



- **Method 2**

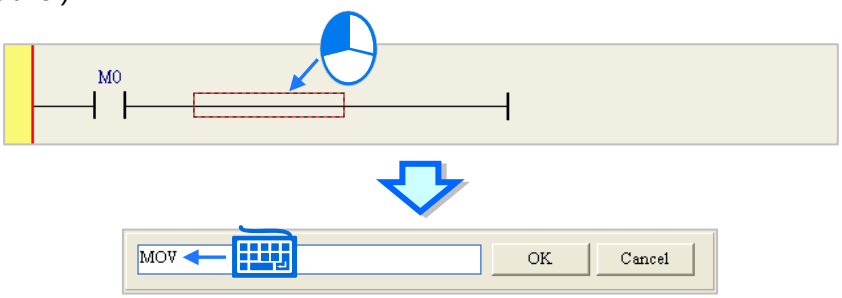
  Click the position in which an applied instruction or a function block will be inserted, click on the toolbar, select the item which will be inserted in the **API/FB** window, and click **Insert**.



1. Users can select the **All Types** option button, the **Comparison Instruction** option button, the **API** option button, or the **Function Block** option button.
2. If users select the **API** option button in the **Type Class** section, they have to select an instruction type in the **Classification** box.
3. Users can select an applied instruction, a comparison contact, or a function block here.
4. If users unselect the **Auto Close** checkbox, they can insert instructions or function blocks continuously.

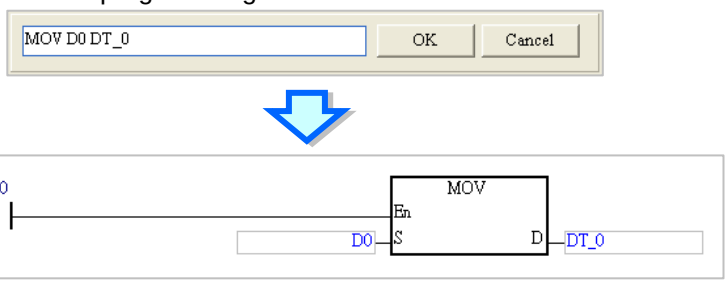
- **Method 3**

  Click the position in which an instruction or a function block will be inserted, and type the applied instruction or the function block definition. As soon as the applied instruction or the function block definition is typed, a box which can be edited appears. After the typing of the applied instruction or the function block definition is complete, users can press Enter on the keyboard or click **OK** at the right side of the box. (The applied instruction and the function block definition are case-insensitive.)

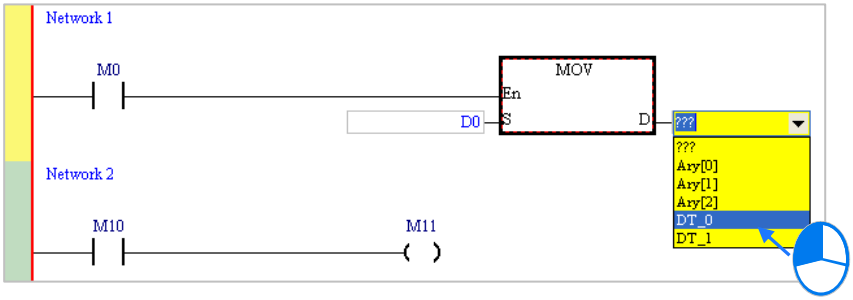


If users want to insert a function block, they have to type the function block definition, and they can not type the operands specifed in the function block. If users want to insert an applied instruction, they have to type the applied instruction, and they can type the operands specified in the applied instruction. The users do not have to type all the operands specified in the applied instruction. The system will assign the operands typed to the corresponding boxes, and ??? will

appear in the boxes where there are no operands. Besides, if the operands typed include a symbol which is not declared, the **Add Symbol** window will appear after the users press Enter on the keyboard. Please refer to programming manuals for more information about instructions.



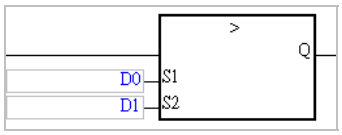
After an applied instruction or a function block is inserted successfully, users have to refer to section 8.2.2, and type operands.



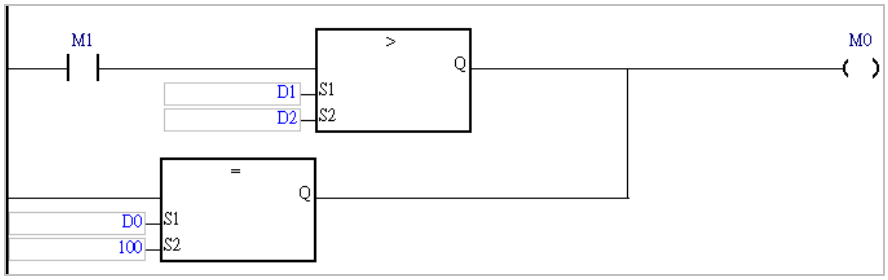
***. If the Auto-leading 'Add Symbol' Dialog checkbox in the Options window is selected, the Add Symbol window will appear automatically after users type a symbol which is not declared. Please refer to section 2.3.1 for more information.**

8.2.5 Creating a Comparison Contact

An example of a comparison contact in ISPSoft is shown below. **>** in the block represents a comparison type. **S1** and **S2** are objects which are compared. The data format which is accepted depends on the comparison type. **Q** can drive a coil or connect to a contact in series.

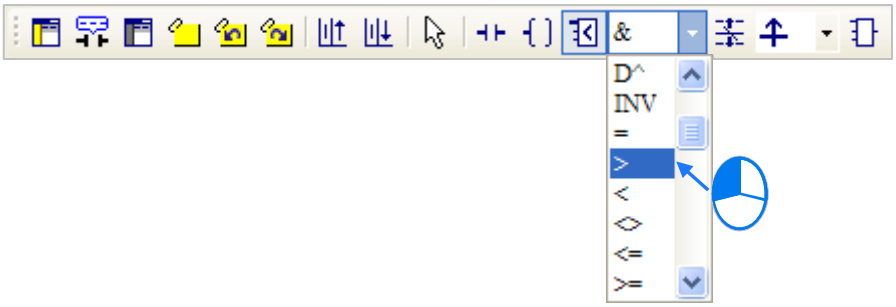


The comparison contacts in a ladder diagram are similar to general contacts in that they can connect to other contacts in parallel or in series.

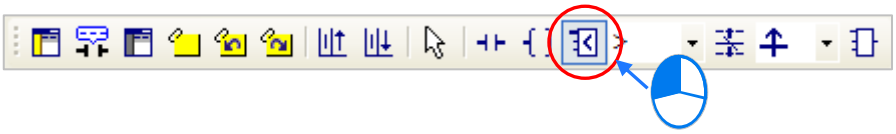


A comparison contact can be inserted not only in one of the three ways described in section 8.2.4, but also by means of the following steps.

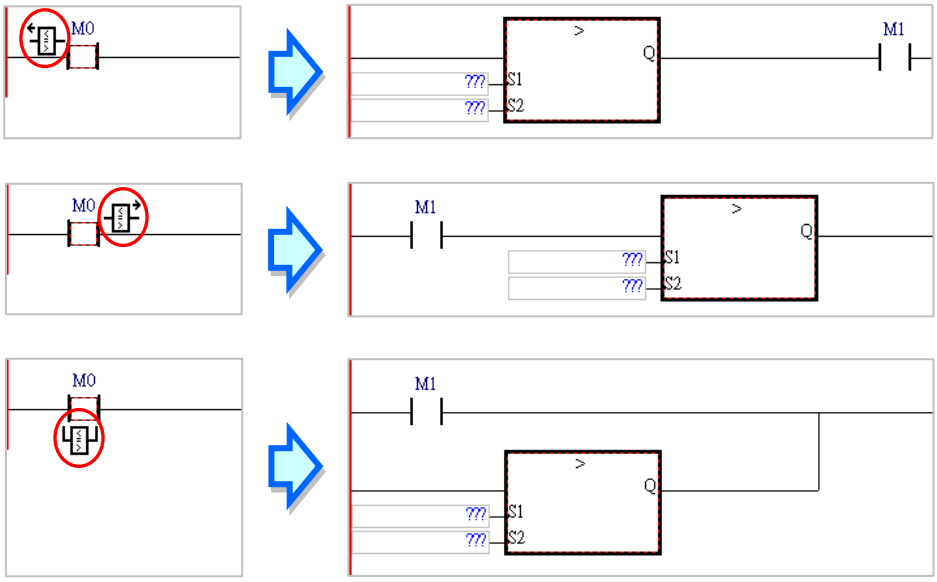
(1) Click on the toolbar, and then select a type.



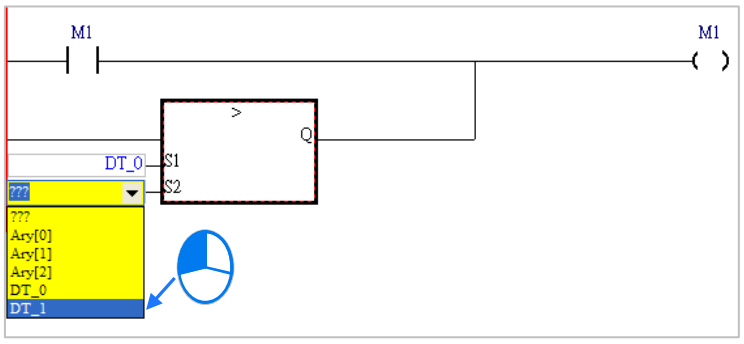
(2) Select a line, a contact, or a group of devices. Before users connect a comparison contact to a group of devices in parallel or in series, they have to draw a frame round the group of devices. After users make sure of the position in which a comparison contact will be inserted, they can click on the toolbar.



(3) The mouse cursor appears as a comparison contact when the mouse cursor is near the position in which a comparison contact will be inserted. Please refer to the figures below. The mouse cursor appears as a contact when the mouse cursor is at the right side of the contact selected, at the left side of the contact selected, or at the bottom of the contact selected. Users can decide where to insert a comparison contact. After users click the position in which a contact will be inserted, the system will insert the contact.

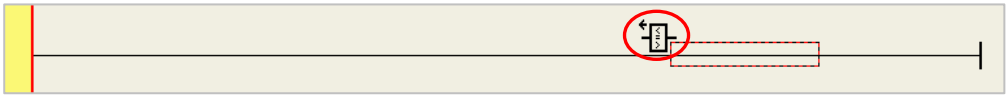


(4) After an comparison contact is inserted successfully, users have to refer to section 8.2.2, and type operands.



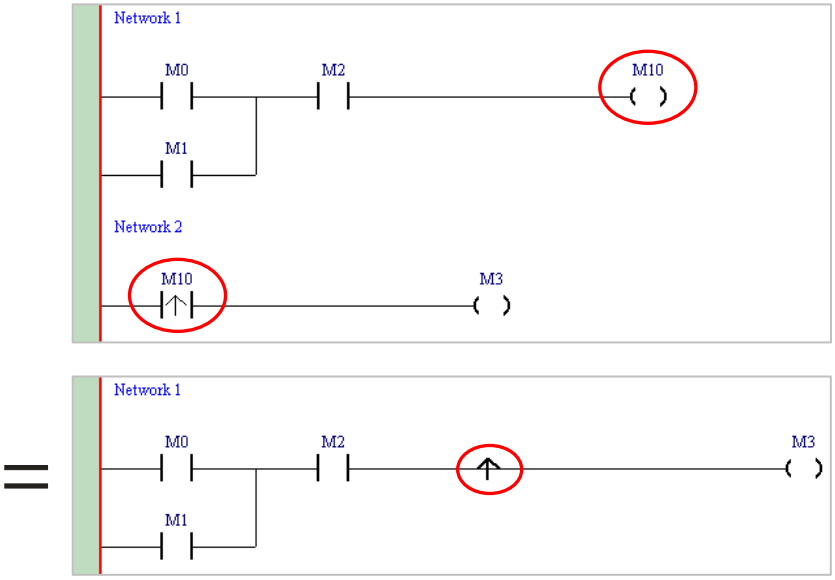
Additional remark

If users want to insert a comparison contact in a new network, the mouse cursor must be near the line selected.



8.2.6 Inserting a Block Logic Instruction

If users want to take the operation result of a block as a condition of a rising edge-triggered circuit, they traditionally have to create two networks in a ladder diagram. However, the users can combine the two networks by means of using a block logic instruction.



The block logic instructions that ISPSoft provides are as follows.

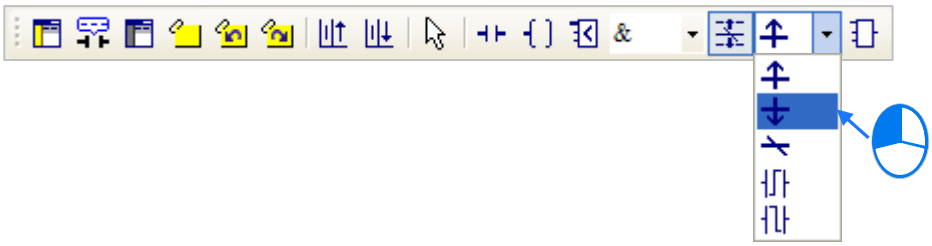
Representation	Instruction	Function
	NP	When the block preceding this block logic instruction is tuned from OFF to ON, the output device is ON.
	PN	When the block preceding this block logic instruction is turned form ON to OFF, the output device is ON.
	INV	The operation result of the block preceding this block logic instruction is inverted, and the inversion result is sent.
	FB_NP	This block logic instruction is similar to NP except that users have to specify a device or a variable in this block logic instruction. (FB_NP can only be used in the function blocks in AH500 series CPU modules.)
	FB_PN	This block logic instruction is similar to PN except that users have to specify a device or a variable in this block logic instruction. (FB_PN can only be used in the function blocks in AH500 series CPU modules.)

*** Please refer to AH500 Programming Manual or section 7.3.2 for more information about FB_NP and FB_PN.**

Users can insert a block logic instruction in one of the two ways described below.

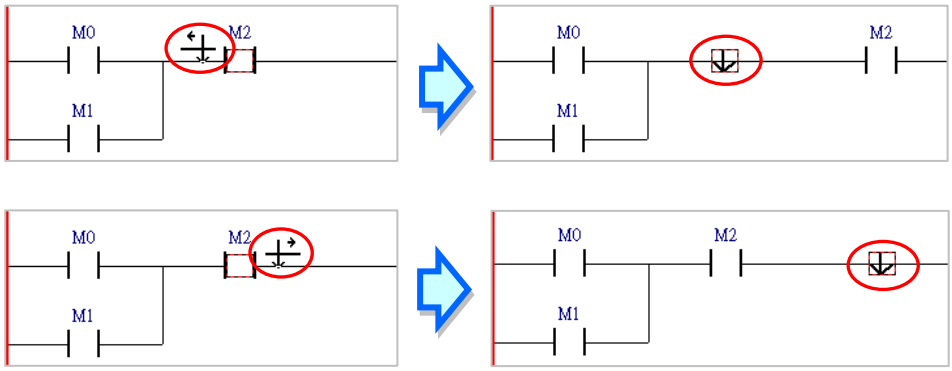
- **Method 1**

  (1) Click on the toolbar, and then select a type.



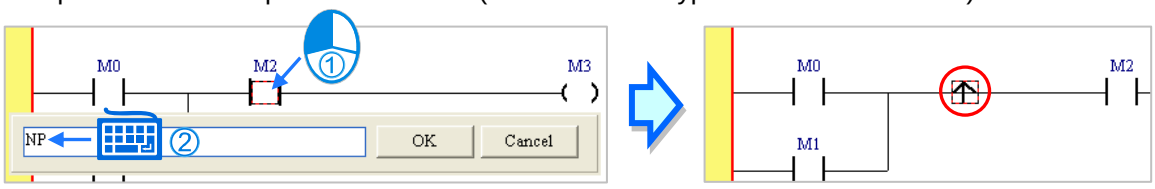
(2) Click on the toolbar.

(3) The mouse cursor appears as the representation of a block logic instruction when the mouse cursor is near the position in which a block logic instruction will be inserted. Please refer to the figures below. The mouse cursor appears as the representation of a block logic instruction when the mouse cursor is at the right side of the contact selected, or at the left side of the contact selected. Users can decide where to insert a block logic instruction. After users click the position in which a block logic instruction will be inserted, the system will insert the block logic instruction.



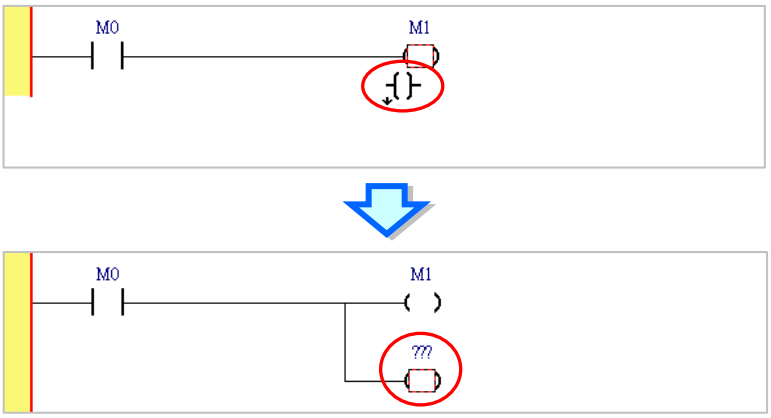
- **Method 2**

  Click the position in which a block logic instruction will be inserted, and type the block logic instruction. As soon as the block logic instruction is typed, a box which can be edited appears. After the typing of the block logic instruction is complete, users can press Enter on the keyboard or click **OK** at the right side of the box. The representation of the block logic instruction typed will be put in front of the position selected. (The instruction typed is case-insensitive.)

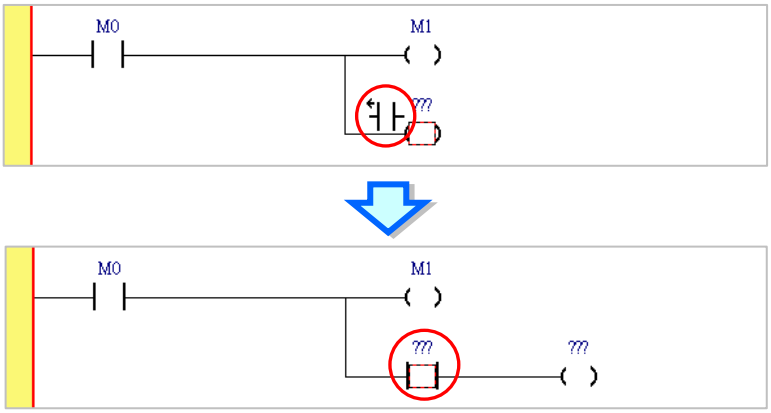


8.2.7 Creating Multiple Outputs

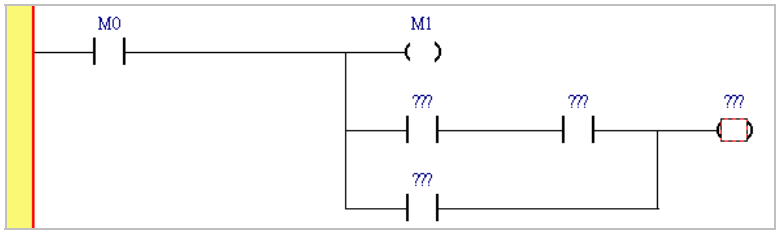
(1) If users want to create multiple outputs, they have to insert a coil or an applied instruction first.



(2) Click or on the toolbar, move the mouse cursor to a position near the output object, and then click the left mouse button.



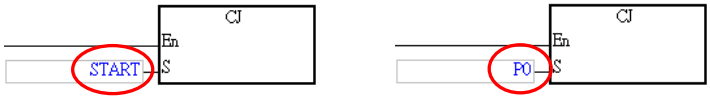
(3) Create multiple outputs according to the description above.



8.2.8 Putting a Label

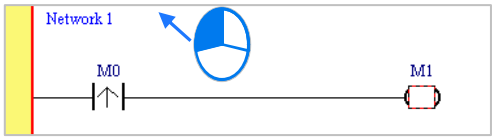
If users want to use a jump instruction, they have to put a label on the network to which the execution of the program jumps. A label is put at the right side of a network number. Besides, if the project created is a project for a DVP series PLC, the label put on a network must be a P device. The P devices in an AH500 series CPU module are automatically assigned by the system, and can not be used by users. If the project created is a project for an AH500 series CPU module, the label symbol users type must be a unique label. The label symbol users type in a project for an AH500 series CPU module does not need to be declared in a symbol table. Please refer to user manuals for more information about P devices.

If users want to specify a label in an instruction, they do not have to type a colon in front of the label.

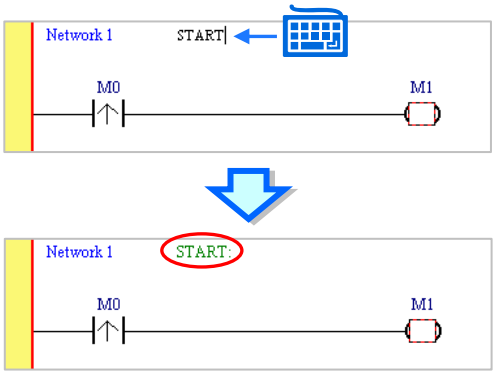


The steps of putting a label are as follows.

(1) Click the right side of the network number at which a label will be put.

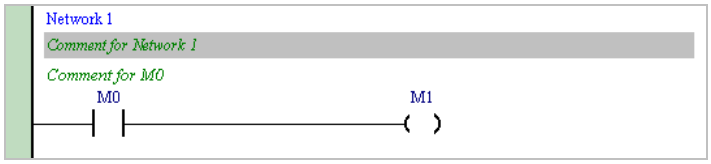


(2) If the project created is a project for a DVP series PLC, users have to type a P device number. If the project created is a project for an AH500 series CPU module, users have to type a label symbol. Press Enter on the keyboard after a P device number or a label symbol is typed. The system will automatically put a colon at the right side of the label.



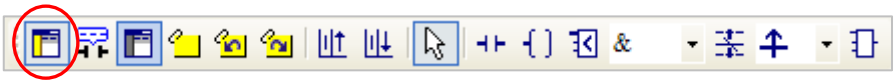
8.2.9 Editing a Comment

In a ladder diagram in ISPSoft, users not only can make comments on the devices, but can make comments on the networks. Users can type comments under the network numbers in a ladder diagram in ISPSoft.

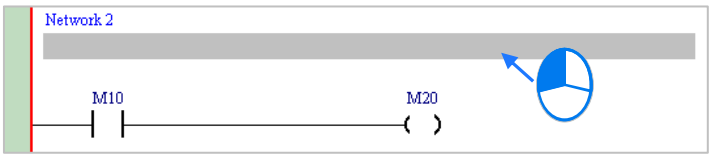


- **Editing the comment on a network**

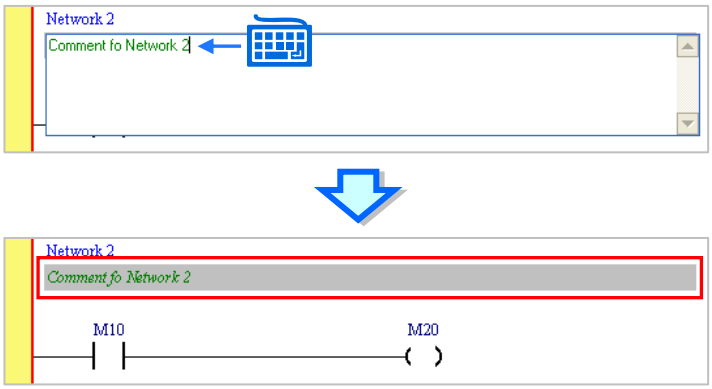
(1) After users click on the toolbar, the comments on the networks will be displayed. If the users click on the toolbar again, the comments on the networks will be hidden.



(2) After the users click the position under a network number, a box will appear.

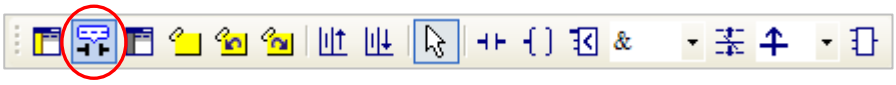


(3) Type a comment in the box. If the users want to start a new line of text at a specific point, they can press Shift+Enter on the keyboard. Press Enter on the keyboard after the editing is complete.

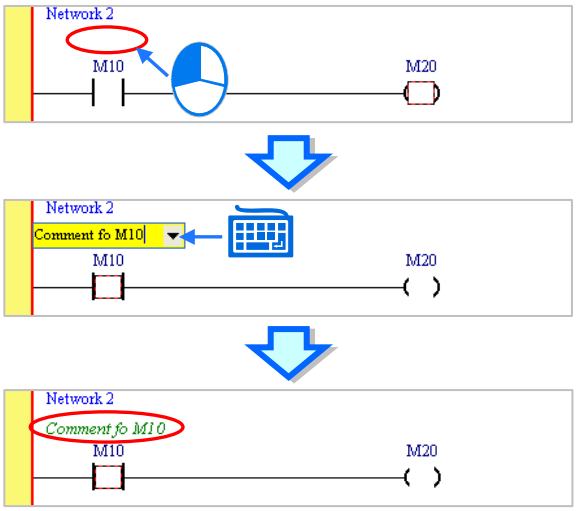


- **Editing the comment on a device**

(1) After users click on the toolbar, the comments on the devices will be displayed. If the users click on the toolbar again, the comments on the devices will be hidden.



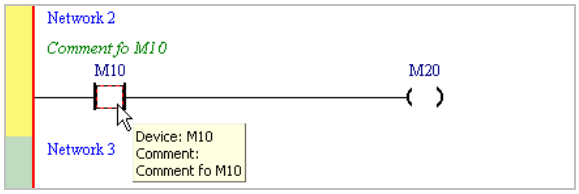
(2) After the users click the position above a device name, they can type a comment in the box which appears. Press Enter on the keyboard after the editing is complete.



Additional remark

If a symbol is assigned to a contact or a coil in the program, the comment users make on the symbol will appear after on the toolbar is clicked. The comments on the devices are different from the comments on the symbols. Users can not edit the comments on the symbols directly. If users want to edit the comment on a symbol, they have to open the **Modify Symbol** window for the symbol. Please refer to section 6.2.4 for more information.

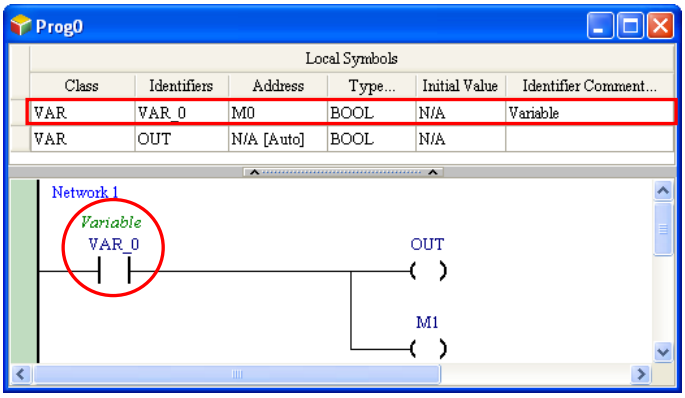
Besides, if on the toolbar is clicked, the hints related to a device will appear after the mouse cursor stays at the device for a while. Users can select hints in the **Options** window in ISPSoft. Please refer to section 2.3.1 for more information.



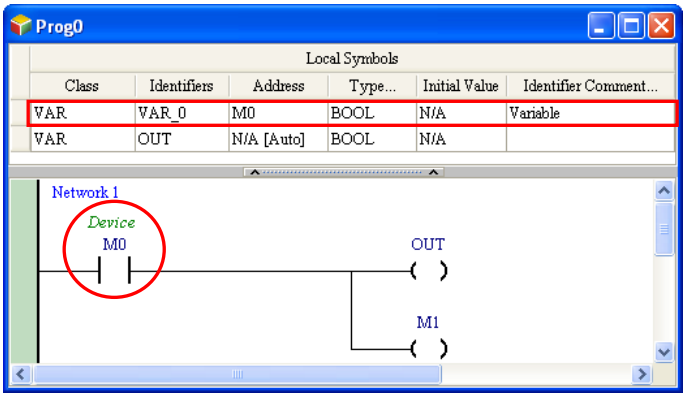
8.2.10 Symbol Mode and Address Mode

If a device address is assigned to a symbol declared in a ladder diagram, users can change the mode in which the symbol is displayed by means of clicking on the toolbar. However, the symbols to which no devices are assigned and the devices to which no symbols are assigned will remain unchanged after on the toolbar is clicked.

If on the toolbar is not clicked, the identifiers of the symbols to which devices are assigned will be displayed, and the comments on the symbols will be displayed.



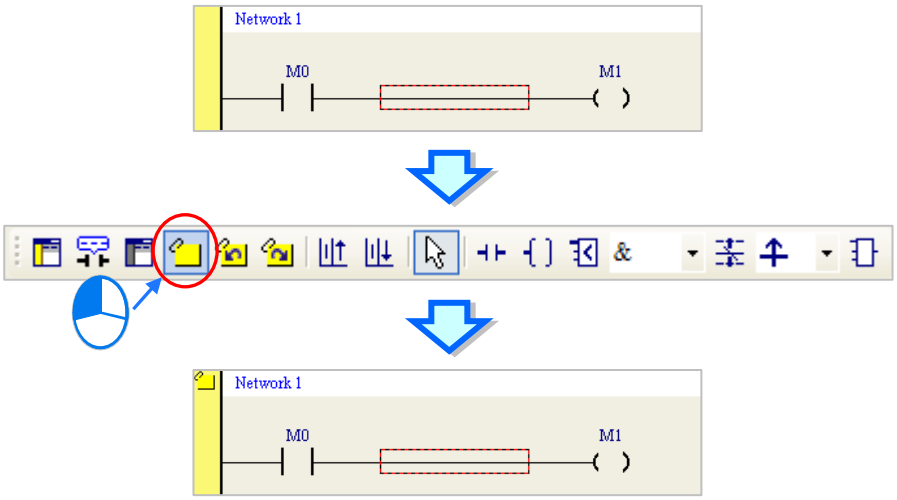
If on the toolbar is clicked, the devices to which symbols are assigned will be displayed, and the comments on the devices will be displayed.



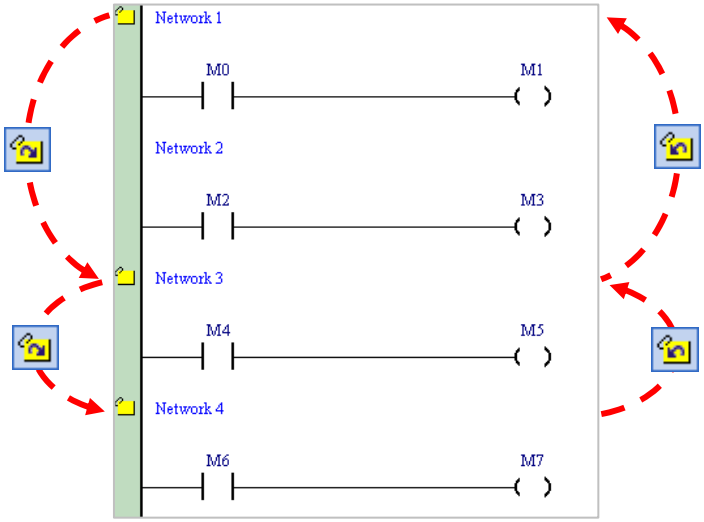
8.2.11 Bookmark

After networks are bookmarked, users can find or go to the networks easily.

(1) After users click on the toolbar, a bookmark will be inserted in the network selected. If the users want to delete the bookmark, they can select the network, and click on the toolbar again.



(2) If there are bookmarks in a program, users can go to the next network which is bookmarked by clicking on the toolbar, and go to the previous network which is bookmarked by clicking on the toolbar.



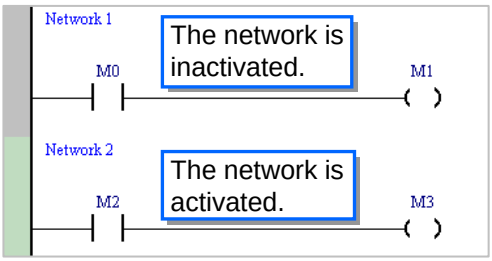
Additional remark

After users point to **Bookmarks** on the **Edit** menu, they can select **Toggle Bookmark**, **Goto the Next Bookmark**, **Goto the Previous Bookmark**, or **Remove All Bookmarks**. If the users click **Remove All Bookmarks**, all the bookmarks in the present program editing window will be deleted.

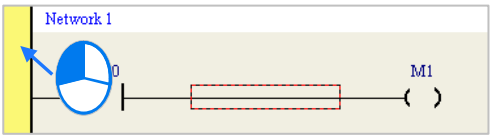
8.2.12 Activating / Inactivating a Network

If a network is inactivated, the compiling of the program will skip the network. When users test or debug a program, they can inactivate some part of the program temporarily by means of this function.

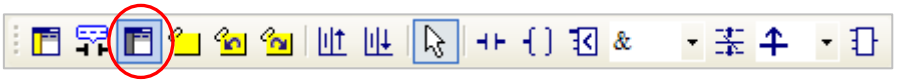
The color at the left side of a network indicates whether the network is activated or inactivated. Users can select colors which will be shown at the left sides of the networks in a ladder diagram in the **Options** window in ISPSoft. Please refer to section 2.3.1 for more information.



If users want to activate or inactivate a network, they have to select the network first. Users can select several networks simultaneously.



Right-click the network selected, and click **Activate/InactivateNetwork** on the context menu. The network selected will be activated/inactivated. Users can also activate/inactivate the network selected by clicking **Activate/InactivateNetwork** on the **Edit** menu, or by clicking on the toolbar.



If users want to activate all the networks which are inactivated in the present program editing window, they can click **Activate All Networks** on the **Edit** menu.

MEMO

Chapter 9 Function Block Diagram

Table of Contents

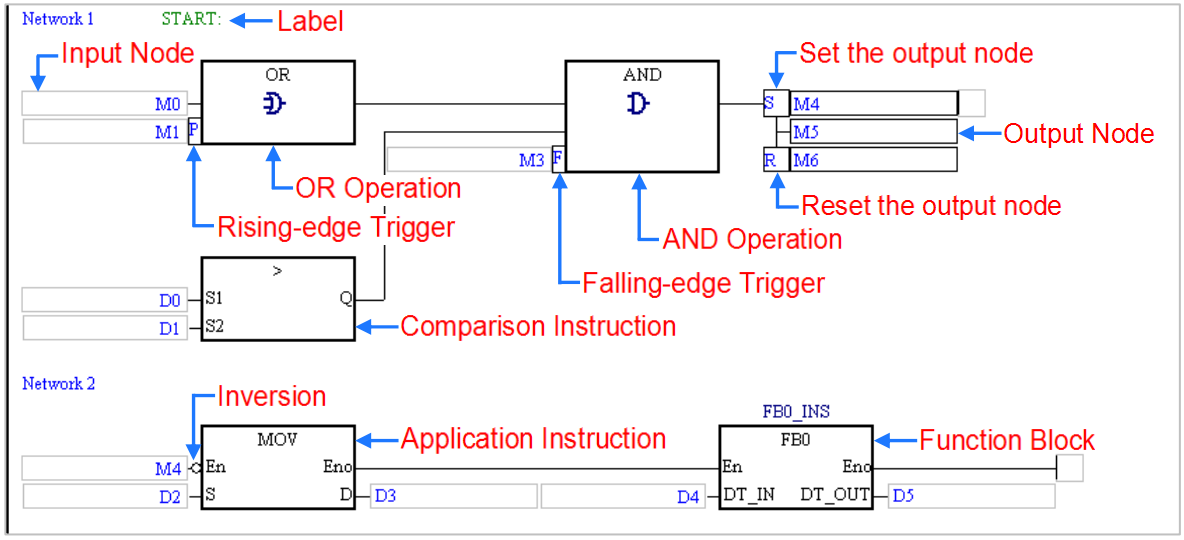
9.1 Introduction of Function Block Diagrams 9-2
9.1.1 Knowing Function Block Diagrams 9-2
9.1.2 Editing Environment 9-2
9.1.3 Networks in a Function Block Diagram 9-3
9.1.4 Selecting Objects 9-4
9.2 Creating a Function Block Diagram in ISPSoft 9-5
9.2.1 Input Nodes and Output Nodes 9-6
9.2.2 Using Devices, Symbols, and Constants in a Function Block Diagram 9-9
9.2.3 AND Block and OR Block 9-10
9.2.4 Inverse Logic 9-13
9.2.5 Rising edge-triggered Input and Falling edge-triggered Input 9-14
9.2.6 Setting an Output and Resetting an Output 9-16
9.2.7 Applied Instruction, Comparison Contact, and Function Block 9-18
9.2.8 Putting a Label 9-22
9.2.9 Comments and Hints 9-22
9.2.10 Symbol Mode and Address Mode 9-24
9.2.11 Bookmark 9-25
9.2.12 Activating/Inactivating a Network 9-26

9.1 Introduction of Function Block Diagrams

9.1.1 Knowing Function Block Diagrams

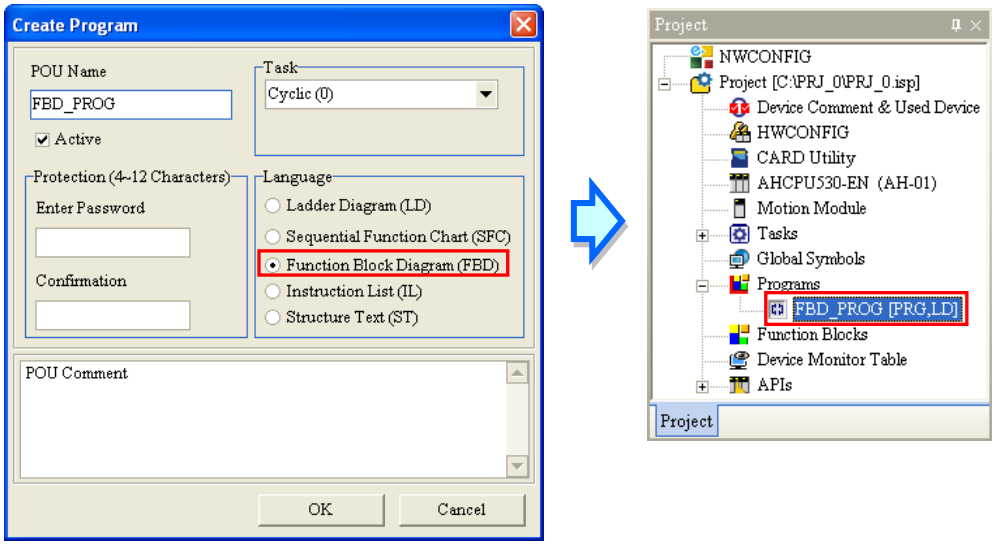
A function block diagram is one of the five languages defined by IEC 61131-3 standard. A function block diagram and a ladder diagram are graphical programming languages. A ladder diagram is a programming language that represents a program by a graphical diagram based on the circuit diagrams of relay logic hardware. A function block diagram is a block diagram that describes a function between input variables and output variables. A program is systematically divided into several function blocks, and these function blocks are arranged in a function block diagram.

A function block diagram in ISPSoft is shown below.

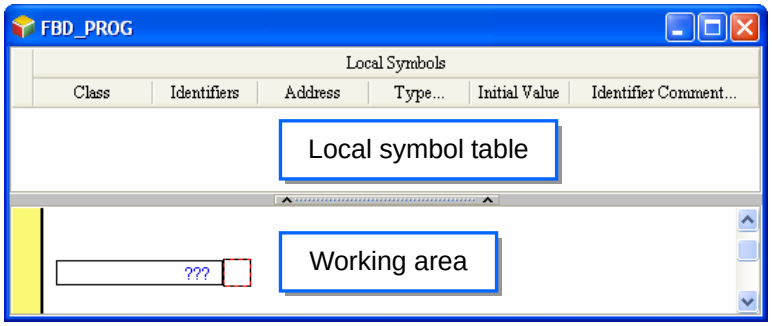


9.1.2 Editing Environment

Select the **Function Block Diagram (FBD)** option button in the **Language** section in the **Create Program** window. Please refer to section 5.4.1 for more information.



The environment in which a function block diagram can be edited is shown below. The table at the upper part of the window is a local symbol table, and the area at the lower part of the window is a working area.



After a program editing window in which a function block diagram can be created is opened, the corresponding toolbar will appear in the ISPSoft window. The functions are described below.

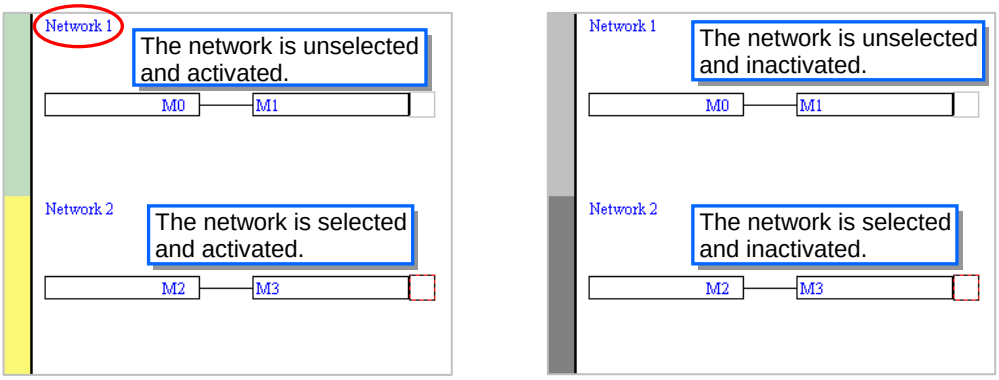
Icon	Keyboard shortcut	Function
	Shift+Ctrl+C	Displaying/Hiding the comments on the networks
	None	Displaying/Hiding the commands on the devices
	Shift+Ctrl+A	Activating/Inactivating the network selected
	Shift+Ctrl+B	Adding a bookmark to the network selected or deleting the bookmark from the network selected
	Shift+Ctrl+P	Going to the previous bookmarked position
	Shift+Ctrl+N	Going to the next bookmarked position
	Ctrl+I	Inserting a network above the network selected
	Shift+Ctrl+I	Inserting a network under the network selected
	Shift+F1	Inserting an AND block
	Shift+F2	Inserting an OR block
	Shift+F3	Inserting an output node
	Shift+F4	Adding an input node to the AND/OR block
	Shift+F5	Deleting the input node from the AND/OR block
	Shift+F6	Inserting inverse logic
	Shift+F7	Setting/Resetting the input
	Shift+F8	The input is rising edge-triggered/falling edge-triggered.
	Shift+Ctrl+U	Inserting an instruction or a function block

***. Instructions and function blocks can be inserted by means of a keyboard. Please refer to section 9.2.7 for more information.**

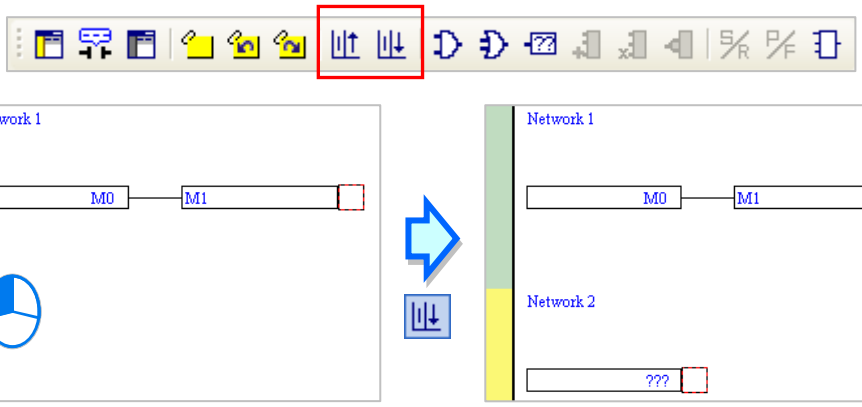
9.1.3 Networks in a Function Block Diagram

A function block diagram consists of networks. Every network is an independent program. Besides, there is no limit on the number of objects which can be inserted in a network in ISPSoft.

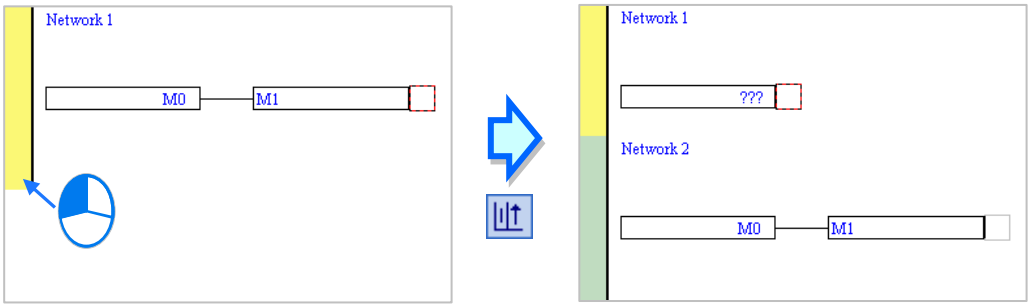
Network numbers are marked on the networks in a ladder diagram. The color at the left side of a network indicates the state of the network. A network can be activated or inactivated, and a network can be selected or unselected. Users can select colors which will be shown at the left sides of the networks in a ladder diagram in the **Options** window in ISPSoft. Please refer to section 2.3.1 for more information.



After a new window in which a ladder diagram will be created is opened, the system will insert a blank network in the window. If users want to add a network, they can select a network, and click on the toolbar. The network added will be under the network selected.

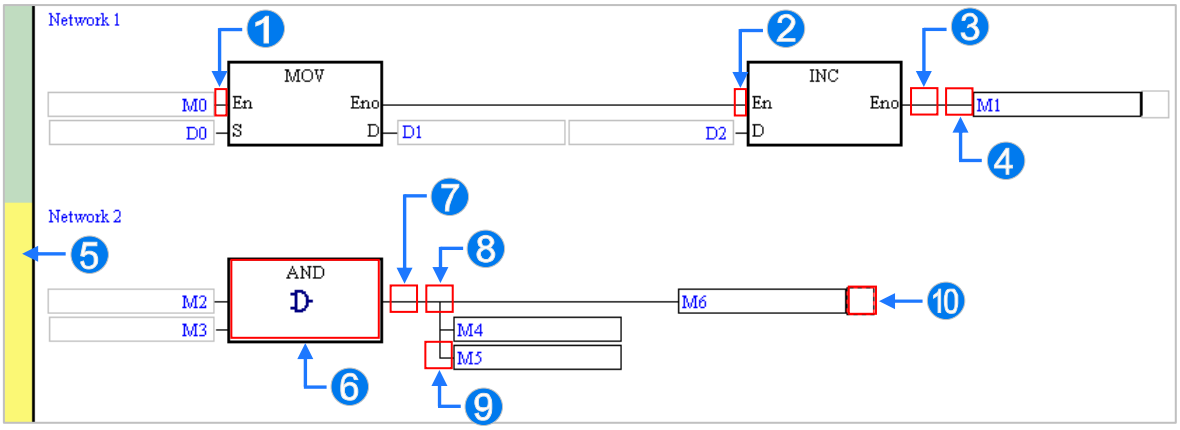


If users click on the toolbar, a blank network will be put above the network selected.



9.1.4 Selecting Objects

The red frame in the working area indicates the objects which are edited. It also indicates the objects which are selected. If users want to edit he program in the figure below, they can only select the positions marked with red frames. Besides, the functions which can be used vary with the position selected. The objects which are copied/cut/pasted/deleted depend on the position selected. The positions which are marked with red frames in the figure are described below.



❶ The input node at the right side of this position is selected. M0 in the figure is selected.

❷ The block before this position is selected.

❸ All the blocks before this position are selected.

❹ The output node at the right side of this position is selected. M1 in the figure is selected.

❺ The whole network is selected.

❻ The block is selected.

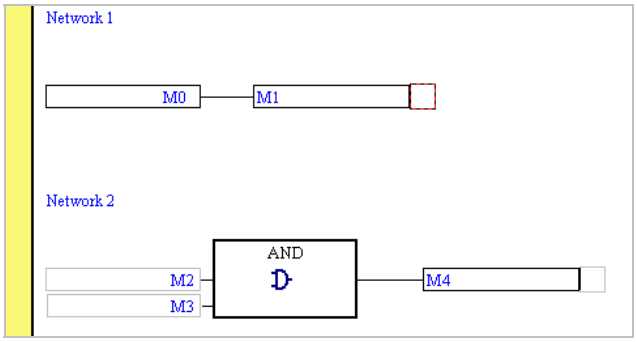
❼ The block before this position is selected.

❽ The output nodes under this position are selected.

❾ The output node at the right side of this position is selected.

❿ The block before this position is selected.

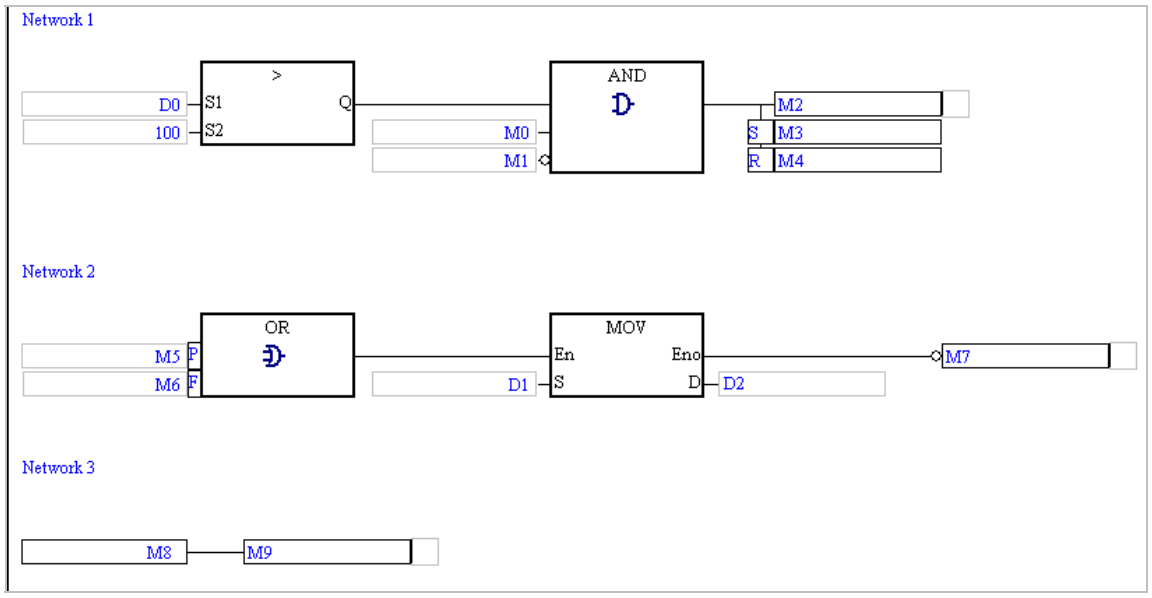
If users want to select several networks, they can hold down Ctrl on the keyboard while they click the networks. The users can also select a range of networks by pressing Shift on the keyboard, clicking the first network within the range, and the last network within the range.



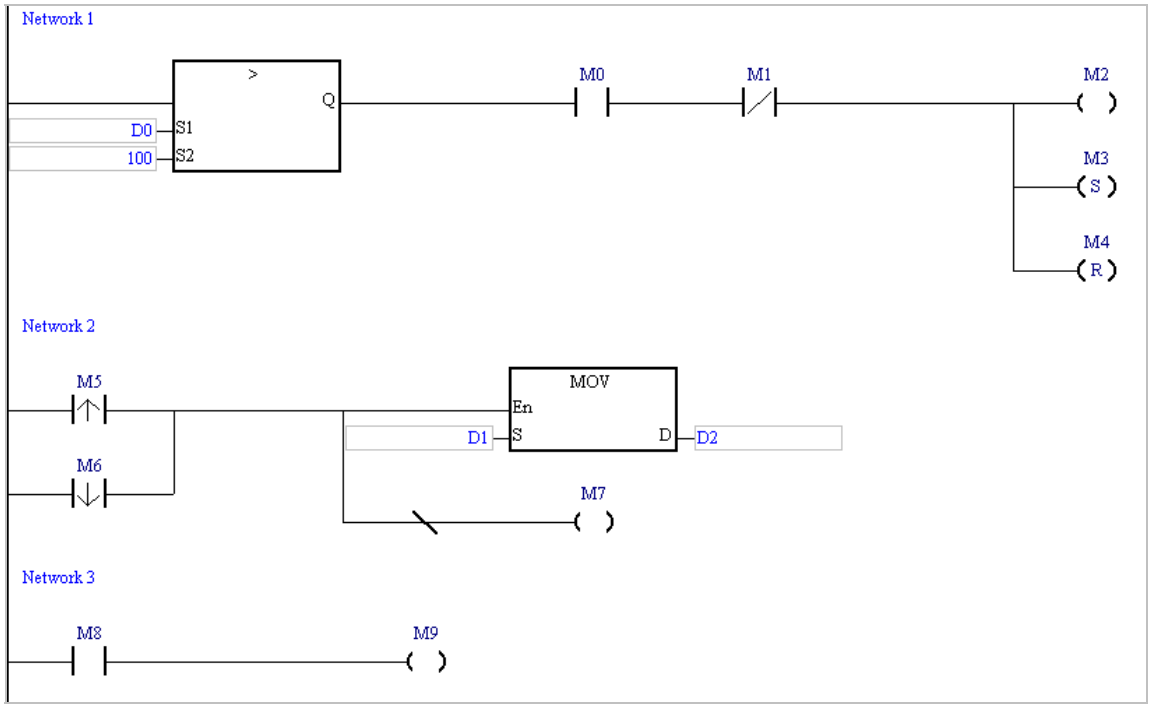
9.2 Creating a Function Block Diagram in ISPSoft

A basic function block diagram and its equivalent ladder diagram are shown below. The graphical representations in a function block diagram will be described in the following sections.

- **Function block diagram**



- **Equivalent ladder diagram**



9.2.1 Input Nodes and Output Nodes

The devices and the symbols in a function block diagram are represented by nodes. An input node and the corresponding output node must conform to the rules listed below.

(a) One node represents an M/S/T/C/HC device, and the other node represents a symbol whose data type is BOOL/STEP/TIMER/COUNTER. (The device assigned to the node which represents a symbol is a bit device.)

(b) One node represents a D/L device, and the other node represents a symbol whose data type is not BOOL/STEP/REAL/STRING.

(c) One node represents a T/C device, and the other node represents a D/L device or a symbol whose data type is WORD/INT/TIMER/COUNTER.

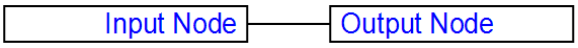
(d) One node represents a D/L/HC device, or a symbol whose data type is COUNTER, and to which a HC device is assigned. The other node represents a symbol whose data type is DWORD/DINT.

(e) One node represents a symbol whose data type is WORD/DWORD/LWORD, and the other node represents a symbol whose data type is INT/DINT/LINT. The data length of the output node must be larger than or equal to the data length of the input node.

(f) Both nodes represent symbols whose data types are REAL/LREAL/STRING. The data lengths of the nodes must be the same.

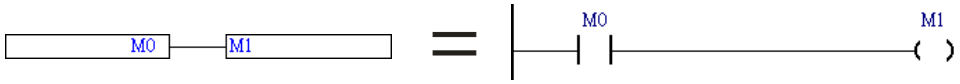
(g) If an input node has multiple output nodes, the output nodes must conform to the rules listed above. 32 output nodes at most form a group. After users add an output node rightward, they can create another group of output nodes.

- **Graphical representation**

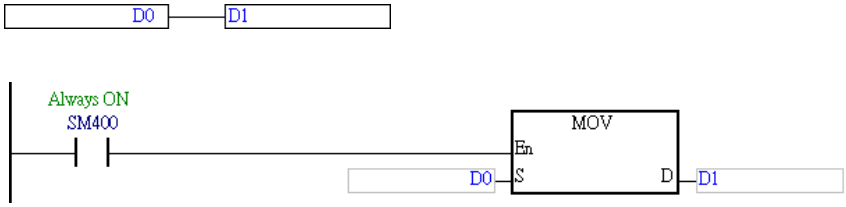


- **Example**

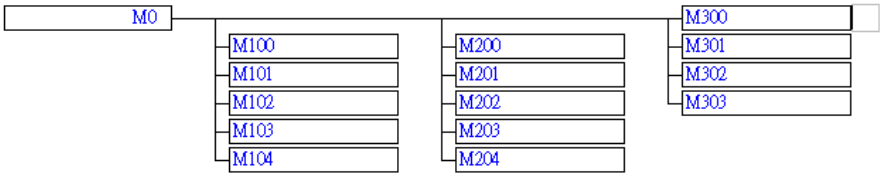
Example 1: The state of the contact M0 is sent to M1.



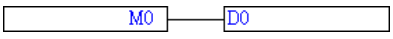
Example 2: The data in D0 is sent to D1.

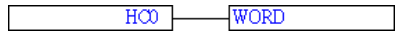
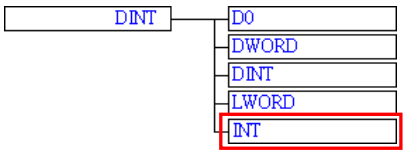


Example 3: There are three groups of output nodes. The first group begins with M100, the second group begins with M200, and the third group begins with M300. 32 output nodes at most form a group. After users add an output node rightward, they can create another group of output nodes. There is no limit on the number of output nodes which can be added rightward.



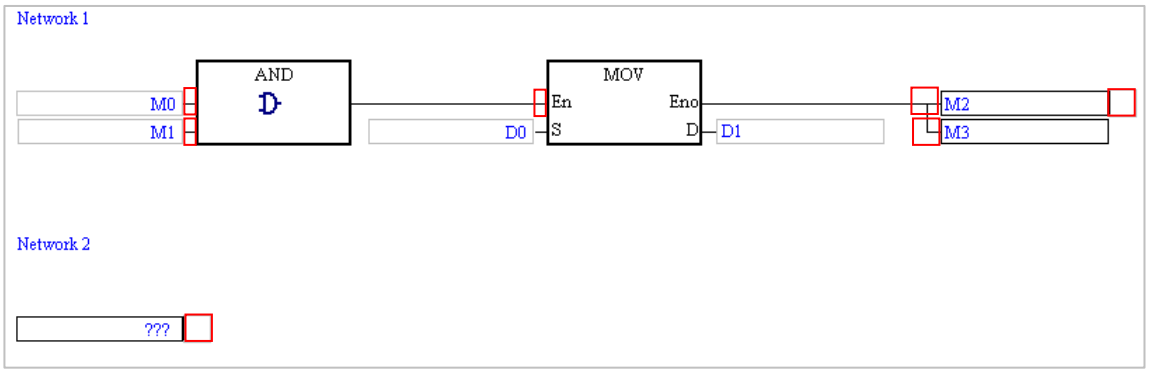
Example 4: Some incorrect examples are listed below.

Incorrect example	Description
	The data type of the input node must not be different from the data type of the output node.
	The data length of the output node must not be less than the data length of the input node.
	The value of a symbol whose data type is REAL can not be sent to a D device.
	If the input node represents a symbol whose data type is REAL, the output node must represent a symbol whose data type is REAL. If the input node represents a symbol whose data type is LREAL, the output node must represent a symbol whose data type is LREAL.
	The data length of the input node representing a symbol whose data type is STRING and the data length of the output node representing a symbol whose data type is STRING must be the same.

Incorrect example	Description
	The value of a symbol whose data type is TIMER can not be sent to a symbol whose data type is DWORD, but can be sent to a symbol whose data type is WORD.
	The value in a HC device can not be sent to a symbol whose data type is WORD, but can be sent to a symbol whose data type is DWORD.
	The multiple output nodes must conform to the rules listed above. The example is incorrect because the data length of the output node representing a symbol whose data type is INT is less than the data length of the input node.

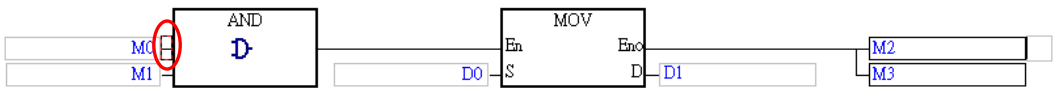
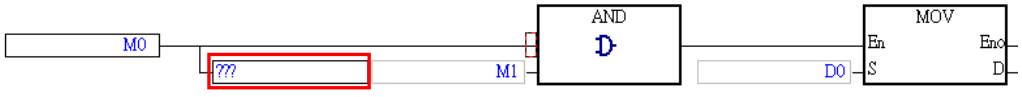
- **Inserting an output node**

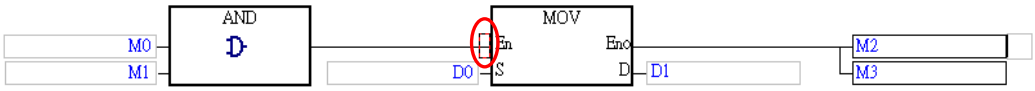
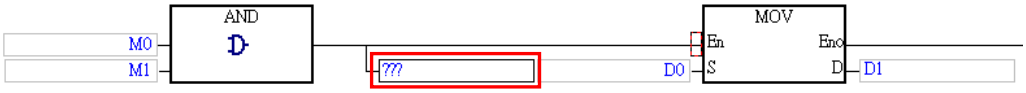
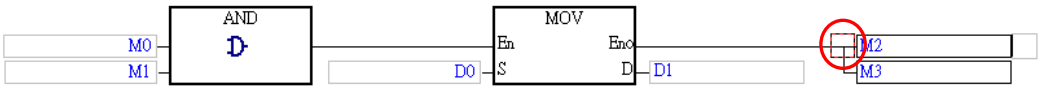
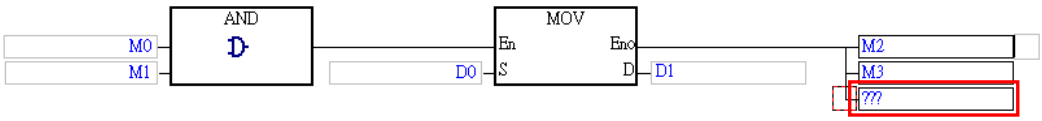
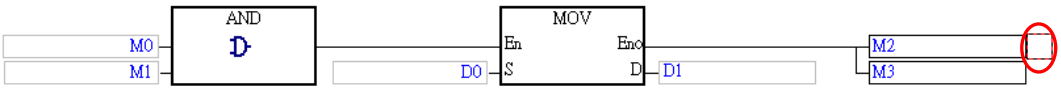
(1) Select a position into which an output node will be inserted in the working area. Users are allowed to insert output nodes into the positions marked with red frames in the figure below.



(2) In addition to the structure in the figure above, users may be allowed to insert an output node into another structure. If users are allowed to insert an output node into the position selected, on the toolbar can be clicked. After is clicked, an output node will be inserted.

An output node can be inserted in the following structures.

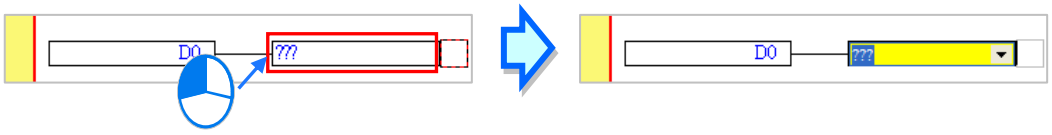
Inserting an output node into an input pin of an AND/OR block	
Before	
After	

Inserting an output node into the output pin of an AND/OR block	
Before	
After	
Adding an output node downward	
Before	
After	
Adding an output node rightward	
Before	
After	

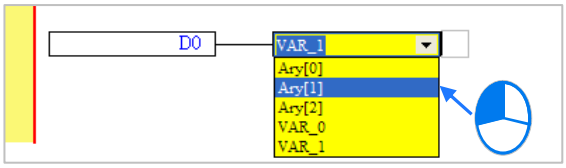
***. Users are not allowed to insert output nodes into the output pins of a function block/an applied instruction to which operands are assigned.**

9.2.2 Using Devices, Symbols, and Constants in a Function Block Diagram

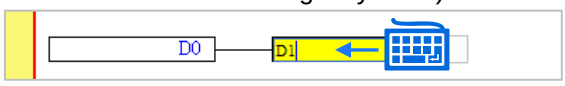
Before users edit a node or an operand of an instruction, they have to click the node or the operand.



Click ▼ in the box, and then select a symbol on the drop-down list. If a symbol whose data type is ARRAY is declared, the elements in the array will be on the drop-down list. Ary[0], Ary[1], and Ary[2] in the figure below are elements.



The users can type a device name or a symbol in the box. If they type a symbol which is not declared, the **Add Symbol** window will appear after they press Enter on the keyboard. (Please refer to section 2.3.1 for more information about declaring a symbol.)



After users click a network and press Enter on the keyboard, they can edit a box. The users can edit the next box in the network after they press Enter on the keyboard. Besides, the next network will be selected after the users press Tab on the keyboard. The users can edit a box with the keyboard. If the users have declared symbols, they can press Page Down on the keyboard when they edit a box. The users can select a symbol by the up/down key on the keyboard. After the editing is complete, the users can press Enter on the keyboard to jump to the next box. If the users want to end the editing, they can press Esc on the keyboard.

Additional remark

(a) If users want to use constants in a function block diagram in ISPSoft, the constants must be represented in the following ways.
- Decimal value: 23456 (A value which is not preceded by any mark will be regarded as a decimal value.)
- Hexadecimal value: 16#5BA0 (A value is preceded by 16#.)
- Octal value: 8#55640 (A value is preceded by 8#.)
- Binary value: 2#101101110100000 (A value is preceded by 2#.)
- String: “XYZ” (Characters are put in double quotes.)
- Boolean value:

  AH500: SM400 (normally-open contact), or SM401 (normally-closed contact) is used.

  DVP: M1000 (normally-open contact), or M1001 (normally-closed contact) is used.

(b) If users want to use a symbol whose data type is ARRAY, the expression format is **Identifier[Index]**. Generally speaking, an index is a decimal constant. The indexes in a function block diagram can not be symbols. The minimum index value for a symbol whose data type is ARRAY must be 0, and the maximum index value for a symbol whose data type is ARRAY can not be larger than the number of elements in the array.

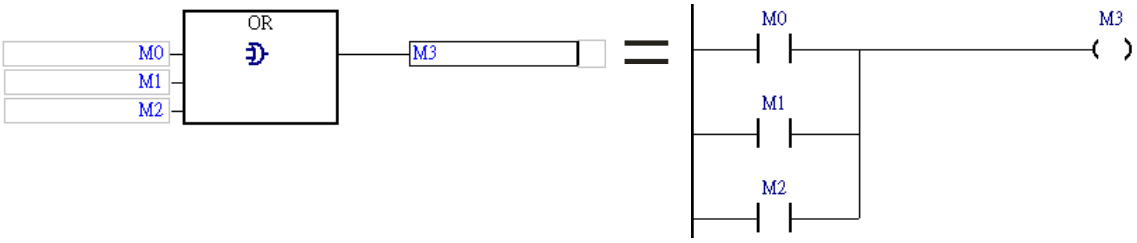
9.2.3 AND Block and OR Block

Contacts in a function block diagram can be connected in series/parallel by means of an AND/OR block. The input nodes and the output nodes of an AND/OR block must represent Boolean devices or symbols whose data types are BOOL. An AND/OR block have 32 input nodes and 32 output nodes at most. Besides, an AND/OR block has at least one output node, or is connected to the next object.

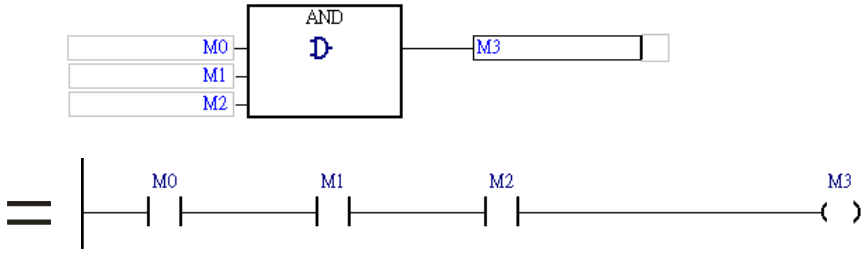
- **Graphical representation**

- **Example**

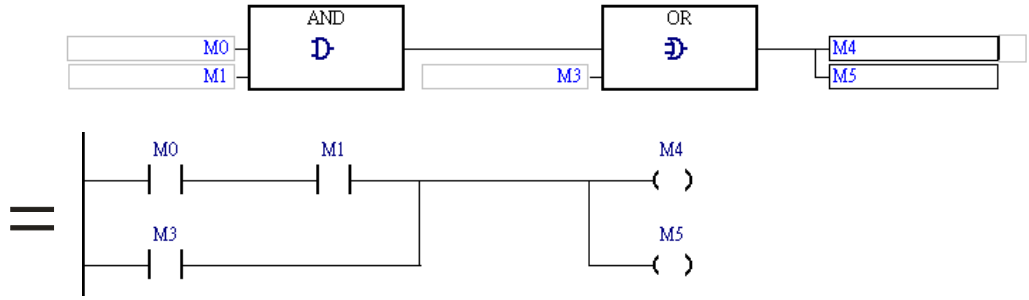
Example 1: The contacts M0, M1, and M2 are assigned to the OR block, and the result of the operation is sent to M3. In other words, after M0, M1, and M2 are connected in parallel, the coil M3 will be driven if M0, M1, or M2 is ON.



Example 2: The contacts M0, M1, and M2 are assigned to the AND block, and the result of the operation is sent to M3. In other words, after M0, M1, and M2 are connected in series, the coil M3 will be driven if M0, M1, and M2 are ON.

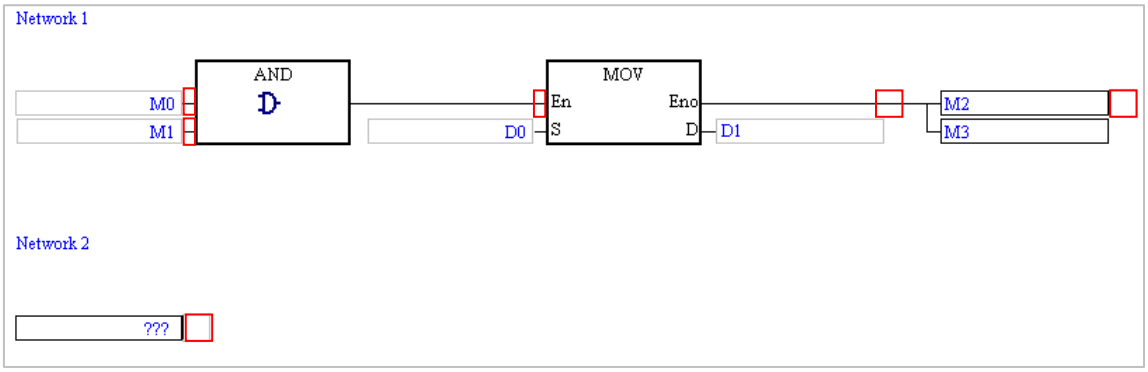


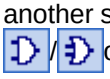
Example 3: The use of an AND block and an OR block in a function block diagram is equivalent to the use of a serial connection and a parallel connection in a ladder diagram.



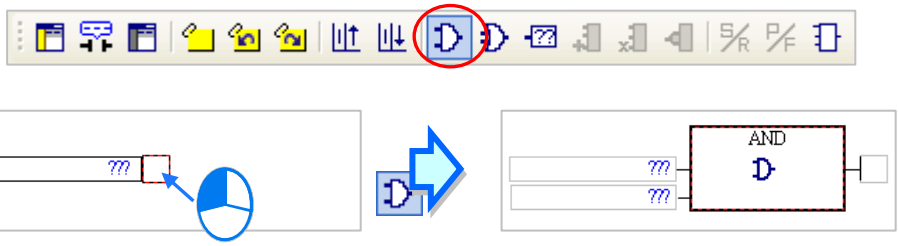
- **Inserting an AND/OR block**

(1) Select a position into which an AND/OR block will be inserted in the working area. Users are allowed to insert AND/OR blocks into the positions marked with red frames in the figure below.

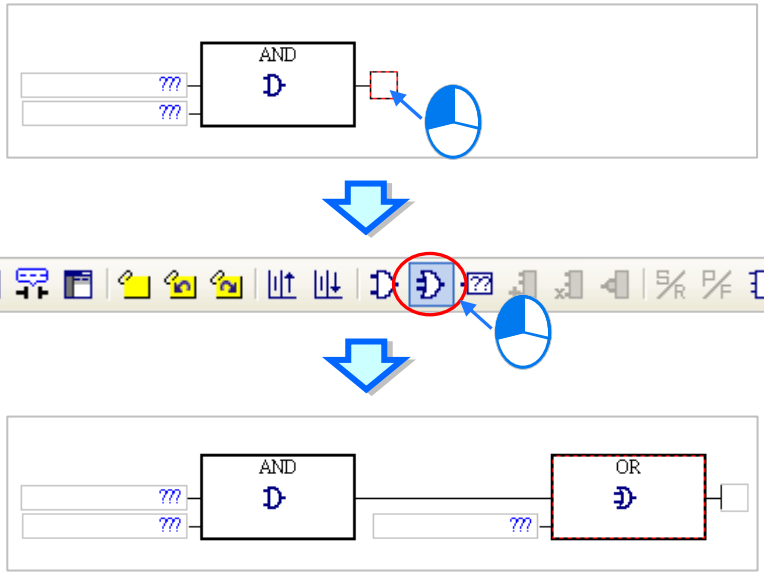


(2) In addition to the structure in the figure above, users may be allowed to insert an AND/OR block into another structure. If users are allowed to insert an AND/OR block into the position selected,  on the toolbar can be clicked.

After is clicked, an AND block will be inserted.

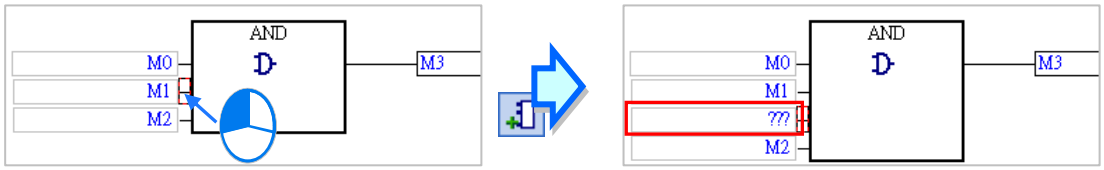


After is clicked, an OR block will be inserted.

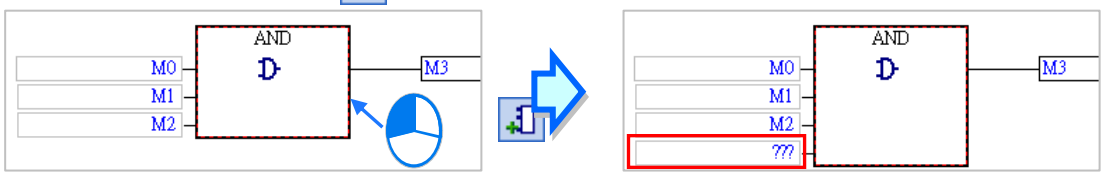


- **Adding an input node to an AND/OR block**

  Select an input node of an AND/OR block, and then click on the toolbar. After is clicked, a node will be added to the bottom of the input node selected. If the input node selected is the topmost input node of an AND/OR block, e.g. M0 in the figure below, users can not add an input node.



If an AND/OR block is selected, an input node will be added to the bottom of the bottommost input node of the block after is clicked.



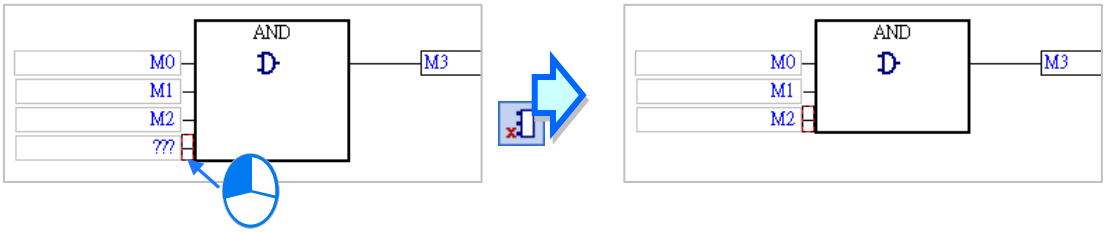
- **Deleting an input node from an AND/OR block**

  Select an input node which will be deleted from an AND/OR block, and then click on the toolbar. After is clicked, the input node will be deleted. Users have to pay attention to the

following two points when they delete an input node.

(a) The topmost input node of an AND/OR block, e.g. M0 in the figure below, can not be deleted.

(b) If an AND/OR block has only two input nodes, neither of the input nodes can not be deleted.



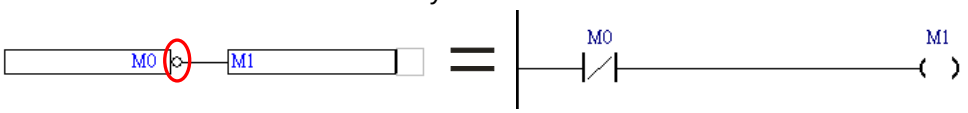
9.2.4 Inverse Logic

Users can insert inverse logic into a Boolean position in a function block diagram, e.g. an input node which represents a Boolean device or a symbol whose data type is BOOL, an output node which represents a Boolean device or a symbol whose data type is BOOL, the En pin of a function block, or the En pint of an applied instruction. However, users have to notice that they are not allowed to insert inverse logic into the pins of a function block/an applied instruction to which operands are assigned.

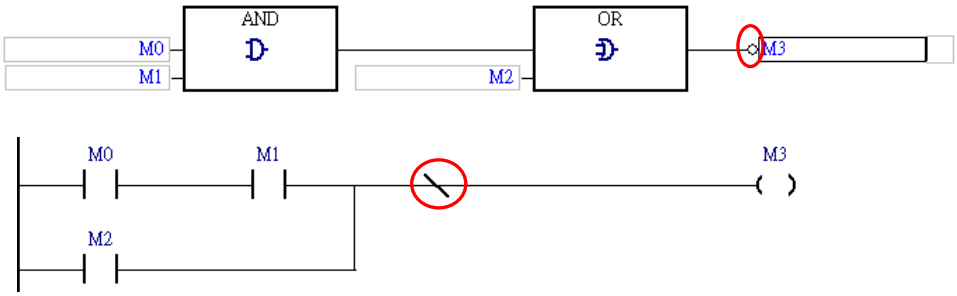
- **Graphical representation**

- **Example**

Example 1: The state of M0 is inverted, and the inversion result is sent to M1. Likewise, the coil M1 will be driven if the normally-closed M0 is OFF.

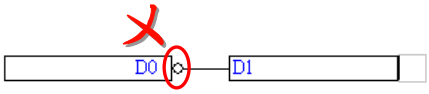


Example 2: The inverse logic inserted into the output node indicates that the result of the operation is inverted. The inversion result is sent to M3.



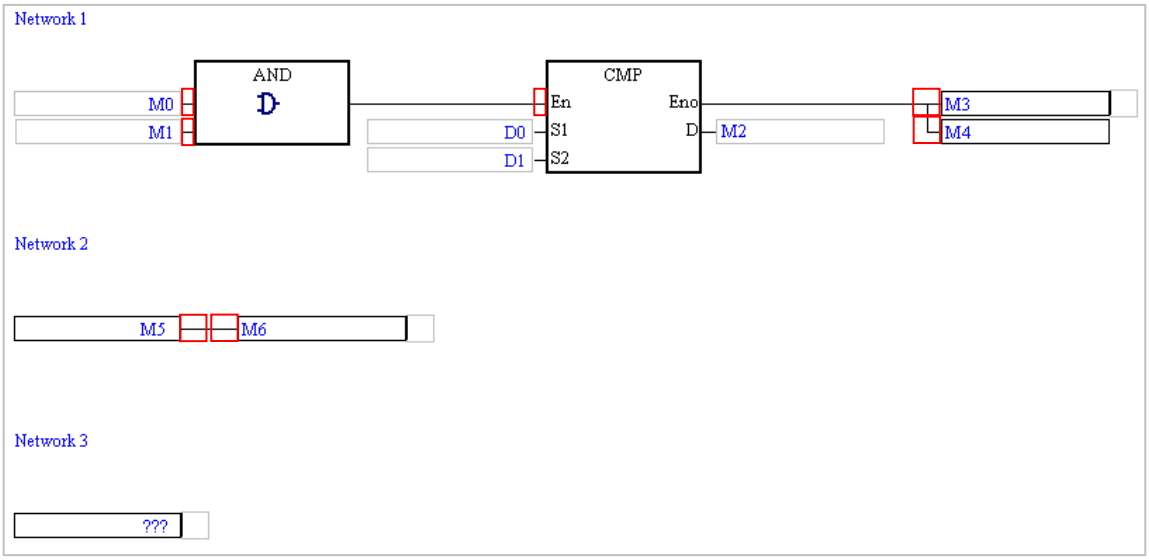
Example 3: If M0 is ON, the function block BLACK will be executed. If M0 is OFF, the function block WHITE will be executed.

Example 4: The figure below is an incorrect example. Inverse logic can not be inserted into a node which does not represent a Boolean device or a symbol whose data type is BOOL.



- **Inserting inverse logic**

(1) Select a position into which inverse logic will be inserted in the working area. Users are allowed to insert inverse logic into the positions marked with red frames in the figure below. Although the operand assigned to the output pin of the instruction CMP is a Boolean operand, inverse logic can not be inserted into the output pin of CMP. Besides, inverse logic can not be inserted into a position which does not have an output node, and a position which is not connected to an object. For example, inverse logic can not be inserted into the input node in network 3.



(2) In addition to the structure in the figure above, users may be allowed to insert inverse logic into another structure. If users are allowed to insert inverse logic into the position selected, on the toolbar can be clicked. After users click , inverse logic will be inserted.

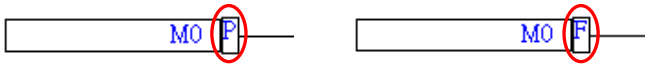
If the users want to cancel the inverse logic, they can click again.

9.2.5 Rising edge-triggered Input and Falling edge-triggered Input

In a function block diagram in ISPSoft, the mark P on an input node indicates that the input node is rising edge-triggered, and the mark F on an input node indicates that the input node is falling edge-triggered. The mark P/F can only be inserted into an input node which represents a Boolean device or a symbol whose data type is BOOL, or an input node of an AND/OR block. Users are not allowed to insert P/F into the operands of a function block/an applied instruction, and the En pin of a

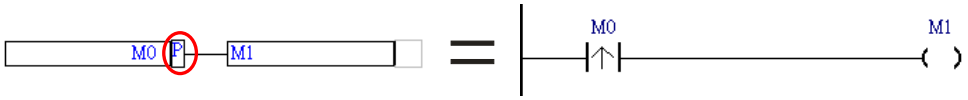
function block/an applied instruction.

- **Graphical representation**

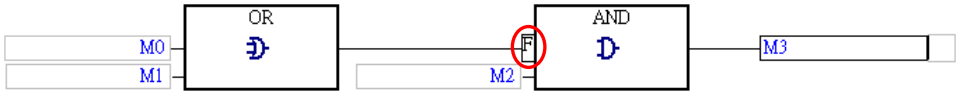


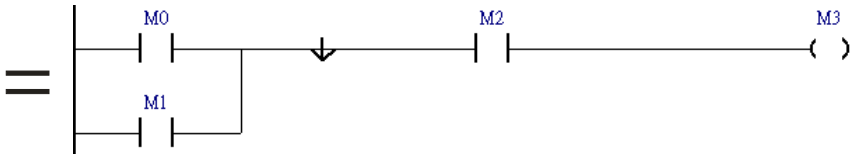
- **Example**

Example 1: When M0 is turned from OFF to ON, M1 is ON. When M0 is turned form ON to OFF, M1 is OFF.

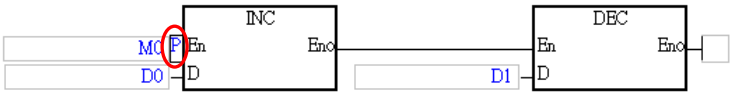


Example 2: The mark F is inserted into an input pin of the AND block.

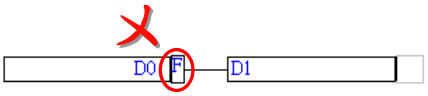




Example 3: The En pin of the first applied instruction is connected to the input node representing M0, and therefore the mark P can be inserted into in the input node representing M0. However, the mark P/F can not be inserted into the En pin of the second applied instruction.

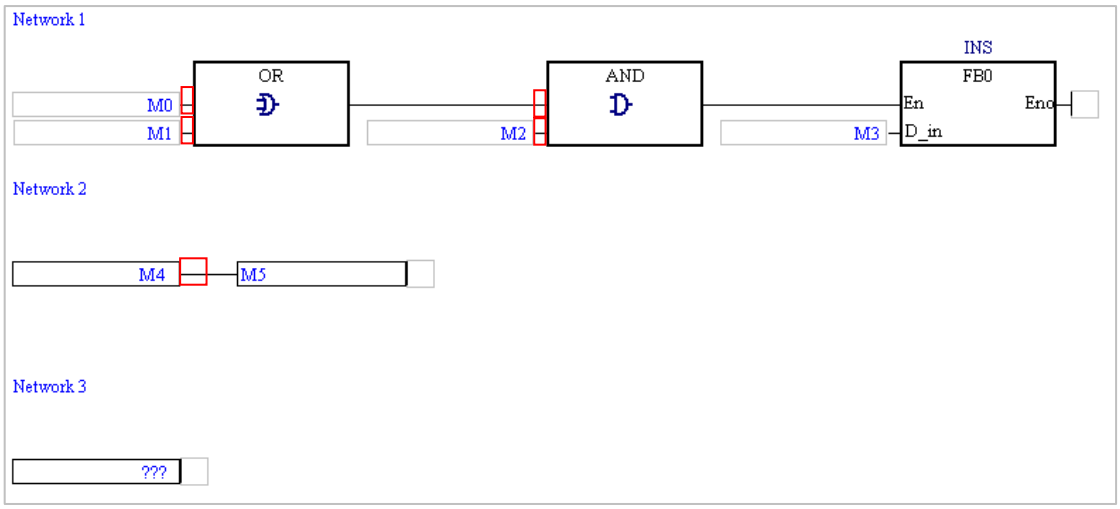


Example 4: The figure below is an incorrect example. The mark F can not be inserted into an input node which does not represent a Boolean device or a symbol whose data type is BOOL.

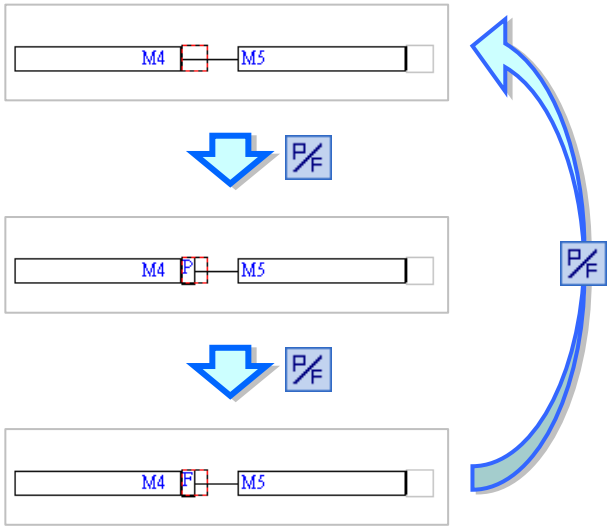


- **Inserting the mark P/F**

(1) Select a position into which the mark P/F will be inserted in the working area. Users are allowed to insert the mark P/F into the positions marked with red frames in the figure below. Although the operand assigned to the input pin of the function block FB0 is a Boolean operand, the mark P/F can not be inserted into the input pin of FB0. Besides, P/F can not be inserted into a position which does not have an output node, and a position which is not connected to an object. For example, the mark P/F can not be inserted into the input node in network 3.



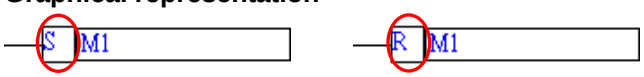
(2) In addition to the structure in the figure above, users may be allowed to insert the mark P/F into another structure. If users are allowed to insert the mark P/F into the position selected, P/F on the toolbar can be clicked. After users click P/F, the mark P will be inserted. If the users click P/F again, the mark F will be inserted.



9.2.6 Setting an Output and Resetting an Output

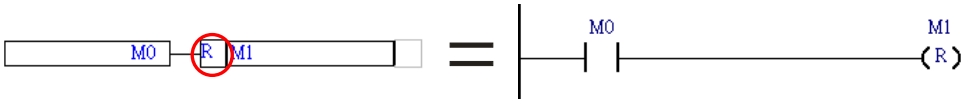
In a function block diagram in ISPSoft, the mark S on an output node indicates that the output node is set to ON, and the mark R on an input node indicates that the output node is reset to OFF. Users are not allowed to insert S/R into the operands of a function block/an applied instruction.

- **Graphical representation**

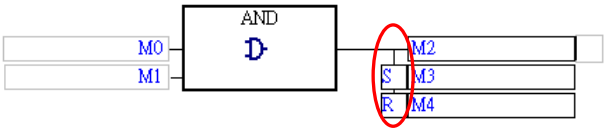


- **Example**

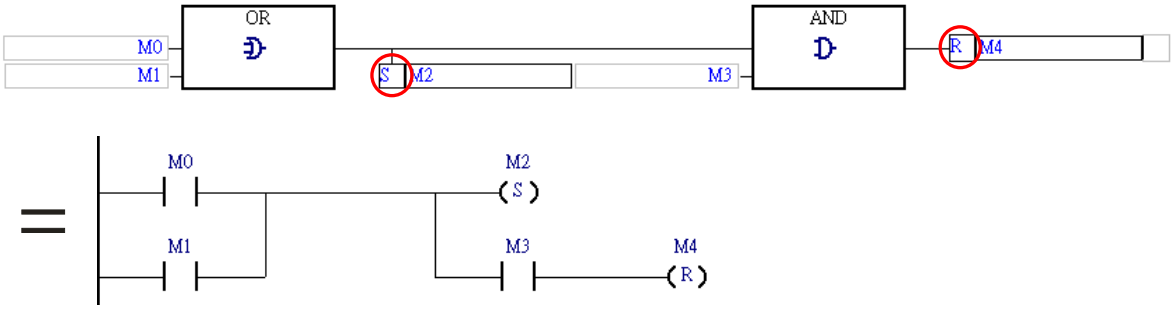
  Example 1: If M0 is ON, M1 will be reset to OFF.



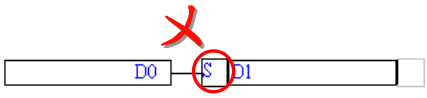
Example 2: In the figure below, the mark S is inserted into the output node representing M3, and the mark R is inserted into the output node representing M4.



Example 3: If M0 or M1 is ON, M2 will be set to ON. If M0 or M1 is ON, and M3 is ON, M4 will be reset to OFF.

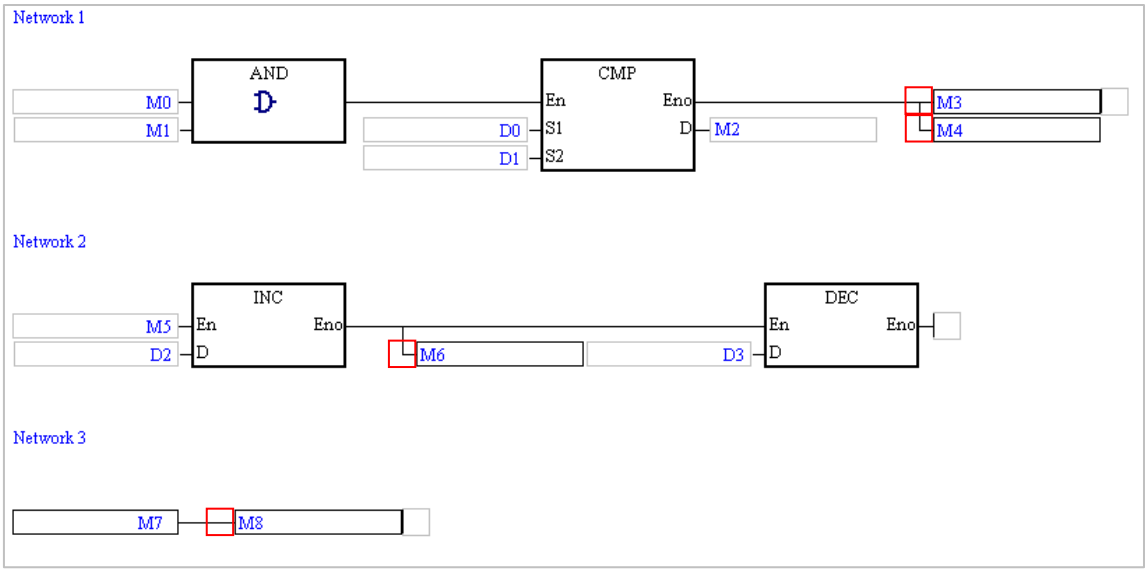


Example 4: The figure below is an incorrect example. The mark S can not be inserted into an output node which does not represent a Boolean device or a symbol whose data type is BOOL.



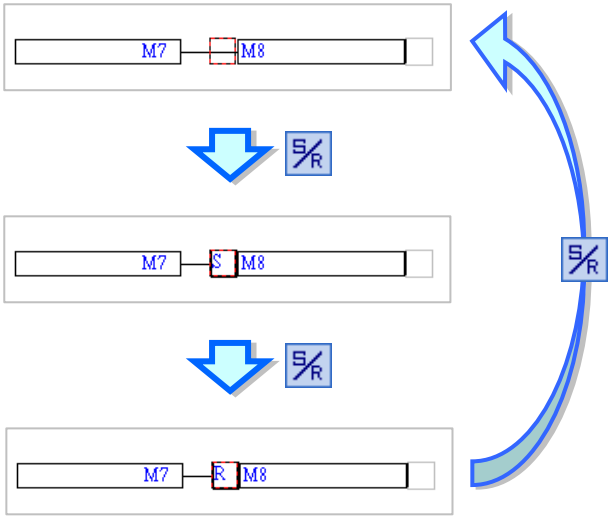
- **Inserting the mark S/R**

(1) Select a position into which the mark S/R will be inserted in the working area. Users are allowed to insert the mark S/R into the positions marked with red frames in the figure below. Although the operand assigned to the output pin of the applied instruction CMP is a Boolean operand, the mark S/R can not be inserted into the output pin of CMP.



(2) In addition to the structure in the figure above, users may be allowed to insert the mark S/R into another structure. If users are allowed to insert the mark S/R into the position selected,

on the toolbar can be clicked. After users click , the mark S will be inserted. If the users click again, the mark R will be inserted.



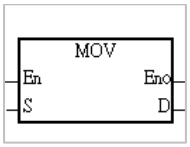
9.2.7 Applied Instruction, Comparison Contact, and Function Block

In a function block diagram in ISPSoft, the applied instructions, the comparison contacts, and the function blocks are represented by blocks. Besides, users can type function block instances in the boxes above the function blocks.

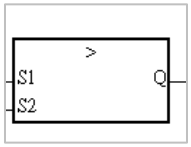
If the logic state connected to the En pin of a block representing an applied instruction or a function block is ON, the applied instruction or the function block will be executed. A function block diagram is flexible in that users can put a logical operation, e.g. inverse logic or an AND/OR block, between two function blocks, between two applied instructions, or between a function block and an applied instruction in a network.

- **Graphical representation**

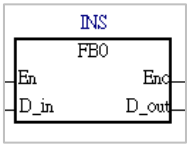
➢ Application Instruction



➢ Comparison Instruction

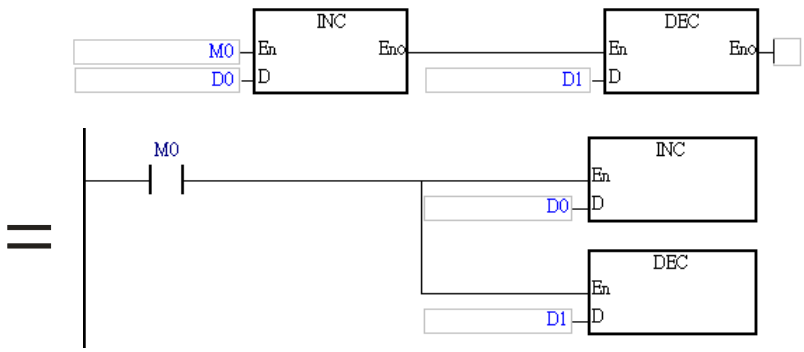


➢ Function Block

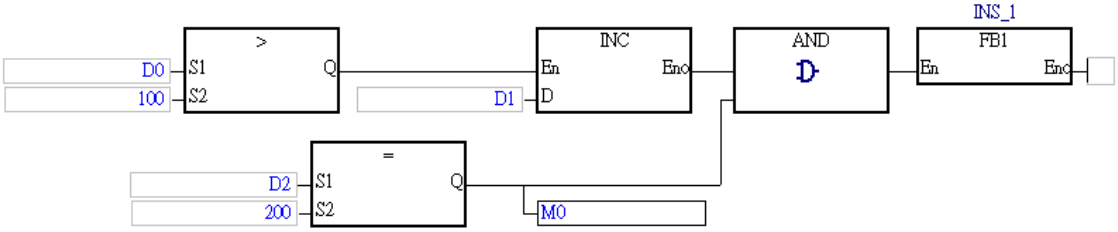


- **Example**

Example 1: The En pin of a block representing an applied instruction in a function block diagram can be connected to anther instruction or object.



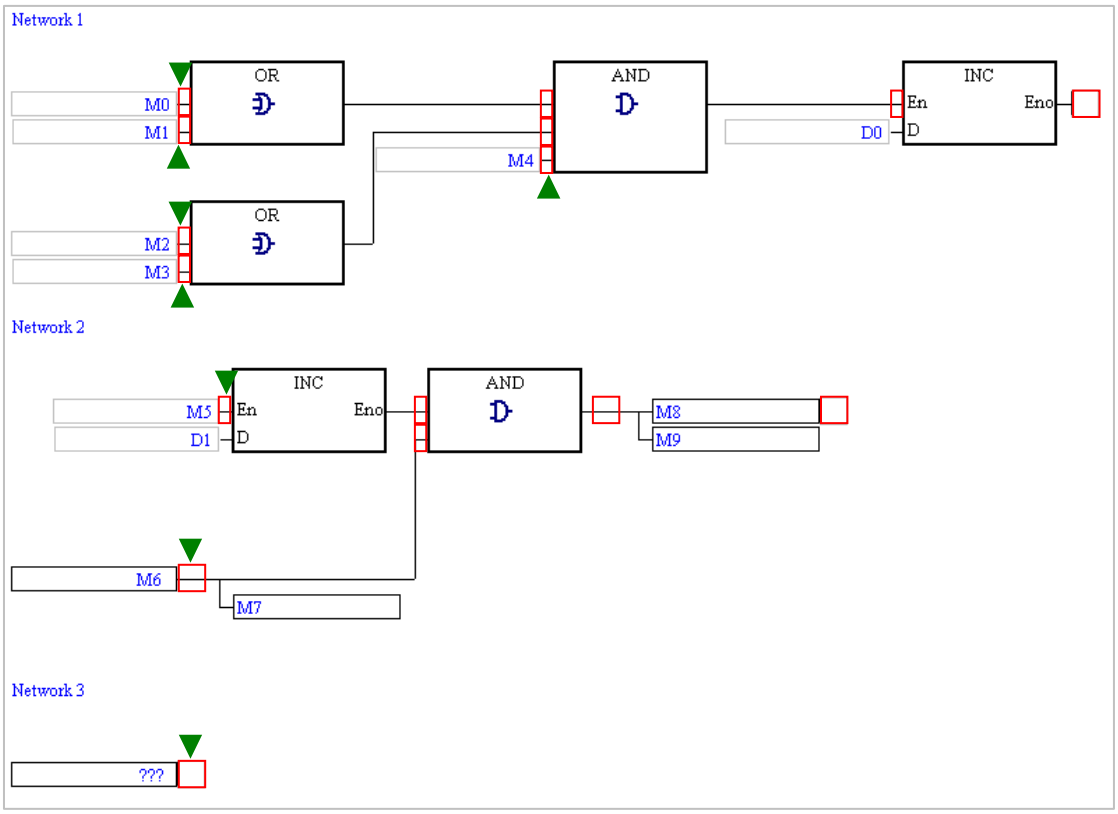
Example 2: In a function block diagram, no object can precede a comparison contact, but the output pin of the comparison contact can connect to an input pin of an AND/OR block, an output node, the En pin of a block representing a function block, or the En pin of a block representing an applied instruction.



***. Please refer to chapter 7 for more information about function blocks.**

- **Inserting an applied instruction, a comparison contact, or a function block**

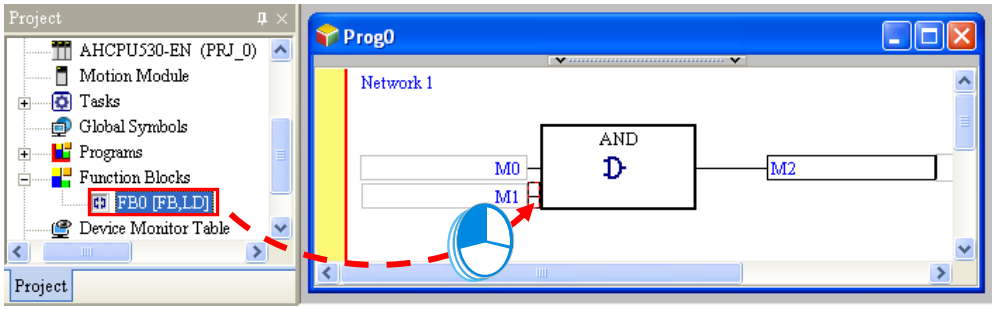
(1) Select a position into which an applied instruction, a comparison contact, or a function block will be inserted in the working area. In the figure below, users are allowed to insert applied instructions and function blocks into the positions marked with red frames, and comparison contacts into the positions marked with green triangles.



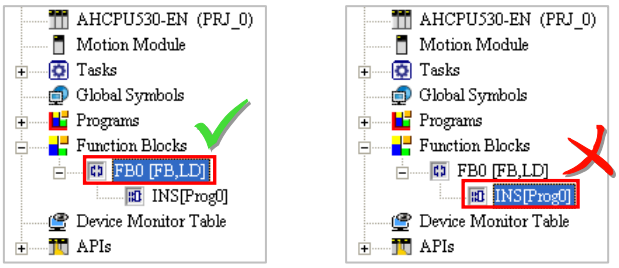
(2) There are three ways to insert an applied instruction, a comparison contact, or a function block in a function block diagram.

➢ **Method 1**

Unfold the **APIs** section or the **Function Blocks** section in the project management area, and find the item which will be inserted. Select the item, and drag it to the position in which it will be inserted.

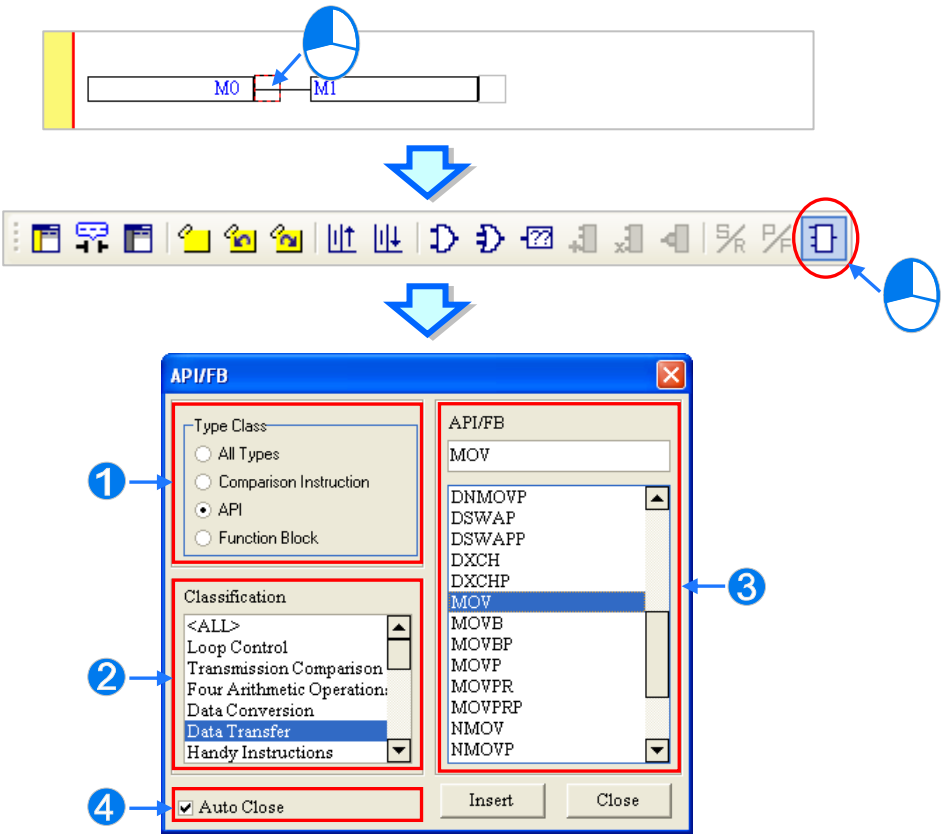


Function block definitions can be dragged, but function block instances can not be dragged.



➢ **Method 2**

Click the position into which an applied instruction, a comparison contact, or a function block will be inserted, click on the toolbar, select the item which will be inserted in the **API/FB** window, and click **Insert**.



❶ Users can select the **All Types** option button, the **Comparison Instruction** option button, the **API** option button, or the **Function Block** option button.

❷ If users select the **API** option button in the **Type Class** section, they have to select

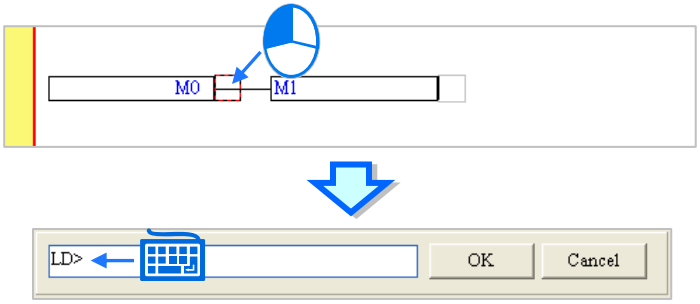
an instruction type in the **Classification** box.

❸ Users can select an applied instruction, a comparison contact, or a function block here.

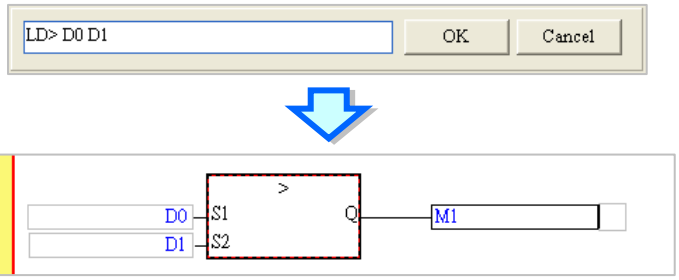
❹ If users unselect the **Auto Close** checkbox, they can insert instructions or function blocks continuously.

➢ **Method 3**

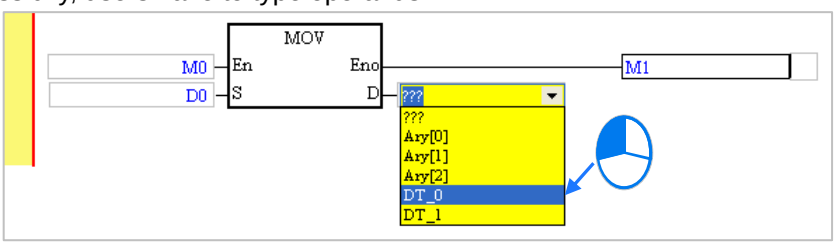
Click the position into which an applied instruction, a comparison contact, or a function block will be inserted, and type the applied instruction, the comparison contact instruction, or the function block definition. As soon as the applied instruction, the comparison contact instruction, or the function block definition is typed, a box which can be edited appears. After the typing of the applied instruction, the comparison contact instruction, or the function block definition is complete, users can press Enter on the keyboard or click **OK** at the right side of the box. (The applied instruction, the comparison contact instruction, and the function block definition are case-insensitive.)



If users want to insert a function block, they have to type the function block definition, and they can not type the operands specifed in the function block. If users want to insert a comparison contact or an applied instruction, they have to type the comparison contact instruciton or the applied instruction, and they can type the operands specified in the comparison contact instruciton or the applied instruction. The users do not have to type all the operands specified in the comparison contact instruciton or the applied instruction. The system will assign the operands typed to the corresponding boxes, and ??? will appear in the boxes where there are no operands. Besides, if the operands typed include a symbol which is not declared, the **Add Symbol** window will appear after the users press Enter on the keyboard. Please refer to programming manuals for more information about instructions.



(3) After an applied instruction, a comparison contact, or a function block is inserted successfully, users have to type operands.

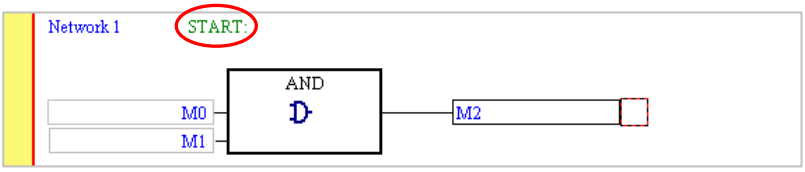


***. If the Auto-leading 'Add Symbol' Dialog checkbox in the Options window is selected, the Add Symbol window will appear automatically after users type a symbol which is not declared. Please refer to section 2.3.1 for more**

information.

9.2.8 Putting a Label

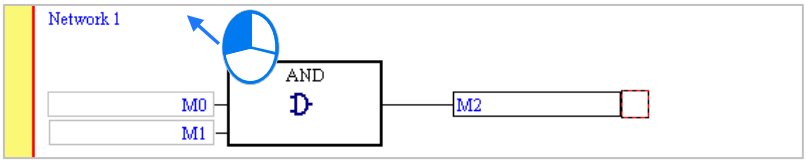
If users want to use a jump instruction, they have to put a label on the network to which the execution of the program jumps. A label is put at the right side of a network number. Besides, if the project created is a project for a DVP series PLC, the label put on a network must be a P device. The P devices in an AH500 series CPU module are automatically assigned by the system, and can not be used by users. If the project created is a project for an AH500 series CPU module, the label symbol users type must be a unique label. The label symbol users type in a project for an AH500 series CPU module does not need to be declared in a symbol table. Please refer to user manuals for more information about P devices.



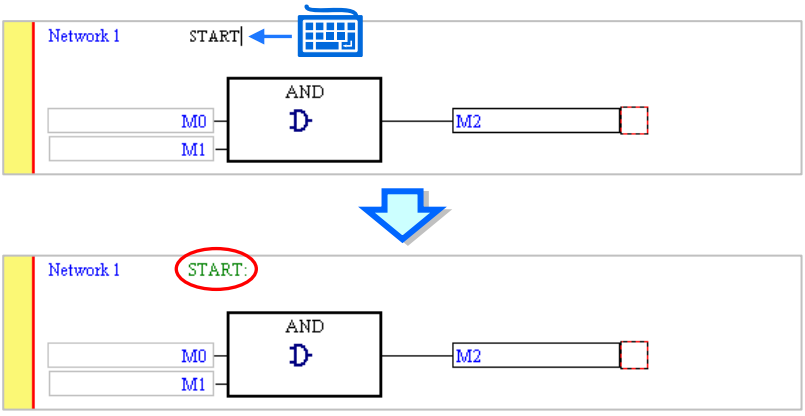
If users want to specify a label in an instruction, they do not have to type a colon in front of the label.

The steps of putting a label are as follows.

(1) Click the right side of the network number at which a label will be put.

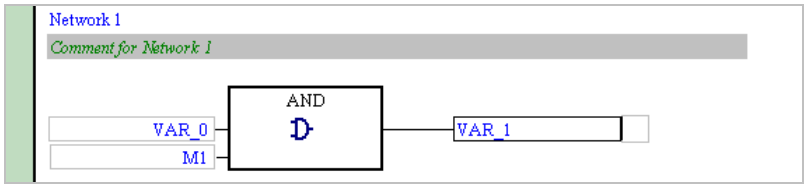


(2) If the project created is a project for a DVP series PLC, users have to type a P device number. If the project created is a project for an AH500 series CPU module, users have to type a label symbol. Press Enter on the keyboard after a P device number or a label symbol is typed. The system will automatically put a colon at the right side of the label.



9.2.9 Comments and Hints

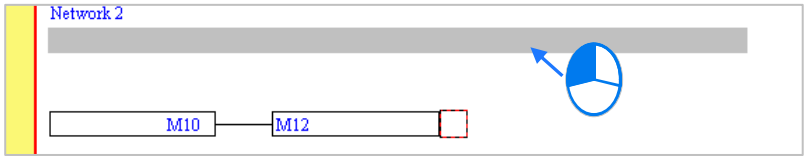
Users can make comments on the networks in a function block diagram in ISPSoft. They can type comments under the network numbers in a function block diagram in ISPSoft.



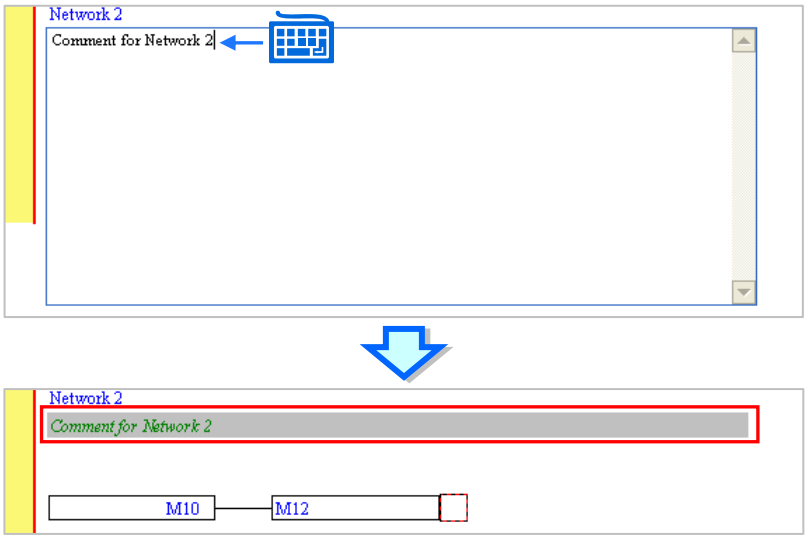
- **Editing the comment on a network**

(1) After users click on the toolbar, the comments on the networks will be displayed. If the users click on the toolbar again, the comments on the networks will be hidden.

(2) After the users click the position under a network number, a box will appear.

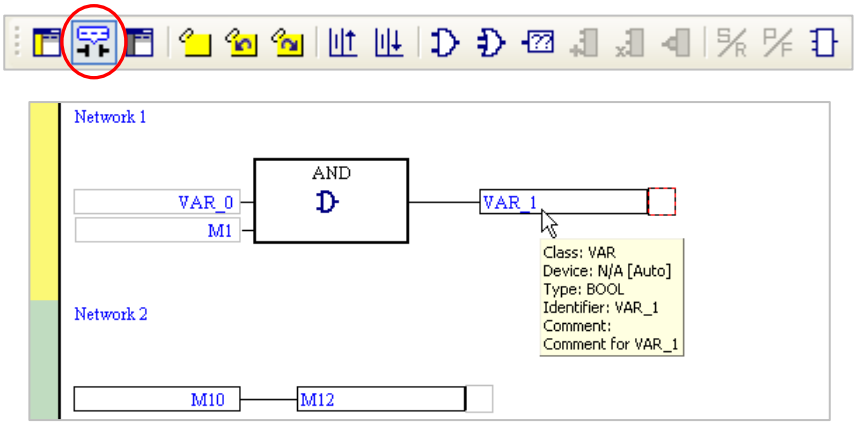


(3) Type a comment in the box. If the users want to start a new line of text at a specific point, they can press Shift+Enter on the keyboard. Press Enter on the keyboard after the editing is complete.

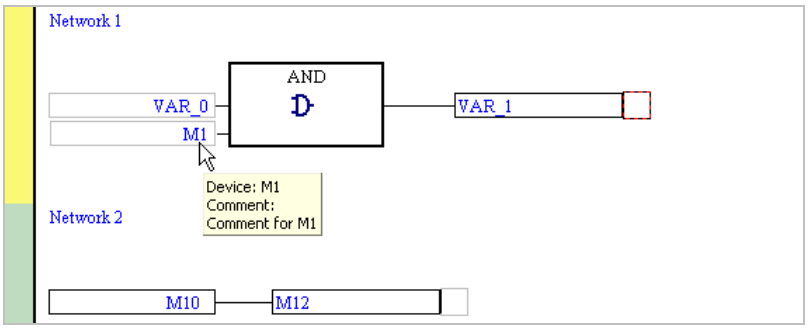


- **Hints related to the devices and the symbols in a function block diagram**

In a function block diagram, the comments on the devices and the comments on the symbols are not displayed. If users want to view the comment on a device or a symbol, they have to make sure that on the toolbar is clicked, and move the mouse cursor to the device or the symbol. After the mouse cursor stays at the device or the symbol for a while, the hints related to the device or the symbol will appear. The users can select hints in the **Options** window in ISPSoft. Please refer to section 2.3.1 for more information. If the users want to hide the hints, they can click again.



Besides, owing to fact that the devices and the symbols are different objects in ISPSoft, the hints related to the devices are different from the hints related to the symbols.



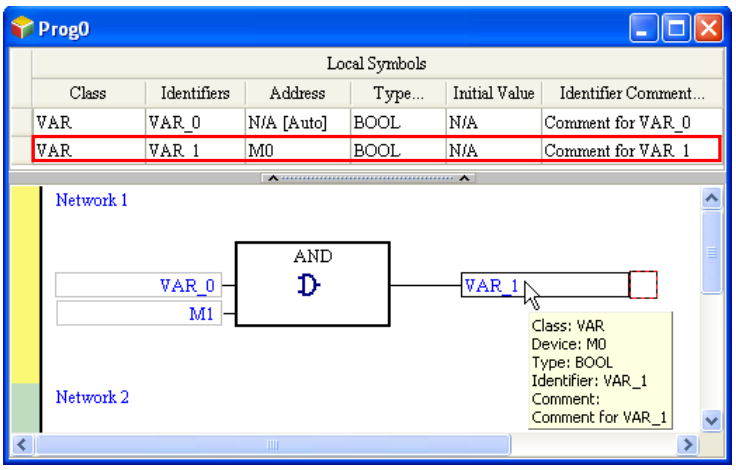
Additional remark

Users can make a comment on a device by means of the comment management tool provided by ISPSoft. Please refer to section 13.2 for more information.

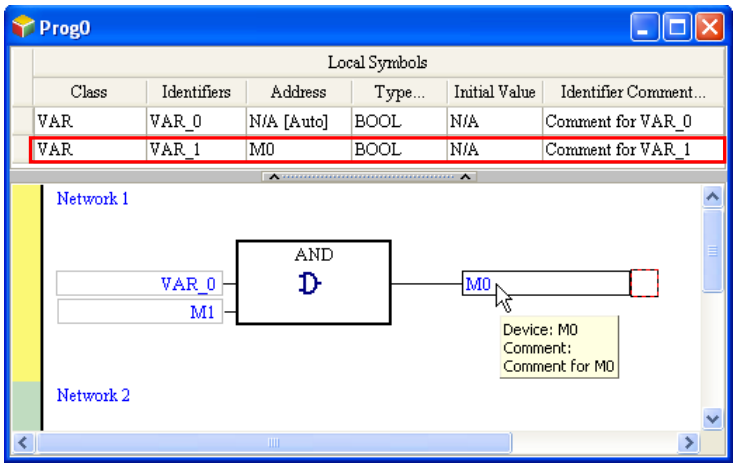
9.2.10 Symbol Mode and Address Mode

If a device address is assigned to a symbol declared in a function block diagram, users can change the mode in which the symbol is displayed by means of clicking on the toolbar. However, the symbols to which no devices are assigned and the devices to which no symbols are assigned will remain unchanged after on the toolbar is clicked.

If on the toolbar is not clicked, the identifiers of the symbols to which devices are assigned will be displayed, and the hints which appear will be the hints related to the symbols.



If on the toolbar is clicked, the devices to which symbols are assigned will be displayed, and the hints which appear will be the hints related to the devices.



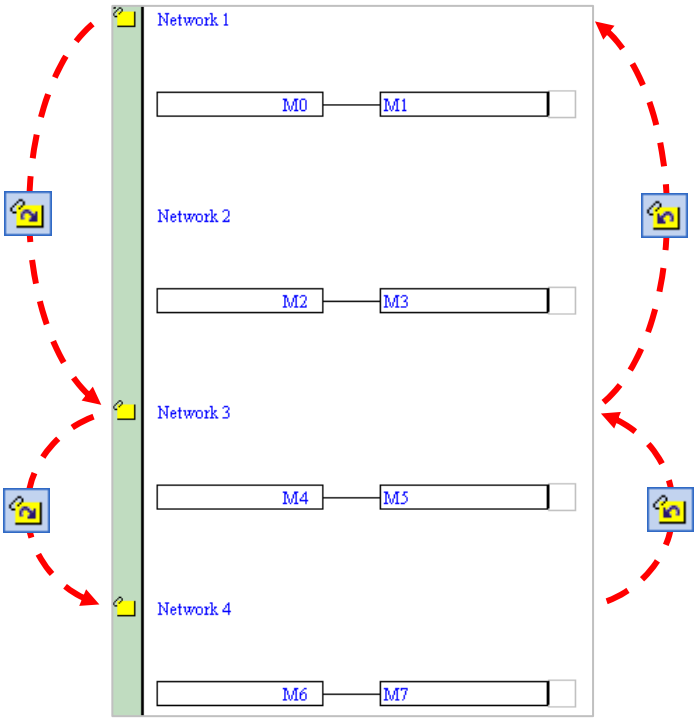
9.2.11 Bookmark

After networks are bookmarked, users can find or go to the networks easily.

(1) After users click on the toolbar, a bookmark will be inserted in the network selected. If the users want to delete the bookmark, they can select the network, and click on the toolbar again.

(2) If there are bookmarks in a program, users can go to the next network which is bookmarked by clicking on the toolbar, and go to the previous network which is bookmarked by clicking

on the toolbar.



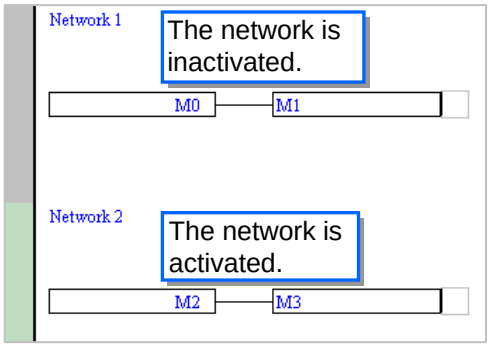
Additional remark

After users point to **Bookmarks** on the **Edit** menu, they can select **Toggle Bookmark**, **Goto the Next Bookmark**, **Goto the Previous Bookmark**, or **Remove All Bookmarks**. If the users click **Remove All Bookmarks**, all the bookmarks in the present program editing window will be deleted.

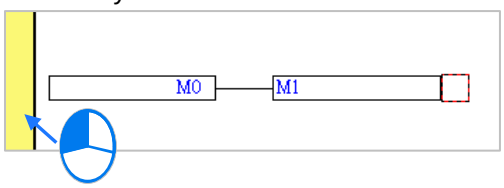
9.2.12 Activating / Inactivating a Network

If a network is inactivated, the compiling of the program will skip the network. When users test or debug a program, they can inactivate some part of the program temporarily by means of this function.

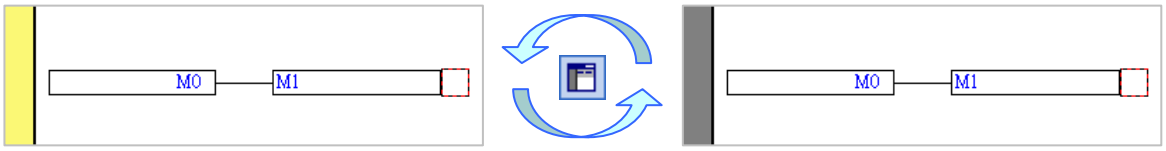
The color at the left side of a network indicates whether the network is activated or inactivated. Users can select colors which will be shown at the left sides of the networks in a ladder diagram in the **Options** window in ISPSoft. Please refer to section 2.3.1 for more information.



If users want to activate or inactivate a network, they have to select the network first. Users can select several networks simultaneously.



Right-click the network selected, and click **Activate/InactivateNetwork** on the context menu. The network selected will be activated/inactivated. Users can also activate/inactivate the network selected by clicking **Activate/InactivateNetwork** on the **Edit** menu, or by clicking on the toolbar.



If users want to activate all the networks which are inactivated in the present program editing window, they can click **Activate All Networks** on the **Edit** menu.

MEMO

Chapter 10 Instruction List

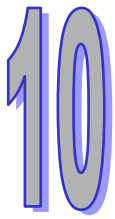


Table of Contents

10.1 Introduction of Instruction Lists 10-2
10.1.1 Structure of an Instruction List 10-2
10.1.2 Calling a Function Block 10-3
10.1.3 Important Points About Instruction Lists 10-4
10.2 Creating an Instruction List in ISPSoft 10-5
10.2.1 Editing Environment.....10-5
10.2.2 Creating an Instruction List 10-6
10.2.3 Inserting Applied Instructions and Function Blocks.....10-7
10.2.4 Bookmark 10-9
10.2.4.1 Adding Bookmarks 10-9
10.2.4.2 Removing Bookmarks 10-10
10.2.4.3 Going to a Bookmark.....10-11

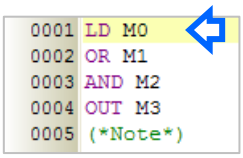
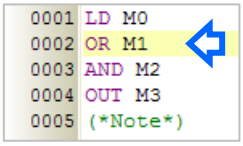
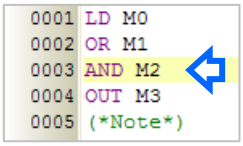
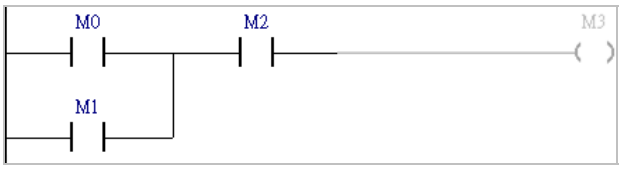
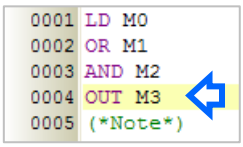
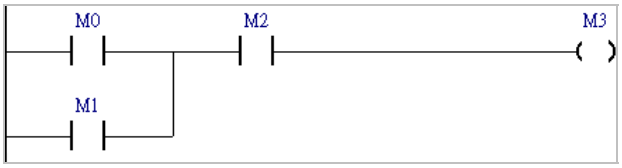
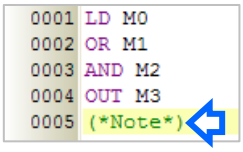
10.1 Introduction of Instruction Lists

An instruction list is one of the five languages supported by the IEC 61131-3 standard. It is a low level language and resembles an assembly language. In earlier times, users entered control instructions into a PLC through a programming panel, and the programming language used was an instruction list.

10.1.1 Structure of an Instruction List

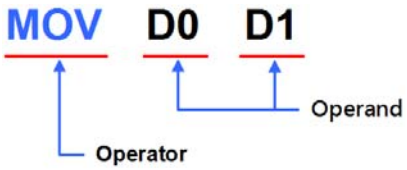
An instruction list is composed of statements. Every statement represents an action. A task which can be accomplished by a complete ladder diagram can be accomplished by an instruction list composed of many instructions.

An example of an instruction list is shown below. There are five lines of program code. The ladder diagram corresponding to the instruction list is shown at the right side of the instruction list.

LD M0: Loading the state of M0

OR M1: The logical OR operation is performed on M1.

AND M2: The logical AND operation is performed on M2.
 
OUT M3: M3 The result of the operation is sent to M3.
 
(*Note*): It is a comment on the program. When the program is compiled, this line is skipped automatically.
 No action

Every statement on an instruction list is composed of an operator and operands. An operator is a function representing an operation, and an operand is the object of an operation. An operand can be a device, a symbol, or a constant. Besides, the operands for an operator are separated by spaces. The operator MOV in the example below denotes that an action of transferring data will be executed.

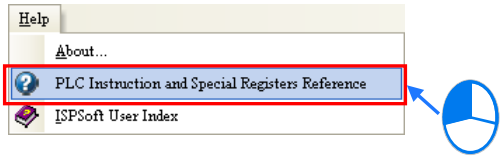
The operands D0 and D1 are the source device and the destination device which take part in the transfer.



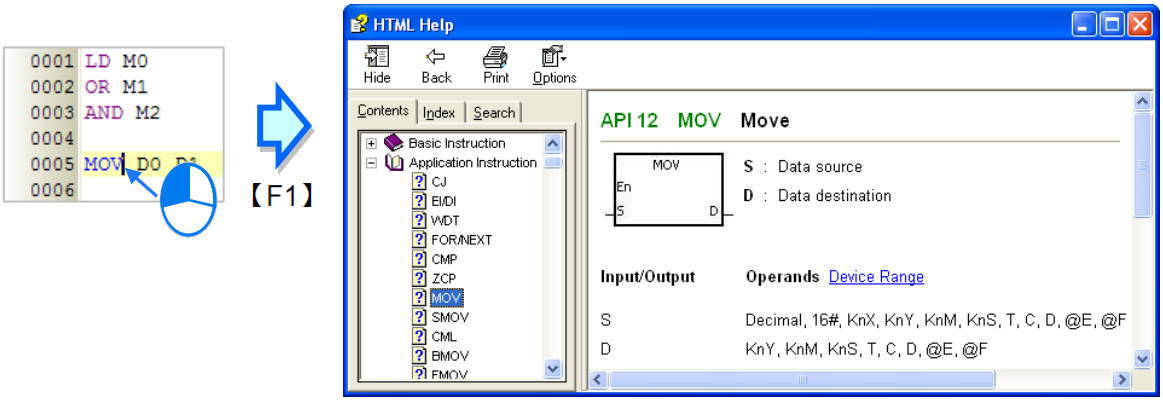
On an instruction list, the words between **(*** and ***)** are regarded as comments. When a program is compiled, the system automatically skips the words between **(*** and ***)**. Besides, as long as the structure of an instruction list is not destroyed, comments can be put anywhere on the instruction list. However, users have to be careful about the readability of the instruction list. The comments in the example below are legal.

```
0001 LD M0 (*Note*)
0002 (*Note*)  OR M1
0003 AND   (*Note*)  M2
0004 OUT M3
0005 (*Note 1
0006   NOte 2
0007   Note 3*)
```

Instructions are divided into basic instructions and advanced instructions. Please refer to programming manuals for more information about the usage of instructions. If users want to check the usage of an instruction, they can click **PLC Instruction and Special Registers Reference** on the **Help** menu.



Besides, if users press F1 after they click an instruction, the usage of the instruction will be displayed in the window which appears.

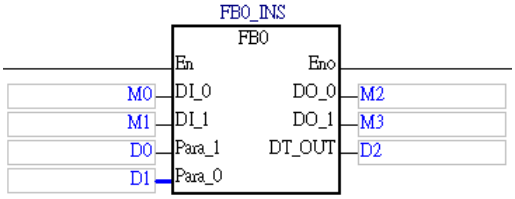


10.1.2 Calling a Function Block

The important points about calling a function block on an instruction list are listed below.

```
CAL  Name of FB instance  (  Input pin 1  :=  Input operand 1  ,
                           Input pin 2  :=  Input operand 2  ,
                              .........
                           Output pin1  =>  Output operand 1  ,
                           Output pin 2  =>  Output operand 2  ,
                              .........
                           Output pin n  =>  Output operand n  )
```

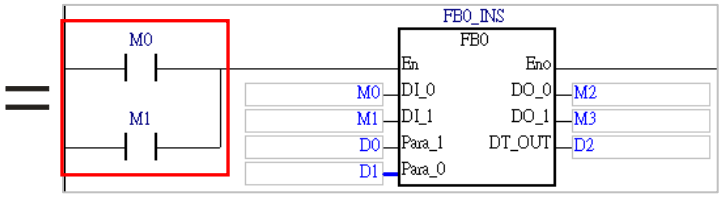
- CAL, the function block instance and the left parenthesis must be in the same line while the other parts can be in different lines.
- The input/output pins of the function block and the corresponding operands must be put in parentheses. A pin and the corresponding operand form a group. The groups are separated by commas. The data types of the operands are specified in the function block definition.
- The function block after CAL is a function block instance rather than a function block definition, that is, the symbol after CAL is a symbol whose data type is a function block. Please refer to chapter 7 for more information about function block definitions and function block instances.
- The assignment mark for an input pin is **:=**, and the assignment mark for an output pin is **=>**.
- The pins and the operands are arranged according to the definition of the function block. The arrangement can not be changed. The En pin and the Eno pin are excluded. The input pins are listed above all the output pins. A pin of the VAR_IN_OUT class is an input pin. Please refer to the ladder diagram and the instruction list below for more information.




```
0003 CAL FB0_INS ( DI_0 := M0 ,
0004               DI_1 := M1 ,
0005               Para_1 := D0 ,
0006               Para_0 := D1 ,
0007               DO_0 => M2 ,
0008               DO_1 => M3 ,
0009               DT_OUT => D2  )
```

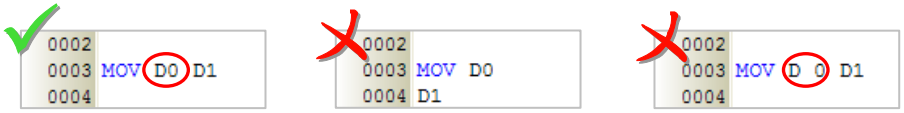
- The state of the En pin of the function block is not specified in the instruction CAL. When the instruction is executed, the result of the logical operation before the instruction determines whether the function block will be enabled.

```
0001 LD M0
0002 OR M1
0003
0004 CAL FB0_INS ( DI_0 := M0 ,
0005               DI_1 := M1 ,
0006               Para_1 := D0 ,
0007               Para_0 := D1 ,
0008               DO_0 => M2 ,
0009               DO_1 => M3 ,
0010               DT_OUT => D2  )
0011
```



10.1.3 Important Points About Instruction Lists

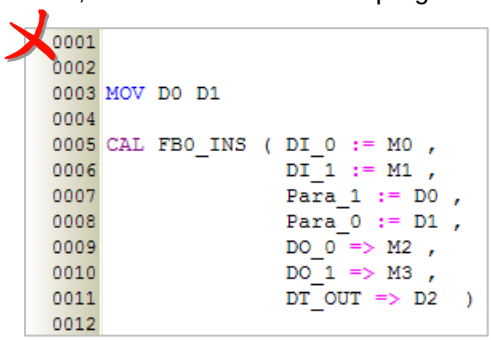
- Every line on an instruction list is regarded as an instruction except the instruction CAL, and therefore an operator and the operands for the operator can not be in different lines. If an operator and the operands for the operator are in two lines, these two lines will be regarded as two incomplete instructions. Besides, no spaces can be inserted into the instruction names, the function block names, and the operand names on an instruction list. Otherwise, the instructions, the function blocks, and the operands will not be recognized if the program is compiled.

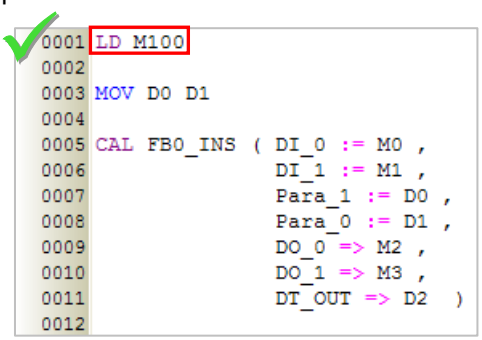


- As long as the format of an instruction is not destroyed, leaving blank spaces and starting new lines are allowed.
- Fullwidth characters and halfwidth characters in ISPSoft are regarded as different characters. Please avoid using fullwidth characters lest they should be identified incorrectly.
- Instruction lists are case-insensitive. “OUT” and “Out” are considered to be the equivalents of “out”.
- If users want to use constants in a program created by means of an instruction list in ISPSoft, the constants must be represented in the following ways.
  - Decimal value: 23456 (A value which is not preceded by any mark will be regarded as a decimal value.)
  - Hexadecimal value: 16#5BA0 (A value is preceded by 16#.)
  - Octal value: 8#55640 (A value is preceded by 8#.)
  - Binary value: 2#101101110100000 (A value is preceded by 2#.)
  - String: “XYZ” (Characters are put in double quotes.)
  - Boolean value:

    AH500: SM400 (normally-open contact), or SM401 (normally-closed contact) is used.

    DVP: M1000 (normally-open contact), or M1001 (normally-closed contact) is used.
- If users want to use a symbol whose data type is ARRAY, the expression format is **Identifier[Index]**. Generally speaking, an index is a decimal constant. The indexes on an instruction list can not be symbols. The minimum index value for a symbol whose data type is ARRAY must be 0, and the maximum index value for a symbol whose data type is ARRAY can not be larger than the number of elements in the array.
- If a function block or an applied instruction is not preceded by the instruction LD or a logical operation, an error will occur if the program is compiled.



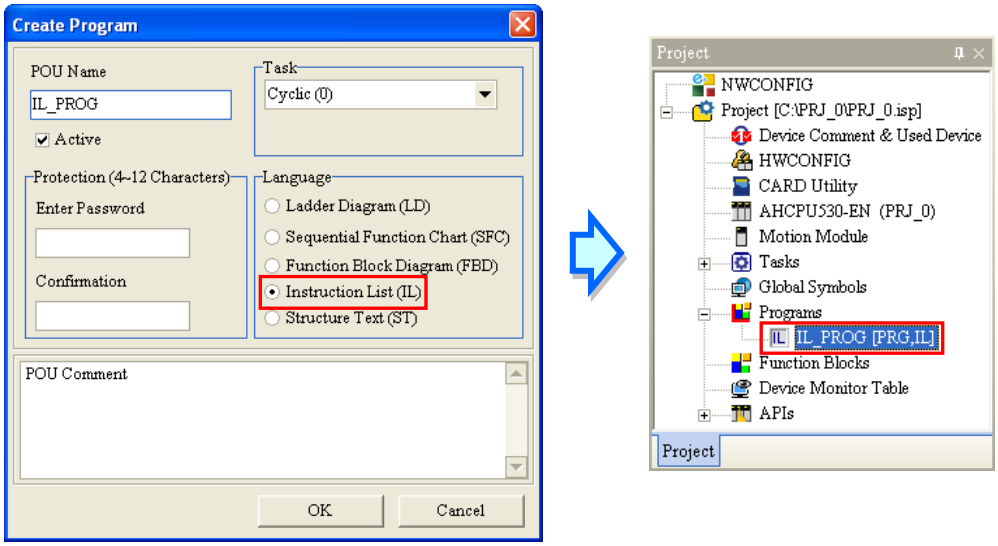


- Although there is no limit on the number of lines, users still have to consider the capacity of the memory in the PLC.
- The sections of an instruction list in ISPSoft can be copied/cut/pasted. Users can copy the text in a file edited with a text editor into an instruction list in ISPSoft.

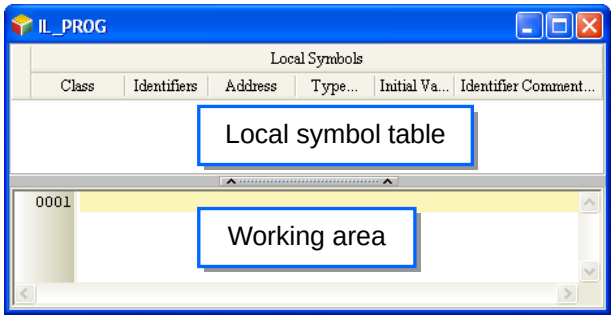
10.2 Creating an Instruction List in ISPSoft

10.2.1 Editing Environment

Select the **Instruction List (IL)** option button in the **Language** section in the **Create Program** window. Please refer to section 5.4.1 for more information.



The environment in which an instruction list can be edited is shown below. The table at the upper part of the window is a local symbol table, and the area at the lower part of the window is a working area. There are line numbers at the left side of the working area. Users can set the way in which the line numbers are displayed. Please refer to section 2.3.1 for more information. After a program editing window in which an instruction list can be created is opened, the corresponding toolbar will appear in the ISPSoft window. The functions are described below.

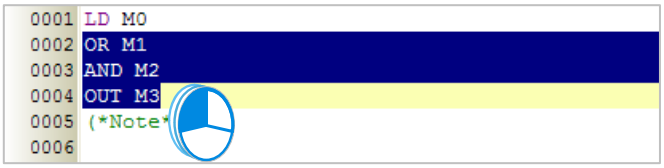


Icon	Keyboard shortcut	Function
	Shift+Ctrl+B	Adding a bookmark to the network selected or deleting the bookmark from the network selected
	Shift+Ctrl+P	Going to the previous bookmarked position
	Shift+Ctrl+N	Going to the next bookmarked position
	Shift+Ctrl+U	Inserting an instruction or a function block

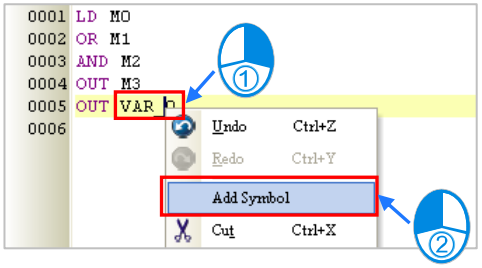
10.2.2 Creating an Instruction List

The way in which an instruction list file is edited is similar to the way in which a text file is edited with a general text editor. If users want to create an instruction list, they can type or modify text in the working area. If the users want to start a new line of text at a specific point, they can press Enter on the keyboard.

If users want to select any amount of text, they can click where they want to begin the selection, hold down the left mouse button, and drag the pointer over the text that they want to select. The users can also click at the start of the selection, scroll to the end of the selection, and hold down Shift on the keyboard while they click where they want the selection to end.



If users type a symbol which has not been declared, they can click the symbol, click the right mouse button, and click **Add Symbol** on the context menu to open the **Add Symbol** window.

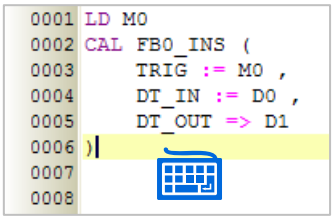


10.2.3 Inserting Applied Instructions and Function Blocks

Users can insert an applied instruction or a function block in an instruction list in ISPSoft in one of the three ways described below.

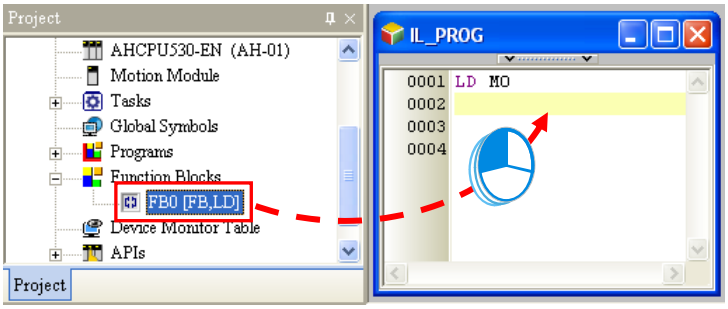
- **Method 1**

  An applied instruction or a function block is inserted according to the format of the applied instruction or the format of the function block.

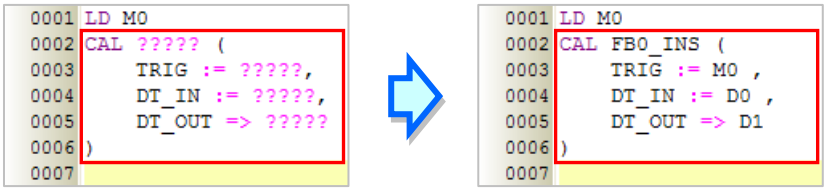


- **Method 2**

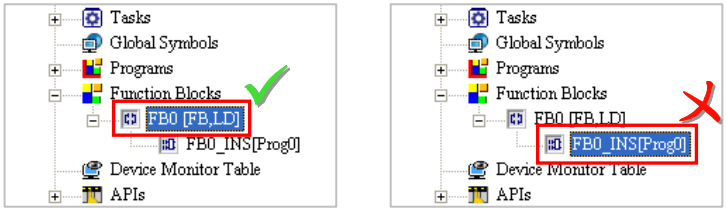
  Unfold the **APIs** section or the **Function Blocks** section in the project management area, and find the item which will be inserted. Select the item, and drag it to the position in which it will be inserted.



After the applied instruction or the function block is inserted, users have to type the corresponding operands according to the format of the applied instruction or the format of the function block.

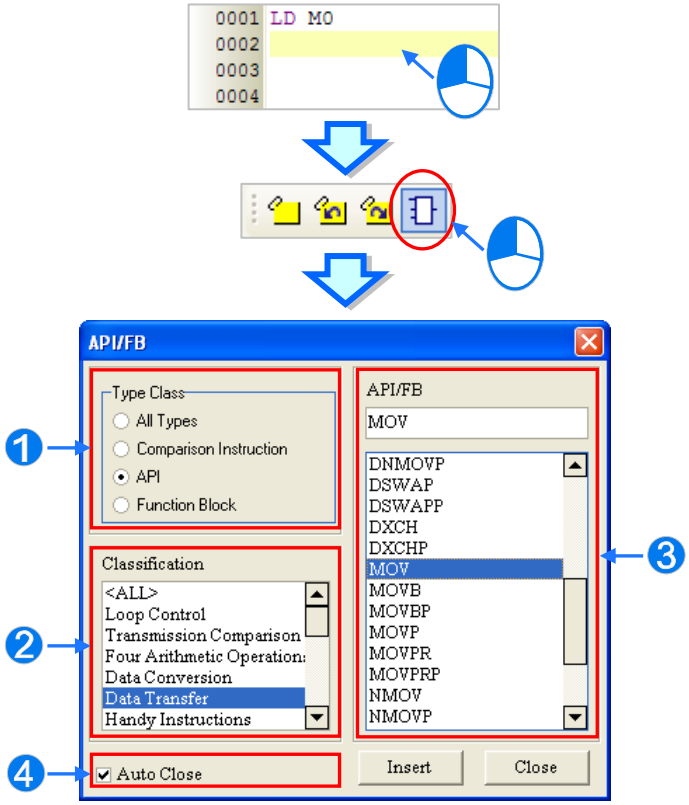


Function block definitions can be dragged, but function block instances can not be dragged.



- **Method 3**

  Click the position in which an applied instruction or a function block will be inserted, click on the toolbar, select the item which will be inserted in the **API/FB** window, and click **Insert**.



❶ Users can select the **All Types** option button, the **Comparison Instruction** option button, the **API** option button, or the **Function Block** option button.

❷ If users select the **API** option button in the **Type Class** section, they have to select an instruction type in the **Classification** box.

❸ Users can select an applied instruction, a comparison contact, or a function block here.

❹ If users unselect the **Auto Close** checkbox, they can insert instructions or function blocks continuously.

After the applied instruction or the function block is inserted, users have to type the corresponding operands according to the format of the applied instruction or the format of the function block.

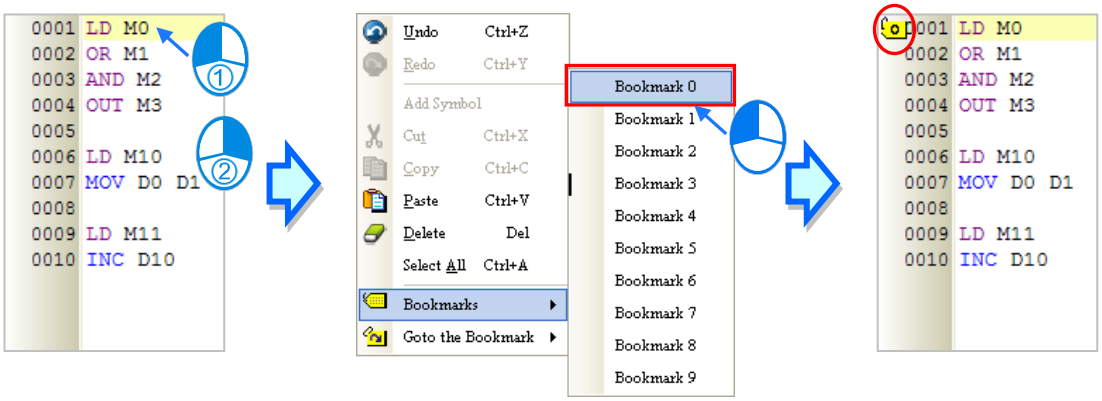
10.2.4 Bookmark

After lines are bookmarked, users can find or go to the lines easily. Users can add ten bookmarks at most to an instruction list.

10.2.4.1 Adding Bookmarks

- **Method 1**

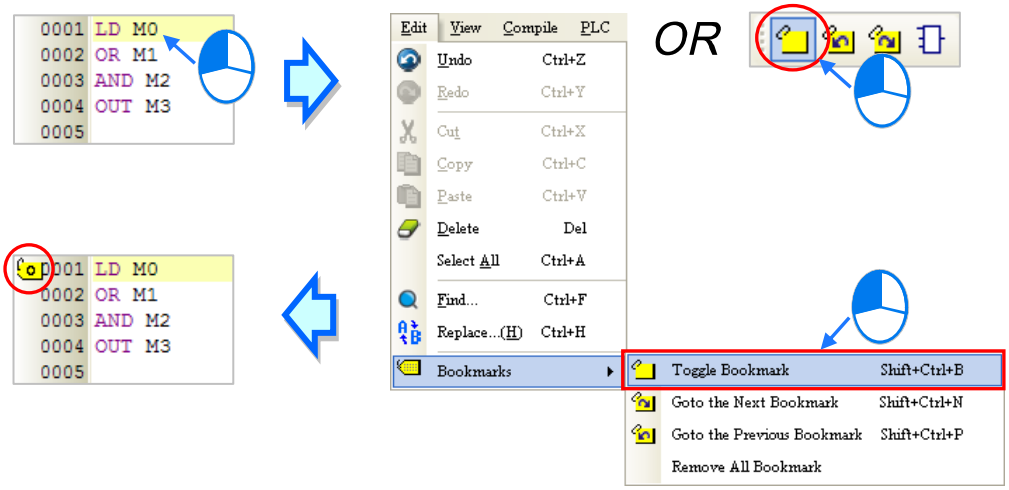
  If users want to add a bookmark to a line, they can select the line, click the right mouse button, click **Bookmarks** on the context menu, and select a bookmark number. If the bookmark number selected has been added to another line, the bookmark number will be moved to the line selected presently.



- **Method 2**

  If users want to add a bookmark to a line, they can click the line, and click on the toolbar.

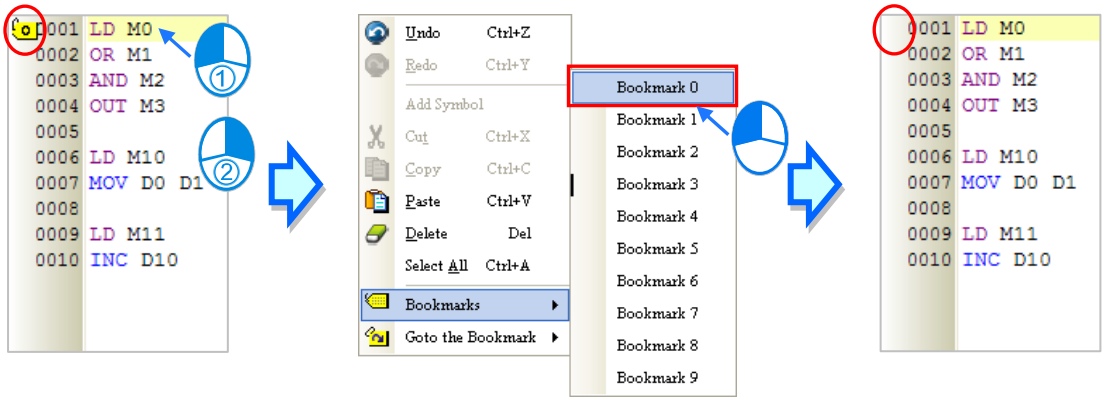
  The users can also add a bookmark to a line by clicking the line, clicking the **Edit** menu, pointing to **Bookmarks**, and clicking **Toggle Bookmark**. The bookmarks added in this way are numbered in sequence. The users can not specify bookmark numbers.



10.2.4.2 Removing Bookmarks

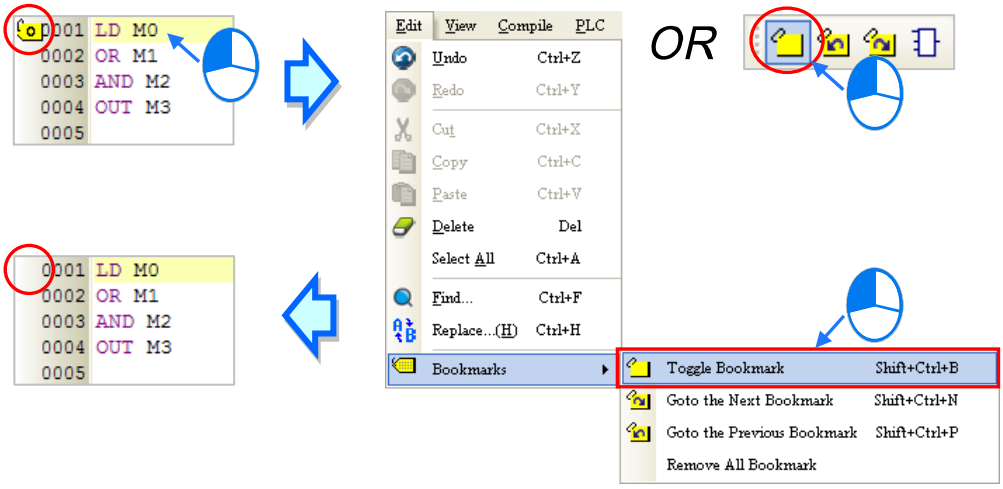
- **Method 1**

  If users want to remove the bookmark from a line, they can select the line, click the right mouse button, click **Bookmarks** on the context menu, and click the bookmark number corresponding to the bookmark added to the line. If the bookmark number clicked does not correspond to the bookmark added to the line, the bookmark number added to the line will become the bookmark number clicked. Besides, if the bookmark number clicked has been added to another line, the bookmark number will be moved to the line selected presently.



- **Method 2**

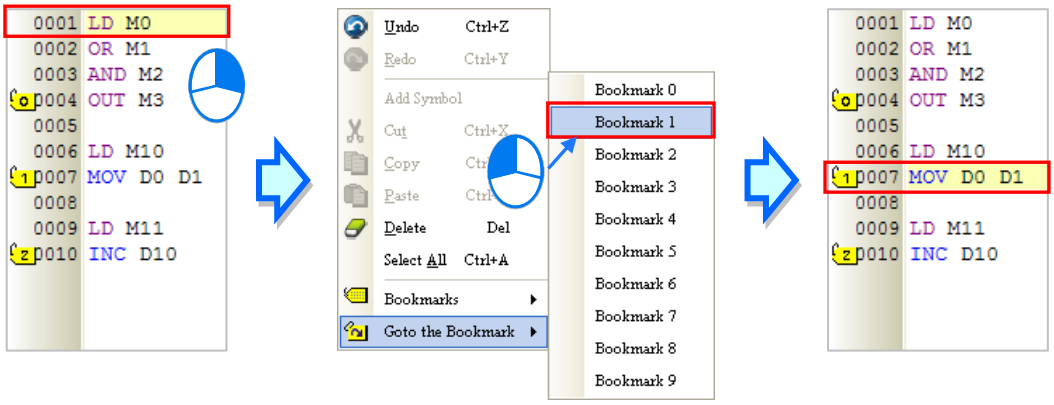
  If users want to remove the bookmark from a line, they can click the line, and click on the toolbar. The users can also remove the bookmark from a line by clicking the line, clicking the **Edit** menu, pointing to **Bookmarks**, and clicking **Toggle Bookmark**. If the users click the **Edit** menu, point to **Bookmarks**, and click **Remove All Bookmarks**, all the bookmarks in the present program editing window will be deleted.



10.2.4.3 Going to a Bookmark

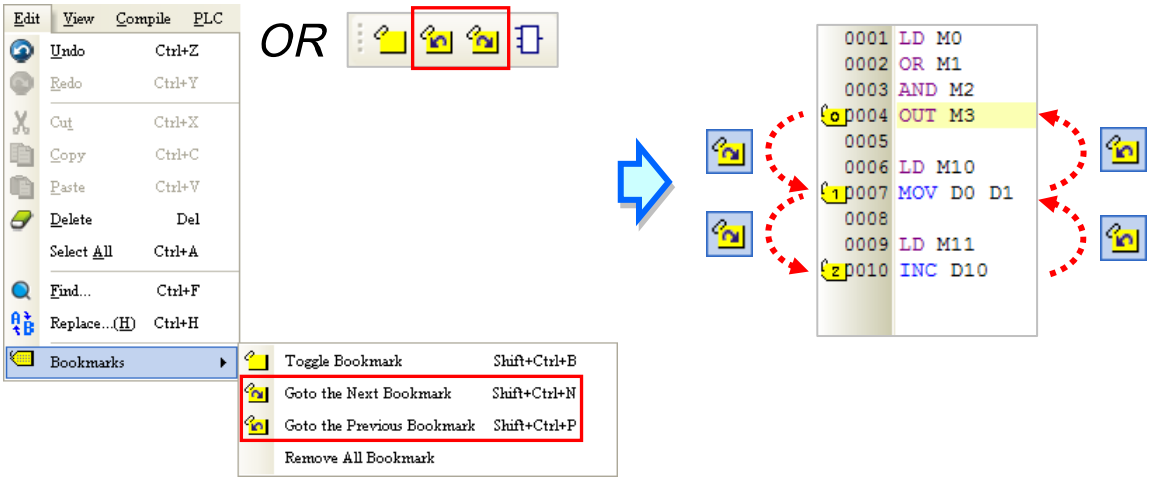
- **Method 1**

  Right-click the working area, point to **Goto the Bookmark** on the context menu, and click a bookmark number.



- **Method 2**

  If there are bookmarks in a program, users can go to the next line which is bookmarked by clicking on the toolbar, and go to the previous line which is bookmarked by clicking on the toolbar. The users can also go to the next/previous line which is bookmarked by clicking the **Edit** menu, pointing to **Bookmarks**, and clicking **Goto the Next Bookmark/Goto the previous Bookmark**.



MEMO

Chapter 11 Structured Text

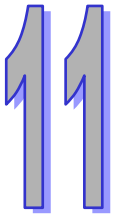


Table of Contents

11.1 Knowing Structured Texts 11-2
11.1.1 Basic Structure of a Structured Text 11-2
11.1.2 Statement 11-2
11.1.3 Expression 11-3
11.1.4 Operand and Operator 11-4
11.1.5 Keyword and Comment 11-5
11.1.6 Using a Symbol Whose Data Type is ARRAY in a Structured Text 11-6
11.1.7 Important Points About Structured Texts 11-6
11.2 Structure of a Statement 11-7
11.2.1 Assignment Structure—:= 11-7
11.2.2 Conditional Structure—IF 11-10
11.2.3 Conditional Structure—CASE 11-12
11.2.4 Loop Structure—REPEAT 11-14
11.2.5 Loop Structure—WHILE 11-15
11.2.6 Loop Structure—FOR 11-16
11.2.7 Applied Instruction Structure 11-17
11.2.8 Function Block Structure 11-18
11.2.9 Blank Statement 11-19
11.2.10 RETURN Statement 11-19
11.2.11 EXIT Statement 11-20
11.3 Creating a Structured Text in ISPSoft 11-21
11.3.1 Editing Environment 11-21
11.3.2 Creating a Structured Text 11-22
11.3.3 Inserting Applied Instructions and Function Blocks 11-23
11.3.4 Bookmark 11-24
11.4 Example of a Structured Text 11-25
11.4.1 Explanation 11-25
11.4.2 Planning Hardware 11-25
11.4.3 Planning a Program 11-25
11.4.4 Creating a Program 11-25

11.1 Knowing Structured Texts

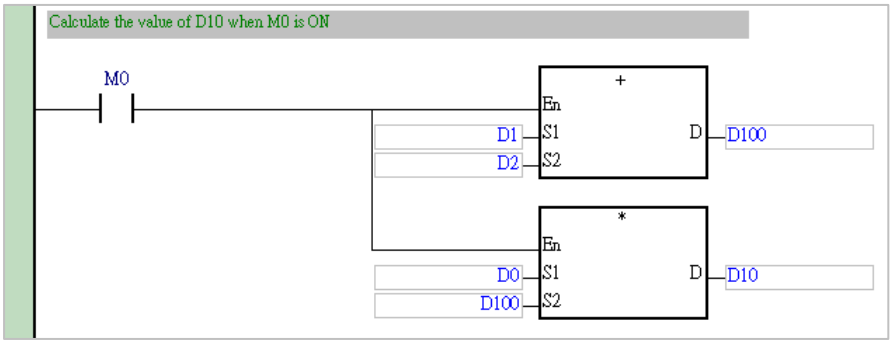
A structured text is a high level language that is syntactically resembles C or Pascal. It is composed of text. Compared with instruction lists, structured texts are more readable. A structured text is very flexible and intuitive for writing control algorithms. People who are trained in programming languages often find it the easiest language to use for programming control logic. Besides, structured texts provide a familiar programming environment for programmers who are familiar with the C language.

11.1.1 Basic Structure of a Structured Text

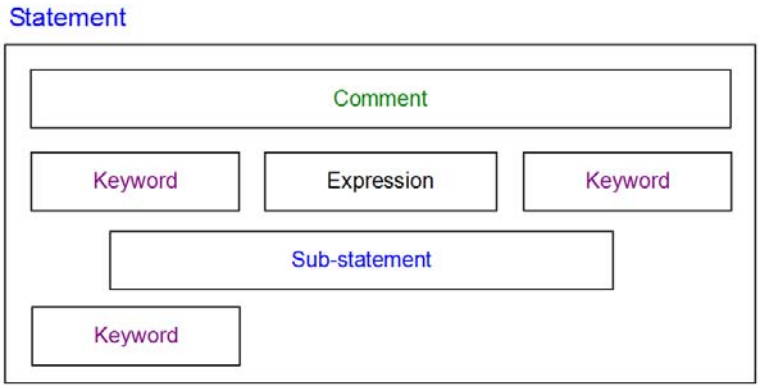
The structured text shown below is a basic structured text. It has the same function as the ladder diagram under it. The structure of the structured text is similar to the syntax of human language. A structured text is composed of statements.

```
0001 (*Calculate the value of D10 when M0 is ON*)
0002 IF M0 THEN
0003    D10 := D0*(D1+D2) ;
0004 END_IF ;
0005
```

=



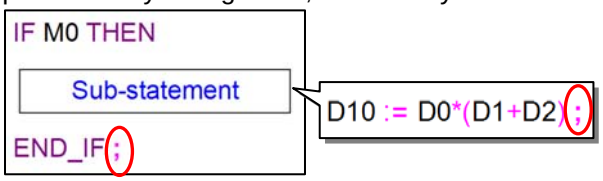
The structure of the structured text above is shown below.



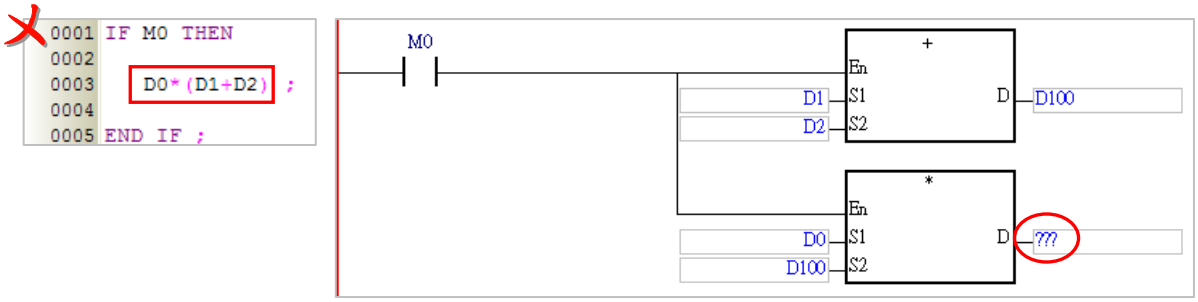
*** In a structured text, "**: = **" means "to assign", and "** = **" means "to be equal to".**

11.1.2 Statement

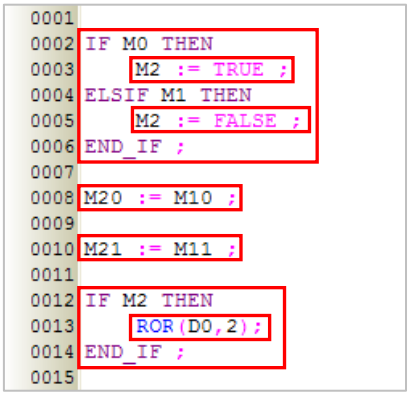
The statement is a basic unit. It represents a complete task which can be executed. A complete statement may not be represented by a single line, but it always ends with a semicolon.



A complete statement corresponds to a network in a ladder diagram. It represents a task. Take the sub-statement shown above for example. The value of the arithmetic expression D0* (D1+D2) is calculated, and the result of the operation is stored in D10. The arithmetic expression in the red frame in the figure below is not a legal statement. It is an expression. It is not a concrete task. It is like the ladder diagram shown below. The ladder diagram lacks an output operand. It is not a complete program which can be executed.



A structured text is composed of statements, and a statement is composed of elements. An example of a structured text is shown below. The complete statements in the structured text are in the red frames.



*** In a structured text, "**:= **" means "to assign", and "** = **" means "to be equal to".**

11.1.3 Expression

An expression is an important element of a statement. It represents a value. An expression can be an arithmetic expression, a constant, a symbol, or a device. Examples are listed below.

- M0 & M1 (Boolean value)

  The logical operator **&** takes **M0** and **M1**, and performs the logical **&** operation on **M0** and **M1**. The result of the operation is a Boolean value.
- M0 = FALSE (Boolean value)

  The expression is a conditional expression. If M0 is ON, the Boolean value is TRUE. If M0 is OFF, the Boolean value is TRUE.
- M0 (Boolean value)

  If M0 is ON, the Boolean value is TRUE. If M0 is OFF, the Boolean value is FALSE.
- D1 + D2 (Value)

  The value of the arithmetic expression D1+D2 is calculated.
- D0 (Value)

  The value represented by the expression if the value stored in D0.
- D2 = D0 + D1 (Boolean value)

  If the value of the arithmetic expression D0+D1 is equal to the value stored in D2, the Boolean value is TRUE. If the value of the arithmetic expression D0+D1 is not equal to the value stored in D2, the Boolean value is FALSE.

- D2 := D0 + D1; (Statement)
  This is a statement rather than an expression. The value of the arithmetic expression D0+D1 is calculated, and the result of the operation is stored in D2. This statement is composed of the expression **D2** and the expression **D0+D1**.

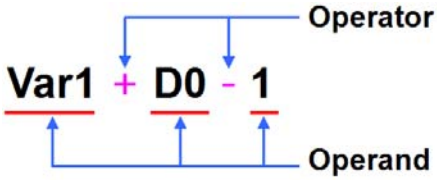
Please refer to the following example. The program is mainly composed of two IF statements. When the first IF statement is executed, the system evaluates the expression M0. If M0 is ON, the value of the arithmetic expression D21+D22 will be calculated, and the result of the operation will be stored in D20. When the second IF statement is executed, the system judges whether the value of the arithmetic expression D0* (D1+D2) is equal to the value stored in D20. If the value of the arithmetic expression D0* (D1+D2) is equal to the value stored in D20, the Boolean value is TRUE, and the value of M1 is TRUE. If the value of the arithmetic expression D0* (D1+D2) is not equal to the value stored in D20, the value of M1 is FALSE.

```
0001 IF M0 THEN
0002     D20 := D21 + D22 ;
0003 END_IF;
0004
0005 IF D20 = D0*(D1+D2) THEN
0006     M1 := TRUE ;
0007 ELSE
0008     M1 := FALSE ;
0009 END_IF;
```

11.1.4 Operand and Operator

Operands and operators are basic elements of an expression. An operand is the object of an operation, and an operator is a symbol or function representing an operation. For example, in the expression D0 + D1, D0 and D1 are operands, and + is an operator.

An expression can be composed of operands and operators. It can also be composed of a single operand. The operands in an expression can be devices, symbols, or constants.



Rules of precedence affect which values form operands for which operators. If the operators in an expression share the same precedence, the order of the operation is carried out from left to right. The operands which can be used in a structured text in ISPSoft are listed below.

Symbol	Function	Data format		Example of an expression		Precedence
		Operand	Result of an operation (Value of an expression)	Expression	Value	
( )	The parenthetical part of an expression is calculated first.	Unspecified	Unspecified	(D0 + 6)* 3	33	Highest
**	Exponentiation	Floating-point number	Floating-point number	2.0 ** 5.0	3.2E+1	
-	Negative sign	Signed number	Signed number	-(D0 + 3)	-8	
NOT	Logical negation	Boolean value	Boolean value	NOT M0	TRUE	
*	Multiplication	Any value	Any value	D0 * 3	15	
/	Division	Any value	Any value	15 / D0	3	
MOD	Remainder	Integer	Integer	D0 MOD 3	2	
+ , -	Addition, subtraction	Any value	Any value	D0 + 3	8	
< , >,<=, >=	Comparison	Any value	Boolean value	D0 > 2	TRUE	
=, <>	Equal to, not equal to	Any value	Boolean value	D0 <> 2	TRUE	
		Boolean value		M0 = TRUE	FALSE	
AND, &	Logical AND operation	Boolean value	Boolean value	M0 & M1	FALSE	
XOR	Logical exclusive OR operation	Boolean value	Boolean value	M0 XOR M1	TRUE	
OR	Logical inclusive OR operation	Boolean value	Boolean value	M0 OR M1	TRUE	Lowest

***1. In the table above, the default value of M0 is FALSE, the default value of M1 is TRUE, and the default value of D0 is 5.**

***2. The operands for an exponentiation operator must be constants or floating-point symbols, and must not be devices.**

11.1.5 Keyword and Comment

In computer programming, a keyword is a word which has a special meaning to the programming language. For example, TRUE and FALSE in a program represent Boolean values, and the IF statements in the example above represent tasks which will be executed. Besides, applied instructions are regarded as keywords. In order to prevent a program from being compiled incorrectly, the words which are regarded as keywords in the program can not be declared to be symbols. However, the words which are regarded as keywords in a program can be parts of the symbols in the program. For example, "**FIFO**" and "**_IF**" are allowed.

In a structured text, the words between **(*** and ***)** are regarded as comments. When a program is compiled, the system automatically skips the words between **(*** and ***)** in the program. Besides, as long as the structure of a structured text is not destroyed, comments can be put anywhere in the structured text. However, users have to be careful about the readability of the structured text.

Please refer to the two examples below. The comments in the left example are legal. The comment in the right example divides the keyword “END_IF” into two parts, and therefore an error will occur if the program is compiled.


```
0001 (*Note*)
0002
0003 IF (*Note*) M0 THEN
0004
0005     D10(*Note*) := D10*(D1+D2) ;
0006
0007 END_IF(*Note*);(*Note*)
0008
0009 (*Note*)
```

```
0001
0002 IF M0 THEN
0003
0004     D10 := D10*(D1+D2) ;
0005
0006 END(*Note*)_IF ;
0007
0008
0009
```

11.1.6 Using a Symbol Whose Data Type is ARRAY in a Structured Text

If users want to use a symbol whose data type is ARRAY in a structured text, the expression format is **Identifier[Index]**. The index is a constant or a symbol. However, only the indexes in a project for an AH500 series CPU module can be symbols. The minimum index value must be 0, whether the index is a constant or a symbol. The maximum index value can not be larger than or equal to the number of elements in the array. For example, if the number of elements in an array is 10, the index value must be within the range between 0 and 9. Please refer to section 6.2.3 for more information.

```
0001 IF M0 THEN
0002
0003 Ary_A[0] := Ary_B[IDX] ;
0004
0005 END_IF ;
```

Besides, users have to pay attention to the following points when they use a symbol whose data type is ARRAY.

- Users can only manipulate a single element in an array. A symbol which represents an array can not be used as an operand.
- The arrays that ISPSoft supports are one-dimensional arrays.
- If an index is a symbol, and the value of the symbol is not within a legal range, the PLC will continue running and no error will occur. However, the data taking part in the operation may be incorrect.
- If the index for a symbol is constant, the symbol can be modified by an index register. If the index for a symbol is a symbol, the symbol can not be modified by an index register. Please refer to section 6.1.5 for more information.

11.1.7 Important Points About Structured Texts

- If the structure of the keywords or the statements in a structured text is destroyed, starting new lines and leaving blank spaces will be allowed.
- In principle, fullwidth characters and halfwidth characters in ISPSoft are regarded as different characters. Please avoid using fullwidth characters lest they should be identified incorrectly.
- Structured texts are case-insensitive. “IF” and “If” are considered to be the equivalents of “if”. (The C language is case-sensitive.)
- If users want to use constants in a structured text in ISPSoft, the constants must be represented in the following ways.
  - Decimal value: 23456 (A value which is not preceded by any mark will be regarded as a decimal value.)
  - Hexadecimal value: 16#5BA0 (A value is preceded by 16#.)
  - Octal value: 8#55640 (A value is preceded by 8#.)
  - Binary value: 2#101101110100000 (A value is preceded by 2#.)
  - String: “XYZ” (Characters are put in double quotes.)
  - Boolean value: TRUE or FALSE
- In a structured text in ISPSoft, the data types of the operands in an expression must be the same.

However, if the data type of one operand is INT and the data type of the other operand is WORD, the operands can take part in the operation. Likewise, if the data type of one operand is DINT and the data type of the other operand is DWORD, the operands can take part in the operation. If the data lengths of the operands in an expression are different, the operands can not take part in the operation.

```
0001 Var_0 := Var_INT + Var_WORD ;
0002
0003 Var_1 := Var_DINT + Var_DWORD ;
0004
0005 Var_2 := Var_INT + Var_DINT ;
```

- If the data type of a symbol is WORD, DWORD, LWORD, INT, DINT, or LINT, the system regards it as a signed integer during the operation.
- Although there is no limit on the number of lines, users still have to consider the capacity of the memory in the PLC.
- The sections of a structured text in ISPSoft can be copied/cut/pasted. Users can copy the text in a file edited with a text editor into a structured text in ISPSoft.
- The instructions that a structured text supports are as follows.

DVP series PLC	AH500 series CPU module
All the basic instructions are not supported. For example, LD, LDI, OUT, SET, RST, LD&, and etc are not supported.	
All the pulse instructions are not supported. For example, ROLP, RORP, BCDP, TRDP, and etc are not supported.	
PLS, PLF, FOR, NEXT, CJ, MC, MCR, STL, RET, and etc are not supported.	The Four Arithmetic Operations instructions, the Logical Operation instructions, the Contact Type Logic Operation instructions, and the Structure Creation instructions are not supported. PLS, PLF, MOV, DMOV, DFMOV, $MOV, MC, MCR, CJ, JMP, GOEND, and etc are not supported.

11.2 Structure of a Statement

11.2.1 Assignment Structure—:=

- **Format**

Device or Symbol := Expression ;

- **Description**

The value represented by the expression at the right side of := is assigned to the device or the symbol at the left side of :=.

- **Rules and limits**
  - The expression at the right side of := can be a constant, a device, a symbol, or an arithmetic expression. Only a device or a symbol can be at the left side of :=.
  - The operands at both sides of := must conform to the rules listed below.
    (a) One operand is an M/S/T/C/HC device, and the other operand is a symbol whose data type is BOOL/STEP/TIMER/COUNTER. (The device assigned to the operand which is a symbol is a bit device.)
    (b) One operand is a D/L device, and the other operand is a symbol whose data type is not BOOL/STEP/REAL/STRING.
    (c) One operand is a T/C device, and the other operand is a D/L device or a symbol whose data type is WORD/INT/TIMER/COUNTER.
    (d) One operand is a D/L/HC device, or a symbol whose data type is COUNTER, and to which a HC device is assigned. The other operand is a symbol whose data type is DWORD/DINT.

(e) One operand is a symbol whose data type is WORD/DWORD/LWORD, and the other operand is a symbol whose data type is INT/DINT/LINT. The data length of the operand at the left side of := must be larger than or equal to the data length of the operand at the right side of :=.

(f) Both operands are symbols whose data types are REAL/LREAL/STRING. The data lengths of the operands must be the same.

- **Example**

Example 1: The value of M0 is assigned to the symbol OUT_0.

Example 2: M0 is set to ON.

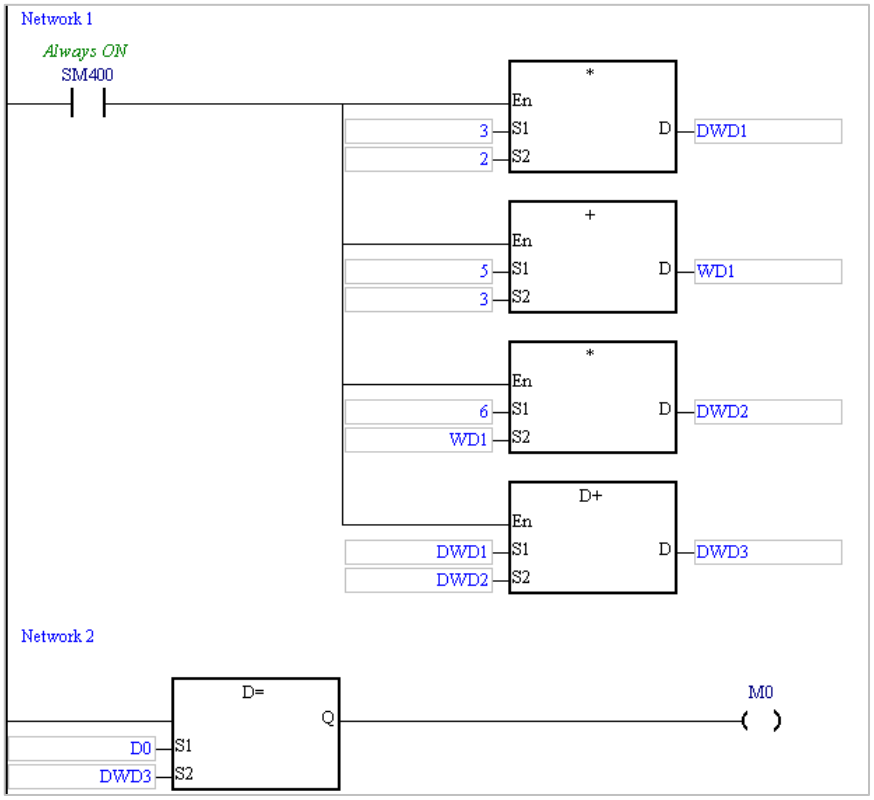
Example 3: The value in D2 is added to the value in D1. The sum is stored in D0.

Example 4: 3 is assigned to the symbol DATA.

Example 5: The system judges whether the value of the arithmetic expression 3*2+6* (5+3) is equal to the value stored in D0. The judgement result is sent to M0.

```
0001
0002 M0 := (D0=3*2+6*(5+3)) ;
0003
```

=



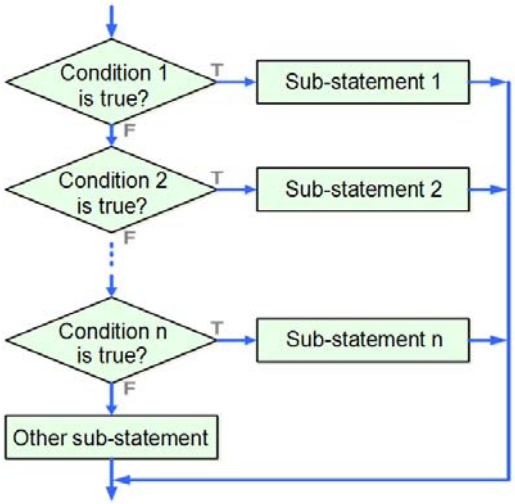
Example 6: Examples of incorrect use of the assignment structure are as follows.

Statement	Description
D0 := M0 ;	The data type of one operand must not be different from the data type of the other operand.
V_WORD := V_DWORD ;	The data length of the operand at the left side of := must not be less than the data length of the operand at the right side of :=.
D0 := V_REAL ;	The value of a symbol whose data type is REAL can not be assigned to a D device.
V_LREAL := V_REAL ;	If one operand is a symbol whose data type is REAL, the other operand must be a symbol whose data type is REAL. If one operand is a symbol whose data type is LREAL, the other operand must be a symbol whose data type is LREAL.
V_DWORD := V_TIMER ;	The value of a symbol whose data type is TIMER can not be assigned to a symbol whose data type is DWORD, but can be assigned to a symbol whose data type is WORD.
V_WORD := HC0 ;	The value in a HC device can not be assigned to a symbol whose data type is WORD, but can be assigned to a symbol whose data type is DWORD.

11.2.2 Conditional Structure—IF

- **Format**

```
IF   Bool expression 1   THEN
     Sub-statement 1 ;
ELSIF  BooL expression 2   THEN
     Sub-statement 2 ;
     :
ELSIF  BooL expression n   THEN
     Sub-statement n ;
ELSE
     Other sub-statement ;
END_IF ;
```



- **Description**

  If the value of **Boolean expression 1** is TRUE, **Sub-statement 1** will be executed. If the value of **Boolean expression 1** is FALSE, **Boolean expression 2** will be evaluated. If the values of all the Boolean expressions are FALSE, **Other sub-statement** will be executed.

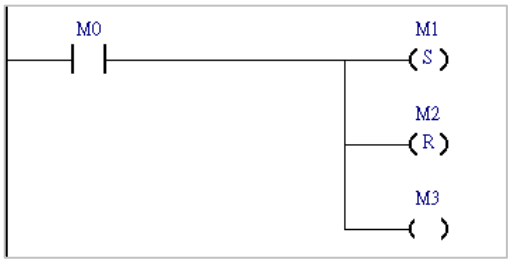
- **Rules and limits**
  - A Boolean expression can be a device, a symbol, or an arithmetic expression. It can not be a constant.
  - A sub-statement can be a legal statement, or an IF statement.
  - There is no limit on the number of sub-statements following **THEN** or **ELSE**.
  - The **ELSIF** sections and the **ELSE** section can be omitted. If the **ELSE** section is omitted, and the values of all the Boolean expressions are FALSE, no sub-statement will be executed.
  - Users can add **ELSIF** sections to a conditional structure at will. There is only one **ELSE** section in a conditional structure. The **ELSE** section in a conditional structure is at the last section of the conditional structure.

- **Example**

  Example 1: If M0 is ON, M1 will be set to ON, and M2 will be set to OFF. If M0 is not ON, the step will end. No matter what the value of M0 is, the value of M0 will be assigned to M0.

```
0001 IF M0 THEN
0002     M1 := TRUE ;
0003     M2 := FALSE ;
0004 END_IF;
0005
0006 M3 := M0 ;
```

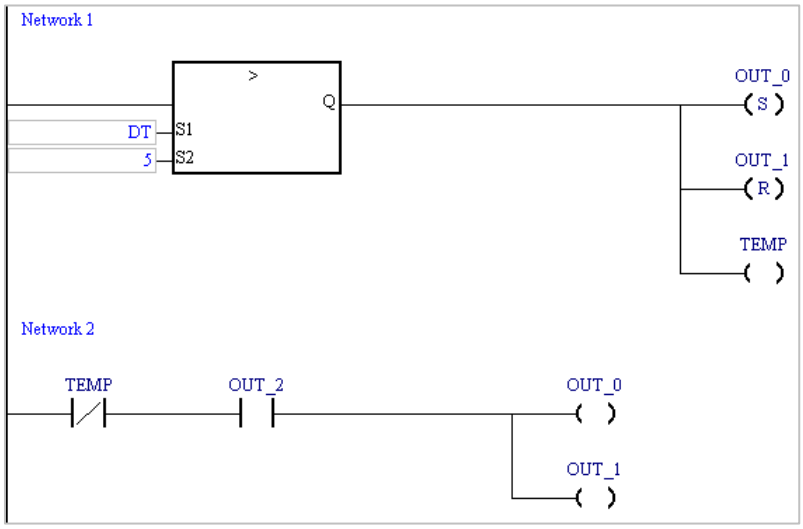
=



  Example 2: If the value of DT is larger than 5, OUT_0 will be set to ON, and OUT_1 will be reset to OFF. If the value of DT is not larger than 5, the value of OUT_2 will be assigned to OUT_0 and OUT_1.

```
0001 IF DT>5 THEN
0002     OUT_0 := TRUE ;
0003     OUT_1 := FALSE ;
0004 ELSE
0005     OUT_0 := OUT_2 ;
0006     OUT_1 := OUT_2 ;
0007 END_IF;
```

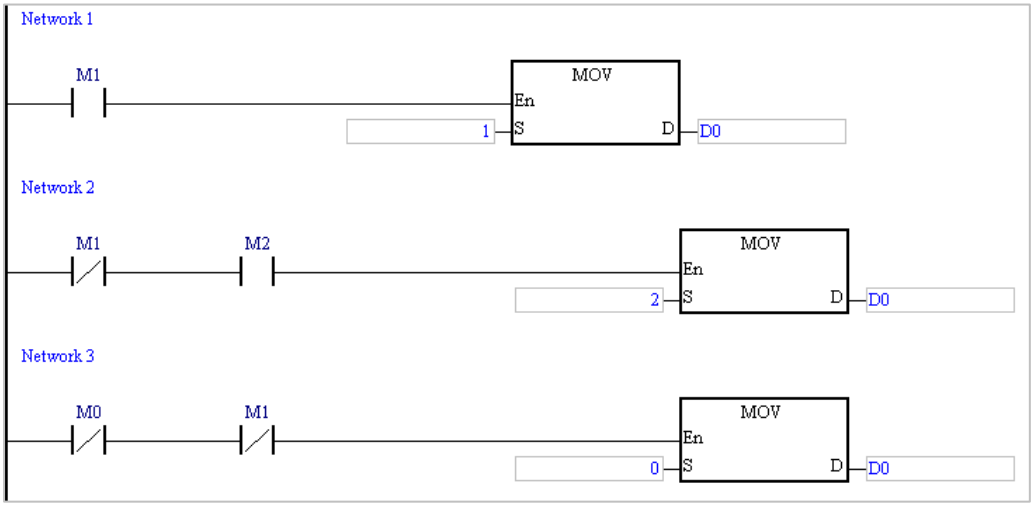
=



Example 3: If M1 is ON, the value in D0 will be 1. If M1 is not ON, the PLC will evaluate M2. If M2 is ON, the value in D0 is 2. If M2 is not ON, the value in D0 will be 0.

```
0001 IF M1 THEN
0002     D0 := 1 ;
0003 ELSIF M2 THEN
0004     D0 := 2 ;
0005 ELSE
0006     D0 := 0 ;
0007 END_IF;
```

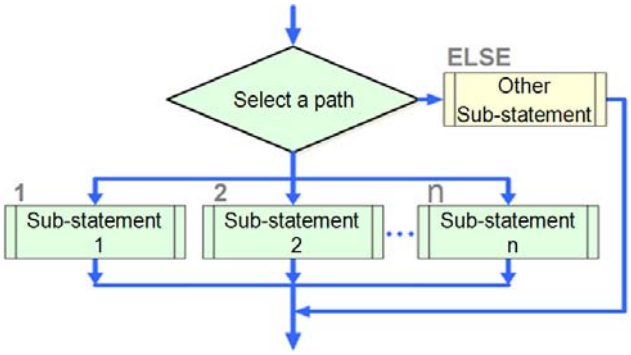
=



11.2.3 Conditional Structure—CASE

- **Format**

```
CASE  Integer expression  OF
    Conditional value 1  :  Sub-statement 1  ;
    Conditional value 2  :  Sub-statement 2  ;
                :
    Conditional value n  :  Sub-statement n  ;
ELSE
    Other sub-statement  ;
END_CASE ;
```



- **Description**

  The PLC judges whether the value of **Integer expression** is on the list of condition values. If the value of **Integer expression** is the same as a condition value on the list, the sub-statement following the condition value will be executed. If the value of **Integer expression** is not on the list of condition values, **Other sub-statement** under the keyword **ELSE** will be executed.

- **Rules and limits**

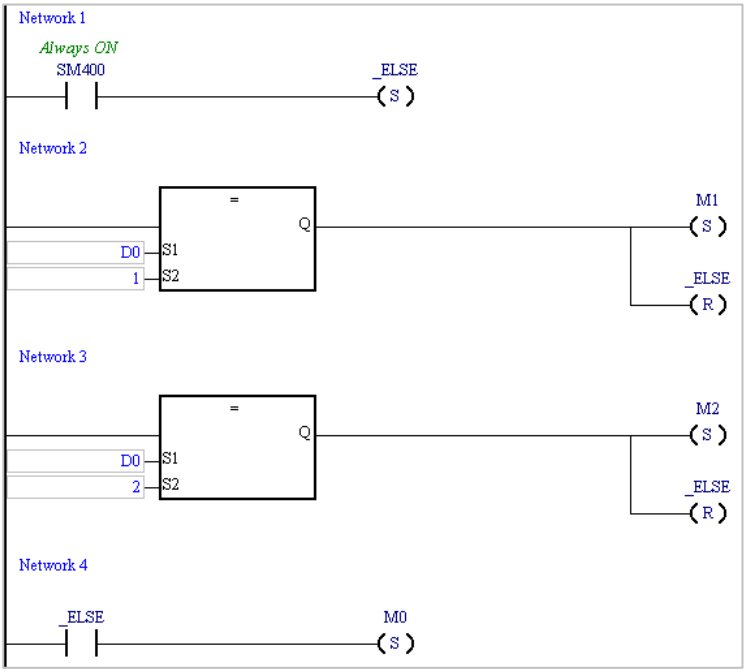
  - **Integer expression** in the figure above can be a device, a symbol, or an arithmetic expression. It can not be a constant. The data type of **Integer expression** must be INT (-32768~32767) or DINT (-2147483648~2147483647). If a device is used as **Integer expression**, the system will automatically regard the value in the device as an integer. Besides, if an arithmetic expression is used as **Integer expression**, and the result of the operation is not an integer, the result will be rounded down to the nearest whole digit.
  - A condition value on the list can not be the same as another condition value on the list, and the condition values on the list must be integers. The range of condition values depends on the data type of **Integer expression**. If the data type is INT, the condition values must be in the range of -32768 to 32767. If the data type is DINT, the condition values must be in the range of -2147483648 to 2147483647.
  - If the sub-statements following some condition values are the same, users can combine them.
    (a) If the condition values are not continuous values, they are separated by commas. For example, **1, 3, 5: Sub-statement** means that the sub-statement will be executed if the value of **Integer expression** is 1, 3, or 5.
    (b) If the condition values are continuous values, the range of condition values can be indicated by the mark "**..**". For example, **3..6: Sub-statement** means that the sub-statement will be executed if the value of **Integer expression** is in the range of 3 to 6 (including 3 and 6). If the mark "**..**" is used, the value at the left side of "**..**" must be less than the value at the right side of "**..**". For example, **6..3 :** is an illegal range.
  - A sub-statement can be a legal statement, or another structure.
  - There is no limit on the number of sub-statements following a condition value.
  - The **ELSE** section can be omitted. If the **ELSE** section is omitted, and the value of **Integer expression** is not on the list of condition values, no sub-statement will be executed.
  - After a sub-statement is executed, the execution of the program will jump out of the CASE structure. No jump instruction is needed.

- **Example**

  Example 1: If the value in D0 is 1, M1 will be ON. If the value in D0 is 2, M2 will be ON. If the value in D0 is neither 1 nor 2, M0 will be ON.

```
0001 CASE D0 OF
0002   1: M1 := TRUE ;
0003   2: M2 := TRUE ;
0004 ELSE
0005   M0 :=TRUE ;
0006 END_CASE;
```

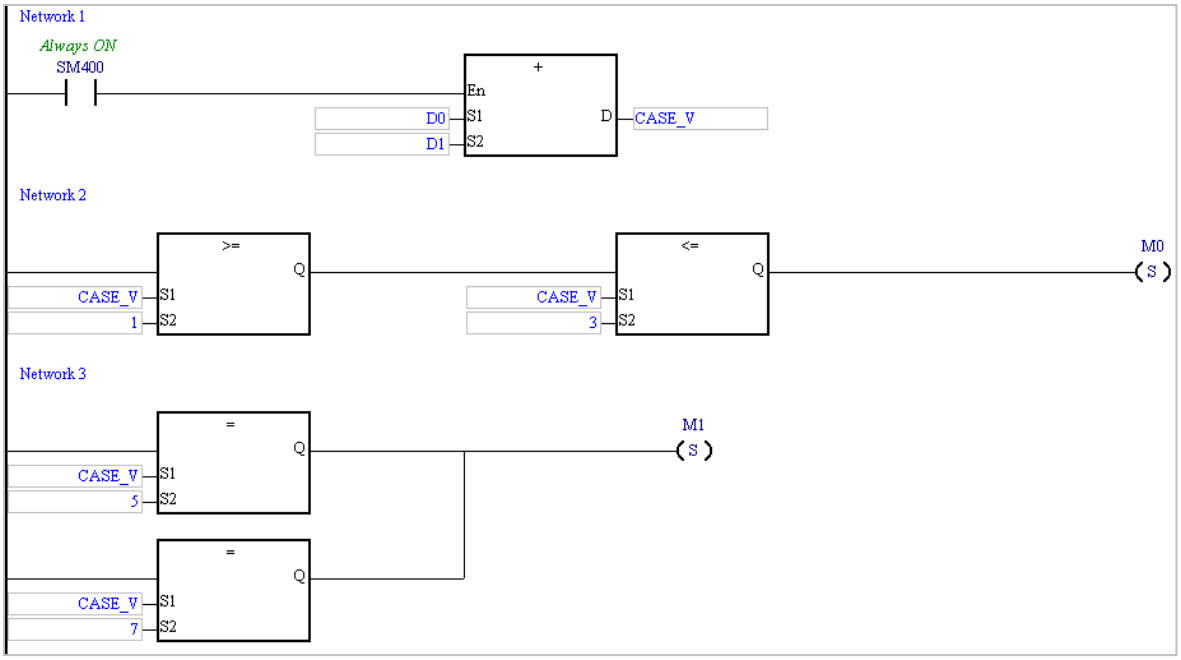
=



Example 2: The value of the arithmetic expression D0+D1 is calculated first. If the result of the operation is in the range of 1 to 3, M0 will be ON. If the result of the operation is 5 or 7, M1 will be ON. If the result of the operation is not on the list of condition values, no sub-statement will be executed, and the execution of the program will jump out of this structure.

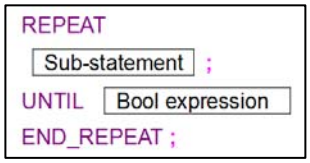
```
0001 CASE D0+D1 OF
0002    1..3: M0 := TRUE ;
0003    5,7 : M1 := TRUE ;
0004 END_CASE;
```

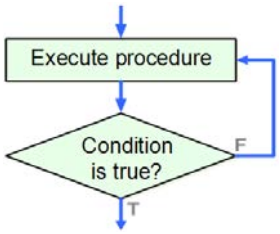
=



11.2.4 Loop Structure—REPEAT

- **Format**





- **Description**

  The sub-statement is executed, and then the Boolean expression is evaluated. If the Boolean expression is evaluated as TRUE, the loop will be terminated. If the value of the Boolean expression is FALSE, the program in the structure will be executed repeatedly until the value of the Boolean expression is TRUE.

- **Rules and limits**
  - The Boolean expression can be a device, a symbol, or an arithmetic expression. It can not be a constant.
  - The sub-statement is executed before the Boolean expression is evaluated, and therefore the sub-statement is executed at least once.
  - The sub-statement can be a legal statement, or another structure. There is no limit on the number of sub-statements in a **REPEAT** structure. One loop structure can be embedded in another loop structure. There are at most 64 loop structures.
  - If the value of the Boolean expression following **UNTIL** is TRUE, the loop will be terminated. To prevent an infinite loop, the operand in the Boolean expression following **UNTIL** can not be a device or a symbol which can not be updated.
  - The X devices in a program are updated after the program is scanned. If users want to use an X device as the operand of the Boolean expression, they have to add an I/O update instruction to the structure, or update the X through an interrupt subroutine.

- **Example**

  Example 1: The initial value of DT is 0. In the loop structure, the value of DT increases progressively by 2. The loop will not be terminated until the value of DT is larger than 100.

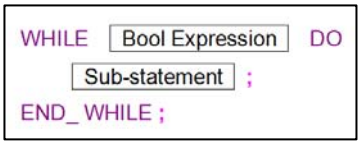
```
0001 DT := 0 ;
0002 REPEAT
0003   DT := DT + 2 ;
0004 UNTIL DT > 100
0005 END_REPEAT;
```

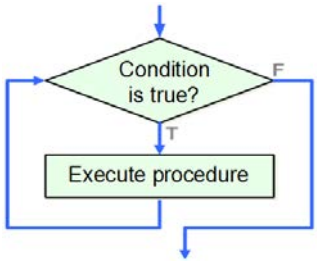
  Example 2: The initial value in D0 is 1, and the initial value in D10 is 5. After the loop is terminated, the value in D0 will become 5*4*3*2*1.

```
0001 D10 := 5 ;
0002
0003 D0 := 1 ;
0004 REPEAT
0005   D0 := D0 * D10 ;
0006   D10 := D10 - 1 ;
0007 UNTIL  D10 = 0
0008 END_REPEAT ;
```

11.2.5 Loop Structure—WHILE

- **Format**





- **Description**

  The Boolean expression is evaluated. If the value of the Boolean expression is TRUE, the sub-statement will be executed. This repeats until the value of the Boolean expression becomes FALSE.

- **Rules and limits**
  - The Boolean expression can be a device, a symbol, or an arithmetic expression. It can not be a constant.
  - The sub-statement will be executed if the value of the Boolean expression is TRUE, and therefore the sub-statement may not be executed.
  - The sub-statement can be a legal statement, or another structure. There is no limit on the number of sub-statements in a **WHILE** structure. One loop structure can be embedded in another loop structure. There are at most 64 loop structures.
  - If the value of the Boolean expression following **WHILE** is FALSE, the loop will be terminated. To prevent an infinite loop, the operand in the Boolean expression following **WHILE** can not be a device or a symbol which can not be updated.
  - The X devices in a program are updated after the program is scanned. If users want to use an X device as the operand of the Boolean expression, they have to add an I/O update instruction to the structure, or update the X through an interrupt subroutine.

- **Example**

  Example 1: The initial value of DT is 0. In the loop structure, the value of DT increases progressively by 2. The loop will not be terminated until the value of DT is larger than 100.

```
0001 DT := 0 ;
0002
0003 WHILE DT <= 100 DO
0004   DT := DT + 2 ;
0005 END_WHILE;
```

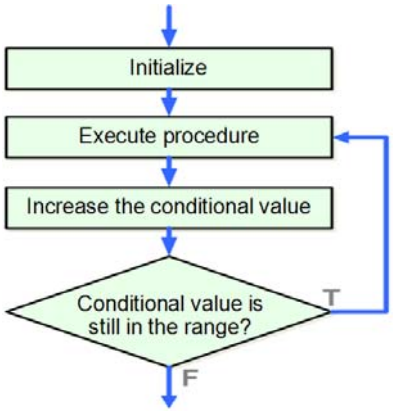
  Example 2: The REPEAT loop checks the Boolean expression after the sub-statement is executed. The sub-statement in the REPEAT structure is executed once. After the REPEAT loop is terminated, the value in D0 will become 1. The WHILE loop checks the Boolean expression before the sub-statement is executed. The Boolean expression in the WHILE structure is evaluated as FALSE, and therefore the sub-statement in the WHILE structure is not executed. After the WHILE loop is terminated, the value in D1 will still be 0.

```
0001 M0 := TRUE ;
0002 D0 := 0 ;
0003 D1 := 0 ;
0004
0005 REPEAT
0006  D0 := D0 + 1 ;
0007 UNTIL  M0
0008 END_REPEAT ;
0009
0010 WHILE (NOT M0) DO
0011    D1 := D1 + 1 ;
0012 END_WHILE ;
```

11.2.6 Loop Structure—FOR

- **Format**

```
FOR Device or Symbol := Start value TO End value BY Increment DO
    Sub-statement ;
END_FOR ;
```



- **Description**

  The initial value of the symbol or the initial value in the device is the start value. Whenever the loop body is executed, the value of the symbol or the value in the device increases by the increment. If the value of the symbol or the value in the device is in the range of the start value to the end value, the loop body will be executed. If the value of the symbol or the value in the device is not in the range of the start value to the end value, the loop will be terminated.

- **Rules and limits**
  - The start value/end value/increment can be a device, a symbol, an arithmetic expression, or a constant. The data type of the start value/end value/increment must be INT (-32768~32767) or DINT (-2147483648~2147483647). If a device is used as the start value/end value/increment, the system will automatically regard the value in the device as an integer. Besides, if an arithmetic expression is used as the start value/end value/increment, and the result of the operation is not an integer, the result will be rounded down to the nearest whole digit.
  - If the end value is less than the start value, the increment must be a negative number. Otherwise, there will be an infinite loop.
  - The data type of the symbol must be INT or DINT. If the symbol or the device takes part in the operation, please prevent the value of the symbol or the value in the device from resulting in an infinite loop.
  - The sub-statement can be a legal statement, or another structure. There is no limit on the number of sub-statements in a **FOR** structure. One loop structure can be embedded in another loop structure. There are at most 64 loop structures.
  - After the value of the symbol or the value in the device increases by the increment, the system will check whether the value of the symbol or the value in the device is in the range of the start value to the end value. If an overflow occurs after the value of the symbol or the value in the device increases by the increment, there will be an infinite loop.

- **Example**

  The initial value of the symbol INDEX is 1. Whenever the loop body is executed, the value of INDEX increases by 2. After the loop body is executed for the fifth time, the value of INDEX will increase by 2, and the value of INDEX will become 11. Owing to the fact that the value INDEX is not in the range of 1 to 9, the loop is terminated. The initial value of the symbol SU is 0. After the loop is terminated, the value of SU will become $1^2+3^2+5^2+7^2+9^2$.

```
0001 SU := 0 ;
0002
0003 FOR  INDEX := 1  TO  9  BY  2  DO
0004      SU :=  INDEX * INDEX + SU ;
0005 END_FOR ;
0006
```

11.2.7 Applied Instruction Structure

- **Format**

Name of API (Operand 1 , Operand 2 , ... , Operand n);

- **Description**

The applied instruction is executed.

- **Rules and limits**
  - The state of the En pin does not need to be specified in the applied instruction. When the statement is executed, the applied instruction is executed. An IF statement functioning as an En pin can be used with the applied instruction.
  - As long as the structure of the operands and the structure of the keywords are not destroyed, starting new lines is allowed.
  - The applied instruction is a statement, and therefore it ends with a semicolon.
  - The operands specified in the applied instruction must be put in parentheses.
  - The order in which the operands are arranged can not be changed. Please refer to instruction manuals for more information.
  - The applied instructions in ISPSoft do not support the assignment of values, and therefore `M0 := CMP ( D0, D10, M0 ) ;` is an illegal statement.
  - As long as the structure of the applied instruction is unchanged, starting new lines is allowed.

```
0001 CMP(
0002      D0,
0003      D10,
0004      M0
0005      ) ;
```

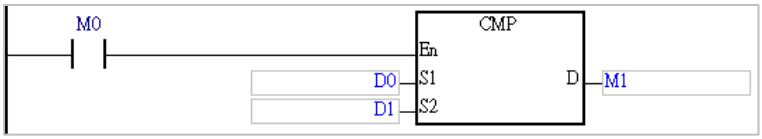
```
0001 CMP
0002 ( D0, D10, M0 ) ;
0003
0004
0005
```

```
0001 CM
0002 P( D0, D10, M0 ) ;
0003
0004
0005
```

- **Example**

Example 1: The instruction CMP is used with an IF statement.

```
0001 IF M0 THEN
0002   CMP(D0, D1, M1);
0003 END_IF;
```



Example 2: The state of the En pin does not need to be specified in an applied instruction. If the instruction TMR is used, a statement about resetting the timer is required. Otherwise, the value of the counter will not become zero. For example, line 0004 in the figure below is a statement about resetting the timer.

```
0001 IF M0 THEN
0002   TMR (T0, 50);
0003 ELSE
0004   T0 := FALSE ;
0005 END_IF ;
```

11.2.8 Function Block Structure

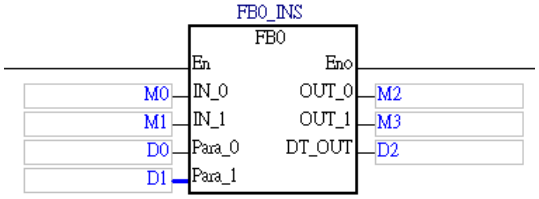
- **Format**

```
Name of FB instance ( Input pin 1 := Input operand 1 ,
                      Input pin 2 := Input operand 2 ,
                      .....
                      Output pin1 => Output operand 1 ,
                      Output pin 2 => Output operand 2 ,
                      .....
                      Output pin n => Output operand n ) ;
```

- **Description**

  The function block is executed.

- **Rules and limits**
  - The state of the En pin of the function block does not need to be specified in the function block. When the statement is executed, the function block is executed. An IF statement functioning as an En pin can be used with the applied instruction.
  - As long as the structure of the operands and the structure of the keywords are not destroyed, starting new lines is allowed.
  - The function block is a statement, and therefore it ends with a semicolon.
  - The input/output pins of the function block and the corresponding operands must be put in parentheses. A pin and the corresponding operand form a group. The groups are separated by commas. The data types of the operands are specified in the function block definition.
  - The function block before the parentheses is a function block instance rather than a function block definition, that is, the symbol before the parentheses is a symbol whose data type is a function block. Please refer to chapter 7 for more information about function block definitions and function block instances.
  - The assignment mark for an input pin is **:=**, and the assignment mark for an output pin is **=>**.
  - The pins and the operands are arranged according to the definition of the function block. The arrangement can not be changed. The En pin and the Eno pin are excluded. The input pins are listed above all the output pins. A pin of the VAR_IN_OUT class is an input pin. Please refer to the ladder diagram and the structured text below for more information.



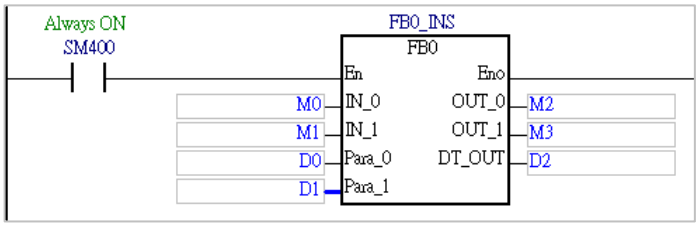

```
0001 FB0_INS ( IN_0 := M0 ,
0002           IN_1 := M1 ,
0003           Para_0 := D0 ,
0004           Para_1 := D1 ,
0005           OUT_0 => M2 ,
0006           OUT_1 => M3 ,
0007           DT_OUT => D2 );
```

- **Example**

  Example 1: When a statement which calls a function block is executed, the function block is executed.

```
0001 FB0_INS ( IN_0 := M0 ,
0002           IN_1 := M1 ,
0003           Para_0 := D0 ,
0004           Para_1 := D1 ,
0005           OUT_0 => M2 ,
0006           OUT_1 => M3 ,
0007           DT_OUT => D2 );
```

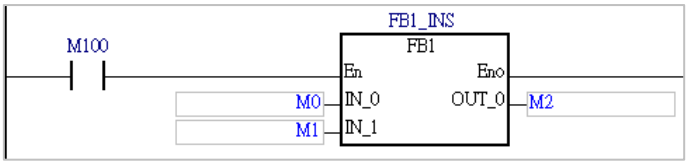
=



  Example 2: A function block is used with an IF statement.

```
0001 IF M100 THEN
0002 FB1_INS ( IN_0 := M0 ,
0003           IN_1 := M1 ,
0004           OUT_0 => M2 );
0005 END_IF;
```

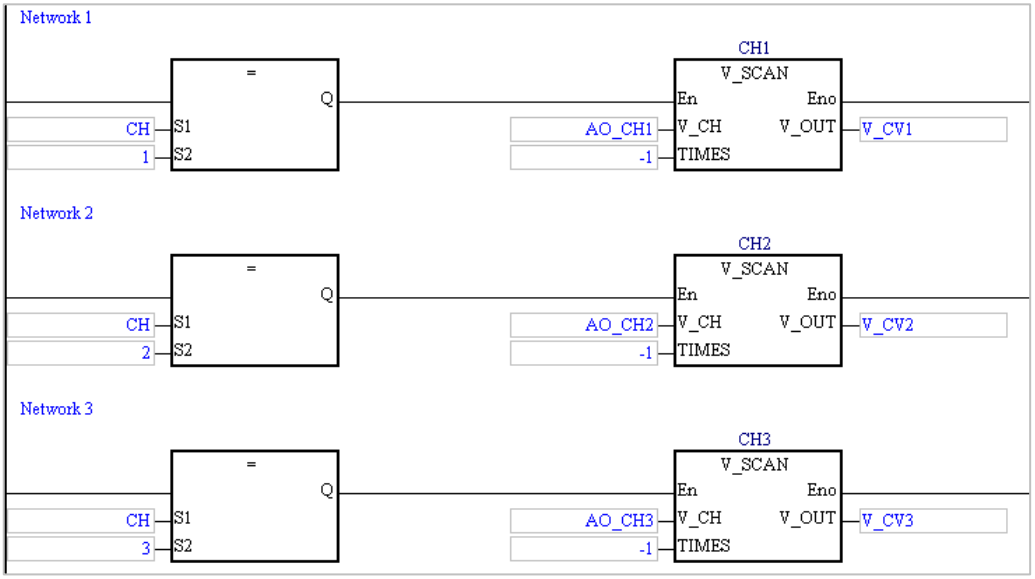
=



Example 3: The value of the symbol CH determined which function block will be executed.

```
0001 CASE CH OF
0002   1: CH1 ( V_CH := AO_CH1,
0003            TIMES := -1 ,
0004            V_OUT => V_CV1 );
0005
0006   2: CH2 ( V_CH := AO_CH2,
0007            TIMES := -1 ,
0008            V_OUT => V_CV2 );
0009
0010   3: CH3 ( V_CH := AO_CH3,
0011            TIMES := -1 ,
0012            V_OUT => V_CV3 );
0013 END_CASE;
```

=



11.2.9 Blank Statement

- **Format**

```
;
```

- **Description**

  If a statement is needed, but no action needs to be executed, a blank statement is used.

- **Example**

  The For loop below shows that no action is executed. Owing to the blank statement, the execution of the program is delayed. Users can adjust the time for which the execution of the program is delayed by adjusting the number of times the loop body is executed.

```
0001 FOR D100:=1 TO 3000 BY 1 DO
0002     ;
0003 END_FOR;
```

11.2.10 RETURN Statement

- **Format**

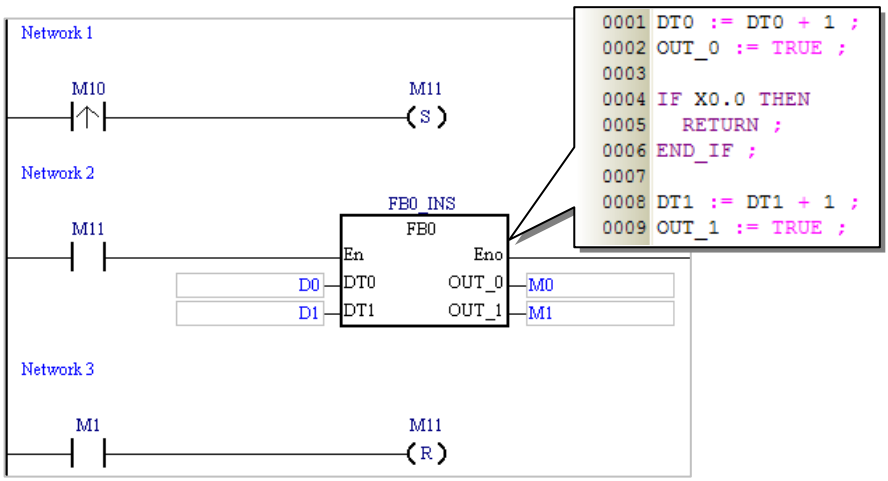
```
RETURN ;
```

- **Description**

  A RETRUN statement is used in a function block in a structured text. It ends the program in the function block.

- **Example**

  The programming language used to create the program in the function block FB0 is a structured text. After the upper part of the program is executed, the state of X0.0 will be judged. If X0.0 is ON, the execution will leave the current program. If X0.0 is OFF, the whole program in the function block will be executed.



11.2.11 EXIT Statement

- **Format**

```
EXIT ;
```

- **Description**

  An EXIT statement is used in a REPEAT loop, a WHILE loop, or a FOR loop. It is used to exit the loop.

- **Example**

  Example 1: The loop below is a REPEAT loop. The loop will not be terminated until the value of DT is 0. The IF structure in the loop is used with an EXIT statement. The initial value in D0 is 0. It increases by 1. The loop will not be terminated until the value in D0 is 100. As a result, the loop body will be executed 100 times at most. In the example below, the initial value of DT is 110. After the loop body is executed for the 100th time, the value of DT will become 10. In other words, the final value of DT will be 10. The value of DT will not decrease to 0.

```
0001 DT := 110 ;
0002 D0 := 0 ;
0003
0004 REPEAT
0005
0006    DT  :=  DT - 1 ;
0007    D0  :=  D0 + 1 ;
0008
0009    IF  D0=100  THEN
0010      EXIT ;
0011    END_IF ;
0012
0013 UNTIL  DT = 0
0014 END_REPEAT ;
```

  Example 2: The loop below is a FOR loop. The value of the arithmetic expression D0* D10 is calculated, and the result of the operation is stored in D0. The IF structure in the loop is used with an EXIT statement. The loop will not be terminated until the value in D0 is

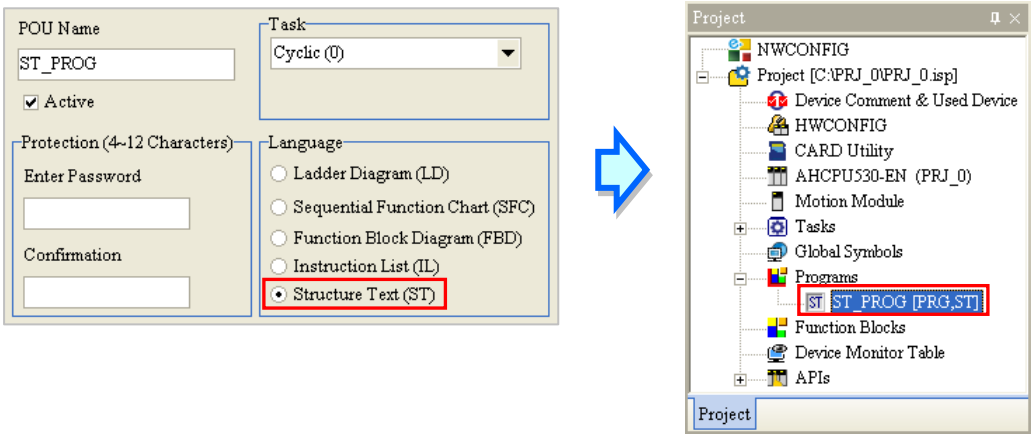
larger than 100. When the value in D10 increases to 5, the value in D0 is 1*2*3*4*5, and the loop is terminated.

```
0001 DT := 10 ;
0002 D0 := 1 ;
0003
0004 FOR D10 := 1 TO DT BY 1 DO
0005
0006     D0 := D0 * D10 ;
0007
0008     IF  D0>100  THEN
0009       EXIT ;
0010     END_IF ;
0011
0012 END_FOR ;
```

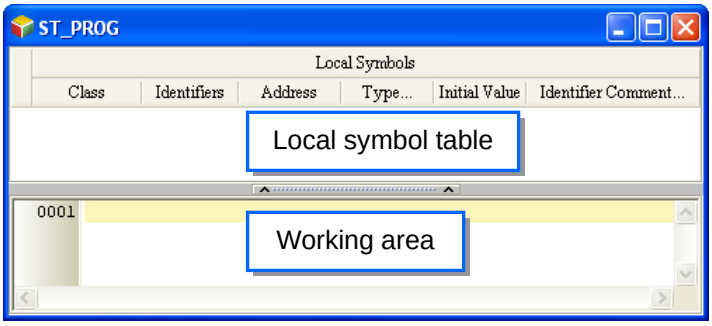
11.3 Creating a Structured Text in ISPSoft

11.3.1 Editing Environment

Select the **Structured Text (ST)** option button in the **Language** section in the **Create Program** window. Please refer to section 5.4.1 for more information.



The environment in which a structured text can be edited is shown below. The table at the upper part of the window is a local symbol table, and the area at the lower part of the window is a working area. There are line numbers at the left side of the working area. Users can set the way in which the line numbers are displayed. Please refer to section 2.3.1 for more information. After a program editing window in which a structured text can be created is opened, the corresponding toolbar will appear in the ISPSoft window. The functions are described below.



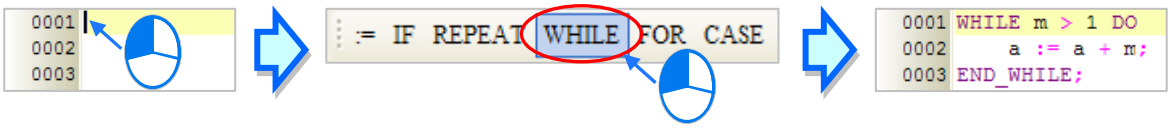
Icon	Function
:=	Inserting an example of an assignment structure

Icon	Function
IF	Inserting an example of an if statement
REPEAT	Inserting an example of a repeat loop
WHILE	Inserting an example of a while loop
FOR	Inserting an example of a for loop
CASE	Inserting an example of a case statement

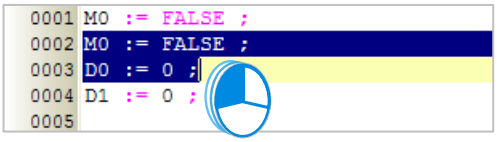
11.3.2 Creating a Structured Text

The way in which a structured text file is edited is similar to the way in which a text file is edited with a general text editor. If users want to create a structured text, they can type or modify text in the working area. If the users want to start a new line of text at a specific point, they can press Enter on the keyboard.

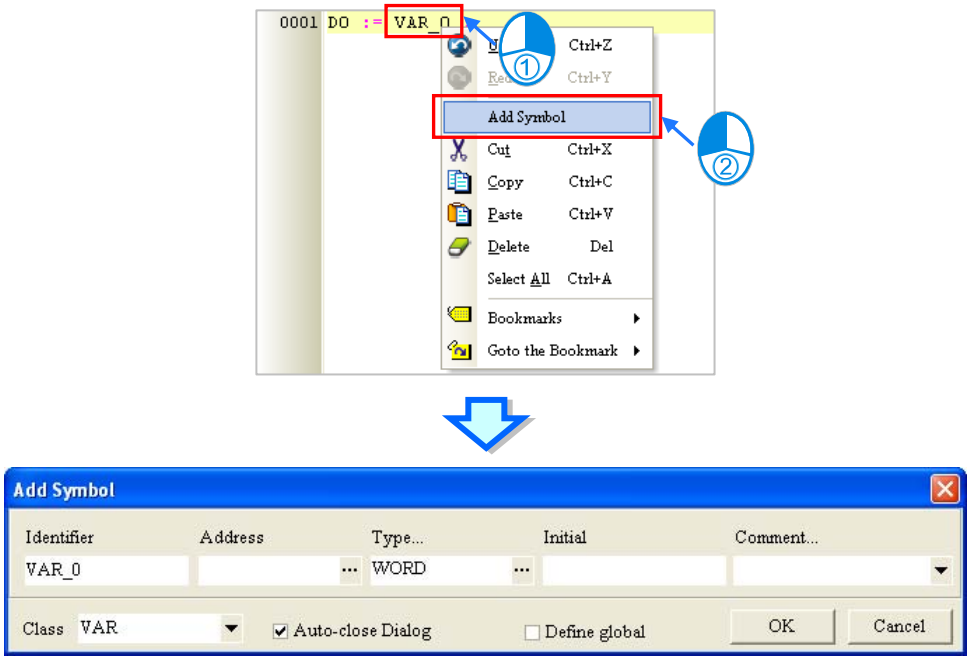
Besides, the users can click a line which will be edited, insert an example of a structure, and modify the structure.



If users want to select any amount of text, they can click where they want to begin the selection, hold down the left mouse button, and drag the pointer over the text that they want to select. The users can also click at the start of the selection, scroll to the end of the selection, and hold down Shift on the keyboard while they click where they want the selection to end.



If users type a symbol which has not been declared, they can click the symbol, click the right mouse button, and click **Add Symbol** on the context menu to open the **Add Symbol** window.

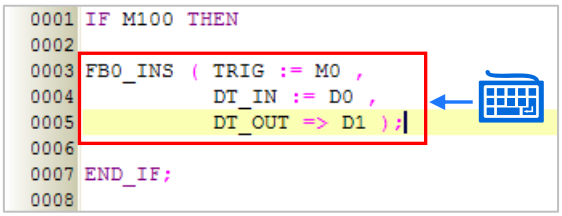


11.3.3 Inserting Applied Instructions and Function Blocks

Users can insert an applied instruction or a function block in a structured text in ISPSoft in one of the two ways described below.

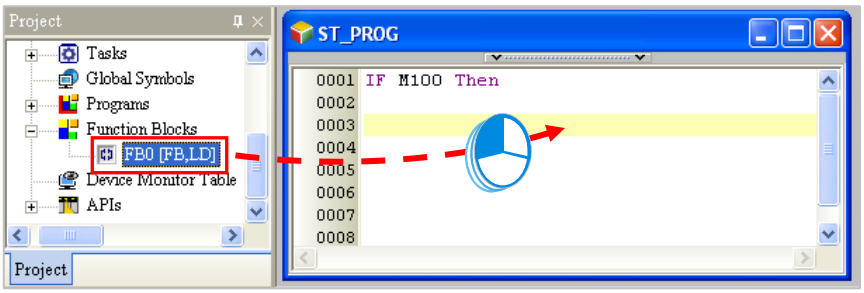
- **Method 1**

  An applied instruction or a function block is inserted according to the format of the applied instruction or the format of the function block.

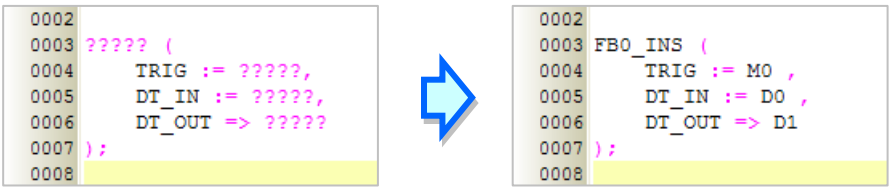


- **Method 2**

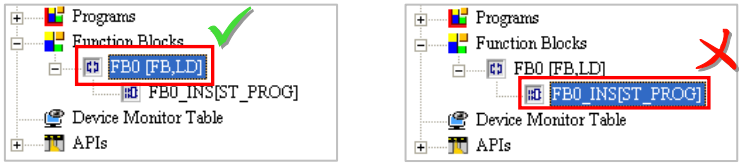
  Unfold the **APIs** section or the **Function Blocks** section in the project management area, and find the item which will be inserted. Select the item, and drag it to the position in which it will be inserted.



After the applied instruction or the function block is inserted, users have to type the corresponding operands according to the format of the applied instruction or the format of the function block.



Function block definitions can be dragged, but function block instances can not be dragged.



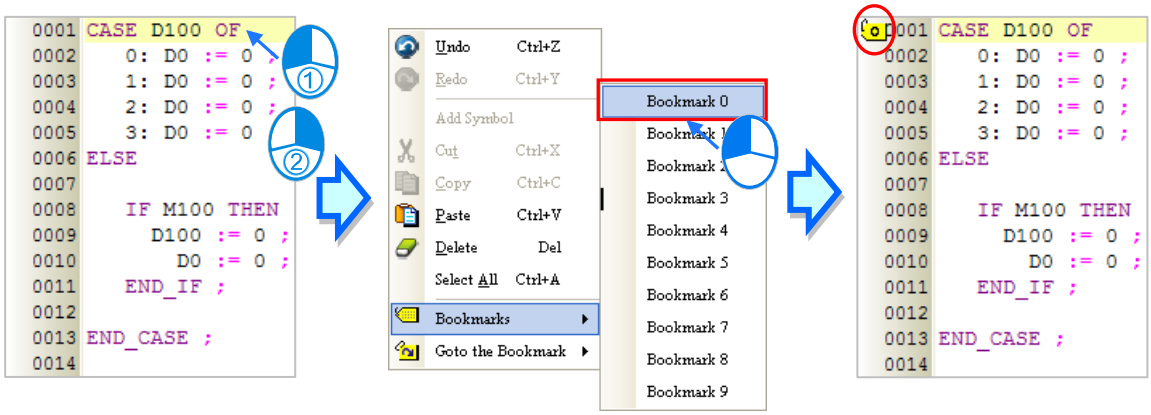
11.3.4 Bookmark

After lines are bookmarked, users can find or go to the lines easily.

- **Adding/Removing bookmarks**

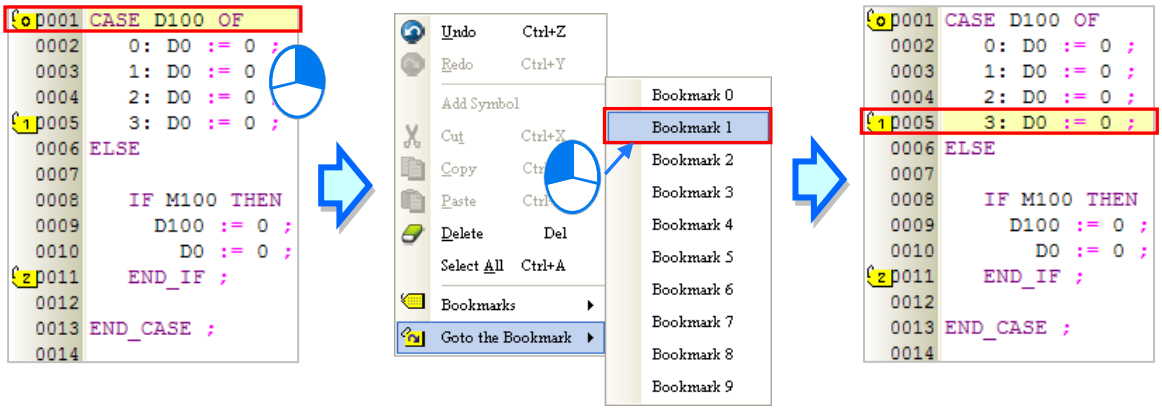
  If users want to add a bookmark to a line, they can select the line, click the right mouse button, click **Bookmarks** on the context menu, and select a bookmark number. If users want to remove the bookmark from a line, they can select the line, click the right mouse button, click **Bookmarks** on the context menu, and click the bookmark number corresponding to the bookmark added to the line. If the bookmark number clicked does not correspond to the bookmark added to the line, the bookmark number added to the line will become the bookmark number clicked. Besides, if users click the **Edit** menu, point to **Bookmarks**, and click **Remove All Bookmarks**, all the bookmarks in the working area will be removed.

  If users click a bookmark number which has been added to another line, the bookmark number will be moved to the line selected presently.



- **Going to a bookmark**

  Right-click the working area, point to **Goto the Bookmark** on the context menu, and click a bookmark number.



11.4 Example of a Structured Text

11.4.1 Explanation

The example is about the control of a tower light. A tricolor light and a buzzer are used. The control mechanisms for various device statuses are listed below. If more than one device status appears, the device status which is given precedence over the others is displayed.

State code	Device status	Light				Precedence
		Red light	Yellow light	Green light	Buzzer	
4	The device is malfunctions.	ON	OFF	OFF	Flash	Highest
3	The device malfunctions. (The buzzer is turned off.)	Flash	OFF	OFF	OFF	↑
2	The device is operating.	OFF	OFF	ON	OFF	
1	The work is done.	OFF	Flash	OFF	OFF	
0	The device is idling.	OFF	ON	OFF	OFF	Lowest

11.4.2 Planning Hardware

In this example, the AH500 series CPU module **AHCPU530-EN**, the digital I/O module **AH16AP11R-5A**, and the four-slot main backplane **AHBP04M1-5A** are used. The table below is an I/O allocation table.

Type	ID	Description
Digital output	Y0.0	Red light
Digital output	Y0.1	Yellow light
Digital output	Y0.2	Green light
Digital output	Y0.3	Buzzer

11.4.3 Planning a Program

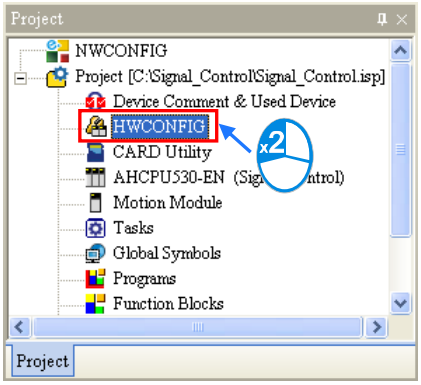
(1) If the flag ERROR is ON, and the flag BUZZER_OFF is OFF, the state code will be 4.
(2) If the flag ERROR is ON, and the flag BUZZER_OFF is ON, the state code will be 3.
(3) If the flag ERROR is OFF, and the flag RUNNING is ON, the state code will be 2.
(4) If the flags ERROR and RUNNING are OFF, and the flag COMPLETE is ON, the state code will be 1.
(5) If the flags ERROR, RUNNING, and COMPLETE are OFF, the state code will be 0.
(6) Every state code corresponds to a light. The control of the tricolor light has many practical applications, and the tricolor light can be displayed in various ways. Considering that a program can be expanded and reused, a function block is created.

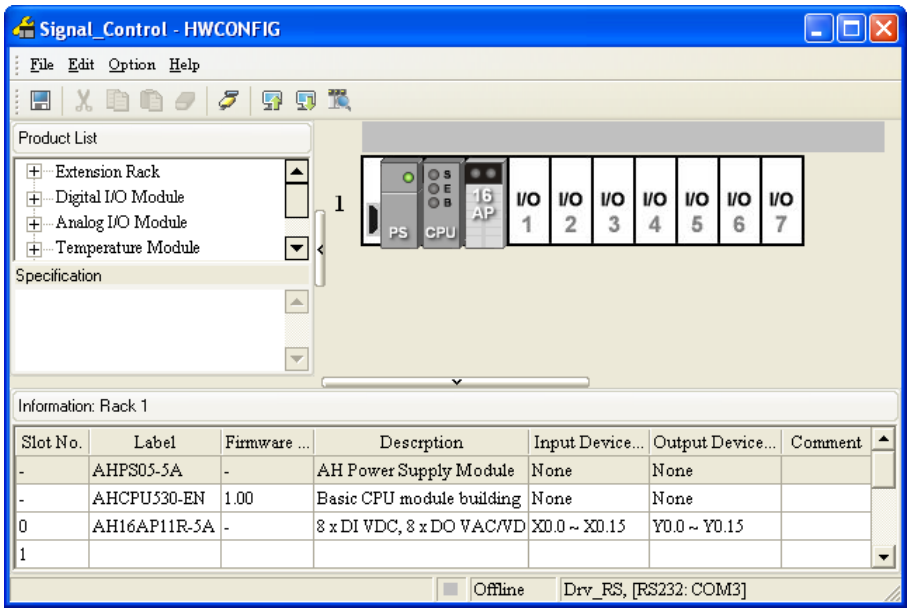
11.4.4 Creating a Program

Start ISPSoft, and then create a new project.

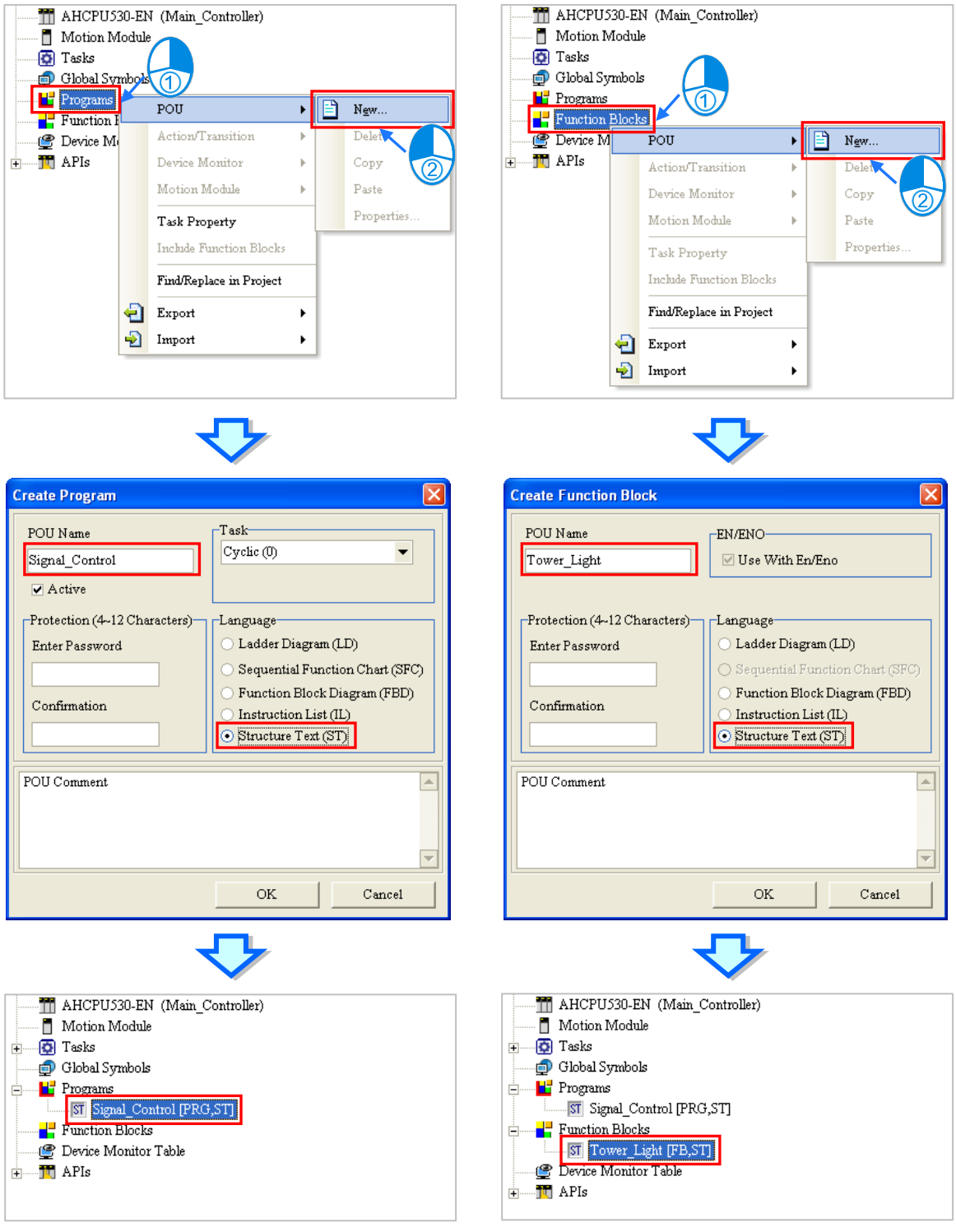
The hardware configuration in this example is the same as the hardware configuration in chapter 4.

Please refer to chapter 4, and complete the hardware configuration.





Create a POU of the program type and a POU of the function block type in the project management area. The programming language used to create the POU of the program type and the POU of the function block type are structured texts.



Create the flags and the digital outputs in the global symbol table.

Global Symbols

Global Symbols

Class	Identifiers	Address	Type...	Initial Value	Identifier Comment...
VAR	ERROR	N/A [Auto]	BOOL	FALSE	
VAR	RUNNING	N/A [Auto]	BOOL	FALSE	
VAR	COMPLETE	N/A [Auto]	BOOL	FALSE	
VAR	BUZZER_OFF	N/A [Auto]	BOOL	FALSE	
VAR	RED	Y0.0	BOOL	FALSE	
VAR	YELLOW	Y0.1	BOOL	FALSE	
VAR	GREEN	Y0.2	BOOL	FALSE	
VAR	BUZZER	Y0.3	BOOL	FALSE	

When users edit the POU of the function block type, they have to create an input pin which receives the state code specified by the main program, and output pins which correspond to the three lights and the buzzer.

Owing to the fact that the corresponding light will be displayed after the function block receives the state code specified by the main program, a CASE statement can be used in the function block. If there is another display mode, the users can add a state code, and insert a condition in the function block.

The users can insert an example of a CASE statement in the working area, and modify the statement. They can also type the structured text shown below in the working area.

Tower_Light

Local Symbols

Class	Identifiers	Address	Type...	Initial Value	Identifier Comment...
VAR_INPUT	MODE	N/A [Auto]	INT	N/A	
VAR_OUTPUT	RL	N/A [Auto]	BOOL	FALSE	
VAR_OUTPUT	YL	N/A [Auto]	BOOL	FALSE	
VAR_OUTPUT	GL	N/A [Auto]	BOOL	FALSE	
VAR_OUTPUT	BR	N/A [Auto]	BOOL	FALSE	

```
0001 CASE MODE OF
0002
0003 (* SM407:0.5S on/ 0.5S off *)
0004
0005    0:  RL := FALSE ;
0006        YL := TRUE  ;
0007        GL := FALSE ;
0008        BR := FALSE ;
0009
0010    1:  RL := FALSE ;
0011        YL := SM407 ;
0012        GL := FALSE ;
0013        BR := FALSE ;
0014
0015    2:  RL := FALSE ;
0016        YL := FALSE ;
0017        GL := TRUE  ;
0018        BR := FALSE ;
0019
0020    3:  RL := SM407 ;
0021        YL := FALSE ;
0022        GL := FALSE ;
0023        BR := FALSE ;
0024
0025    4:  RL := TRUE  ;
0026        YL := FALSE ;
0027        GL := FALSE ;
0028        BR := SM407 ;
0029
0030 END_CASE;
```

When the users edit the POU of the program type, they have to declare the symbol SC, which represents a state code, and the function block instance in the local symbol table. Owing to the fact that there is a precedence mechanism for the control of the tricolor light, an IF statement can be used in the main program. After a state code is specified, the function block must be called. The state code is sent to the function block, and the result of the operation is sent to the digital output

which is specified.
The main program is shown below.

Signal_Control

Local Symbols

Class	Identifiers	Address	Type...	Initial Value	Identifier Comment...
VAR	SC	N/A [Auto]	INT	N/A	
VAR	TL_INS	N/A [Auto]	Tower_Light	N/A	

```
0001 IF ERROR THEN   (*Alarm*)
0002
0003   IF BUZZER_OFF THEN
0004      SC := 3 ; (*Buzzer OFF*)
0005   ELSE
0006      SC := 4 ; (*Buzzer ON*)
0007   END_IF;
0008
0009 ELSIF RUNNING THEN
0010   SC := 2 ;     (*Running*)
0011
0012 ELSIF COMPLETE THEN
0013   SC := 1 ;     (*Finish*)
0014
0015 ELSE
0016   SC := 0 ;     (*Idle*)
0017
0018 END_IF;
0019
0020 TL_INS (
0021
0022    MODE := SC ,
0023      RL => RED ,
0024      YL => YELLOW ,
0025      GL => GREEN ,
0026      BR => BUZZER  );
0027
```

After the POU of the program type and the POU of the function block type are edited, the users can compile the program, download the parameters and the program to the CPU module, and test the program.

MEMO

Chapter 12 Sequential Function Chart

Table of Contents

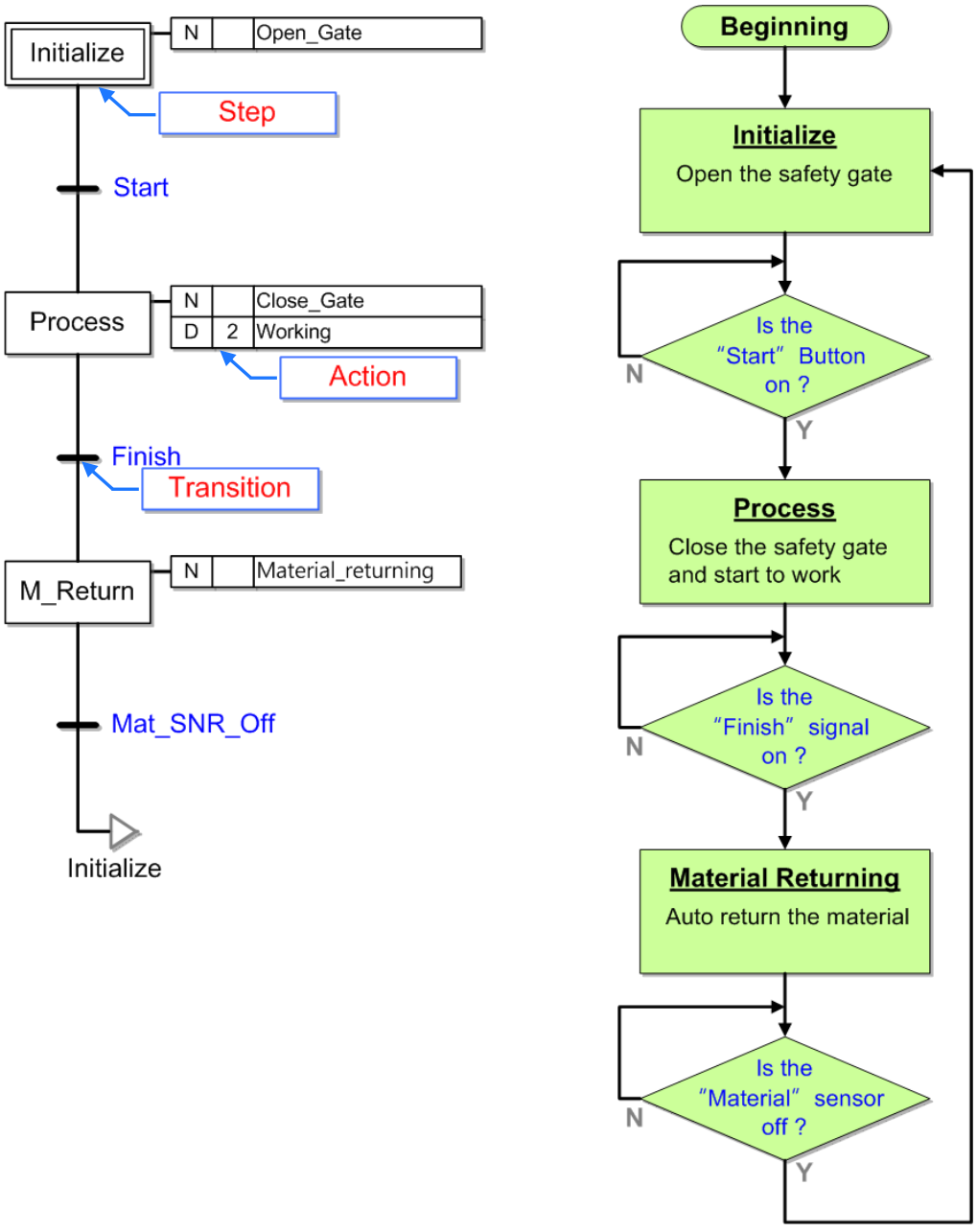
12.1 Knowing Sequential Function Charts 12-2
12.1.1 Structure of a Sequential Function Chart 12-2
12.1.2 Principle of a Sequential Function Chart 12-2
12.2 Sequential Function Chart is ISPSoft 12-3
12.2.1 Steps and Actions 12-3
12.2.2 Transitions 12-6
12.2.3 Simultaneous Divergence and Divergence of Sequence Selection 12-8
12.2.4 Simultaneous Convergence and Convergence of Sequence Selection 12-9
12.2.5 Jump 12-10
12.2.6 Qualifier of an Action 12-12
12.2.6.1 Qualifier Types 12-13
12.2.6.2 Important Points About Qualifying an Action 12-16
12.2.7 Initial Step 12-17
12.3 Creating a Sequential Function Chart in ISPSoft 12-18
12.3.1 Editing Environment 12-18
12.3.2 Creating and Managing Actions and Transitions 12-20
12.3.3 Adding a Step 12-22
12.3.4 Connecting a Transition in Parallel 12-23
12.3.5 Connecting a Step in Parallel 12-24
12.3.6 Creating a Simultaneous Divergence and a Convergence of Sequence Selection 12-25
12.3.7 Creating a Divergence of Sequence Selection and a Simultaneous Convergence 12-26
12.3.8 Inserting a Jump Point 12-28
12.3.9 Assigning Symbols to the Steps and the Transitions 12-29
12.3.10 Specifying an Initial Step 12-29
12.3.11 Assigning Actions 12-29
12.4 Example of a Sequential Function Chart 12-31
12.4.1 Explanation 12-31
12.4.2 Planning Hardware 12-33
12.4.3 Planning a Program 12-33
12.4.4 Creating a Program 12-34

12.1 Knowing Sequential Function Charts

12.1.1 Structure of a Sequential Function Chart

A sequential function chart is one of the five languages defined by IEC 61131-3 standard. It is a powerful graphical technique for describing the sequential behavior of a control program.

A sequential function chart and its corresponding flowchart are shown below. The main components of a sequential function chart are steps with associated actions, transitions with associated logic conditions, and directed links between steps and transitions.

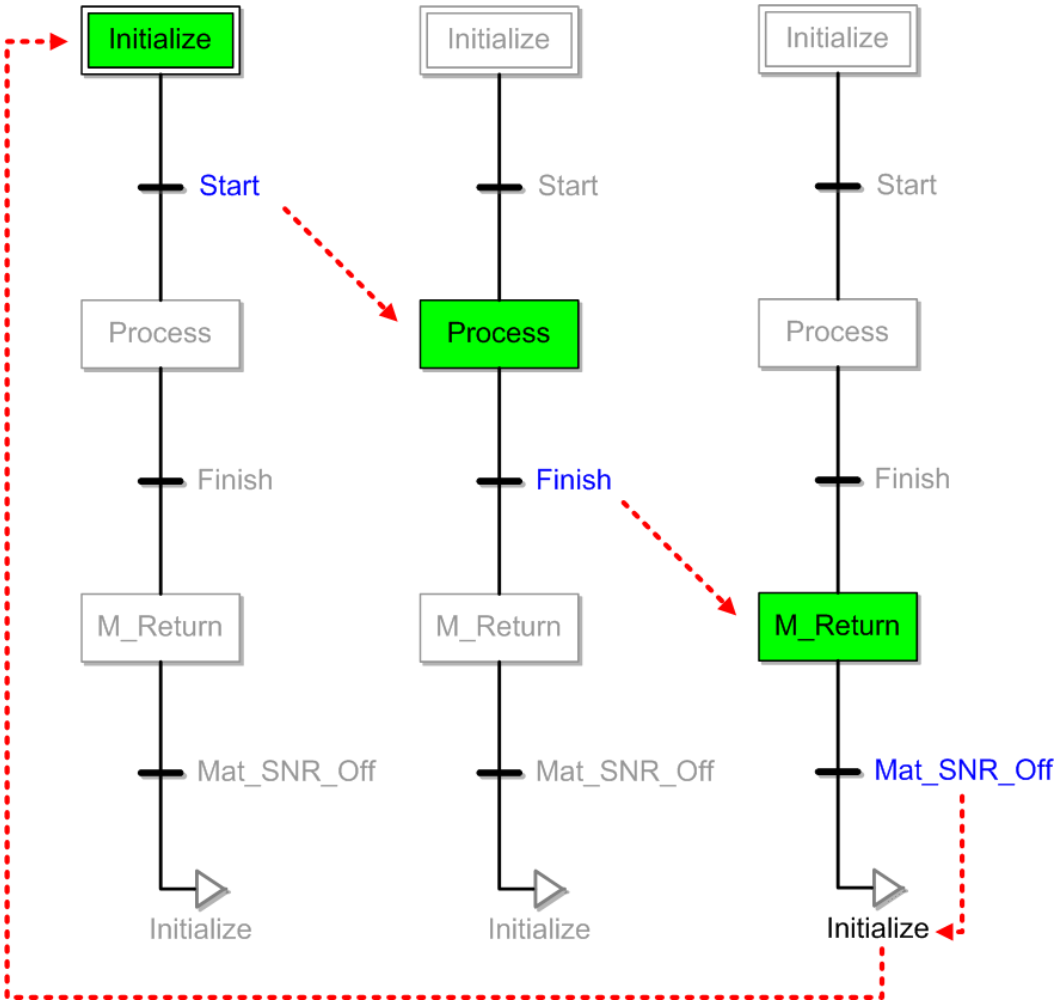


12.1.2 Principle of a Sequential Function Chart

A sequence in a sequential function chart is depicted as a series of steps shown as rectangular boxes connected by vertical lines. Each step represents a particular state of the system being controlled. A transition is associated with a condition which, when true, causes the step before the transition to be deactivated and the step that follows the transition to be activated.

Steps in a sequential function chart can be active or inactive. Actions are only executed for active steps. If a step is inactive, the transition follows it will not affect the execution of the program, and the actions associated with the step will not be executed.

In a sequential function chart in ISPSoft, the steps must alternate with the transitions. If there are two continuous steps or transitions in a sequential function chart, the program can not be compiled.



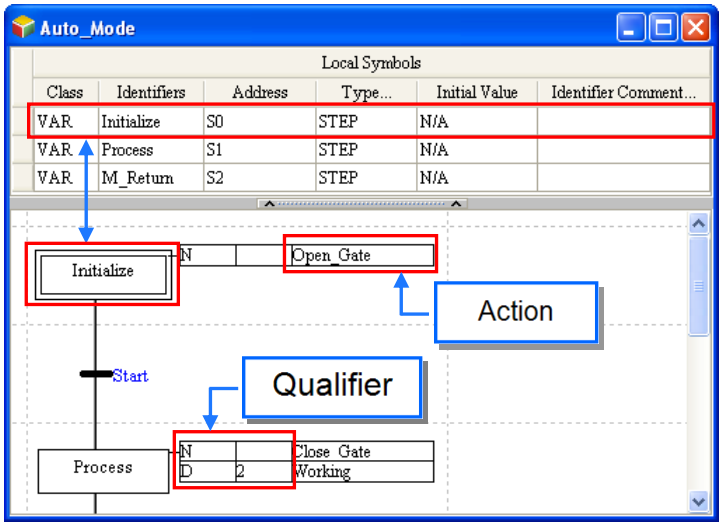
The example above is the structure of a simple sequential function chart. ISPSoft also provides other flexible programming functions, such as the use of divergent and convergent paths, the use of modular actions and transitions, and the qualification of actions. These programming functions will be introduced in the sections below.

12.2 Sequential Function Chart is ISPSoft

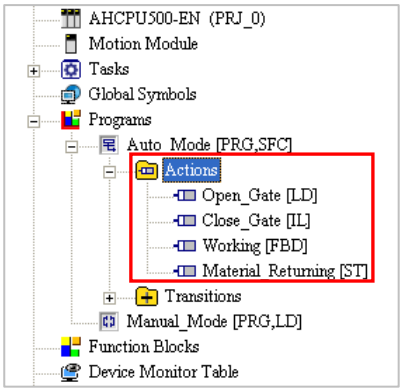
12.2.1 Steps and Actions

Every step in ISPSoft must be assigned a variable whose data type is STEP, and functions as a state flag. If the flag correspond to a step is ON, the step is active. Every symbol whose data type is STEP occupies an S device in the PLC. Please refer to operation manuals for more information about S devices.

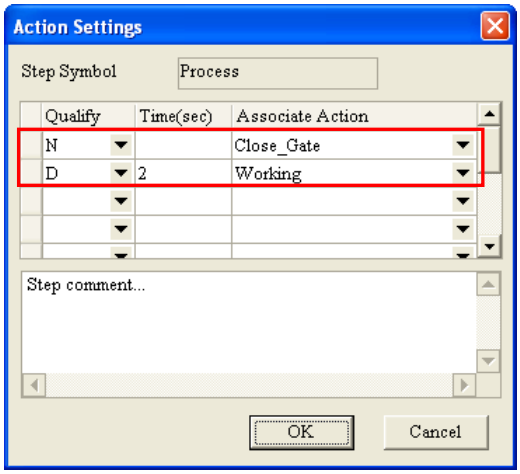
After a step is activated, the actions associated with the step will be executed according to the qualifiers. In ISPSoft, the actions can be modularized, and can be associated with different steps. A step can be associated with more than one action. Besides, every action has a qualifier that determines when the action is executed.

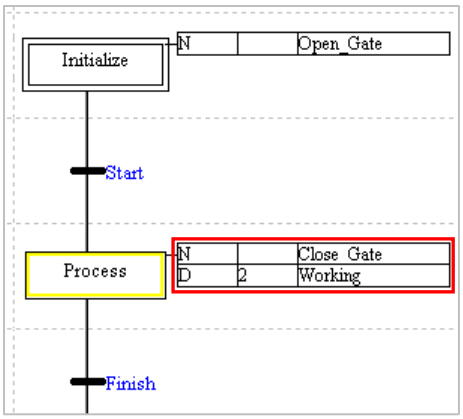


In ISPSoft, an action is a program code, and can be developed by means of a ladder diagram, an instruction list, a function block diagram, or a structured text. The actions which are created are listed in the project management area. The program editing window in which an action can be created differs from the window for a programming language in that there is no local symbol table in the program editing window in which an action can be created. In ISPSoft, the actions and the transitions in a sequential function chart share the same local symbol table.



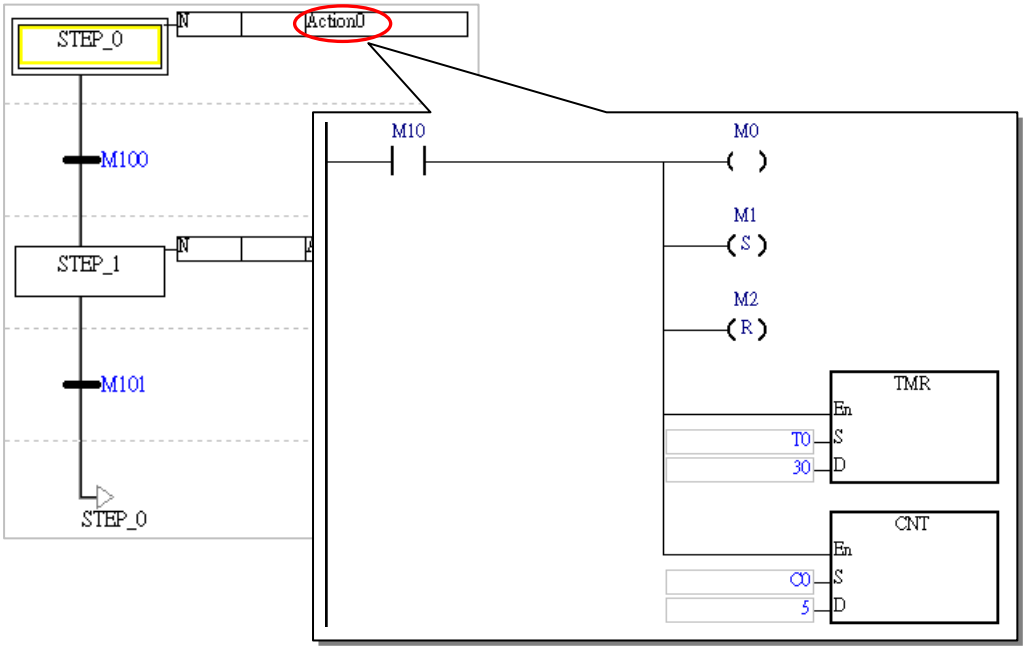
If users want to edit a sequential function chart, they have to create actions, assign the actions to the steps, and assign qualifiers to the actions. The **Action Settings** window and the corresponding action list are shown below.



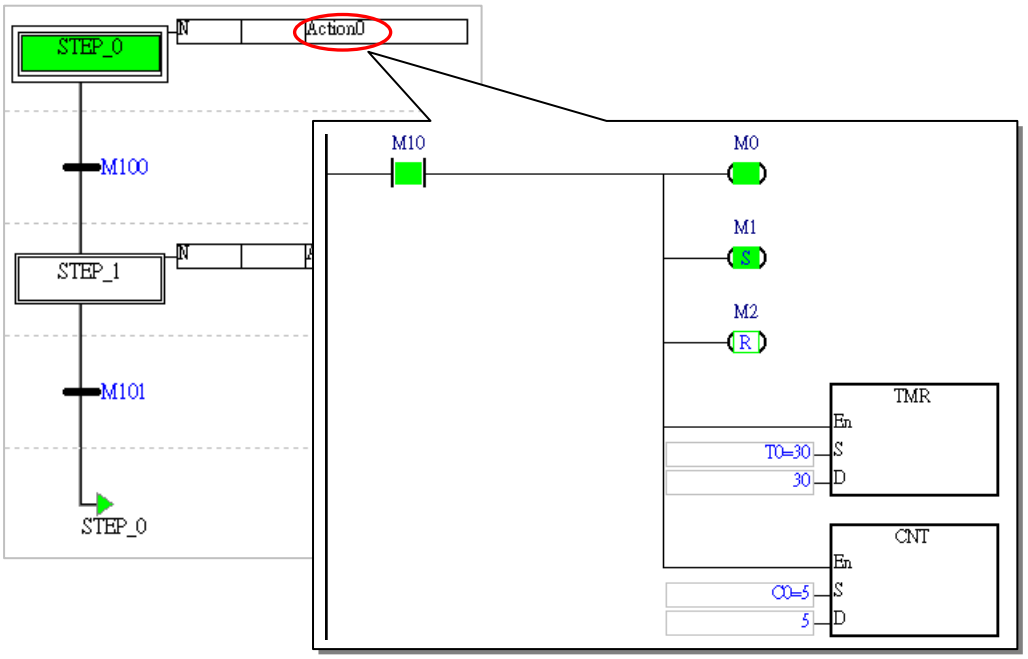


Final scans are important procedures during the execution of a sequential function chart. After the execution of an action stops, the system will automatically execute a procedure to disable all the outputs in the action. If a final scan is executed, the coils driven by the instruction OUT will be OFF, the application instructions and the function blocks will not executed, the timers will be reset, the states of the coils driven by the instruction SET or RESET will remain unchanged, the counters will stop counting, the states of the contacts of the counters will remain unchanged, and the values in the counters will remain unchanged.

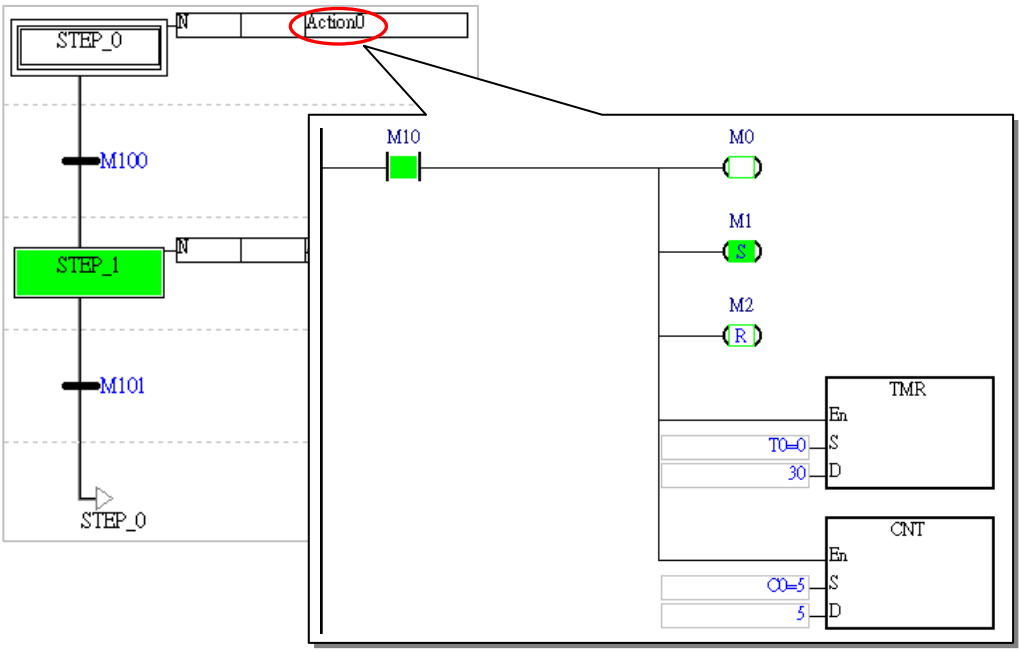
Action0 in the figure below is an action associated with STEP_0, and is an N action. When STEP_0 is deactivated and STEP_1 is activated, the system executes a final scan.



STEP_0 in the figure below is activated. M10 in Action0 is ON, and therefore M0 is ON, M1 is set to ON, M2 is reset to OFF, the value in T0 is 30, and the value in C0 is 5.



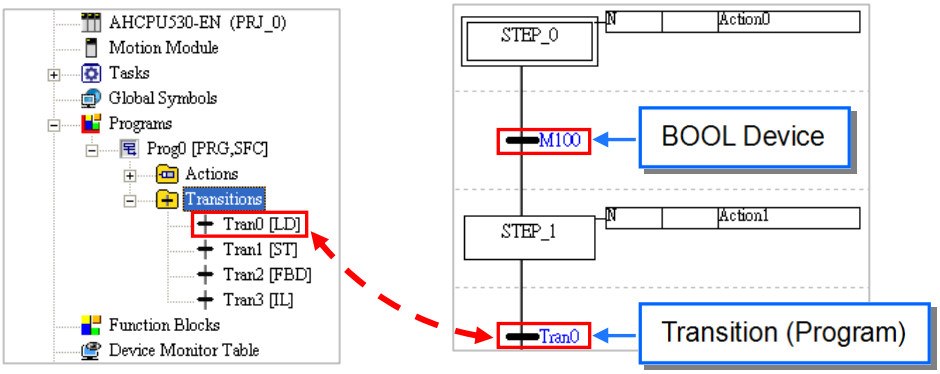
STEP_1 in the figure below is activated. Although M10 is ON, M0 is OFF, and T0 is reset. The state of M1, the state of M2, and the value in C0 remain unchanged.



12.2.2 Transitions

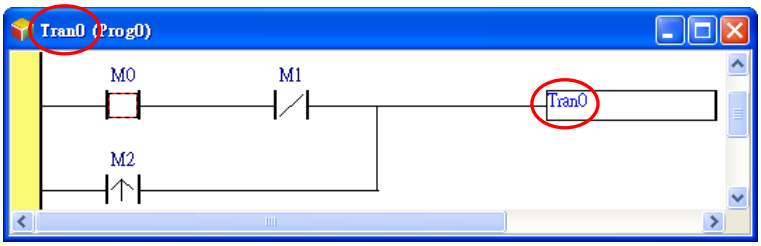
When a transition is true, the active step immediately before the transition becomes inactive and the step immediately after the transition becomes active. A transition can be true anytime, but for a transition to activate the step following it, the step preceding it must be active when the transition is true.

In ISPSoft, a transition can be a Boolean device or symbol, or a program code. If a transition is a logical operation, users can create a transition program, and assign the program to the transition. The transition programs which are created are listed in the project management area.

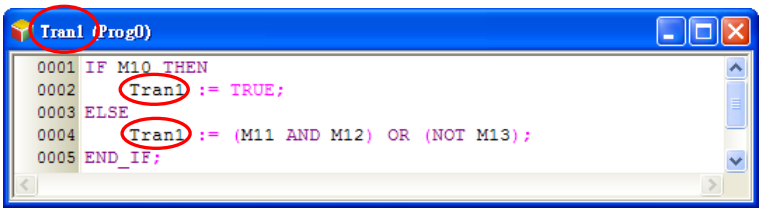


The programming languages which can be used to create transition programs include ladder diagrams, instruction lists, function block diagrams, and structured texts. The Boolean state in a transition program must be sent to a symbol name which is the same as the transition program name, whether the programming language used to create the transition program is a ladder diagram, an instruction list, a function block diagram, or a structured text. The symbol does not need to be declared. However, if there is a symbol name in a symbol table which is the same as a transition program name, an error occurs when the program is compiled. Besides, applied instructions and function blocks can not be used in a transition program, but comparison contacts and block logic instructions (NP, PN, and INV) can be used in a transition program.

If the programming language used to create a transition program is a ladder diagram, the program code of the transition program must be composed of one network, and the output contact must be assigned a symbol name which is the same as the transition program name. Besides, multiple outputs are not allowed in a transition program created by means of a ladder diagram.

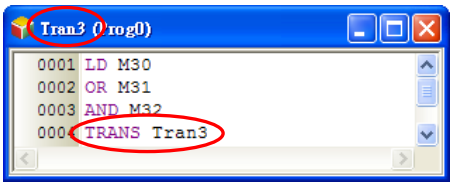


If the programming language used to create a transition program is a structured text, there is no limit on the number of lines of program code. However, a Boolean state in a transition program must be assigned to a symbol name which is the same as the transition program name.



If the programming language used to create a transition program is a function block diagram, the program code of the transition program must be composed of one network, and the output contact must be assigned a symbol name which is the same as the transition program name. Besides, multiple outputs are not allowed in a transition program created by means of a function block diagram.

If the programming language used to create a transition program is an instruction list, there is no limit on the number of lines of program code. OUT, SET, RST, and applied instructions can not be used in a transition program created by means of an instruction list. Besides, the output instruction in a transition program created by means of an instruction list must be TRANS, and the operand for TRANS must be assigned a symbol name which is the same as the transition program name.



When users create a transition program, a symbol name which is the same as the transition program name is added to the program code of the transition program. The users only have to create a conditional operation. Owing to the fact that the actions and the transitions in a sequential function chart share the same local symbol table, there are not local symbol tables in the program editing windows for the transitions.

Users can not create a complex transition program. If a transition condition must undergo a complex operation, users can create the complex operation in an action associated with a step, create a condition flag in the action, and assign the flag to the transition following the step.

Although users do not need to declare the symbols in the transition programs, the system assigns memories which function as flags to the symbols. As a result, users have to make sure that the symbols in the transition programs can be assigned a specific logic state. Otherwise, an error will

occur if the program is executed.

Take the transition program which is created by means of a structured text below for example. During the execution of the transition program for the first time, Tran4 will be ON if the result of (M0 AND M1) is ON. Since Tran4 is ON, it passes control to the next step. After the control is passed to the step preceding the transition program again, the transition program will be executed again. If the result of (M0 AND M1) is OFF, the statement following **THEN** will not be executed. Tran4 will still be ON because it is not updated. The system will evaluate the transition condition as TRUE, and the control will be passed to the next step.

Adding an **ELSE** section to the transition program below is a better way to ensure that the symbol Tran4 is assigned a specific logic state. Users also have to pay attention to the same problem in a CASE sttructure.

```
0001 IF (M0 AND M1) THEN
0002     Tran4 := TRUE;
0003 END_IF;
```

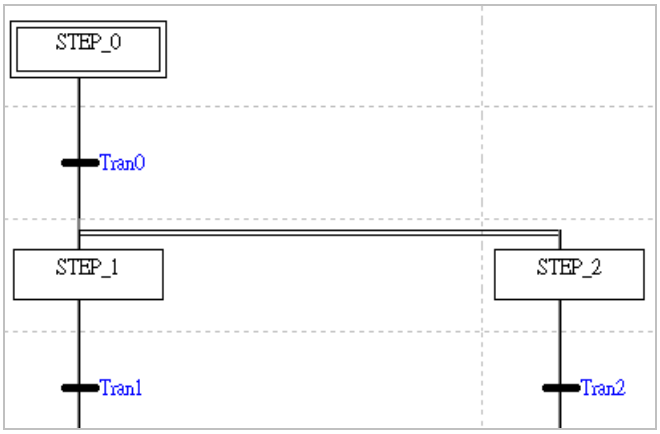
12.2.3 Simultaneous Divergence and Divergence of Sequence Selection

Users can create divergent paths in a sequential function chart. There are two types of divergences. One is a simultaneous divergence, and the other is a divergence of sequence selection

- **Simultaneous divergence**

  A simultaneous divergence is shown below. The divergent steps are connected by a double line. The characteristic of a simultaneous divergence is that the divergent steps share the same transition. When the transition is true, the steps are active.

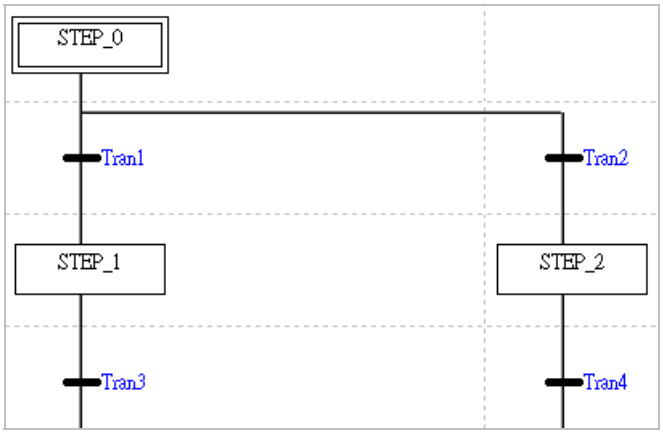
  When Tran0 is true, STEP_0 becomes inactive, and STEP_1 and STEP_2 become active.



- **Divergence of sequence selection**

  A divergence of sequence selection is shown below. The characteristic of a divergence of sequence selection is that the divergent steps have their own transitions. When one transition is true, the step following the transition becomes active, and the other transitions are not evaluated. Consequently, if there are divergent paths, only one path is taken at a time. Besides, the transitions are evaluated from left to right. If several transitions are true simultaneously, only the step following the leftmost transition will be activated.

  When Tran2 is true, STEP_0 becomes inactive, and STEP_2 becomes active. Tran1 is not evaluated, and STEP_1 remains inactive.



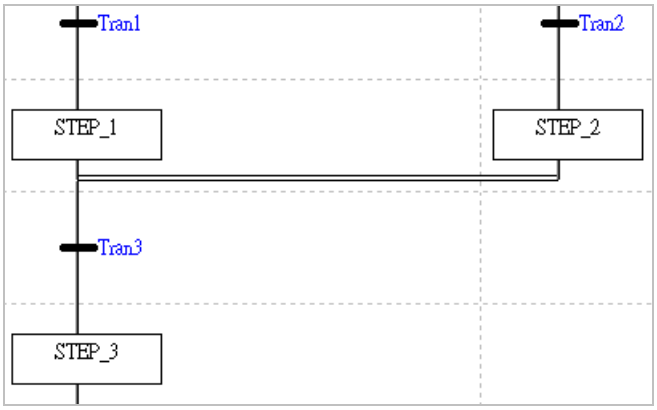
12.2.4 Simultaneous Convergence and Convergence of Sequence Selection

The divergent paths in a sequential function chart must be converged. There are two types of convergences. One is a simultaneous convergence, and the other is a convergence of sequence selection.

- **Simultaneous convergence**

  A simultaneous convergence is shown below. The convergent steps are connected by a double line. The characteristic of a simultaneous convergence is that the convergent steps are connected to the same transition. The transition will not be evaluated until the convergent steps preceding the transition are activated.

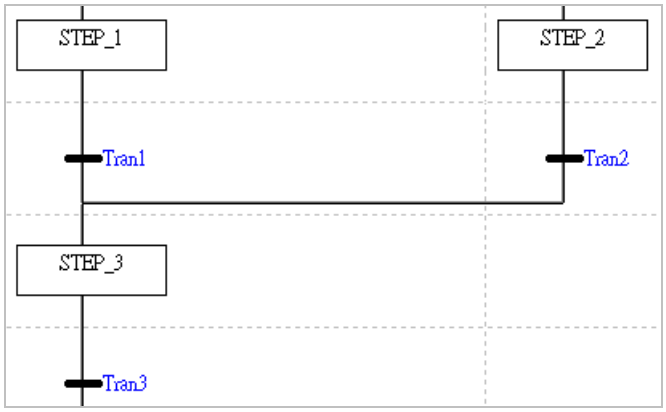
  After STEP_1 and STEP_2 are activated, Tran3 will be evaluated. When Tran3 is true, STEP_1 and STEP_2 become inactive, and STEP_3 becomes active.



- **Convergence of sequence selection**

  A convergence of sequence selection is shown below. The characteristic of a convergence of sequence selection is that the convergent steps have their own transitions. When one transition is true, the next step becomes active. If there is another convergent step which is activated, the step will be active until the transition following the step is true. When the transition is true, the step following the transition becomes active. Consequently, if a convergence of sequence selection is used in a sequential function chart, several steps following the convergence may be activated. If users use a convergence of sequence selection in a sequential function chart, they have to make sure that the program is not confused.

  When Tran1 is true, STEP_1 becomes inactive, and STEP_3 becomes active. If Tran2 is true, STEP_2 will be inactivated, and STEP_3 will be activated again.

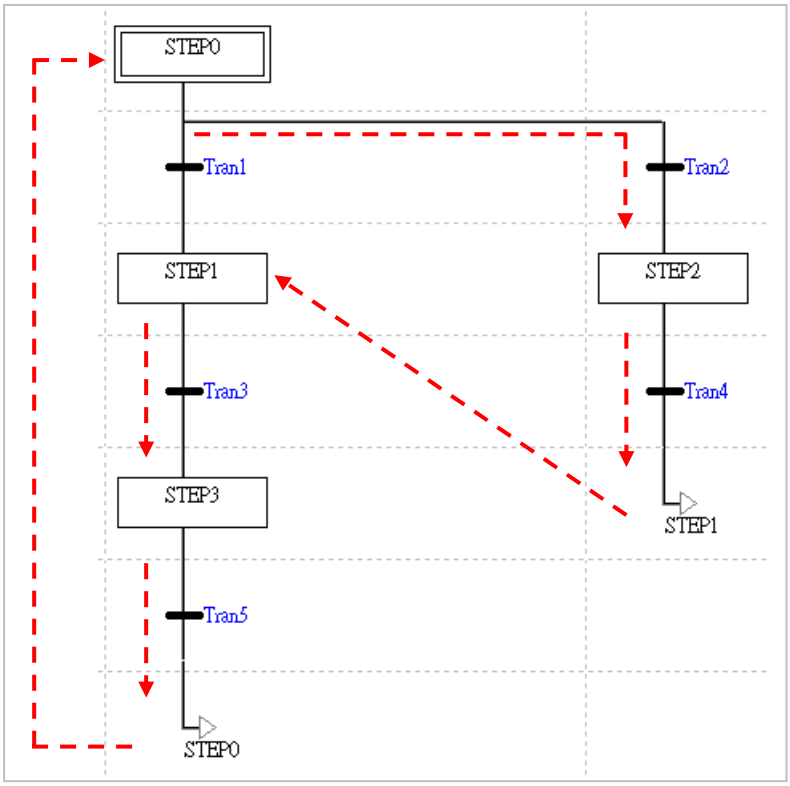


12

12.2.5 Jump

The use of a jump structure in a sequential function chart results in the passing of control to a step which is specified. There is at least one jump point indicating that the passing of control will occur once the transition preceding the jump point is true in a sequential function chart. Users can use jump structures quite freely. If they use a jump structure in a sequential function chart, they have to make sure that the program is not confused.

The program below is a sequential function chart where there is a divergence of sequence selection. If Tran2 is true, control will be passed to STEP2. When Tran4 is true, it passes control to STEP1 through a jump. When Tran5 is true, it passes control to STEP0.

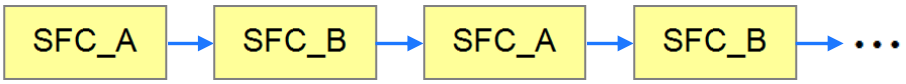


If users use a jump structure in a sequential function chart, the step specified must be a step in the same sequential function chart. If the step specified is not in the same sequential function chart, and is declared in a local symbol table or the global symbol table, an error occurs when the program is compiled

If control needs to be passed from a step in a sequential function chart to a step in another sequential function, users can make use of the characteristics of steps and the characteristics of

transitions. Users can declare the step or the transition which is the destination of a jump in the global symbol table. When the condition of the jump is met, the step or the transition declared is set to ON. The condition of the jump can be created in an action.

The figure below shows that **SFC_A** is executed first. **SFC_B** will be executed after **SFC_A** is executed, and **SFC_A** will be executed after **SFC_B** is executed. User can achieve the jumps by means of the characteristics of transitions.

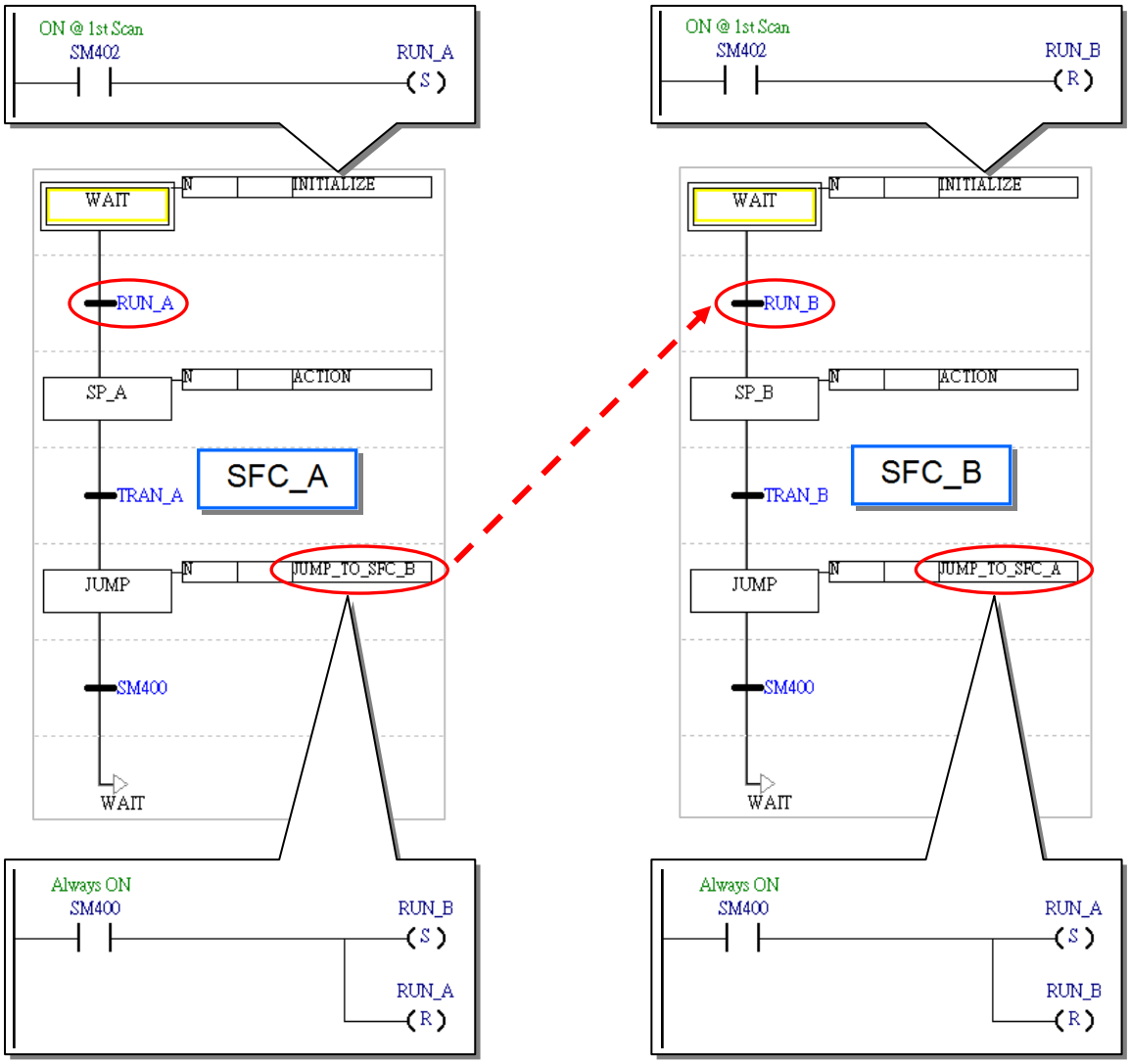


The steps **SP_A** and **SP_B** in the example below are used to express the sequential function chart which will be executed. They are preceded and followed by steps. Besides, the transitions **RUN_A** and **RUN_B** are declared in the global symbol table.

When the program is executed, the step **WAIT** in SFC_A and the step **WAIT** in SFC_B are activated, and the action **INITIALIZE** associated with **WAIT** in SFC_A and the action **INITIALIZE** associated with **WAIT** in SFC_B are executed. **INITIALIZE** in SFC_A sets the transition **RUN_A** to ON during the first scan cycle, and therefore control is passed to **SP_A**. **INITIALIZE** in SFC_B sets the transition **RUN_ B** to OFF during the first scan cycle, and therefore control is not passed to **SP_B**.

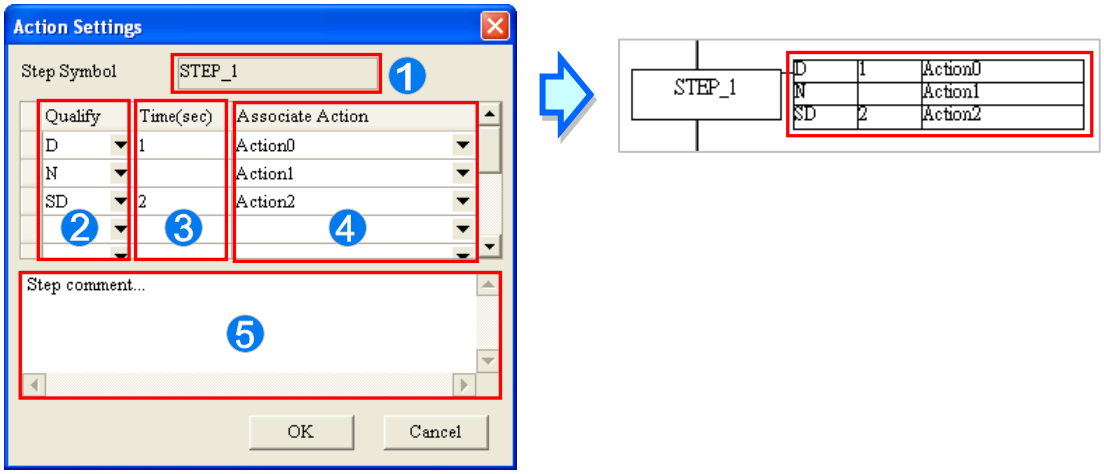
When the step **JUMP** in SFC_A is activated, the action associated with **JUMP** sets the transition **RUN_B** to ON, and resets **RUN_A** to OFF. When **RUN_B** is set to ON, control is passed to the step **SP_B**. Besides, the transition following **JUMP** in SFC_A is a special relay which is always ON, and therefore control is passed to the step **WAIT**. Owing to the fact that **RUN_A** is set to OFF, control is not passed to **SP_A**. **RUN_A** will not be set to ON until the action associated with **JUMP** in SFC_B is executed.

The use of **WAIT** and **JUMP** in SFC_A serves to set **RUN_B** to ON, and the use of **WAIT** and **JUMP** in SFC_B serves to set **RUN_A** to ON. **WAIT** and **JUMP** in SFC_A and **WAIT** and **JUMP** in SFC_B actually have little significance. If the program is applied to an AH500 series CPU module, users can assign the qualifier **P** or **P1** to the actions associated these steps. Please refer to section 12.2.6 for more information.



12.2.6 Qualifier of an Action

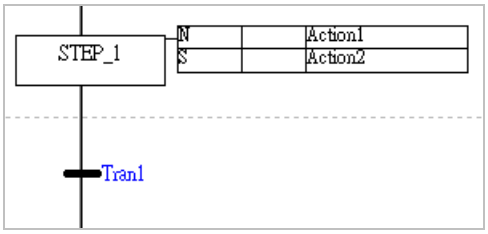
In ISPSoft, users can define the execution of an action by qualifying the action. However, users can not qualify the actions in the program in a DVP series PLC. Consequently, the actions in the program in a DVP series PLC are N actions.



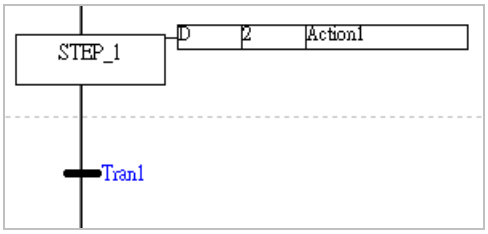
1. The step name is displayed here.
2. Users can select qualifiers for the actions associated with the step.
3. Users can specify qualifiers related to time.
4. Users can select actions which are qualified.
5. Users can make comments on the step.

12.2.6.1 Qualifier Types

- **N (Normal)**
  When a step is activated, the N action associated with the step is executed. When the step is inactivated, the system executes a final scan to disable all the outputs in the N action. However, the outputs which use the instructions similar SET are not disabled. The actions in the program in a DVP series PLC are N actions.
- **S (Set)**
  When a step is activated, the S action associated with the step is executed. Even if the step is inactivated, the S action will still be executed.
  When STEP_1 in the figure below is activated, Action1 and Action2 are executed. When Tran1 is true, STEP_1 becomes inactive. Once STEP_1 is inactivated, the execution of Action1 will stop, the system will execute a final scan, and Action2 will still be executed.



- **D (Delay)**
  The D action associated with a step is not executed at the time when the step is activated. The execution of the D action is delayed for a certain amount of time. If the transition following the step becomes true before the D action is executed, the step will be inactivated, control will be passed to the next step, and the D action will not be executed. When the step is activated again, the delay is measured again.
  Action1 in the figure below is not executed at the time when STEP_1 is executed. The execution of Action1 is delayed for two seconds. If STEP_1 is inactivated before Action1 is executed, Action1 will not be executed.



- **SD (Set Delay)**
  The SD action associated with a step is not executed at the time when the step is activated. The execution of the SD action is delayed for a certain amount of time. If the SD action is executed, it will still be executed after the step is inactivated. Besides, if the transition following the step becomes true before the SD action is executed, control will be passed to the next step, and the delay will still be measured. A certain amount of time has elapsed before the SD action is executed.
- **DS (Delay Set)**
  A DS action is similar to a SD action. However, if the transition following a step becomes true

before the SD action associated with the step is executed, the delay will not be measured, and the SD action will not be executed. When the step is activated again, the delay is measured again. If the DS action is executed, it will still be executed after the step is inactivated.

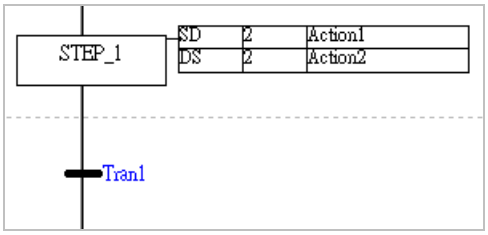
Action1 and Action2 are not executed at the time when STEP_1 is activated. The execution of Action1 and the execution of Action2 are delayed for two seconds. There are two situations which need to be considered.

(a) STEP_1 is activated for more than two seconds.

Action1 and Action2 will be executed after STEP_1 has been executed for two seconds. Even if STEP_1 is inactivated, Action1 and Action2 will still be executed.

(b) STEP_1 is activated for less than two seconds.

After control is passed to the next step, the delay of Action1 will still be measured. Two seconds has elapsed before Action1 is executed. However, after the control is passed to the next step, the delay of Action2 will not be measured, and Action2 will not be executed.



- **L (Limit)**

When a step is activated, the L action associated with the step is executed. If the L action has been executed for a certain amount of time before control is passed to the next step, the execution of the L action will stop automatically, and the system will execute a final scan. If the control is passed to the next step before the time specified has elapsed, the execution of the L action will stop.

- **SL (Set Limit)**

When a step is activated, the SL action associated with the step is executed. If the SL action has been executed for a certain amount of time before control is passed to the next step, the execution of the SL action will stop automatically, and the system will execute a final scan. If the control is passed to the next step before the time specified has elapsed, the execution of the SL action will stop after the time specified has elapsed.

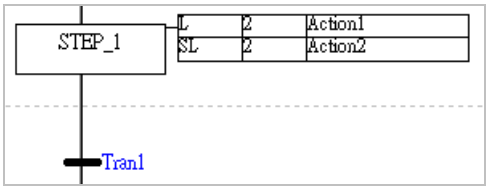
When STEP_1 is activated, Action1 and Action2 are executed. There are two situations which need to be considered.

(a) STEP_1 is activated for more than two seconds.

After Action1 and Action2 have been executed for two seconds, the execution of Action1 and the execution of Action2 will stop, and the system will execute final scans.

(b) STEP_1 is activated for less than two seconds.

After control is passed to the next step, the execution of Action1 will stop, and the system will execute a final scan. The execution of Action2 will stop after two seconds have elapsed.

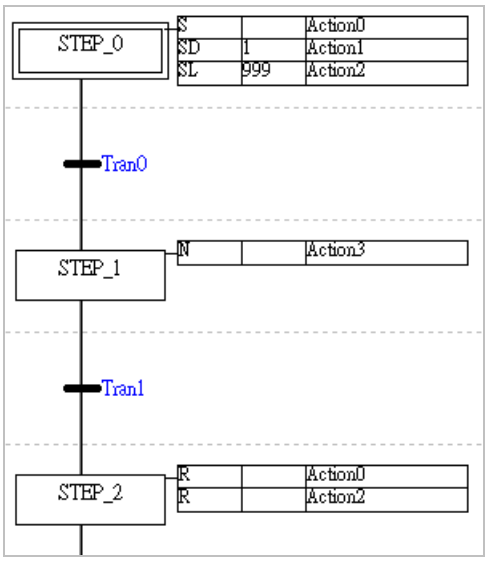


- **R (Reset)**

When a step is activated, the R action related to the step is not executed, and the system executes a final scan. An action qualified by S, SD, DS, or SL will not be executed if it is qualified by R.

Suppose that Action0, Action1, and Action2 are executed during the activation of STEP_0, and will be executed until control is passed to STEP_2. After the control is passed to STEP_2, the execution of Action0 and the execution of Action2 will stop, and the system will execute final

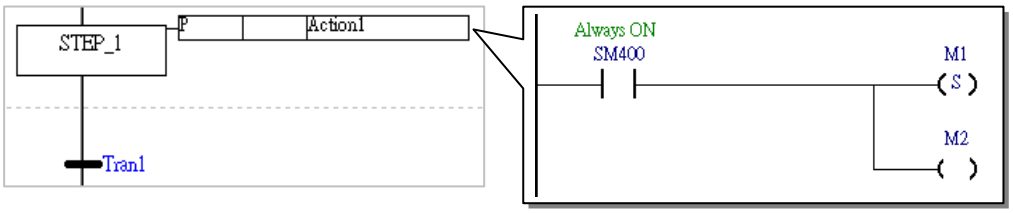
scans. Owing to the fact that Action1 is not qualified by R, it will still be executed after the control is passed to STEP_2.



- **P (Pulse)**

  During the first scan cycle, the P action associated with a step is executed when the step is activated. During the second scan cycle, even if the step is activated, the outputs will be disabled, and the system will execute a final scan.

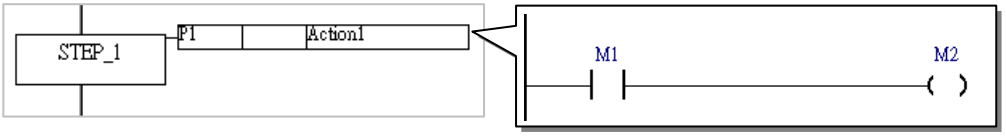
  During the first scan cycle, Action1 is executed when STEP_1 is activated. Owing to the fact that SM400 is ON, M1 and M2 are ON. During the second scan cycle, the system inactivates Action1. Although SM400 is still ON, M2 becomes OFF. M1 is driven by the instruction SET, and therefore it remains ON.



- **P1 (Raising Pulse)**

  The P1 action associated with a step is executed only when the step is activated during the first scan cycle. Besides, the system will not execute a final scan after the execution of the P1 action stops. Consequently, although an action where the instruction OUT or TMR is used is allowed to be qualified by P1, a warning appears when the program is compiled.

  During the first scan cycle, Action1 is executed when STEP_1 is activated. If M1 is ON, M2 will be ON. Action1 is executed only when STEP_1 is activated during the first scan cycle. Besides, the system will not execute a final scan after the execution of Action1 stops. Consequently, the state of M2 will not change even if the state of M1 changes.

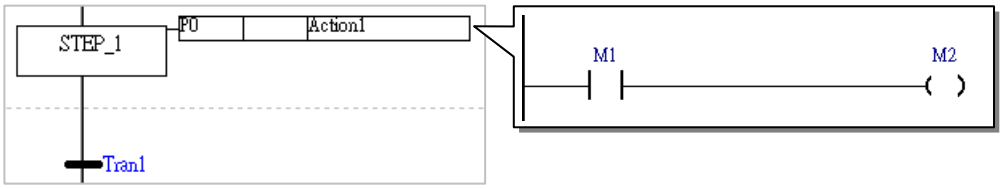


- **P0 (Falling Pulse)**

  The P0 action associated with a step is executed only when the step is inactivated during the first scan cycle. Besides, the system will not execute a final scan after the execution of the P0 action

stops. Consequently, although an action where the instruction OUT or TMR is used is allowed to be qualified by P1, a warning appears when the program is compiled.

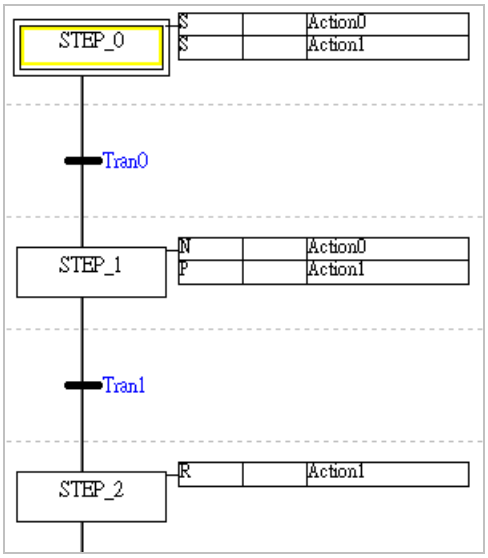
When STEP_1 is activated, Action1 is not executed. Action1 will not be executed until Tran1 is true. If M1 is ON, M2 will be ON. Action1 is executed only when STEP_1 is inactivated during the first scan cycle. Besides, the system will not execute a final scan after the execution of Action1 stops. Consequently, the state of M2 will not change even if the state of M1 changes.



12.2.6.2 Important Points About Qualifying an Action

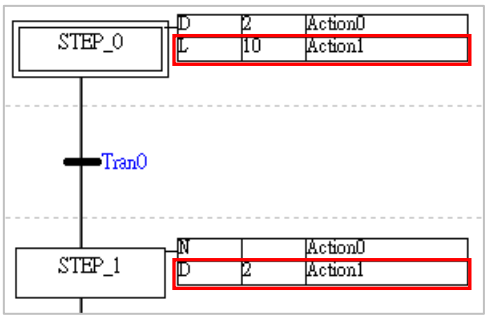
- Once an action qualified by S, SD, DS, or SL is executed, it will still be executed even if it is qualified by N or P. The execution of the action stops only when the action is qualified by R. If there is an action which is qualified by S, SD, DS, or SL, users have to check whether they have to assign R to the action to stop the execution of the action.

  When STEP_0 is activated, Action0 and Action1 are executed. After control is passed to STEP_1, Action0 and Action1 will still be executed even if they are qualified by N and P respectively. The execution of Action1 will stop after control is passed to STEP_2. Action0 is never qualified by R, and therefore it will be executed all the time.



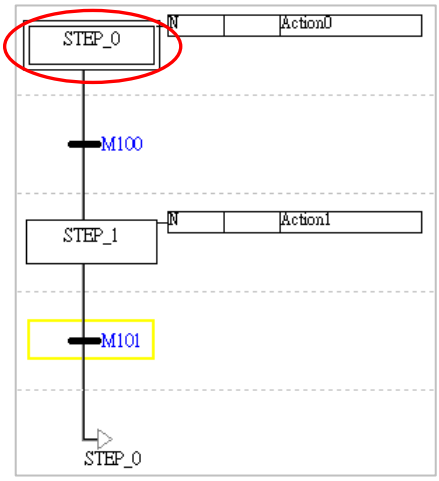
- An action can only be assigned a qualifier related to time once. In other words, if an action is qualified by D, SD, DS, L, or SL, the action can not be assigned another qualifier related to time.

  Action0 and Action1 which are associated with STEP_0 are qualified by D and L respective. Action0 associated with STEP_1 is qualified by N, a qualifier not related to time. The usage is correct. However, Action1 associated with STEP_1 is assigned a qualifier related time again. An error will occur if Action1 is compiled.



12.2.7 Initial Step

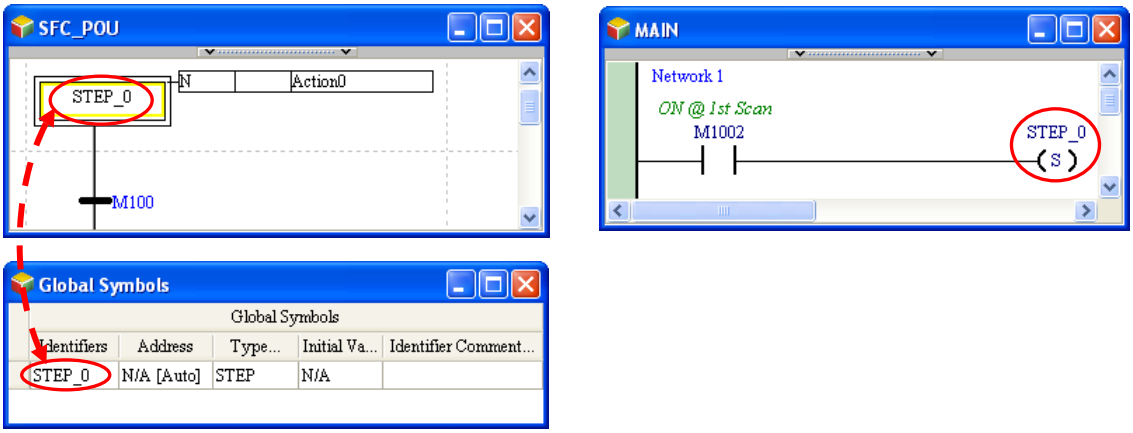
During the execution of a sequential function chart, every step is activated in turn and cyclically. It is necessary to define a step as an initial step which is activated when the program is executed for the first time. In ISPSoft, the initial step in a sequential function chart is drawn as a box with a double line. Besides, there is only one initial step in a sequential function chart.



Any step in a sequential function chart in an AH500 series CPU module can be defined as the initial step in the sequential function chart. When a sequential function chart in an AH500 series CPU module is executed for the first time during the operation of the CPU module, the initial step in the sequential function chart is the step which is activated first.

If there is a sequential function chart in a DVP series PLC, the sequential function chart will not be executed until the instruction SET used in another POU activate a step in the sequential function chart. Users can not define a step in a sequential function chart in a DVP series PLC as the initial step in the sequential function chart, and the first step in a sequential function chart in a DVP series PLC is always the initial step in the sequential function chart. The real initial step in a sequential function chart in a DVP series PLC is the device or the symbol driven by the instruction SET used in another POU. An example of executing a sequential function chart in DVP series PLC is shown below.

Users have to determine which step in a sequential function chart is the initial step in the sequential function chart, declare the step in the global symbol table, and apply the instruction SET to the step in another POU. In the program code where the instruction SET is applied to the step, the condition contact or the logical output preceding the step must be prevented from becoming ON again during the execution of the program. Otherwise, the step will be ON, and the sequential function chart will be executed incorrectly. The special relay M1002 is highly recommended. The characteristic of M1002 is that M1002 is ON during the first scan cycle.



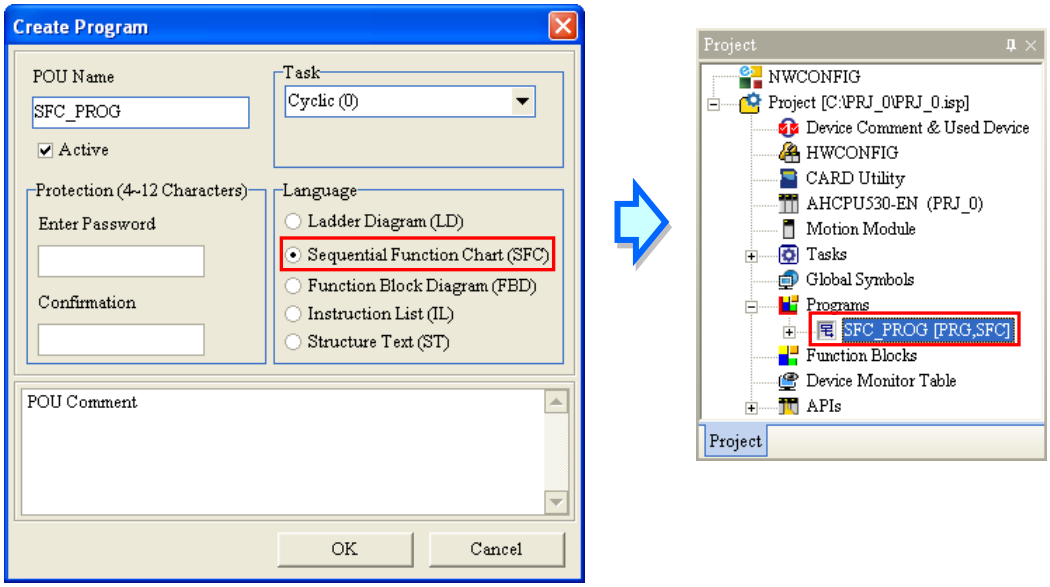
12.3 Creating a Sequential Function Chart in ISPSoft

12.3.1 Editing Environment

Select the **Sequential Function Chart (SFC)** option button in the **Language** section in the **Create Program** window. Please refer to section 5.4.1 for more information about creating a sequential function chart. In addition, users have to pay attention to the following two points when they create a sequential function chart.

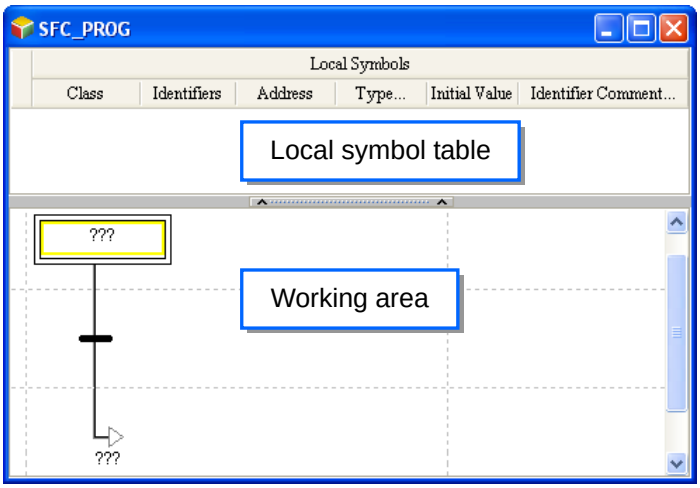
(a) The programming language used to create a POU of the function block type can not be a sequential function chart.

(b) If the programming language used to create a POU of the program type is a sequential function chart, the POU of the program type can only be assigned to a cyclic task.



The environment in which a sequential function chart can be edited is shown below. The table at the upper part of the window is a local symbol table, and the area at the lower part of the window is a working area. The working area is like a checkerboard.

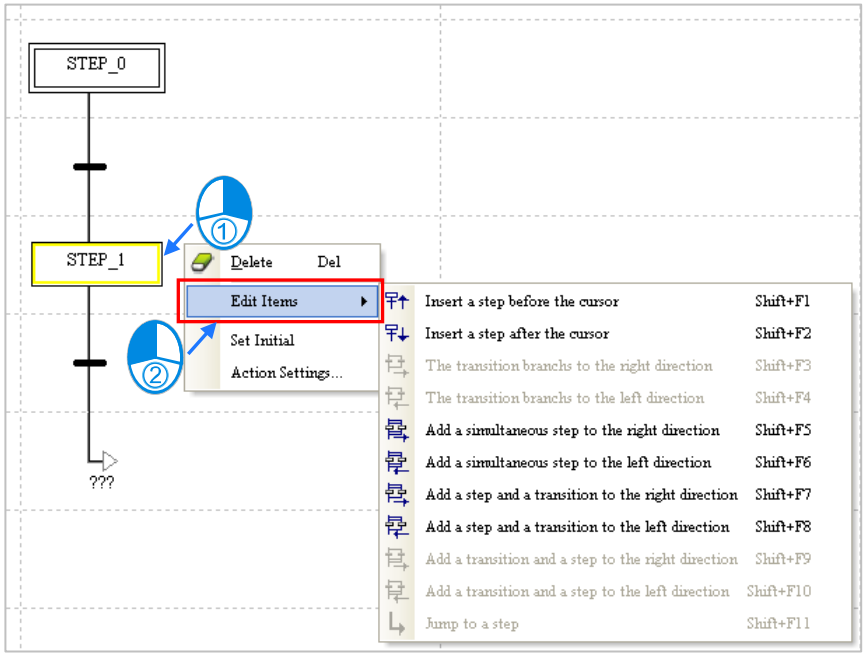
The steps and the transitions are put in squares. If users click an object or the square where the object is put, the object will be selected, and a yellow frame will appear.



After a program editing window in which a sequential function chart can be created is opened, the corresponding toolbar will appear in the ISPSoft window. The functions are described below.

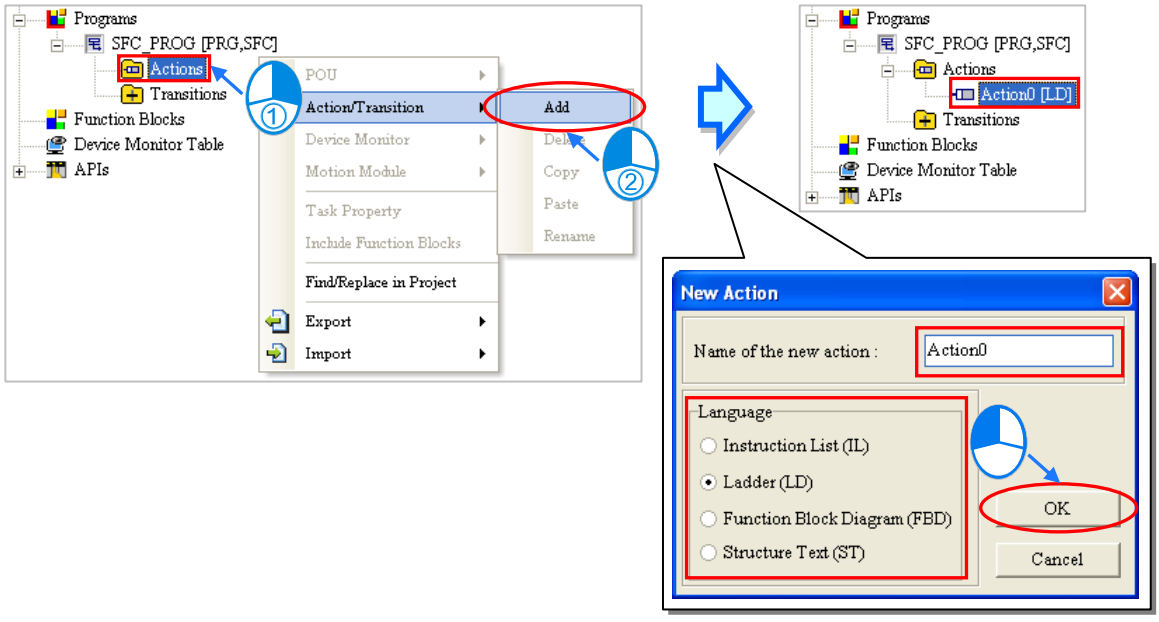
Icon	Keyboard shortcut	Function
	Shift+F1	Adding a step to the top of the object selected.
	Shift+F2	Adding a step to the bottom of the object selected
	Shift+F3	Connecting a transition to the transition selected in parallel, and putting the transition at the right side of the transition selected
	Shift+F4	Connecting a transition to the transition selected in parallel, and putting the transition at the left side of the transition selected
	Shift+F5	Connecting a step to the step selected in parallel, and putting the step at the right side of the step selected
	Shift+F6	Connecting a step to the step selected in parallel, and putting the step at the left side of the step selected
	Shift+F7	Creating a simultaneous divergence and a convergence of sequence selection at the right side of the step selected
	Shift+F8	Creating a simultaneous divergence and a convergence of sequence selection at the left side of the step selected
	Shift+F9	Creating a divergence of sequence selection and a simultaneous convergence at the right side of the transition selected
	Shift+F10	Creating a divergence of sequence selection and a simultaneous convergence at the left side of the transition selected
	Shift+F11	Inserting a jump point

Users can edit the sequential function chart by means of the toolbar. They can also edit the sequential function chart by clicking an object in the working area, and clicking an item on the context menu. To ensure that the sequential function chart is a complete and legal sequential function chart, the editing function which can be used varies with the object selected.

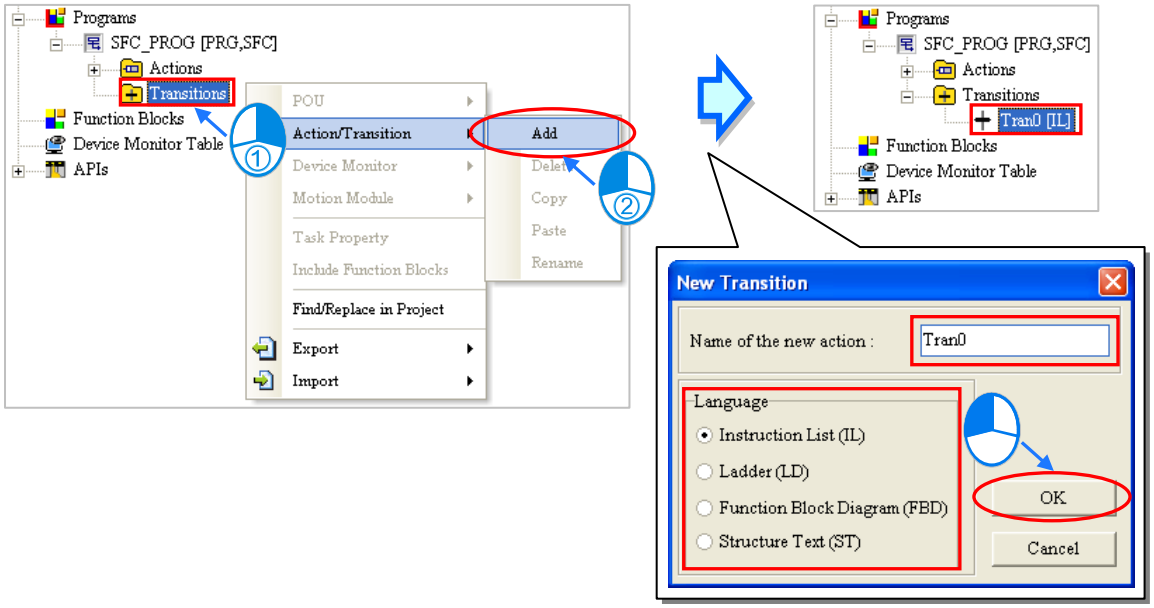


12.3.2 Creating and Managing Actions and Transitions

If users want to create an action, they have to unfold the sequential function chart section in the project management area, select **Actions**, right-click **Actions**, point to **Action/Transition** on the context menu, click **Add**, type an action name in the **New Action** window, and select a programming language in the **New Action** window.



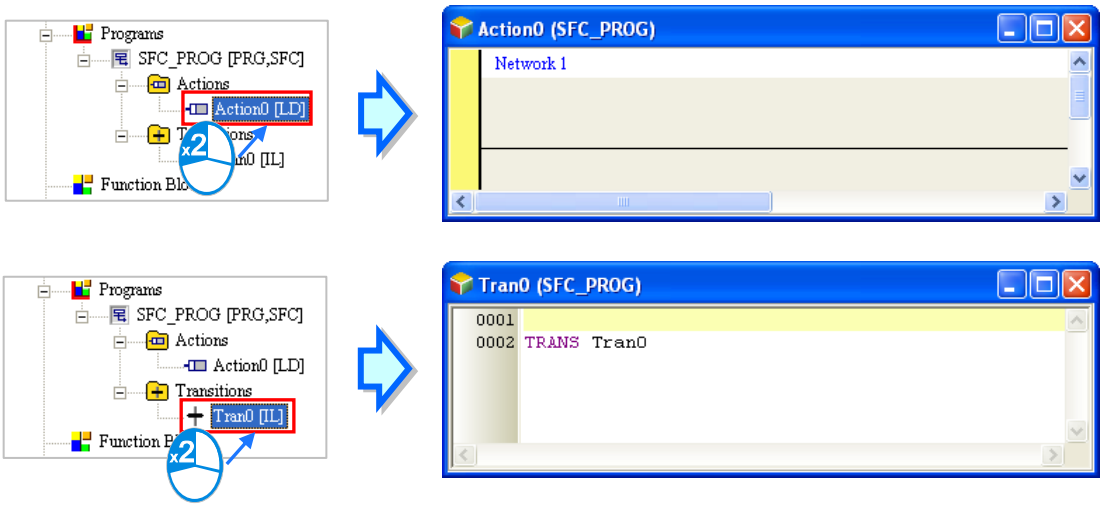
If the users want to create a transition, they have to unfold the sequential function chart section, selection Transitions, right-click Transitions, point to **Action/Transition** on the context menu, click **Add**, type a transition name in the **New Transition** window, and select a programming language in the **New Transition** window.



12

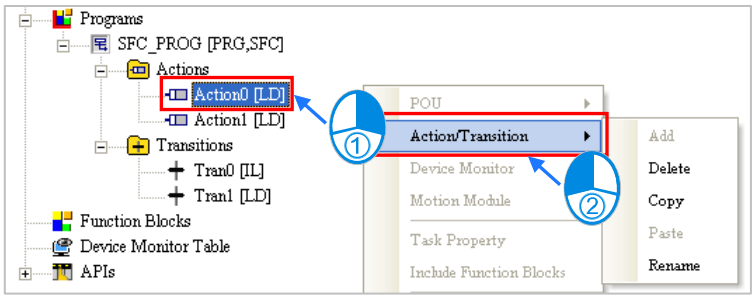
After the users double-click the action or the transition created in the project management area, the corresponding program editing window will appear. In the window for the transition, there is a symbol name which is the same as the transition program name. Please refer to section 12.2.1~section 12.2.2 for more information about editing actions and transitions.

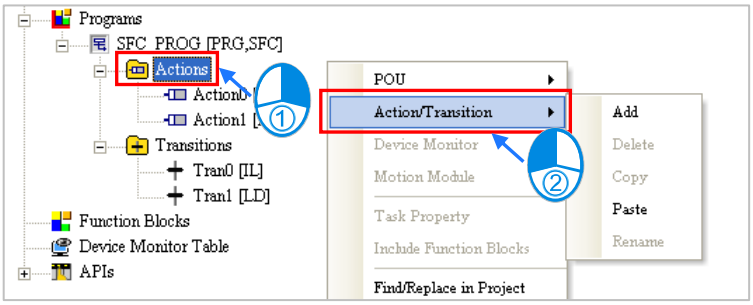
The action and the transition are edited in much the same way as a general POU is. However, there are no local symbol tables in the window for the action and the window for the transition. If users want to declare a local symbol table, they have to declare it in the window for the sequential function chart. Besides, the actions and the transitions in a sequential function chart share the same local symbol table. Please refer to chapter 8~chapter 11 for more information about programming languages.



***. If the transition create is assigned to a transition in the sequential function chart, the window for the transition will be opened after users double-click the transition in the sequential function chart.**

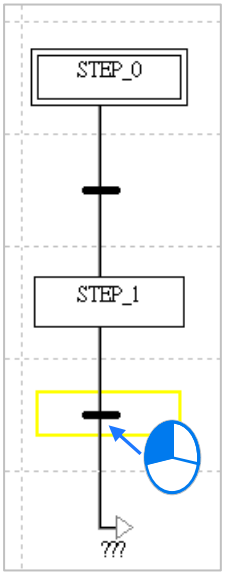
If the users want to manage the action or the transition created in the project management area, they can right-click the action or the transition, point to **Action/Transition** on the context menu, and click an item. The users have to notice that the action can not be copied into the **Transition** section, and the transition can not be copied into the **Action** section.



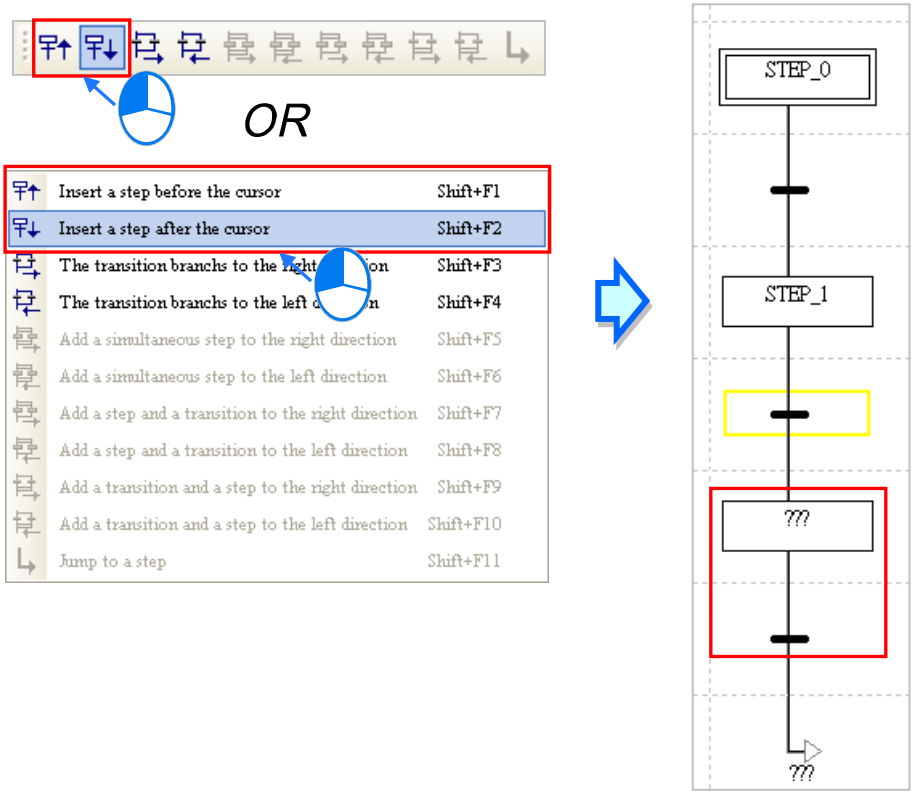


12.3.3 Adding a Step

(1) Select a step in the working area.

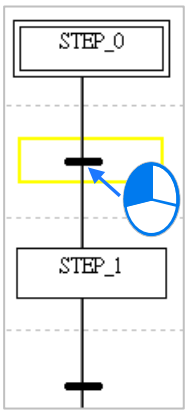


(2) If users want to add a step to the top of the step selected, they can click on the toolbar. They can also add a step to the top of the step selected by right-clicking the step selected, point to **Edit Items** on the context menu, and clicking **Insert a Step before the Cursor**. If the users want to add a step to the bottom of the step selected, they can click on the toolbar. They can also add a step to the bottom of the step selected by right-clicking the step selected, point to **Edit Items** on the context menu, and clicking **Insert a Step after the Cursor**.

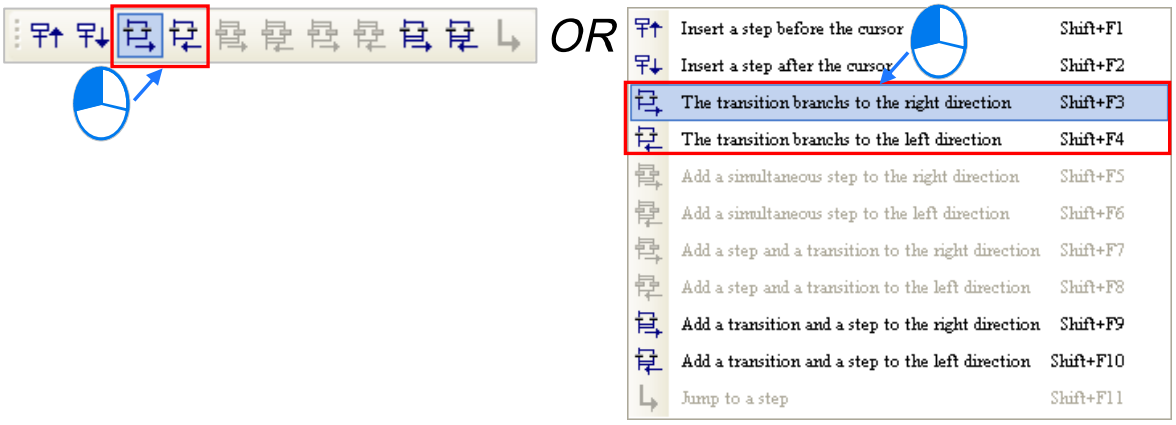


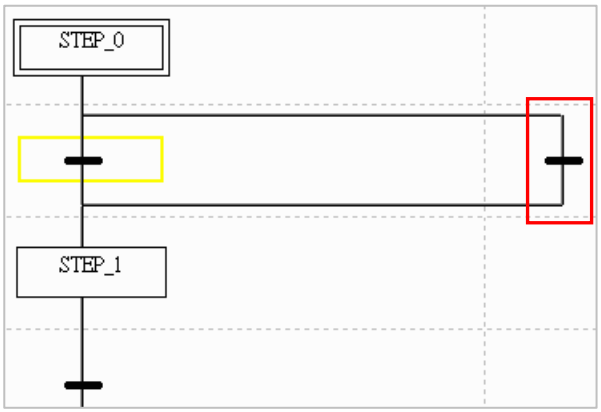
12.3.4 Connecting a Transition in Parallel

(1) Select a transition in the working area.



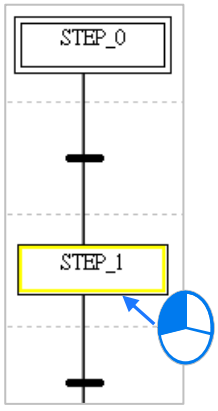
(2) If users want to put a transition at the right side of the transition selected, they can click on the toolbar. They can also put a transition at the right side of the transition selected by right-clicking the transition selected, point to **Edit Items** on the context menu, and clicking **The transition branches to the right direction**. If the users want to put a transition at the left side of the transition selected, they can click on the toolbar. They can also put a transition at the left side of the transition selected by right-clicking the transition selected, point to **Edit Items** on the context menu, and clicking **The transition branches to the left direction**.





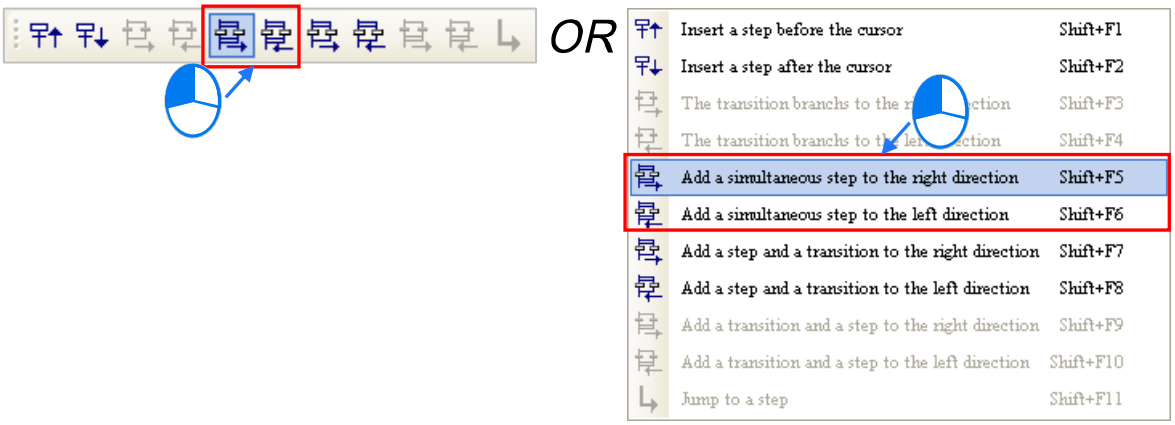
12.3.5 Connecting a Step in Parallel

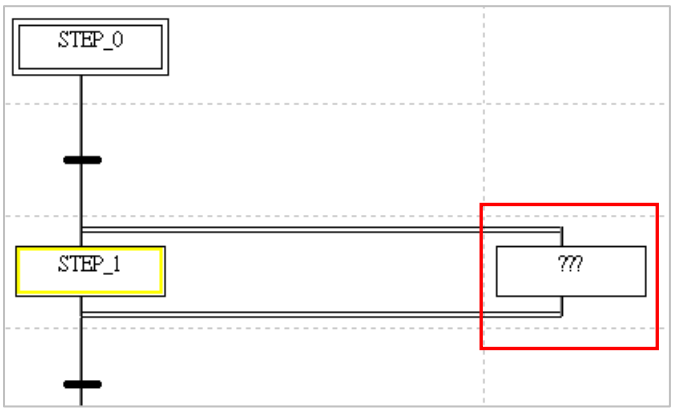
(1) Select a step in the working area. (The topmost step in the working area can not be connected to another step in parallel.)



(2) If users want to put a step at the right side of the step selected, they can click on the toolbar. They can also put a step at the right side of the step selected by right-clicking the step selected, pointing to **Edit Items** on the context menu, and clicking **Add a Simultaneous Step to the Right Direction**. If the users want to put a step at the left side of the step selected, they can click on the toolbar. They can also put a step at the left side of the step selected by right-clicking the step selected, pointing to **Edit Items** on the context menu, and clicking **Add a**

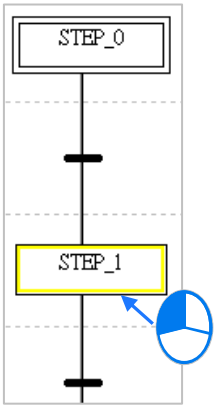
Simultaneous Step to the Left Direction.





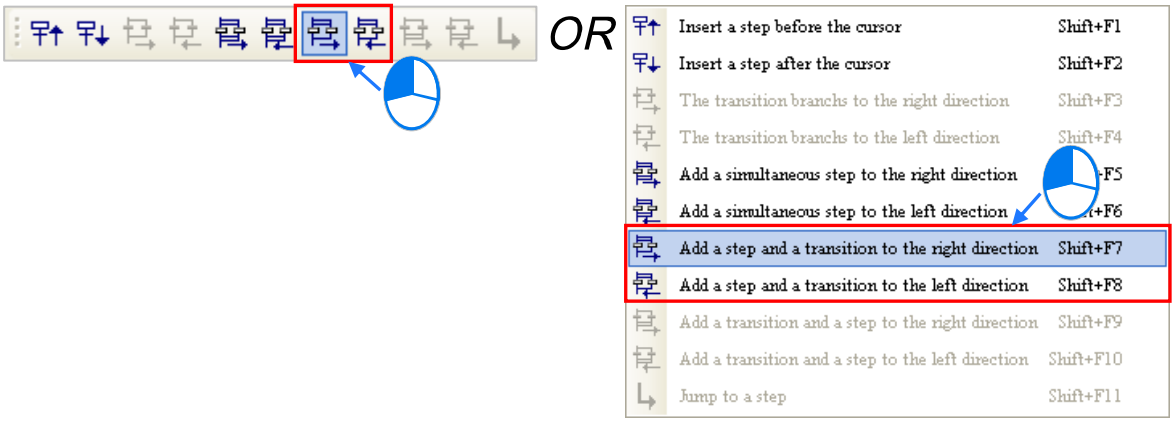
12.3.6 Creating a Simultaneous Divergence and a Convergence of Sequence Selection

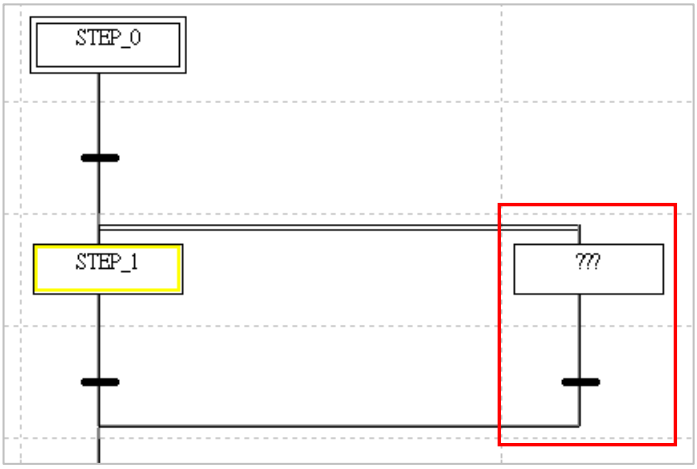
(1) Select a step in the working area. The step selected can not be the topmost step in the sequential function chart.



(2) If users want to create a simultaneous divergence and a convergence of sequence selection at the right side of the step selected, they can click on the toolbar. They can also create a simultaneous divergence and a convergence of sequence selection at the right side of the step

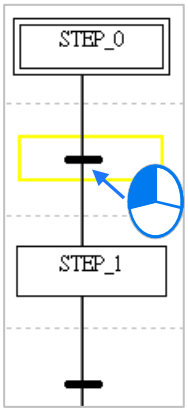
selected by right-clicking the step selected, pointing to **Edit Items** on the context menu, and clicking **Add a Step and a Transition to the Right Direction**. If the users want to create a simultaneous divergence and a convergence of sequence selection at the left side of the step selected, they can click on the toolbar. They can also create a simultaneous divergence and a convergence of sequence selection at the left side of the step selected by right-clicking the step selected, pointing to **Edit Items** on the context menu, and clicking **Add a Step and a Transition to the Left Direction**.



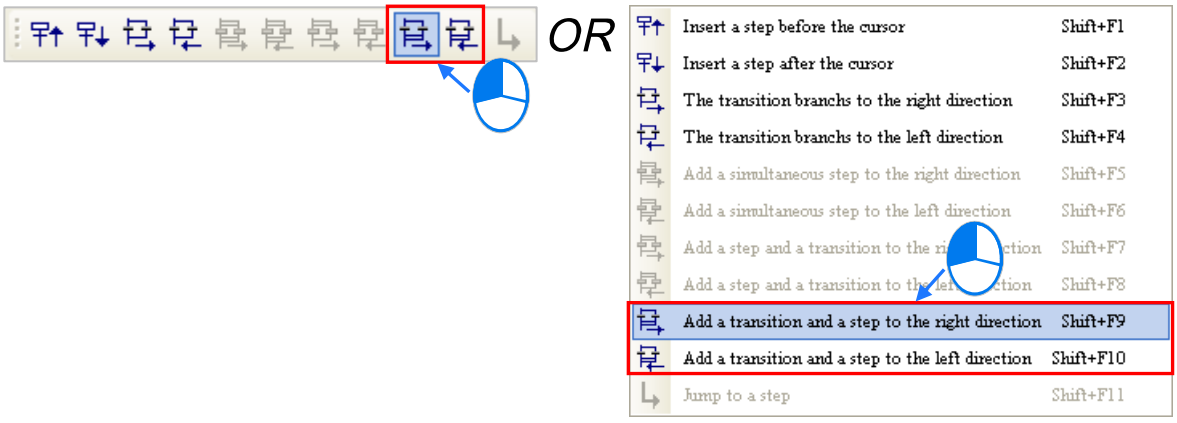


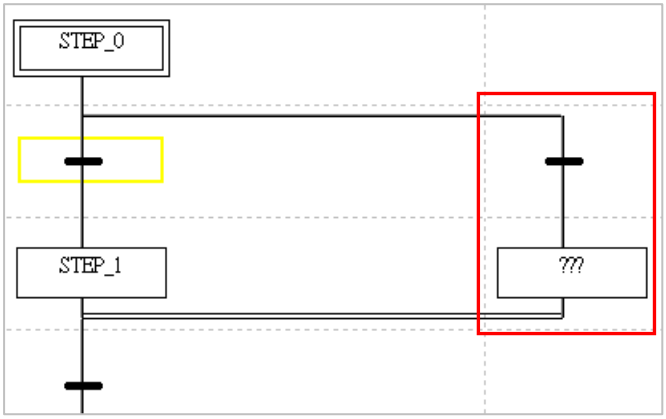
12.3.7 Creating a Divergence of Sequence Selection and a Simultaneous Convergence

(1) Select a transition in the working area. The transition selected can not be the bottommost transition in the sequential function chart.



(2) If users want to create a divergence of sequence selection and a simultaneous convergence at the right side of the transition selected, they can click on the toolbar. They can also create a divergence of sequence selection and a simultaneous convergence at the right side of the step selected by right-clicking the transition selected, pointing to **Edit Items** on the context menu, and clicking **Add a Transition and a Step to the Right Direction**. If the users want to create a divergence of sequence selection and a simultaneous convergence at the left side of the step selected, they can click on the toolbar. They can also create a divergence of sequence selection and a simultaneous convergence at the left side of the step selected by right-clicking the step selected, pointing to **Edit Items** on the context menu, and clicking **Add a Transition and a Step to the Left Direction**.

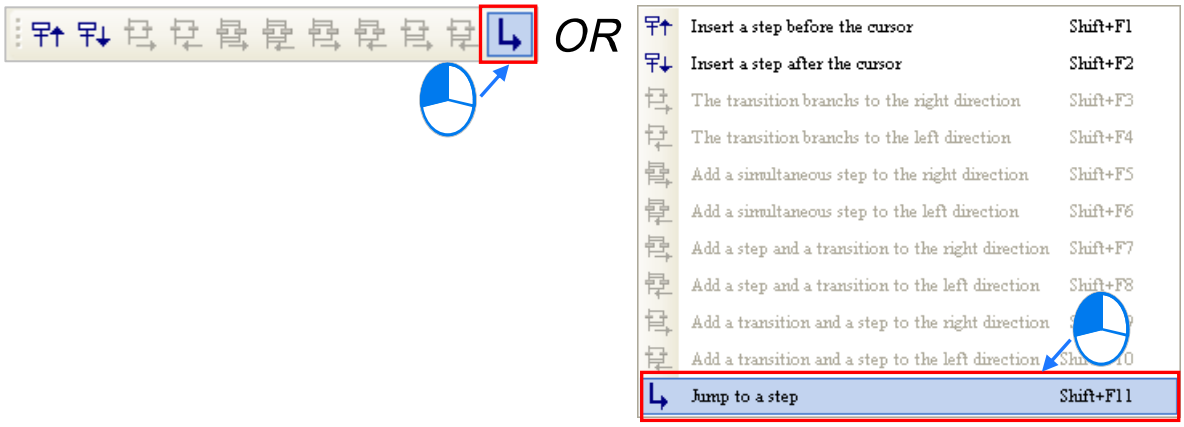


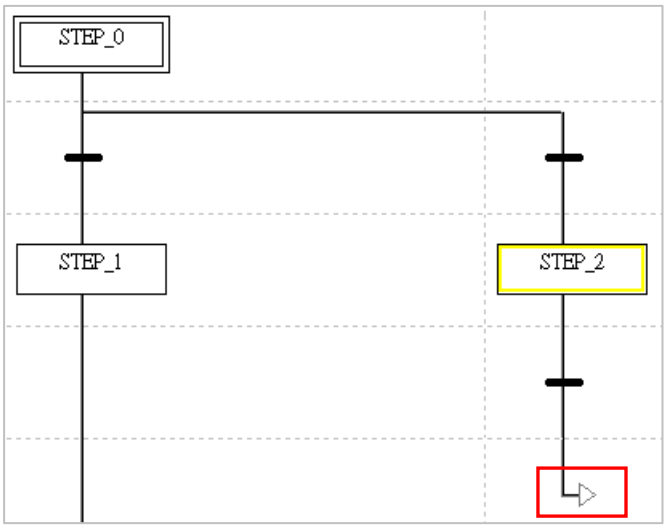


12.3.8 Inserting a Jump Point

(1) Select the bottommost step or the bottommost transition on a divergent path in the working area.

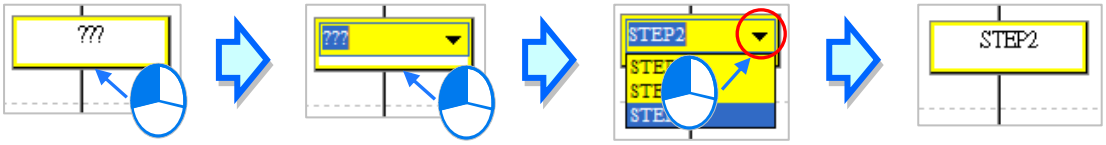
(2) After users click on the toolbar, a jump point will be inserted. The users can also insert a jump point by right-clicking the step or the transition selected, pointing to **Edit Items** on the context menu, and click **Jump to a Step**.



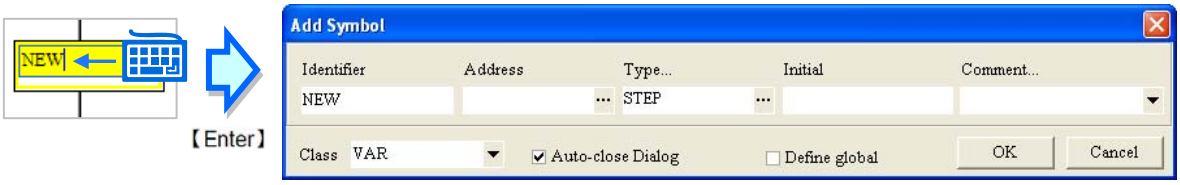


12.3.9 Assigning Symbols to the Steps and the Transitions

After the editing of a sequential function chart is complete, users must assign symbols to the steps and the transitions in the sequential function chart. Click a step or a transition first. After a yellow frame appears, users have to click the step or the transition again. Click the button at the right side of the box which appears, and select an item created.



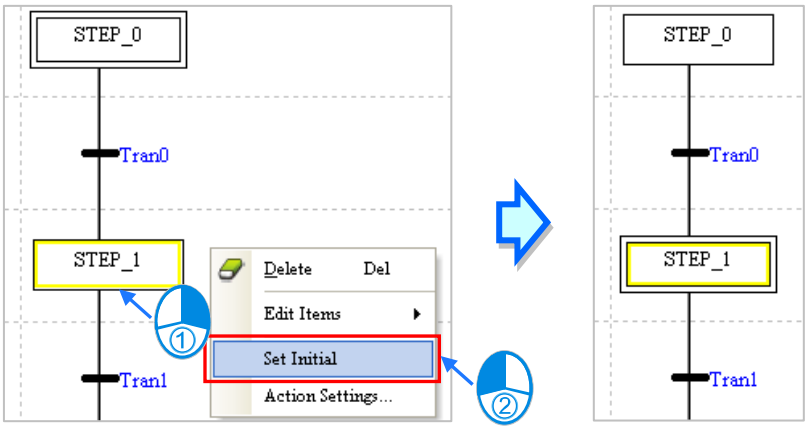
The users can also assign a symbol to a step or a transition by means of the keyboard. They can move the yellow frame by pressing the up/down key on the keyboard. After the users select a step or a transition, they can press Enter on the keyboard. The users can type characters in the box which appears. If they press Page Down on the keyboard, a drop-down list will appear. The users can select a symbol on the drop-down list by the up/down key on the keyboard. After the editing of the box is complete, the users can press Enter on the keyboard to jump to the next box. If the users want to end the editing, they can press Esc on the keyboard. Besides, if the step name the users type in the box for a step is symbol which is not declared, the **Add Symbol** window will appear after the users press Enter on the keyboard. However, if the transition name the users type in the box for a transition is a symbol which is not declared, the **Add Symbol** will not appear.



***. Users can decide whether to allow the system to open the Add Symbol window automatically or not. Please refer to section 2.3.1 for more information about the setting.**

12.3.10 Specifying an Initial Step

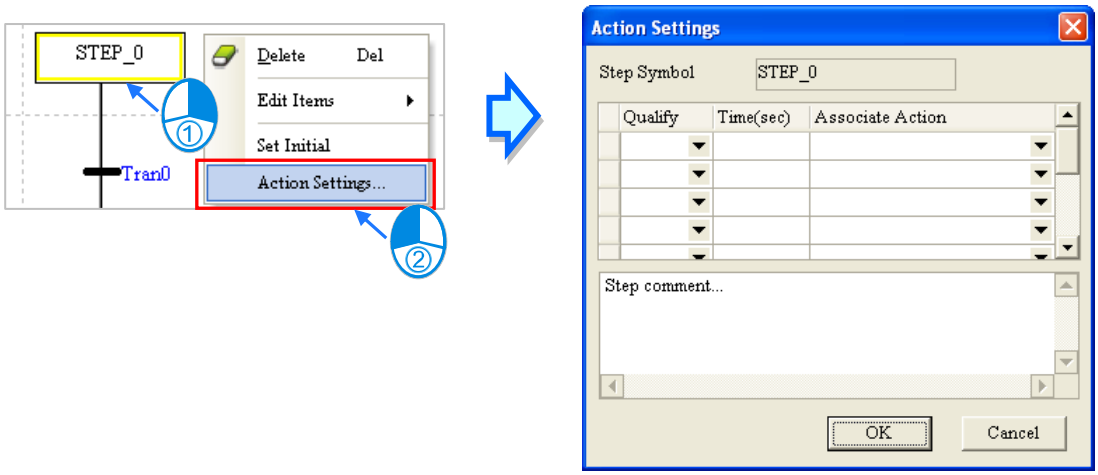
Users can define a step in a sequential function chart in an AH500 series CPU module as the initial step in the sequential function chart whereas they can not define a step in a sequential function chart in a DVP series PLC as the initial step in the sequential function chart. If users want to define a step in a sequential function chart as the initial step in the sequential function chart, they have to select the step in the working area, right-click the step, and click **Set Initial** on the context menu.



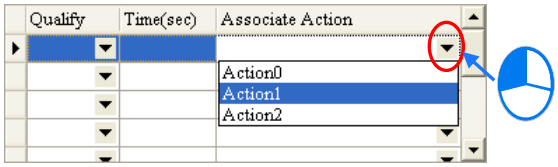
***. Please refer to section 12.2.7 for more information about activating the steps in a DVP series PLC.**

12.3.11 Assigning Actions

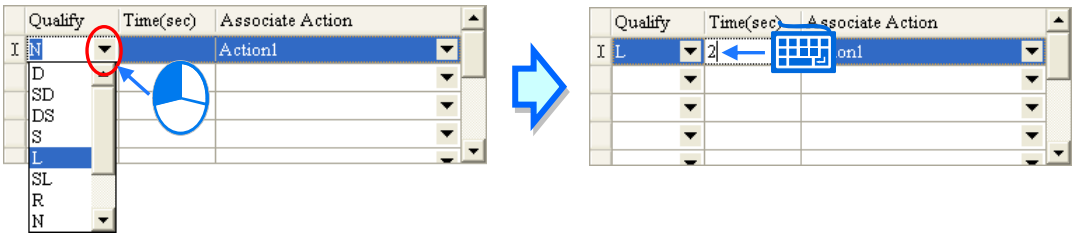
(1) Select the step to which an action will be assigned, right-click the step, and click **Action Settings…** on the context menu to open the **Action Setting**s window. If the step is not declared, the system will not allow the assignment of an action to the step.



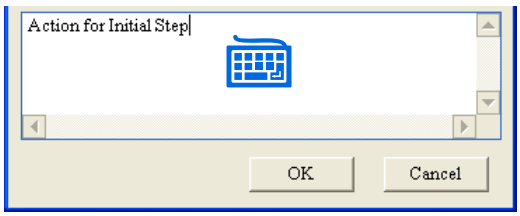
(2) Select an action in an **Associate Action** drop-down list cell in the **Action Settings** window.



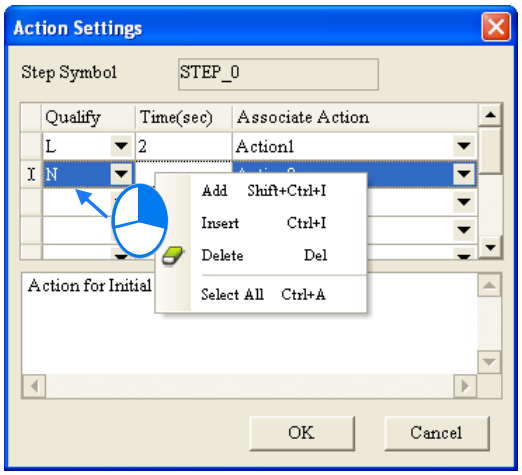
(3) After an action is selected, users have to select a qualifier in the **Qualify** drop-down list cell for the action. If the qualifier selected is related to time, the users have to type a time interval in the **Time (sec)** cell for the qualifier. A second is a measurement unit of a time interval. Besides, the users can only type an integer. (The users can not qualify the actions in the program in a DVP series PLC.)



(4) If the users want to make comments on the step, they can type the comments in the box at the lower part of the window.

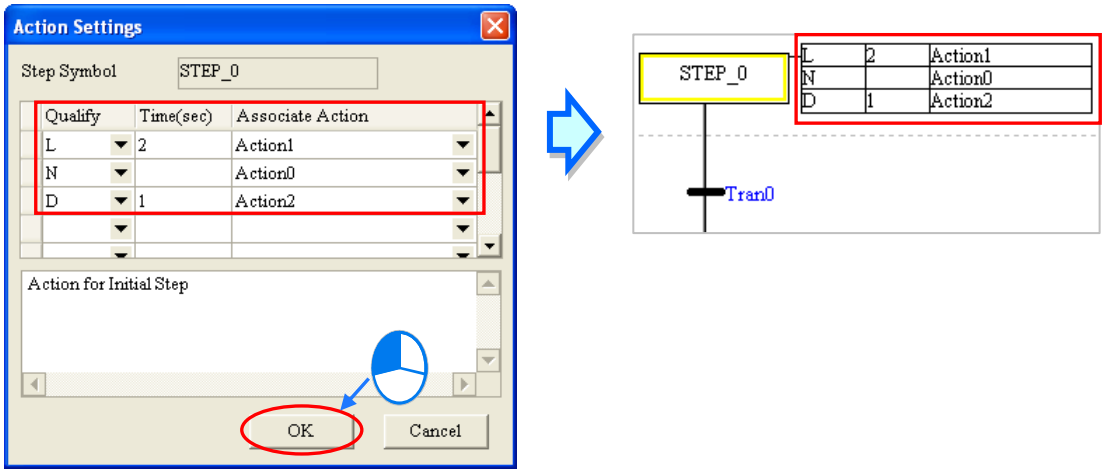


(5) After the users right-click an item on the list, they can click **Add**, **Insert**, **Delete**, or **Select All** on the context menu.



Item	Function
Add	Adding a row to the bottom of the cell selected.
Insert	Inserting a row above the cell selected
Delete	Deleting the row selected
Select All	Selecting all the items

(6) After the assignment of actions is complete, users can click **OK**. A list of the actions will appear at the right side of the step.



Additional remark

If the users want to select several rows in the **Action Settings** window, they can hold down Ctrl on the keyboard while they click the rows. If the users want to select any number of rows, they can click at the start of the selection, scroll to the end of the selection, and hold down Shift on the keyboard while they click where they want the selection to end.

12.4 Example of a Sequential Function Chart

12.4.1 Explanation

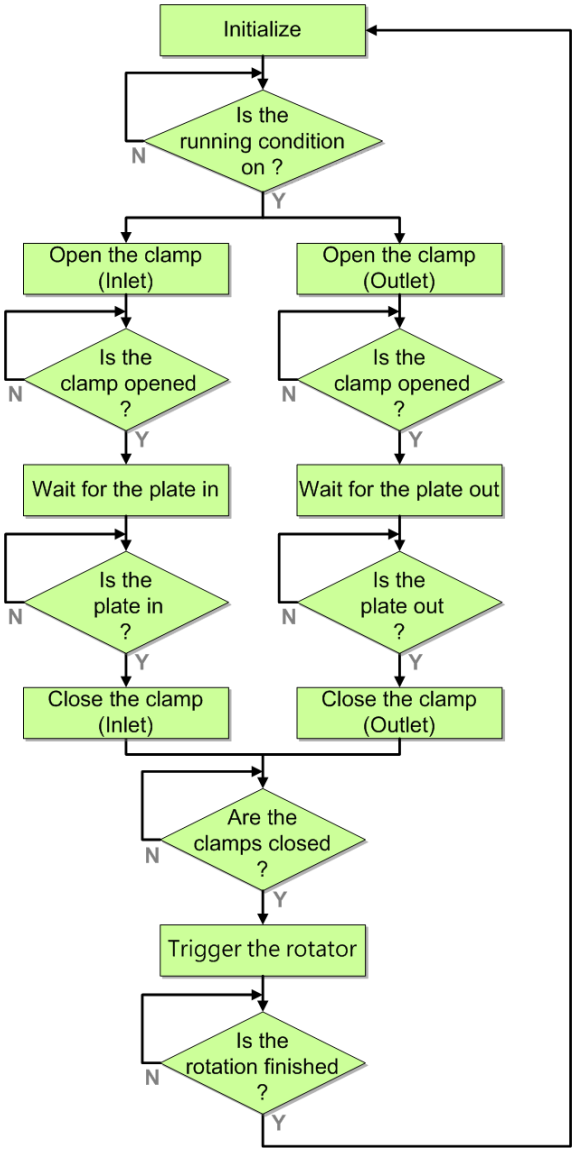
The example is about the rotation of two plates. The figure below is a flow chart. The actual equipment is also diagramed below.

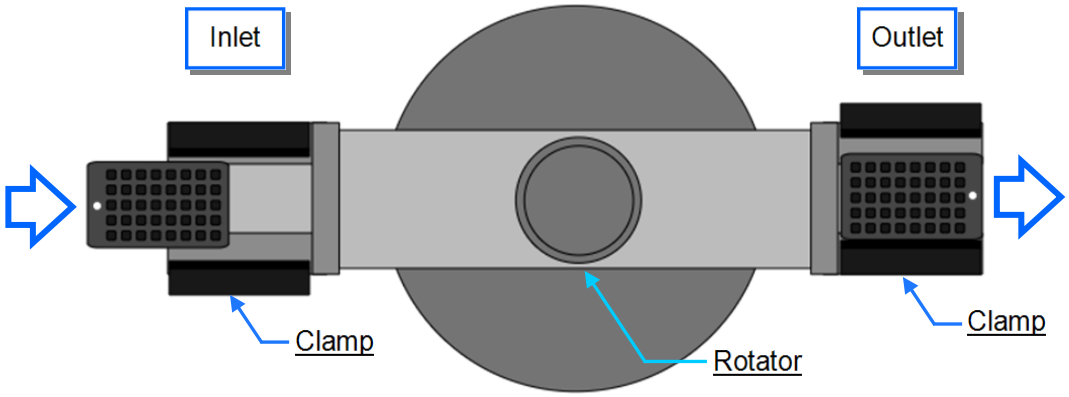
The inlet is at the right side of the platform, and the outlet is at the left side of the platform. After a plate enters the inlet, and the plate in the outlet makes its exit from the outlet, the rotator in the middle of the platform will rotate 180 degrees. After the rotator rotates, the plate which entered the

inlet will be in the outlet, and the clamp from which the plate made its exit will receive another plate. During the whole process, a plate enters the inlet independently, and the plate in the outlet makes its exit from the outlet independently. However, the rotator does not rotate independently.

When the system begins to operate, the cylinders under the clamps open the clamps. After the clamps are opened, the system will send readiness signals. After the readiness signals are sent, a plate will enter the inlet, and the plate in the outlet will make its exit from the outlet.

After a plate enters the inlet, and the plate in the outlet makes its exit from the outlet, the system will receive completion signals, the corresponding readiness signals will be set to OFF, and the clamps will be closed. After the system makes sure that the clamps are closed, the system will send a trigger signal. After the rotator rotates 180 degrees, and the system receives a completion signal, the next cycle will begin.





12.4.2 Planning Hardware

In this example, the AH500 series CPU module **AHCPU530-EN**, the digital I/O module **AH16AP11R-5A**, and the four-slot main backplane **AHBP04M1-5A** are used. The table below is an I/O allocation table.

Type	ID	Description
Digital input	X0.0	Completion signal
Digital input	X0.1	Completion signal
Digital input	X0.2	Completion signal
Digital input	X0.3	(Inlet) Cylinder–Opening sensor
Digital input	X0.4	(Inlet) Cylinder–Closing sensor
Digital input	X0.5	(Outlet) Cylinder–Opening Sensor
Digital input	X0.6	(Outlet) Cylinder–Closing Sensor
Digital output	Y0.0	Readiness signal
Digital output	Y0.1	Readiness signal
Digital output	Y0.2	Trigger signal
Digital output	Y0.3	(Inlet) Clamp
Digital output	Y0.4	(Outlet) Clamp

12.4.3 Planning a Program

(1) The procedure for initializing the system is to reset the trigger signal and the readiness signals to OFF until the condition for the operation of the system is met. In this example, when the operation flag is ON, and the error flag is OFF, the condition for the operation of the system is met.

(2) Once the system begins to operate, the cylinders under the clamps will open the clamps **(Y0.3 and Y0.4)**, and the system will make sure that the opening sensors **(X0.3 and X0.5)** are ON.

(3) After the system makes sure that the clamps are opened, the system will send readiness signals **(Y0.0 and Y0.1)** to inform the work stations connecting to the system.

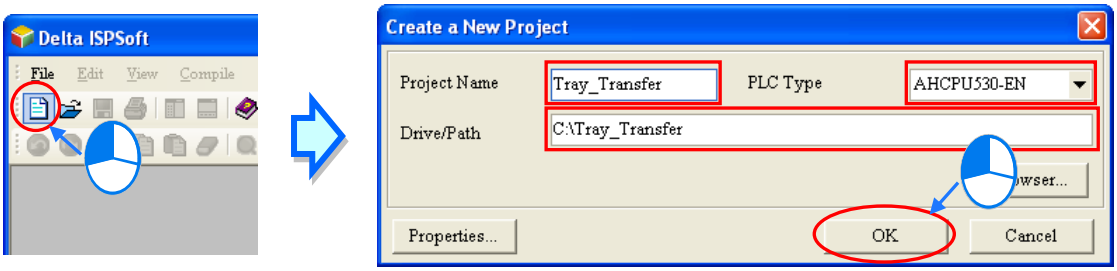
(4) After a plate enters the inlet, the system will receive a completion signal **(X0.0)**, and delay the closing of the clamp in the inlet **(Y0.3)** for one second.

(5) After the plate in the outlet makes its exit from the outlet, the system will receive a completion signal **(X0.1)**, and delay the closing of the clamp in the outlet **(Y0.4)** for one second.

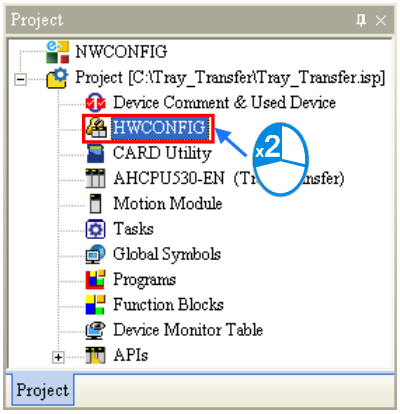
(6) After the system makes sure that the closing sensors **(X0.4 and X0.6)** are ON, the system will send a trigger signal **(Y0.2)**. After the rotator rotates 180 degrees, the system will receive a completion signal **(X0.2)**, and the procedure for initializing the system is carried out again.

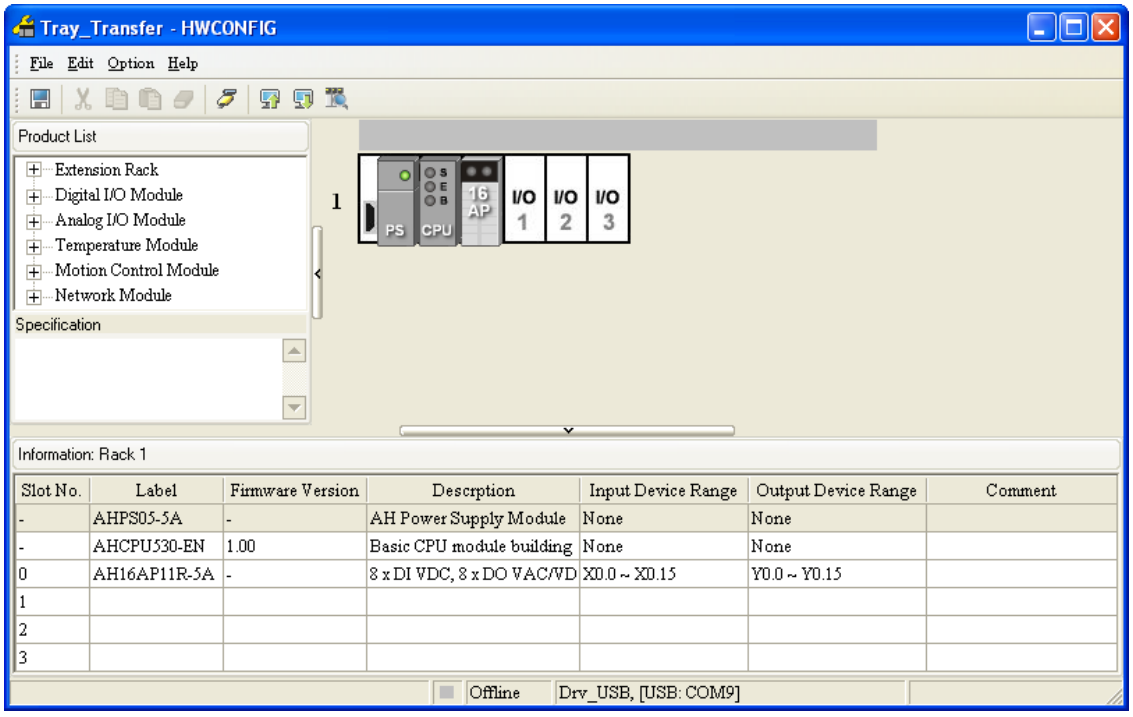
12.4.4 Creating a Program

Start ISPSoft, and then create a new project.



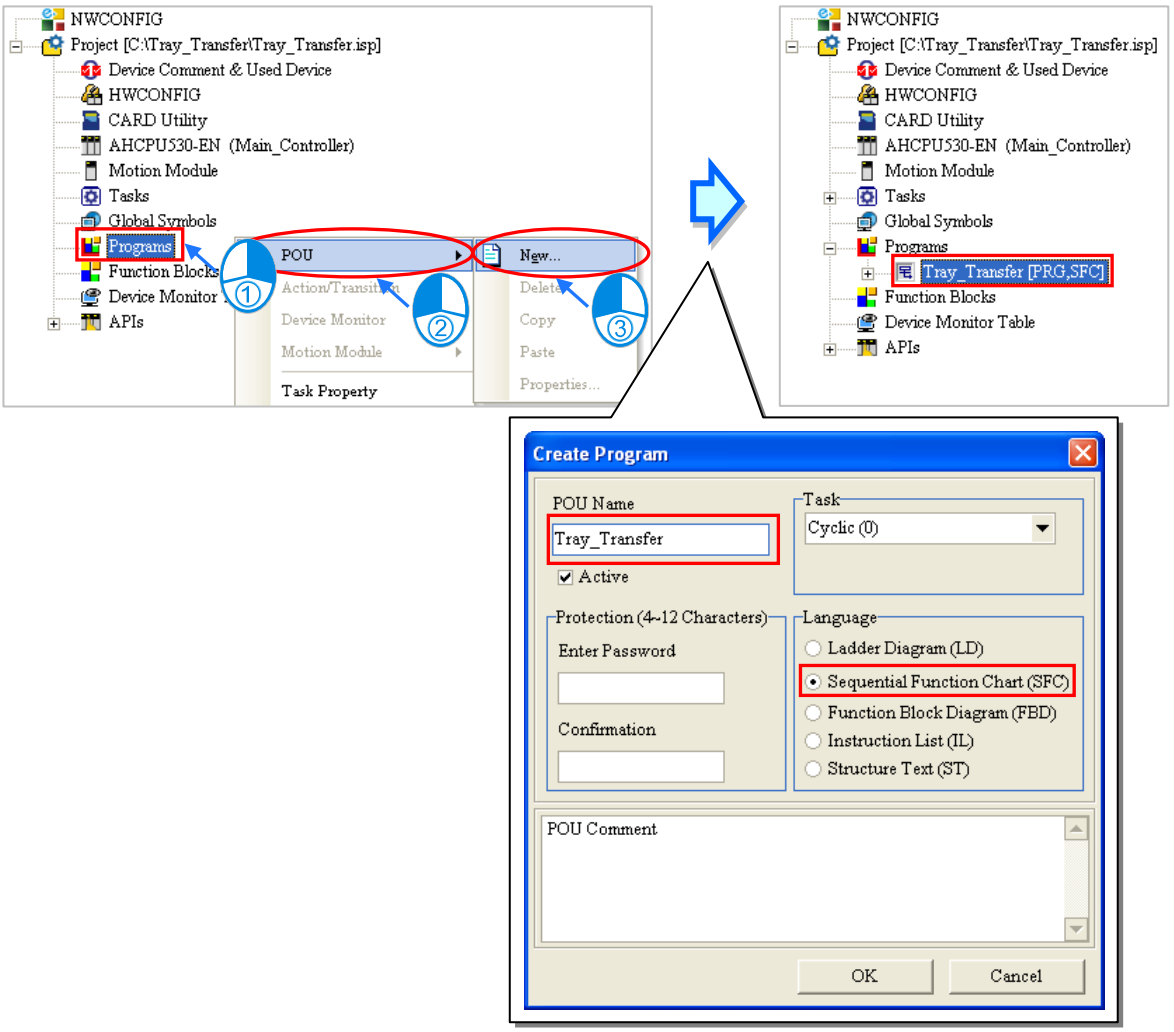
The hardware configuration in this example is the same as the hardware configuration in chapter 4. Please refer to chapter 4, and complete the hardware configuration.





Create a POU of the program type in the project management area. The programming language

used to create the POU of the program type is a structured text.



Create the flags, the digital input, and the digital outputs in the global symbol table.

Global Symbols

Global Symbols

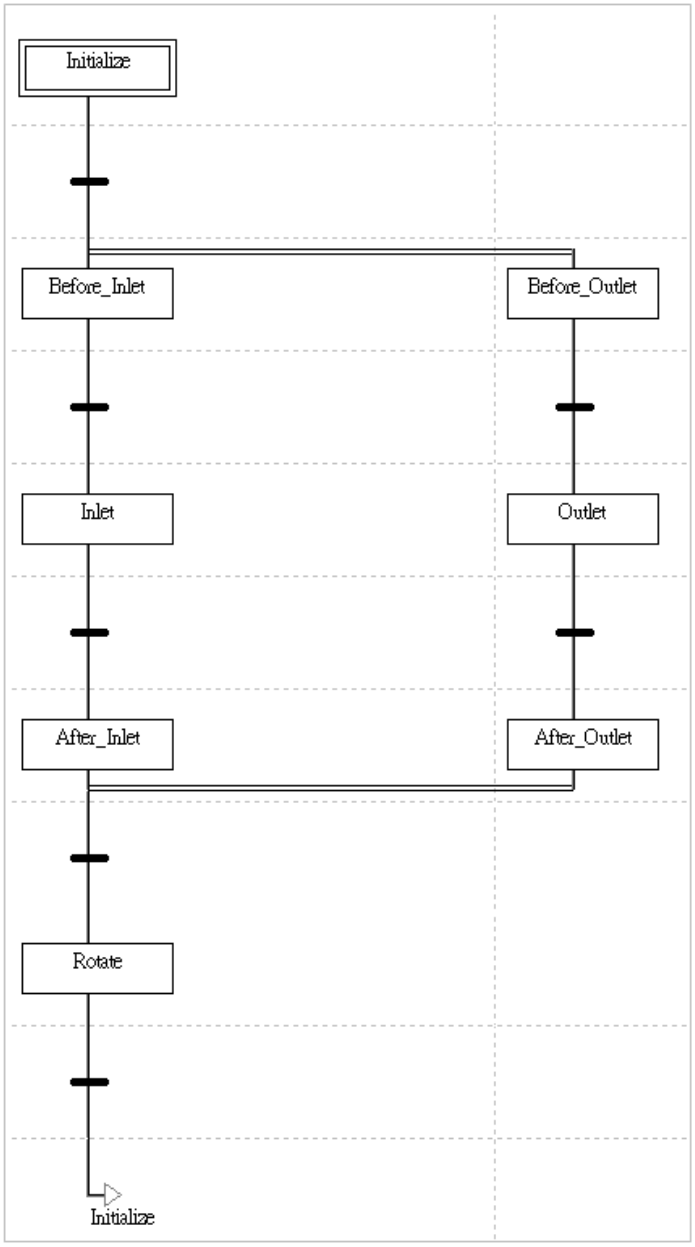
Class	Identifiers	Address	Type...	Initial Value	Identifier Comment...
VAR	IN_Done	X0.0	BOOL	FALSE	
VAR	OUT_Done	X0.1	BOOL	FALSE	
VAR	R_Done	X0.2	BOOL	FALSE	
VAR	Clip_1_OS	X0.3	BOOL	FALSE	
VAR	Clip_1_CS	X0.4	BOOL	FALSE	
VAR	Clip_2_OS	X0.5	BOOL	FALSE	
VAR	Clip_2_CS	X0.6	BOOL	FALSE	
VAR	Inlet_Rdy	Y0.0	BOOL	FALSE	
VAR	Outlet_Rdy	Y0.1	BOOL	FALSE	
VAR	Rotator_Trig	Y0.2	BOOL	FALSE	
VAR	Clip_1	Y0.3	BOOL	FALSE	
VAR	Clip_2	Y0.4	BOOL	FALSE	
VAR	RUN	N/A [Auto]	BOOL	FALSE	
VAR	ERROR	N/A [Auto]	BOOL	FALSE	

Declare the steps shown below in the local symbol table in the POU of the program type.

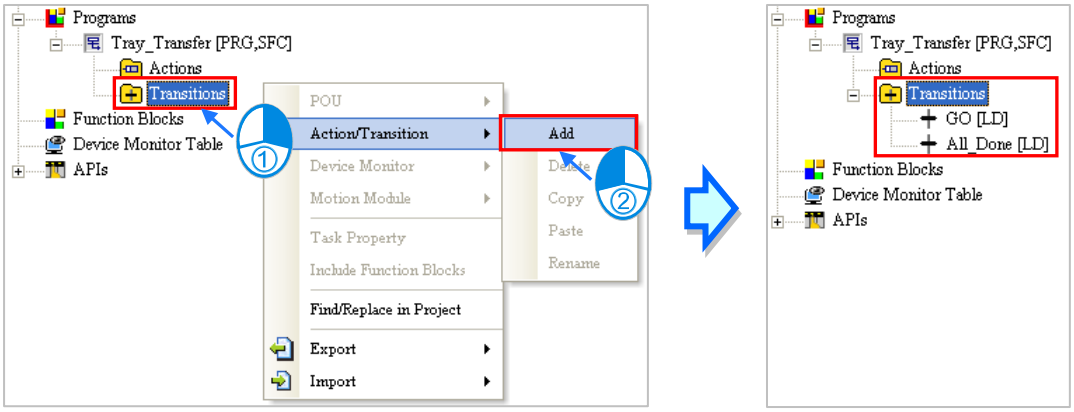
Local Symbols					
Class	Identifiers	Address	Type...	Initial Value	Identifier Comment...
VAR	Initialize	N/A [Auto]	STEP	FALSE	
VAR	Before_Inlet	N/A [Auto]	STEP	FALSE	
VAR	After_Inlet	N/A [Auto]	STEP	FALSE	
VAR	Before_Outlet	N/A [Auto]	STEP	FALSE	
VAR	After_Outlet	N/A [Auto]	STEP	FALSE	
VAR	Inlet	N/A [Auto]	STEP	FALSE	
VAR	Outlet	N/A [Auto]	STEP	FALSE	
VAR	Rotate	N/A [Auto]	STEP	FALSE	

12

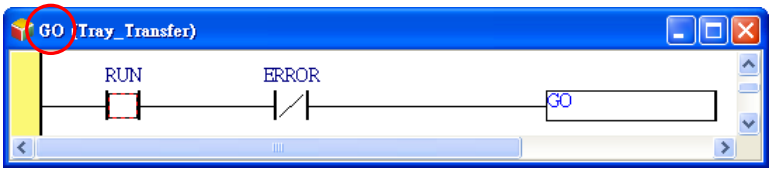
Create the sequential function chart shown below. The initial step in the sequential function chart is the first step in the sequential function chart.

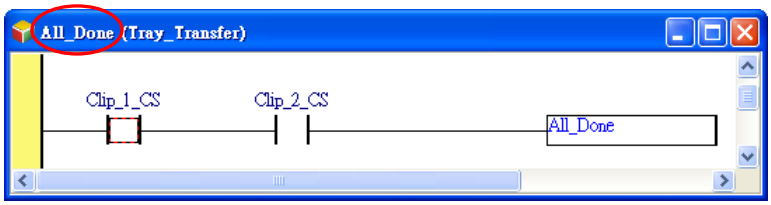


Create a transition called **Go** and a transition called **All_Done** under the **Transitions** section in the project management area.

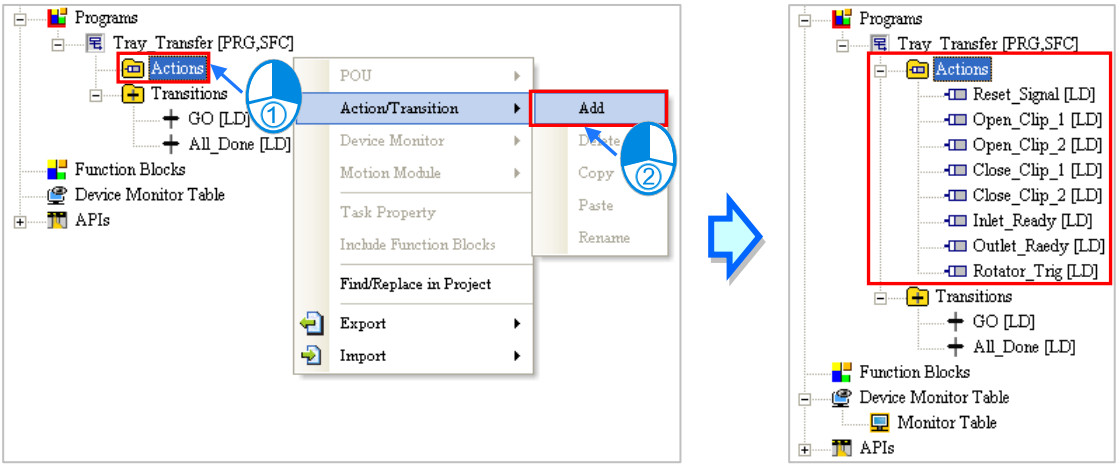


After a transition called **Go** and a transition called **All_Done** are created under the **Transitions** section in the project management area, users have to write the programs shown below.

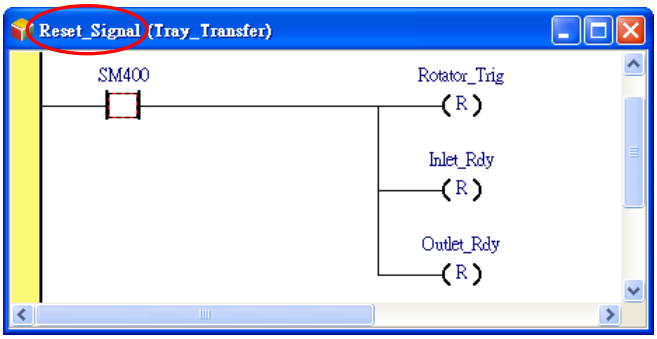


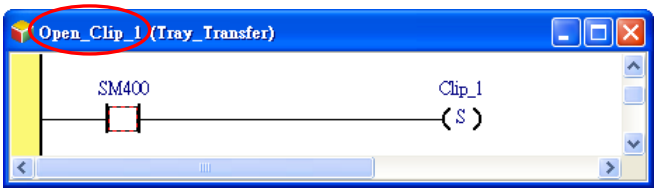


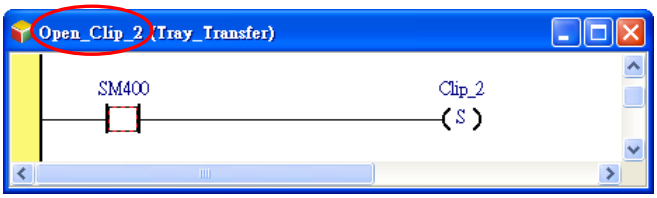
Create the actions which are shown below under the **Actions** section in the project management area.

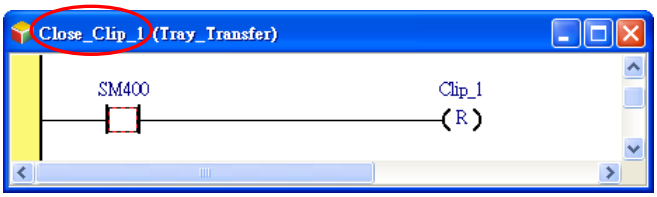


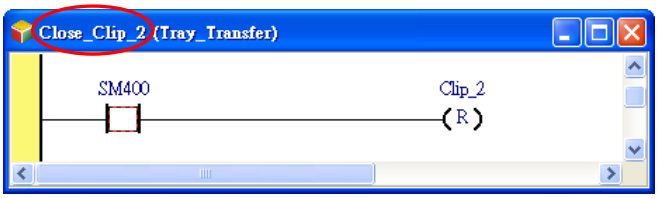
After the actions are created under the **Actions** section in the project management area, the users have to write the programs shown below.

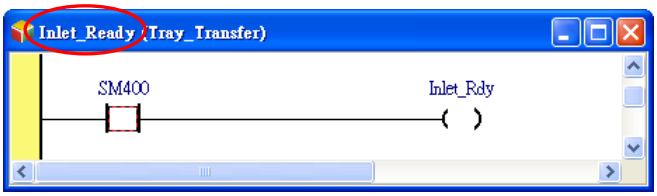

Reset_Signal (Tray_Transfer)
SM400
Rotator_Trig
R
Inlet_Rdy
R
Outlet_Rdy
R

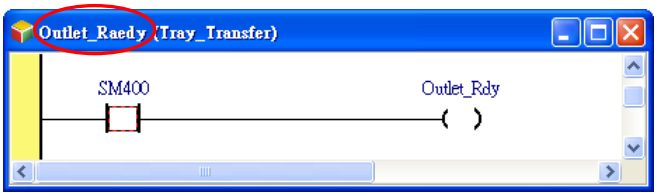

Open_Clip_1 (Tray_Transfer)
SM400
Clip_1
S

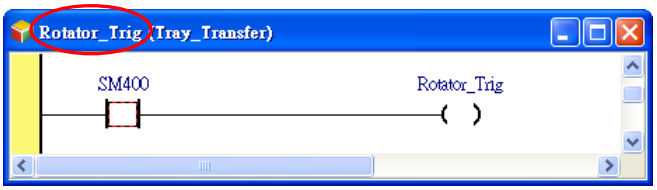

Open_Clip_2 (Tray_Transfer)
SM400
Clip_2
S


Close_Clip_1 (Tray_Transfer)
SM400
Clip_1
R

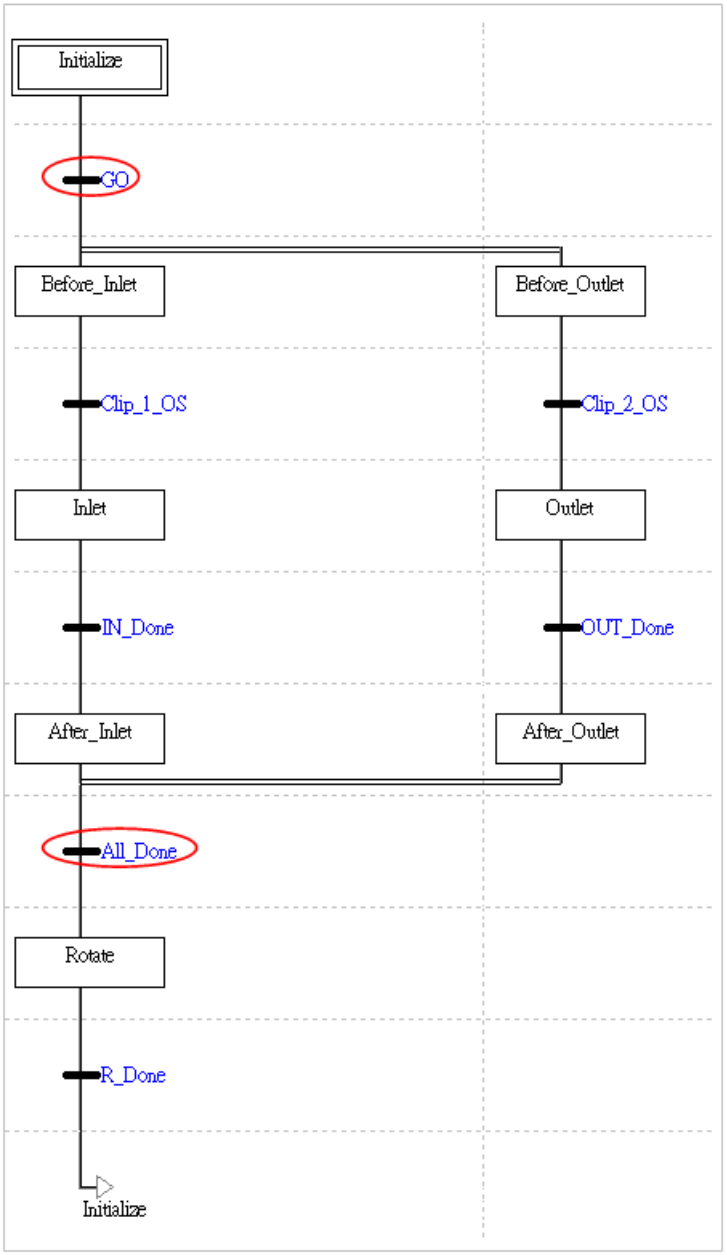

Close_Clip_2 (Tray_Transfer)
SM400
Clip_2
R


Inlet_Ready (Tray_Transfer)
SM400
Inlet_Rdy

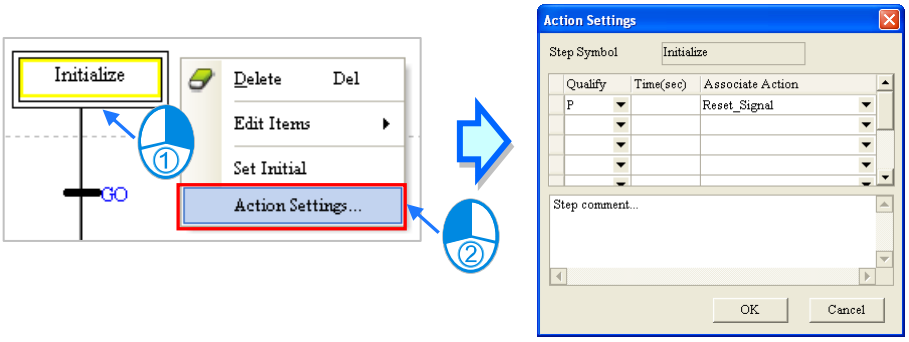

Outlet_Raedy (Tray_Transfer)
SM400
Outlet_Rdy



After the users write the action programs and the transition programs, they have to assign symbols to the transitions in the sequential function chart. All the symbols assigned to the transitions in the sequential function chart are symbols in the global symbol except **GO** and **All_Done**.

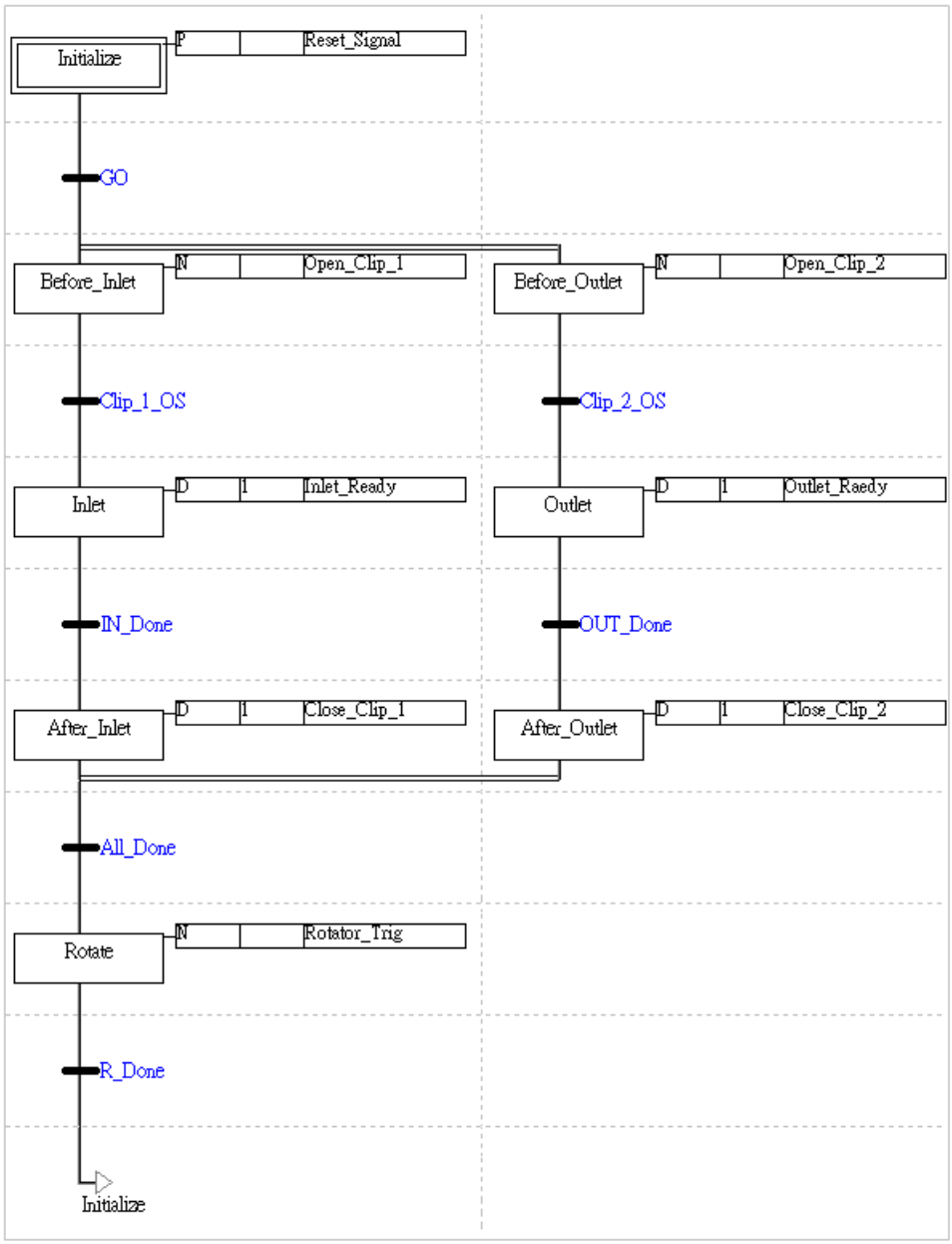


The users have to assign the actions to the steps in the sequential function chart, and assign qualifiers to the actions.



12

The writing of the program has been completed so far. The users can try to compile the program, download the parameters and the program to the CPU module, and test the program by themselves.



Chapter 13 Auxiliary Editing Tools

Table of Contents

13.1 Editing Tools and Auxiliary Functions in ISPSoft 13-2
13.1.1 Changing the PLC Type 13-2
13.1.2 Downloading/Uploading a Project.....13-2
13.1.3 Find/Replace/Goto Function in a Ladder Diagram/Function Block Diagram 13-4
13.1.4 Finding/Replacing Text on an Instruction List/in an Structured Text 13-6
13.1.5 Finding/Replacing a Step or a Transition in a Sequential Function Chart.....13-8
13.1.6 Finding an Identifier in a Symbol Table.....13-10
13.1.7 Finding/Replacing a Device or a Symbol in a Project 13-11
13.1.8 Printing Function.....13-12
13.2 Managing Devices and Registers 13-15
13.2.1 Device Comment List for a DVP Series PLC 13-15
13.2.2 Report on the Devices Used in a DVP Series PLC.....13-18
13.2.3 Managing Devices in an AH500 Series CPU Module 13-20
13.2.4 Editing the Values in the T/C/D Devices in a DVP Series PLC ...13-23
13.2.5 Editing the States of the M/S Devices in a DVP Series PLC 13-29
13.2.6 Editing the Values in the File Registers in a DVP Series PLC 13-33

13.1 Editing Tools and Auxiliary Functions in ISPSoft

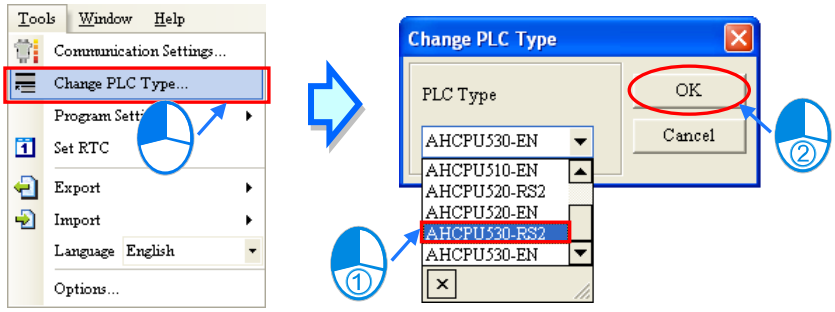
13.1.1 Changing the PLC Type

Users are allowed to change the PLC type in ISPSoft. Users have to pay attention to the following two points before they change the PLC type.

(a) An AH500 system is not compatible with a DVP system, and therefore users can not change the model from an AH500 series CPU module to a DVP series PLC.

(b) If the PLC type is changed, the functions, the range of devices, and the instructions supported may be changed. After the PLC type is changed, users have to check whether the original program, parameter setting, hardware configuration, and network configuration can make the system operate normally.

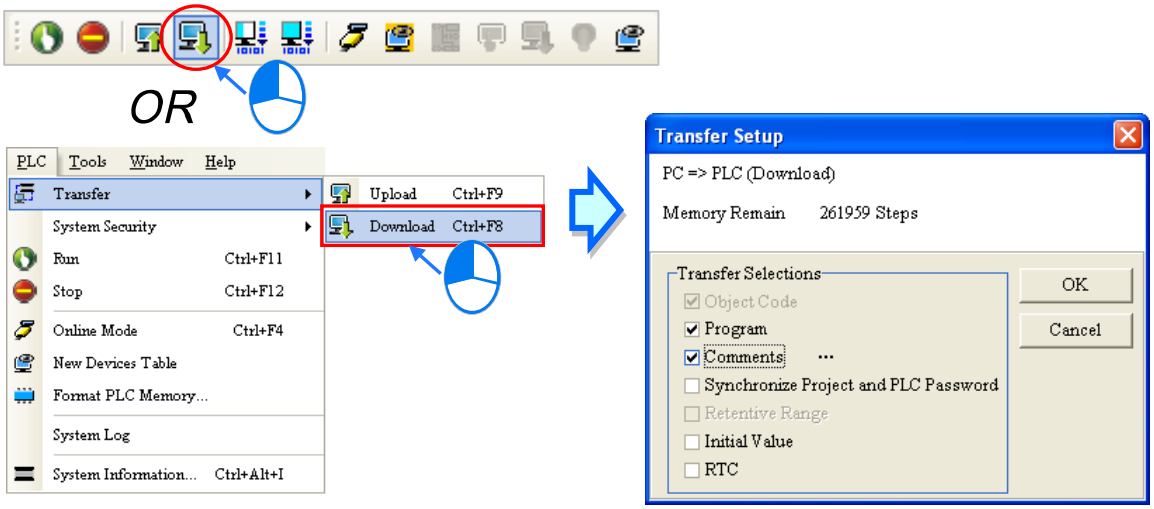
If users want to change the PLC type, they can click **Change PLC Type…** on the **Tools** menu, select a PLC type in the **Change PLC Type** window, and click **OK**.



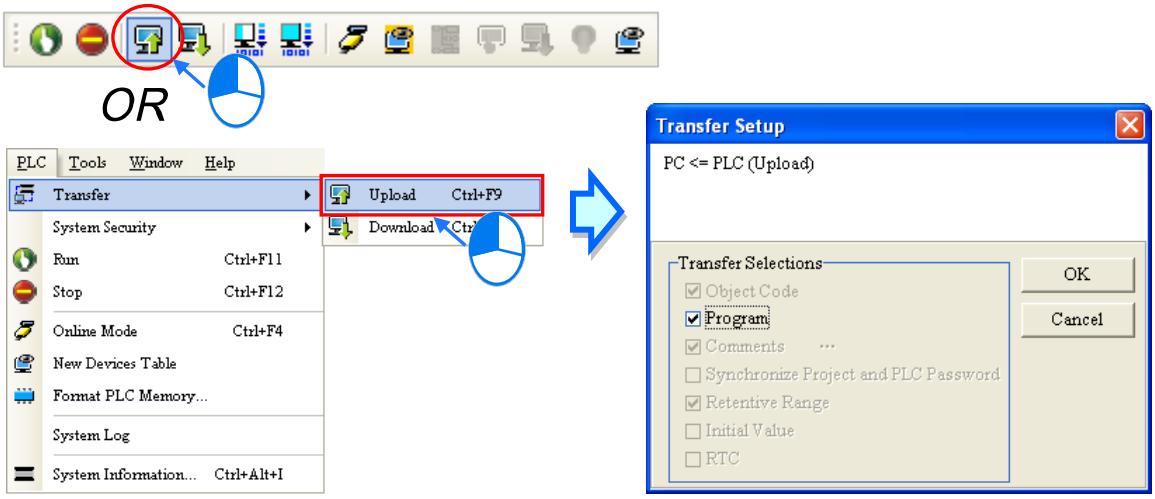
13.1.2 Downloading / Uploading a Project

Users can download/upload not only the program in a project, but also the functions or the parameters in the project.

If users want to download the project, they can click the **PLC** menu, point to Transfer, and click **Download**. The users can also download the project by clicking on the toolbar. Select the items which will be downloaded in the **Transfer Setup** window, and then click **OK**.



If users want to upload the project, they can click the **PLC** menu, point to Transfer, and click **Upload**. The users can also upload the project by clicking on the toolbar. Select the items which will be uploaded in the **Transfer Setup** window, and then click **OK**.



The items in the **Transfer Setup** window for the downloading of a project are the same as the items in the **Transfer Setup** window for the uploading of a project. However, the items which can not be selected and the items which must be selected in the **Transfer Setup** window for the downloading of a project are different from the items which can not be selected and the items which must be selected in the **Transfer Setup** window for the uploading of a project. These items are gray in color. The descriptions of the items in the **Transfer Setup** window for the downloading/uploading of a project are as follows.

Transfer selection	Description
Object Code	Object code is what a computer compiler produces. (The item must be selected.)
Program	Program (Source code) that users create
Comments	Comments added to the project
Synchronize Project and PLC Password	During the transmission of the project, the project password and the PLC password become the same.
Retentive Range	Range of latched devices (The item in the **Transfer Setup** window for the downloading/uploading of a project for an AH500 series CPU module can not be selected.)
Initial Value	When the project is downloaded, the initial values of the symbols are written into the corresponding memories in the PLC.
RTC	When the project is downloaded, the date and the time on the computer's clock is written into the real-time clock in the PLC.

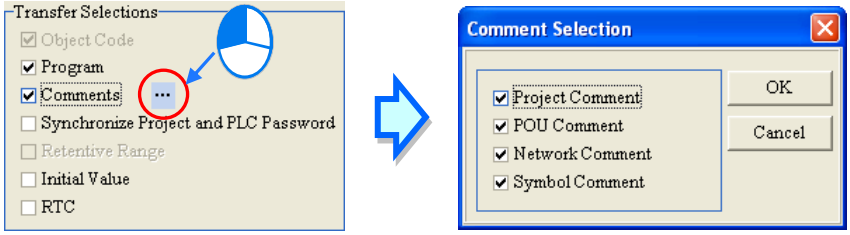
Additional remark

(a) Please refer to section 6.2.6 for more information about downloading the initial values of the symbols.

(b) Please refer to section 15.3 for more information about project passwords and PLC passwords.

(c) The range of latched devices in an AH500 series CPU module is downloaded/uploaded through HWCONFIG. Please refer to section 3.3.2.3 for more information.

(d) If users select the **Comments** checkbox in the **Transfer Setup** window for the downloading of a project, they can select items which will be downloaded in the **Comment Selection** window after they click ... at the right side of the **Comments** checkbox.

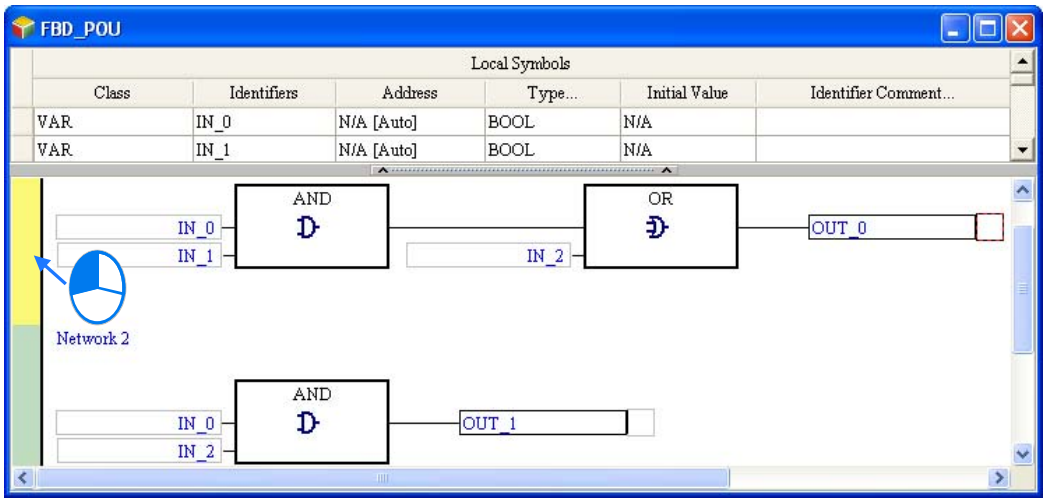


13.1.3 Find / Replace / Goto Function in a Ladder Diagram / Function Block Diagram

The **Find/Replace/Goto** function that ISPSoft provides varies with the program editing window. This section aims at describing the functions used in the window in which a ladder diagram can be created and in the window in which a function block diagram can be created. Besides, the functions described in this section are used in the present program editing window, and are not used in the whole project.

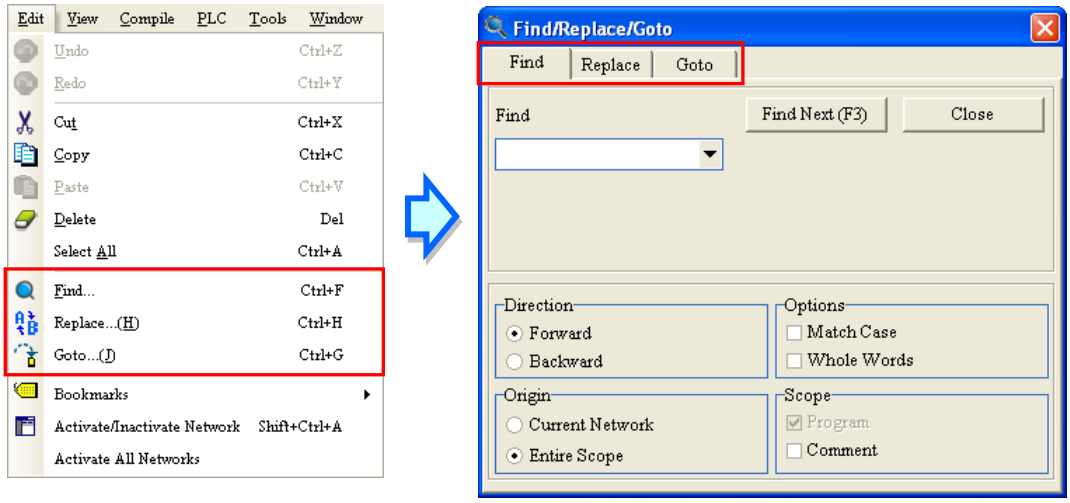
13

Users can open a program editing window, and click the position in the working area at which they want to start searching. If the users click the local symbol table in the program editing window, the local symbol table will be searched. Please refer to section 13.1.6 for more information about using the **Find** function in a symbol table.



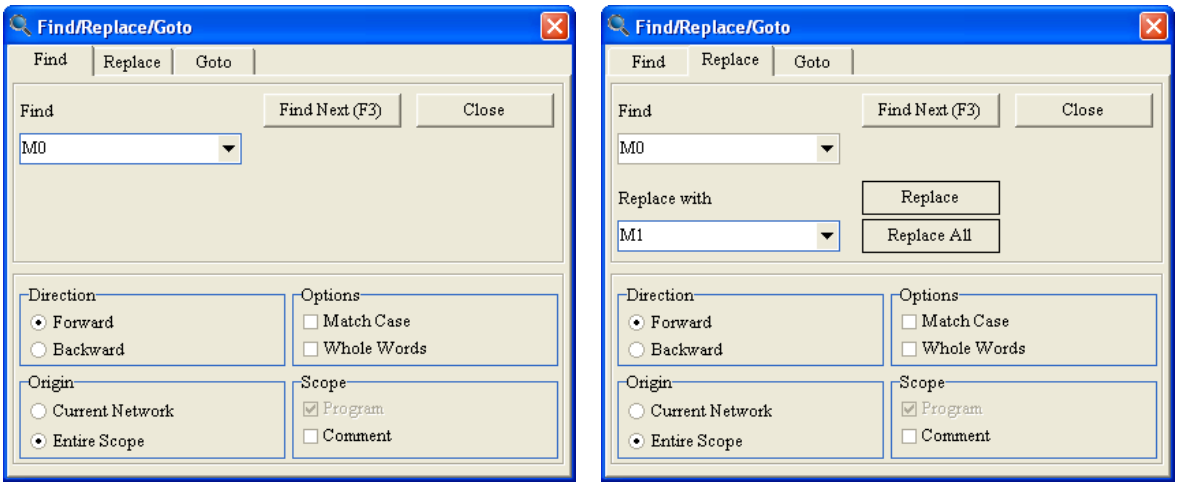
In the editing environment in which a ladder diagram/function block diagram can be created, the **Find** item, the **Replace** item, and the **Goto** item are on the **Edit** menu. The icons corresponding to these items are on the toolbar in the editing environment. The same window is opened, whether users click **Find**, **Replace**, or **Goto**. The page displayed in the window opened varies with the item clicked. There are three tabs in the window opened.

OR



13

The **Find** page and the **Replace** page are shown below. Type the object which will be searched for, and select search conditions. The users can do a search on the object typed in the **Find** box by clicking **Find Next (F3)** or pressing F3 on the keyboard. The users can still do a search on the object typed in the **Find** box by means of pressing F3 on the keyboard even if the **Find/Replace/Goto** window is closed.



Search condition	Description
Find	Users can type the object which will be searched for, or select an object which has been searched for after they click ▼ in the box.
Direction	Users can search down or search up.
Options	**Match Case**: ISPSoft searches only for words that match the case of the word that users type in the **Find** box. **Whole Words**: **Whole Words** instructs ISPSoft to find complete words only, and not to find words that only contain what users type.
Origin	**Current Network**: The search starts at the position selected. **Entire Scope**: The search starts at the beginning of the program or the end of the program.

Search condition	Description
Scope	Users can select ranges which will be searched.

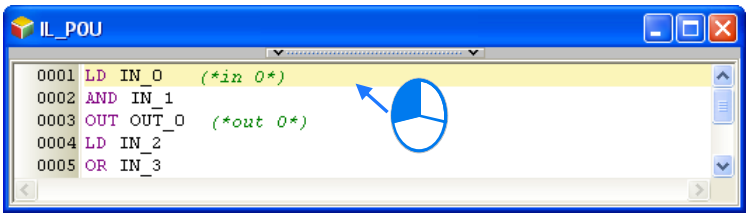
The users can type a replacement object in the **Replace with** box, or select a replacement object which has been typed after they click in the box. If the users click **Replace**, an occurrence of the object typed in the **Find** box will be replaced by the replacement object. If users click **Replace All**, all the occurrences of the object typed in the **Find** box will be replaced.

If the users want to go to a specific position in the program, they can click the **Goto** tab in the **Find/Replace/Goto** window, select **Network Label** or **Bookmark** in the **Location Type** drop-down list box, select an item in the **Index** drop-down list box, and click **Goto**.

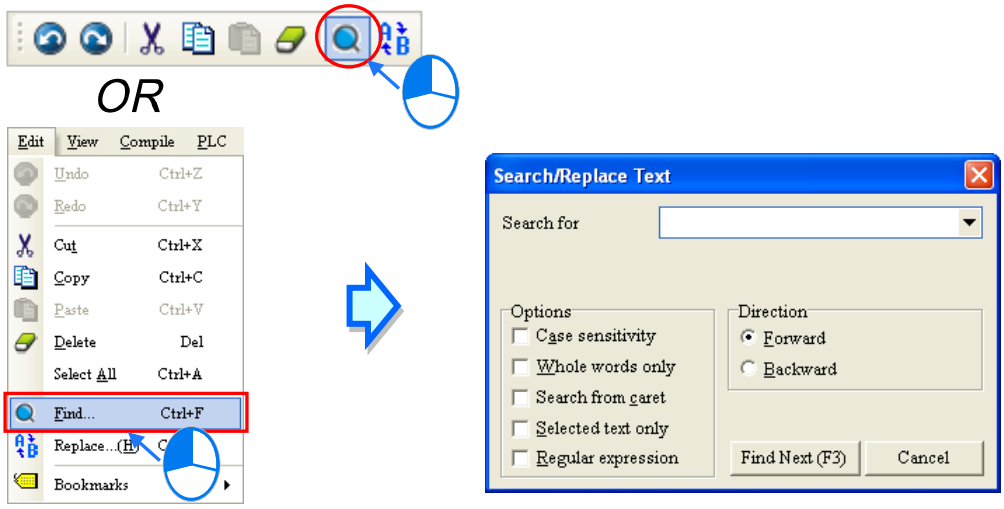
13

13.1.4 Finding/Replacing Text on an Instruction List/in an Structured Text

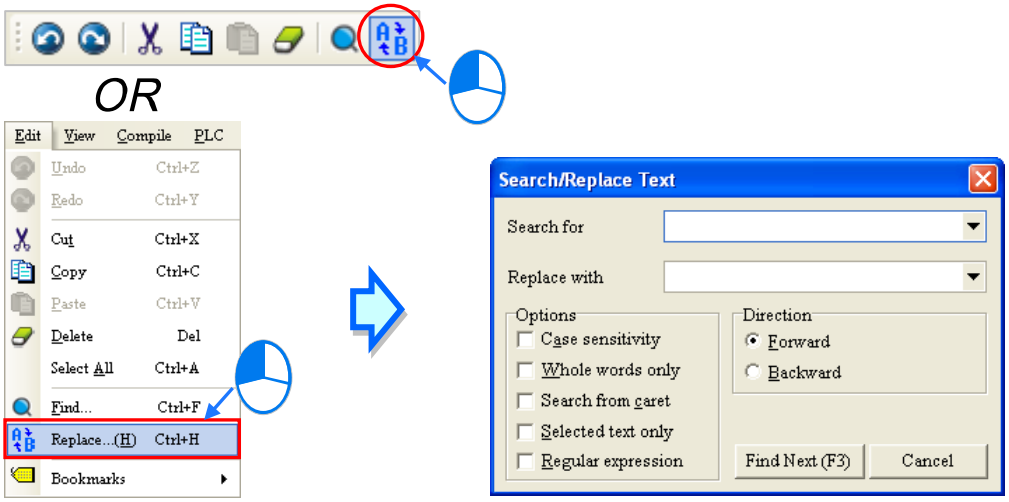
Users can only find and replace text on an instruction list/in a structured text. The functions described in this section are used in the present program editing window. Users can open a program editing window, and click the position in the working area at which they want to start searching. If the users click the local symbol table in the program editing window, the local symbol table will be searched. Please refer to section 13.1.6 for more information about using the **Find** function in a symbol table.



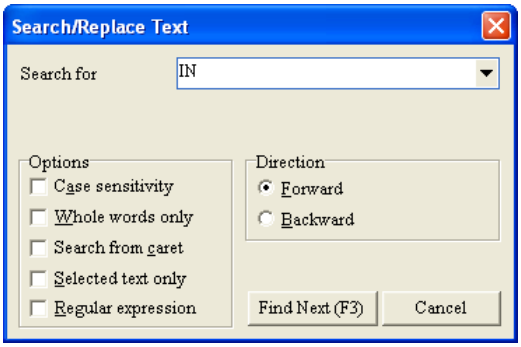
After the users click **Find** on the **Edit** menu, or on the toolbar, the **Search/Replace Text** window will appear.

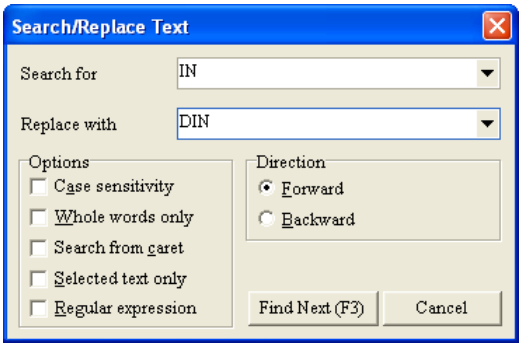


After the users click **Replace** on the **Edit** menu, or on the toolbar, the **Search/Replace Text** window will appear.



The **Find** page and the **Replace** page are shown below. Type the object which will be searched for, and select search conditions. If the users want to use the **Replace** function, they have to type a replacement object in the **Replace with** box. The users can do a search on the object typed in the **Search for** box by clicking **Find Next (F3)** or pressing F3 on the keyboard. The users can still do a search on the object typed in the **Search for** box by means of pressing F3 on the keyboard even if the **Find/Replace/Goto** window is closed.

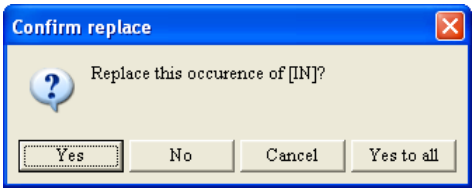




Search condition	Description
Search for	Users can type the object which will be searched for, or select an object which has been searched for after they click ▼ in the box.
Case sensitivity	ISPSoft searches only for words that match the case of the word that users type in the **Search for** box.
Whole words only	**Whole words only** instructs ISPSoft to find complete words only, and not to find words that only contain what users type.
Search from caret	The search starts at the text cursor. (If this checkbox is not selected, the system will search the whole program.)
Selected text only	The search starts at the text selected. (If this checkbox is not selected, the system will search the whole program.)
Regular expression	After this checkbox is selected, regular expressions can be used.
Direction	Users can search down or search up.

*** A regular expression is a type of standard syntax. For example, "*" is a wildcard. Please refer to related technical documents for more information.**

If the users use the **Replace** function, the **Confirm replace** window will appear after an object meeting the search conditions selected is found.

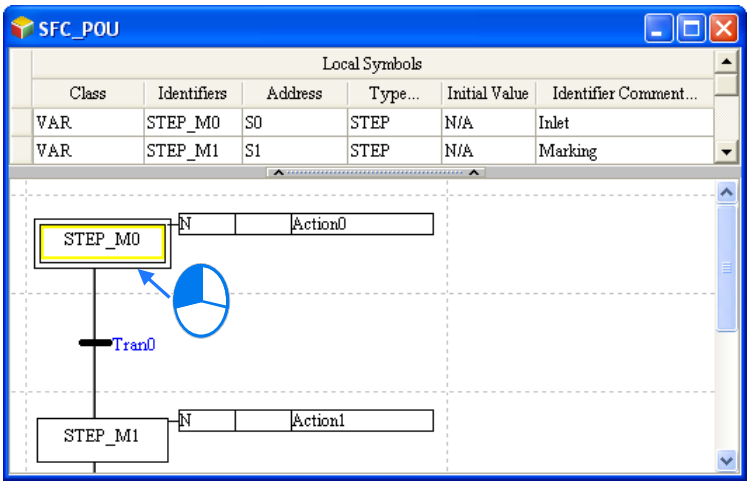


Button	Function
Yes	The object found is replaced by the replacement object, and the system searches for the next occurrence of the object typed in the **Search for** box.
No	The object found is not replaced by the replacement object, and the system searches for the next occurrence of the object typed in the **Search for** box.
Cancel	The search is canceled, and the **Confirm replace** window is closed,
Yes to all	All the objects meeting the search conditions selected are replaced by the replacement object.

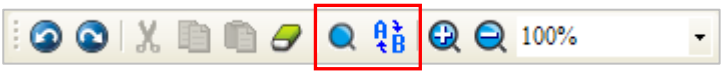
13.1.5 Finding/Replacing a Step or a Transition in a Sequential Function Chart

Users can only use the **Find** function and the **Replace** function in a sequential function chart. Besides, only the steps and the transitions in a sequential function chart can be searched for and replaced. If users want to use the **Find/Replace** function in an action/transition, they have to open the window for the action/transition, make sure of the programming language used to create the action/transition, and refer to section 13.1.3 and section 13.1.4.

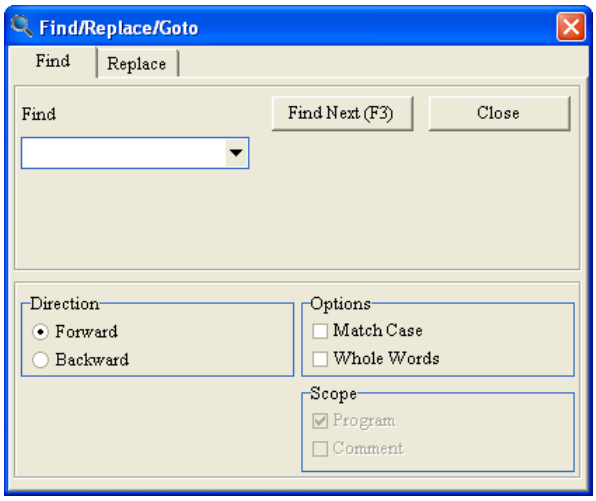
The functions described in this section are used in the present program editing window, and are not used in the whole project. Users can open a program editing window, and click the position in the working area at which they want to start searching. If the users click the local symbol table in the program editing window, the local symbol table will be searched. Please refer to section 13.1.6 for more information about using the **Find** function in a symbol table.



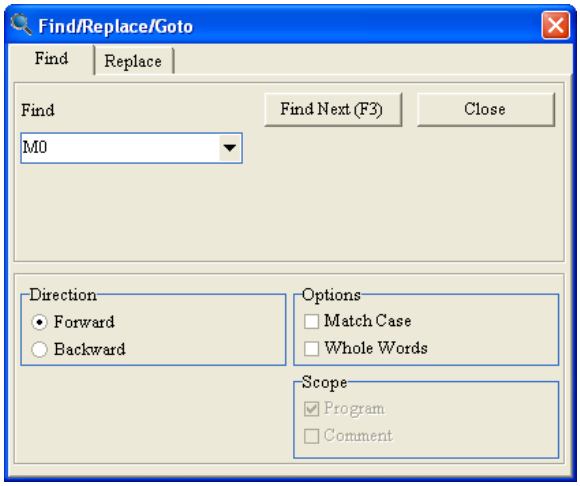
In the editing environment in which a sequential function chart can be created, the **Find** item and the **Replace** item are on the **Edit** menu. The icons corresponding to these items are on the toolbar in the editing environment. The same window is opened, whether users click **Find** or **Replace**. The page displayed in the window opened varies with the item clicked. There are two tabs in the window opened.

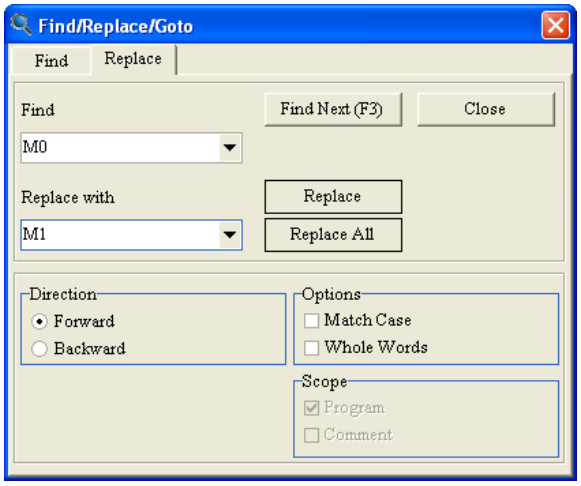


OR



The **Find** page and the **Replace** page are shown below. Type the object which will be searched for, and select search conditions. The users can do a search on the object typed in the **Find** box by clicking **Find Next (F3)** or pressing F3 on the keyboard. The users can still do a search on the object typed in the **Find** box by means of pressing F3 on the keyboard even if the **Find/Replace/Goto** window is closed.

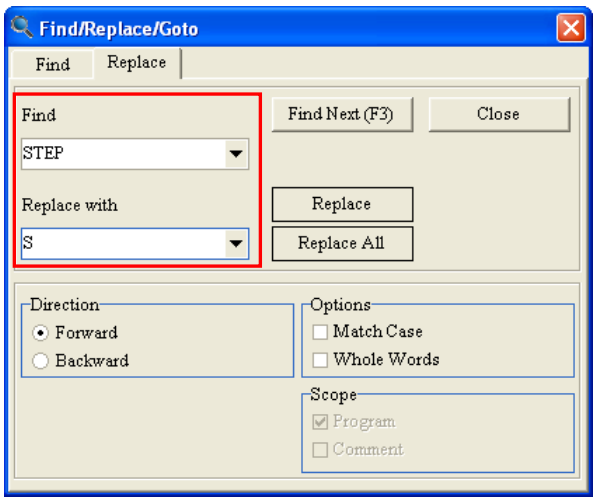




13

Search condition	Description
Find	Users can type the object which will be searched for, or select an object which has been searched for after they click ▼ in the box.
Direction	Users can search down or search up.
Options	**Match Case**: ISPSoft searches only for words that match the case of the word that users type in the **Find** box. **Whole Words**: **Whole Words** instructs ISPSoft to find complete words only, and not to find words that only contain what users type.
Scope	Users can select ranges which will be searched.

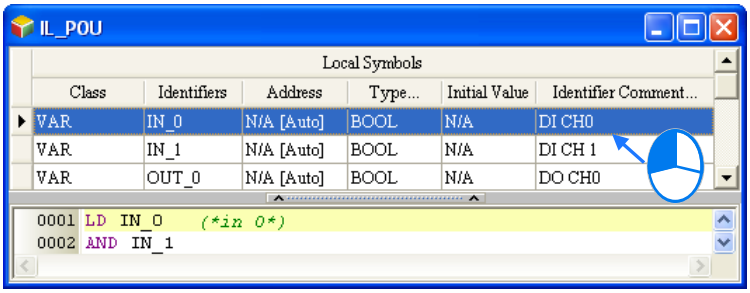
The users can type a replacement object in the **Replace with** box, or select a replacement object which has been typed after they click ▼ in the box. If the users click **Replace**, an occurrence of the object typed in the **Find** box will be replaced by the replacement object. If users click **Replace All**, all the occurrences of the object typed in the **Find** box will be replaced.



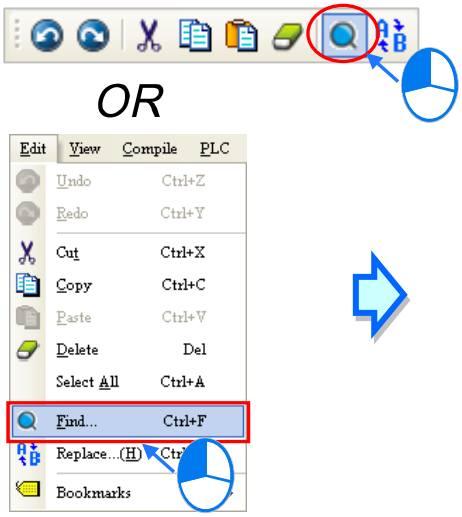
13.1.6 Finding an Identifier in a Symbol Table

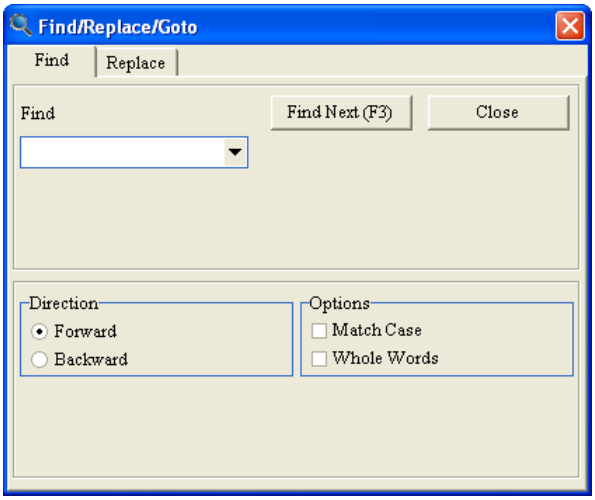
Users can only use the **Find** function in a local symbol table/global symbol table. They can not use the **Replace** function and the **Goto** function in a local symbol table/global symbol table. Besides, only the identifiers in a local symbol table/global symbol table can be searched for. The function described in this section is used in the present program editing window, and is not used in the whole project. Users can open an editing window, and click the position in the symbol table at which they

want to start searching. If the users click the working area in a program editing window, working area will be searched. Please refer to section 13.1.3~section 13.1.5 for more information about using the **Find/Replace** function in a working area.



After the users click **Find** on the **Edit** menu, or on the toolbar, the **Search/Replace/Goto** window will appear.





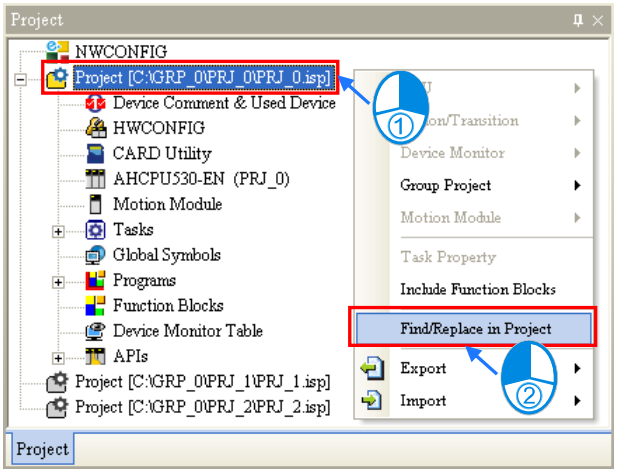
The users can do a search on the object typed in the **Find** box by clicking **Find Next (F3)** or pressing F3 on the keyboard. The users can still do a search on the object typed in the **Find** box by means of pressing F3 on the keyboard even if the **Find/Replace/Goto** window is closed.

Search condition	Description
Find	Users can type the object which will be searched for, or select an object which has been searched for after they click in the box.
Direction	Users can search down or search up.
Options	**Match Case**: ISPSoft searches only for words that match the case of the word that users type in the **Find** box. **Whole Words**: **Whole Words** instructs ISPSoft to find complete words only, and not to find words that only contain what users type.

13.1.7 Finding / Replacing a Device or a Symbol in a Project

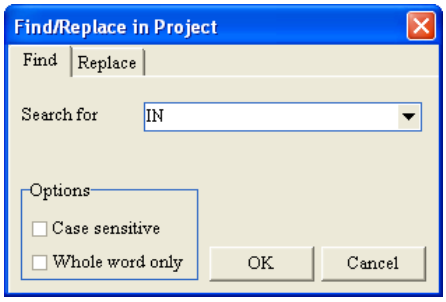
The **Find/Replace** function is used to search for/replace a device or a symbol in a project. The global symbol, the local symbol tables in the POUs, and the working areas in the project are searched. If there is a group of projects, the **Find/Replace** function is used in the project which is being edited.

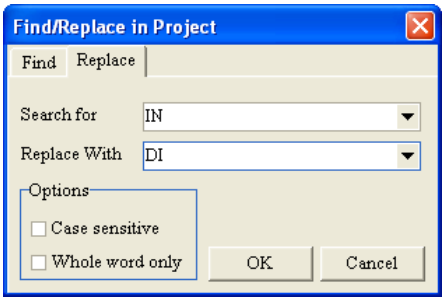
Right-click any item in the project management area, and then click **Find/Replace in Project** on the context menu.



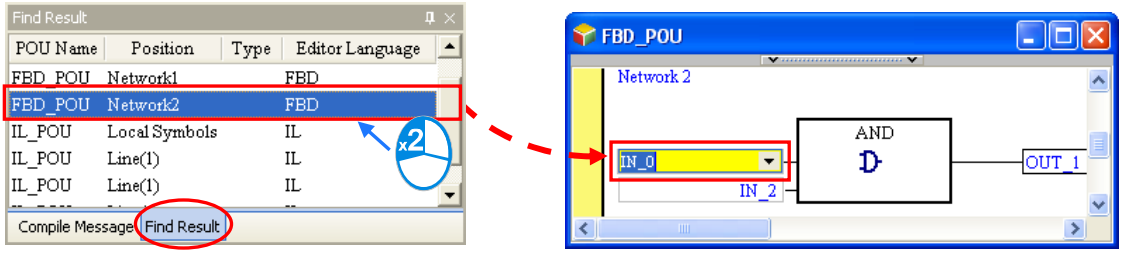
13

The **Find** page and the **Replace** page are shown below. There are two tabs in the **Find/Replace in Project** window. Type the object which will be searched for, and select search conditions. If the users click **OK** on the **Find** page, the system will use the **Find** function. If the users click **OK** on the **Replace** page, all the occurrences of the object typed in the **Search for** box will be replaced by the replacement object. The replacement can not be undone.





After the users use the **Find/Replace** function, the search results will be displayed in the message display area at the bottom of the ISPSoft window. If the users double-click a search result, the system will lead the users to the position of the object corresponding to the search result.

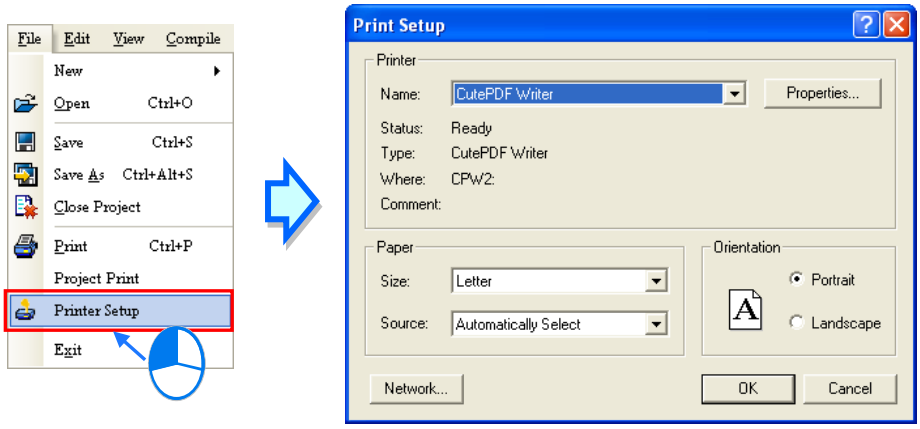


13.1.8 Printing Function

ISPSoft provides a printing function.

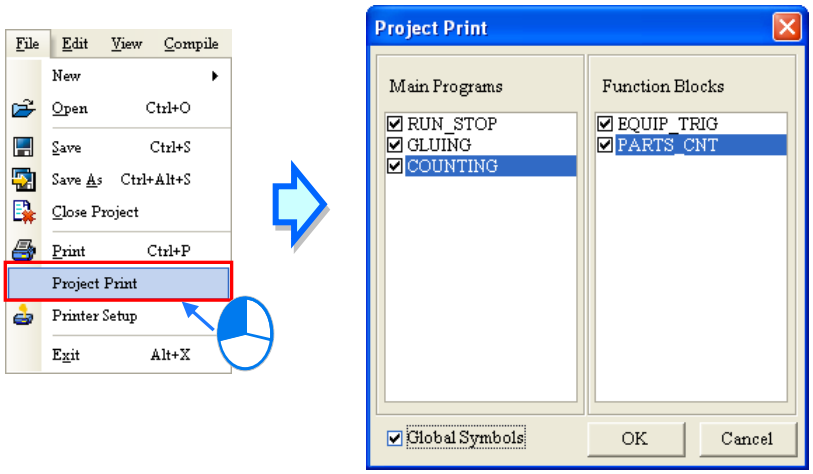
- **Printer Setup**

  After users click **Printer Setup** on the **File** menu, the **Print Setup** window will be opened. Before a document is printed, the users can select a printer, and set a print format.

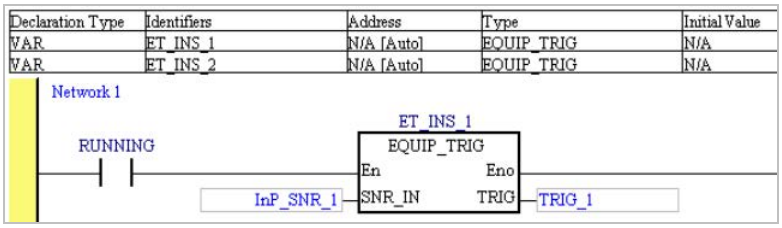


- **Project Print**

  After users click **Project Print** on the **File** menu, users can select items which will be printed in the **Project Print** window. The items which can be selected are the main programs, the function blocks, and the global symbols.

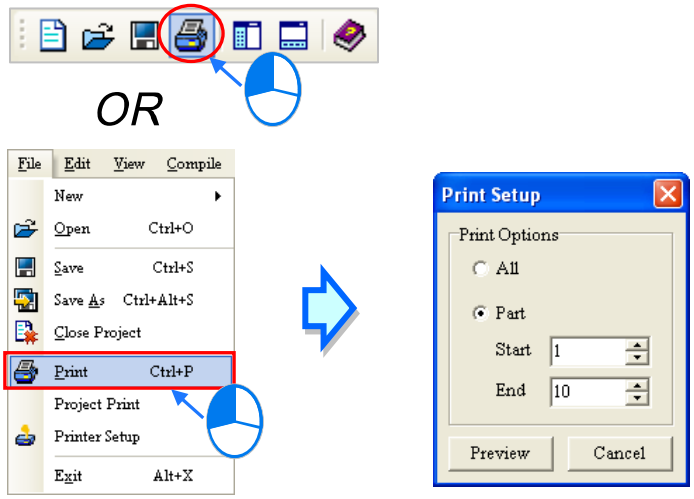


An example of a print result is shown below. The contents which are printed include a local symbol table. If there are comments in the window whose contents are printed, the comments will be printed.



- **Print**

  If users want to print the contents of the present window, they can click **Print** on the **File** menu or on the toolbar. The users can set a print range in the **Print Setup** window. The print result varies with the contents of the present window. Please refer to the table below for more information.

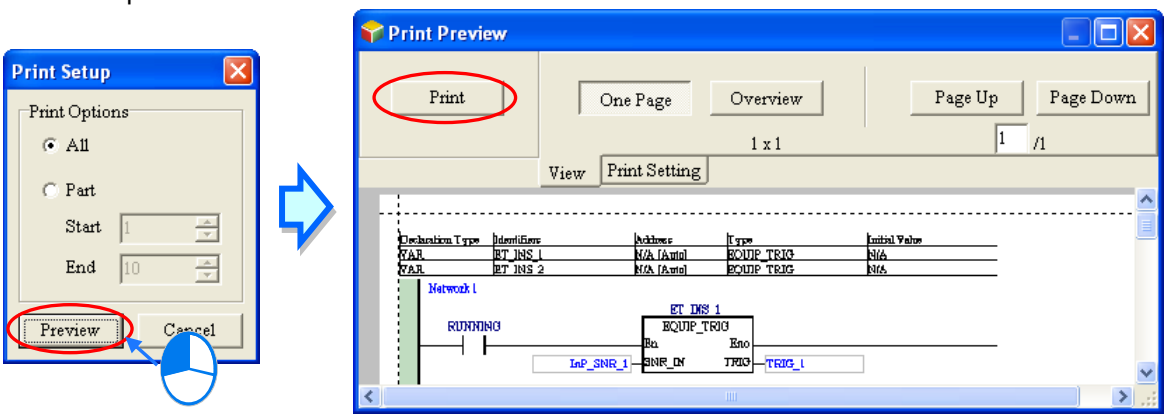


13

Window	All	Part
Global symbol table	The complete global symbol table is printed.	Users can not print a part of the global symbol table.
Ladder diagram/Function block diagram	The complete program and local symbol table are printed.	The complete local symbol table and a part of the program are printed. (The start number and the end number are network numbers.)
Instruction list/Structured text	The complete program and local symbol table are printed.	The complete local symbol table and a part of the program are printed. (The start number and the end number are network numbers.)
Sequential function chart	The complete local symbol table and sequential function chart are printed. (The actions and the transitions are not printed.)	Users can not print a part of the sequential function chart.

***.After users open the window for an action/transition, they can print the action/transition.**

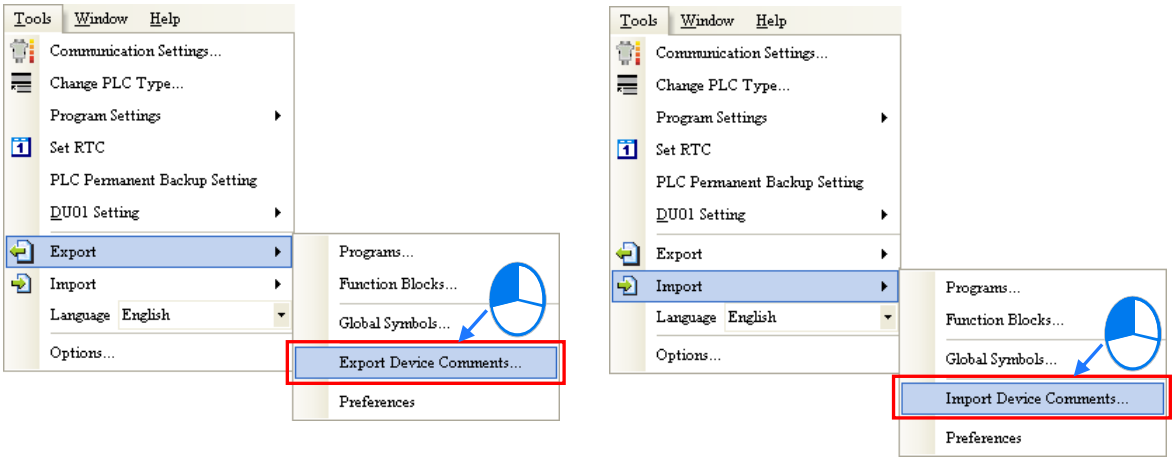
After the users click **Preview** in the **Print Setup** window, the **Print Preview** window will be opened. Please refer to appendix C for more information about the print management tool that ISPSoft provides.



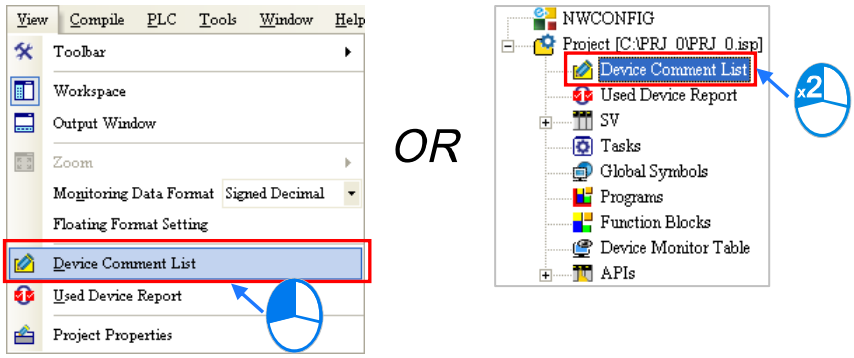
13.2 Managing Devices and Registers

13.2.1 Device Comment List for a DVP Series PLC

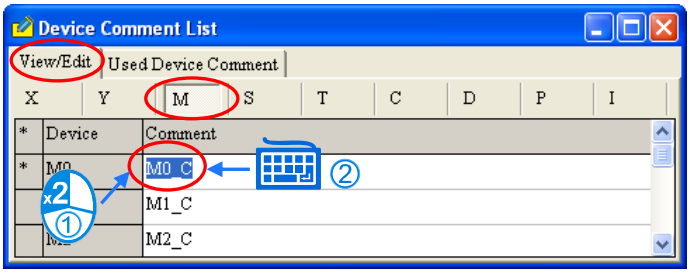
If users want to export the device comment list from a project for a DVP series PLC, they can click the **Tools** menu, point to **Export**, and click **Export Device Comments…**. If the users want to import the device comment list, they can click the **Tools** menu, point to **Import**, and click **Import Device Comments…**.



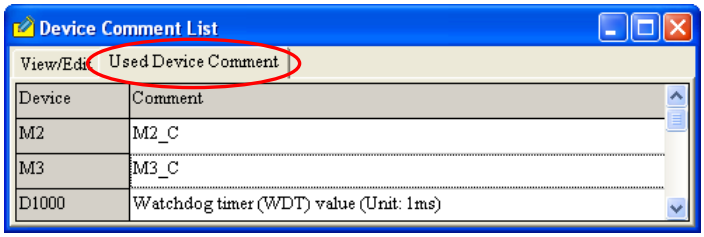
If the users want to manage the comments on devices, they can click **Device Comment List** on the **View** menu, or double-click **Device Comment List** in the project management area. After the users click **Device Comment List** on the **View** menu, or double-click **Device Comment List** in the project management area, the system will open the **Device Comment List** window.



The comments on devices are on a **View/Edit** page. After the users click a tab which indicates a device type in the window, the comments on devices are displayed. The mark "*" at the left side of a device indicates that the device is used in the program. If the users want to make a comment on a device, they can double-click the **Comment** cell for the device, and type the comment.



The devices on which comments are made and the comments on the devices are on the **Used Device Comment** page. After the users double-click the **Comment** cell for a device, they can edit the comment on the device.

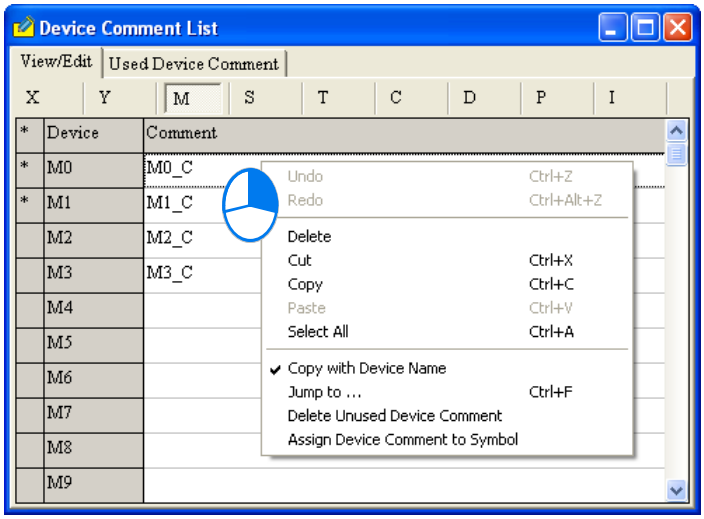


***1. The comments on devices are separate from the comments on symbols. The comments on devices appear in the Device Comment List window, while the comments on symbols do not appear in the Device Comment List window.**

***2. Users can edit the default comments on the special relays and the special registers in a project. However, the new comments are valid only in this project.**

13

After the uses right-click the table in the **Device Comment List** window, a context menu will appear. The context menu comprises basic items and special items.



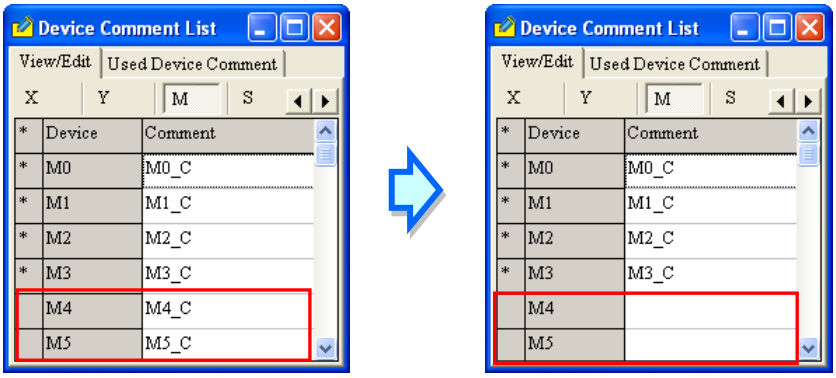
- **Jump to…**

  After the users click **Jump to…** on the context menu, the **Jump to…** window will appear. The users can type a device name in the **Device Name** box. After the users click **OK** in the **Jump to…** window, they can easily view the device name and the comment on the device on a **View/Edit** page.

  The device name typed must be matched with the device type clicked, otherwise, the device name will not be found. If the users click **Jump to…** on the context menu which appears after they right-click the **Used Device Comment** page, the device name typed in the **Device Name** box in the **Jump to…** window must be on the **Used Device Comment** page.

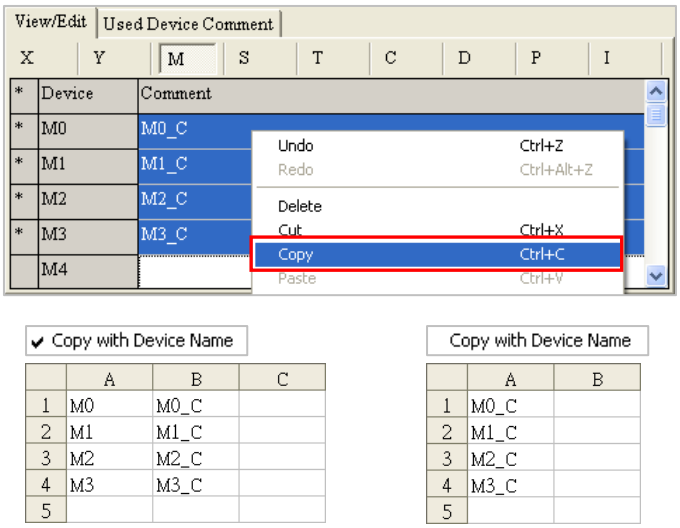
- **Delete Unused Device Comment**

  After the users click **Delete Unused Device Comment** on the context menu, the system will delete the comments on the devices which are not used in the program.



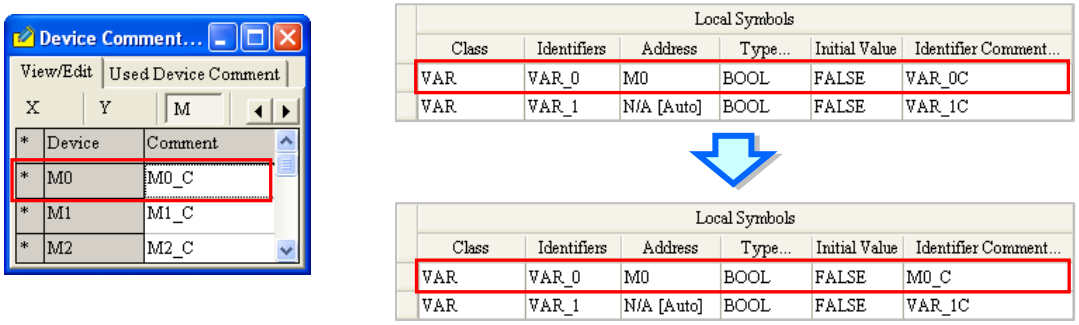
- **Copy with Device Name**

  If the users click **Cut/Copy** on the context menu, the items selected in the **Device Comment List** window will be cut/copied to the Windows Clipboard. The items cut/copied can be pasted into a document in Microsoft Excel. If **Copy with Device Name** on the context menu is clicked before **Cut/Copy** on the context menu is clicked, the contents which will be cut/copied will include device names. Please compare the two Excel tables below.



- **Assign Device Comment to Symbol**

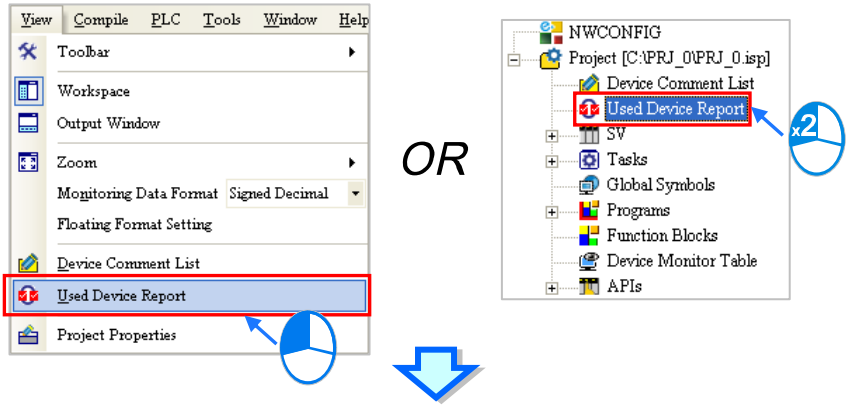
  If symbols are assigned devices, the comments on the devices will be assigned to the symbols after the users click **Assign Device Comment to Symbol** on the context menu.

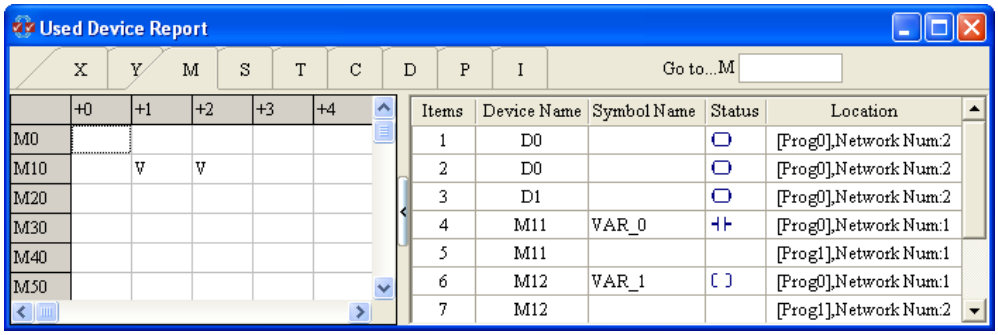


13.2.2 Report on the Devices Used in a DVP Series PLC

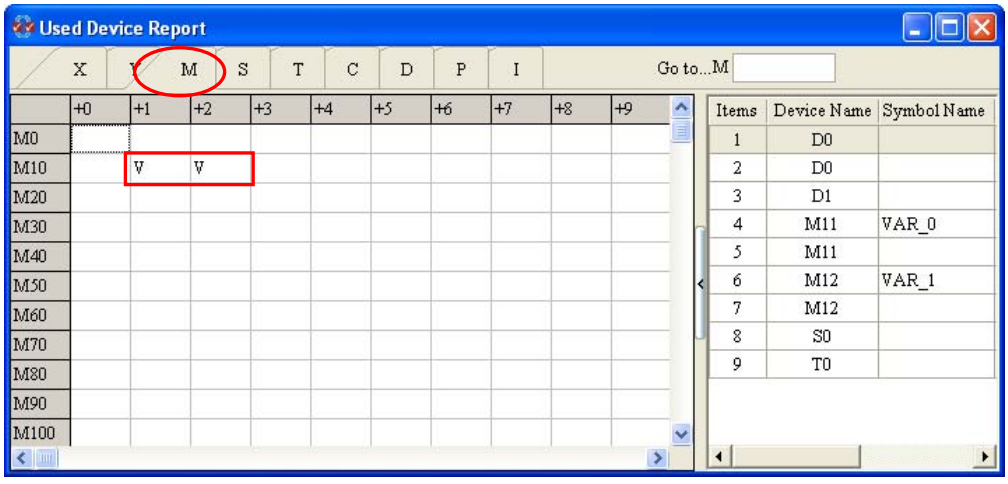
When users edit a project for a DVP series PLC, they can refer to the report on the devices used in the project. However, the users have to compile the program in the project before they refer to the report on the devices used in the project. Besides, if the users modify the program after the window for the report is opened, they have to close the window, compile the new program, and open the window again to get the latest information.

If users click **Used Device Report** on the **View** menu, or double-click **Used Device Report** in the project management area, the **Used Device Report** window will be opened.



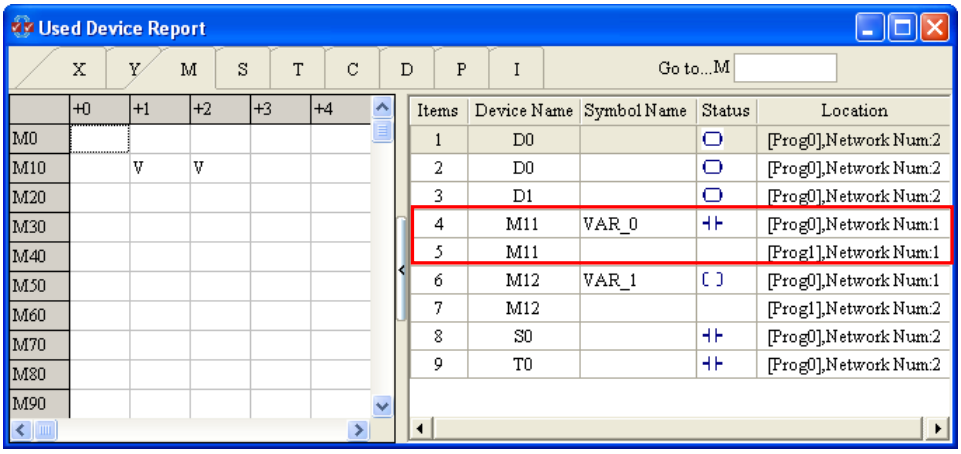


The **Used Device Report** window is composed of two parts. A table of devices is in the left part of the window. The users can click the tabs at the top of the window. The checks in the cells for devices indicate that the devices are used in the program. For example, there are checks in the cells for M11 and M12.



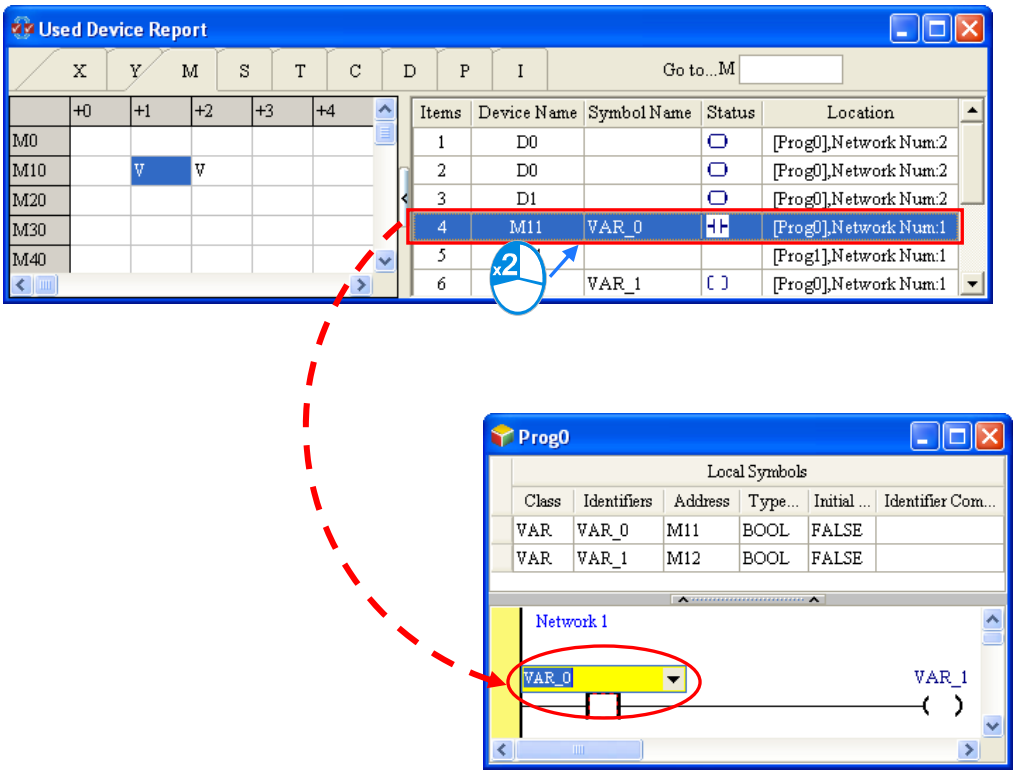
A table of devices used in the program is in the right part of the window. The devices listed are not classified. If a certain device is used in different positions in the program, the information about the

positions will be listed. For example, M11 in the figure below is used in two positions in the program.

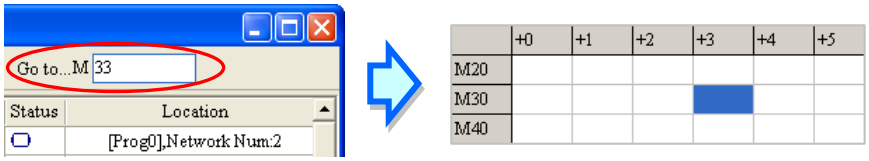


13

After the users double-click a row in the right part of the window, the system will lead the users to the position corresponding to the row.



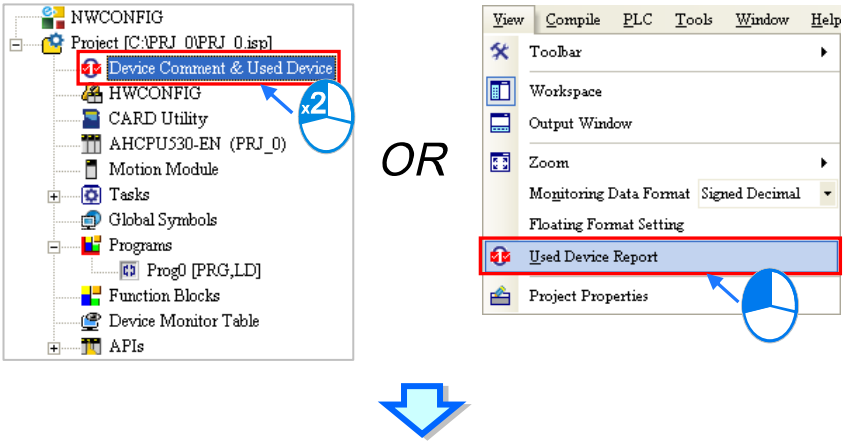
If the users press Enter on the keyboard after they type a device number in the **Go to…** box, the blue cursor in the table in the left part of the window will move to the position corresponding to the device number.

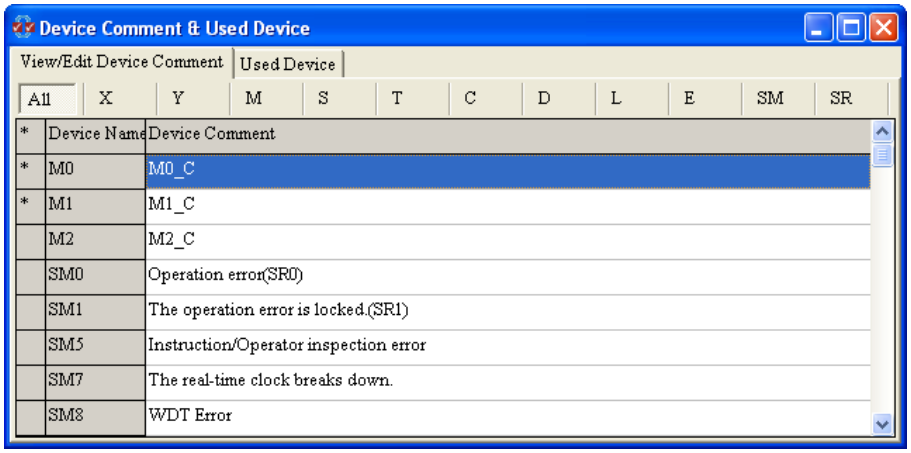


13.2.3 Managing Devices in an AH500 Series CPU Module

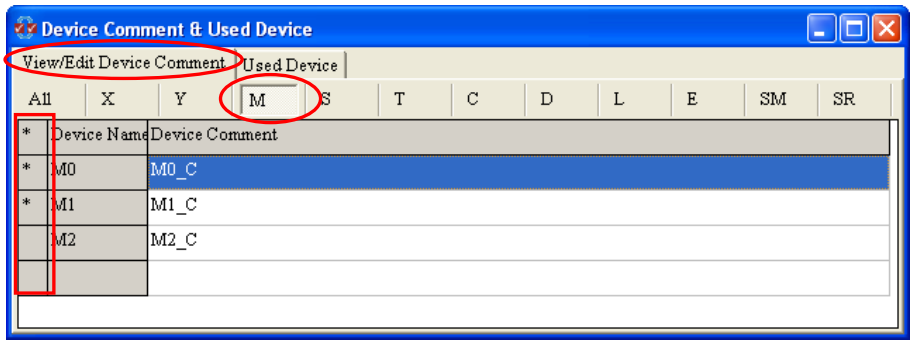
Users can manage the devices in an AH500 series CPU module. After users click **Used Device Report** on the **View** menu, or double-click **Device Comment & Used Device** in the project management area, the **Device Comment & Used Device** window will be opened.

There are two tabs at the top of the **Device Comment & Used Device** window. If the users modify the program or comments after the window is opened, they have to close the window, compile the new program, and open the window again to get the latest information.



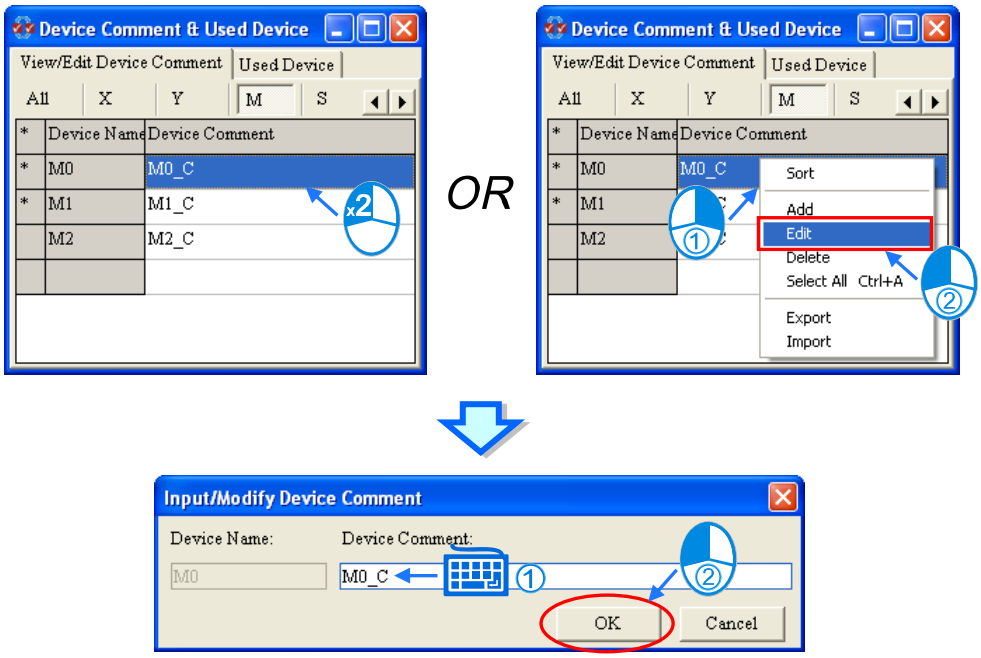


The devices on which comments are made and the comments on the devices are on the **View/Edit Device Comment** pages. The tabs in the window indicate device types. After the users click **All**, the devices on which comments are made are displayed. Besides, the mark "*" at the left side of a device indicates that the device is used in the program.

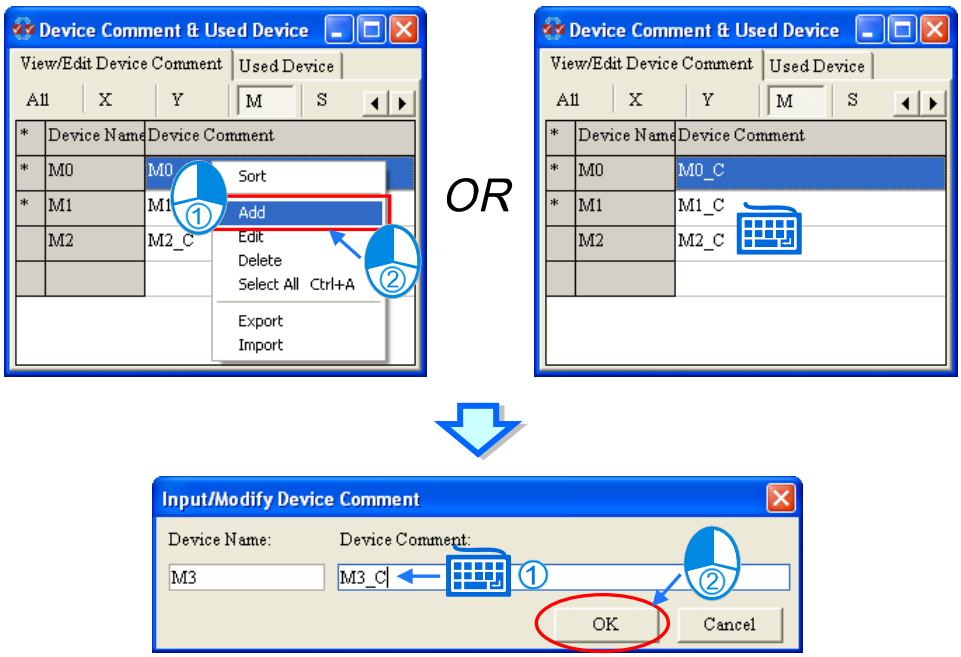


The users can modify the comments on a the **View/Edit Device Comment** page, and add comments to a **View/Edit Device Comment** page. If the users want to modify the comment on a device, they can double-click the comment, and modify the comment in the **Input/Modify Device**

Comment window. The users can also modify the comment on a device by right-clicking the comment, clicking **Edit** on the context menu, and modifying the comment in the **Input/Modify Device Comment** window. The users can edit the default comments on the SR devices and the SM devices in a project. However, the new comments are valid only in this project.

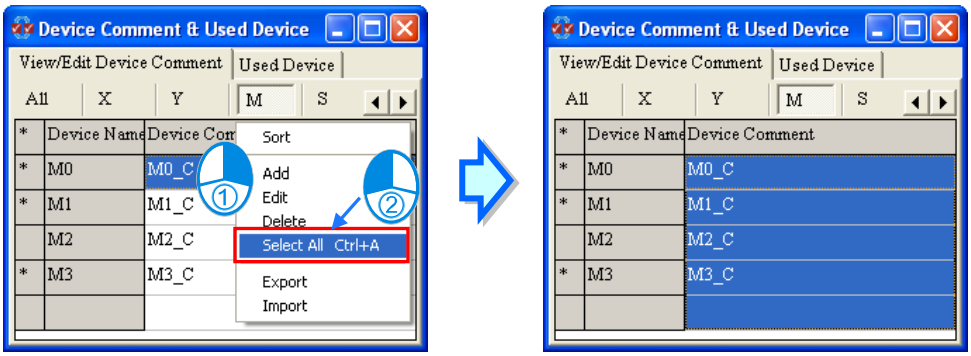


If the users want to add a comment to a **View/Edit Device Comment** page, they can right-click the **View/Edit Device Comment** page, click **Add** on the context menu, and type a device name and a comment in the **Input/Modify Device Comment** window. The users can also add a comment to a **View/Edit Device Comment** page by typing a device name directly, and typing a comment in the **Input/Modify Device Comment** window which appears.



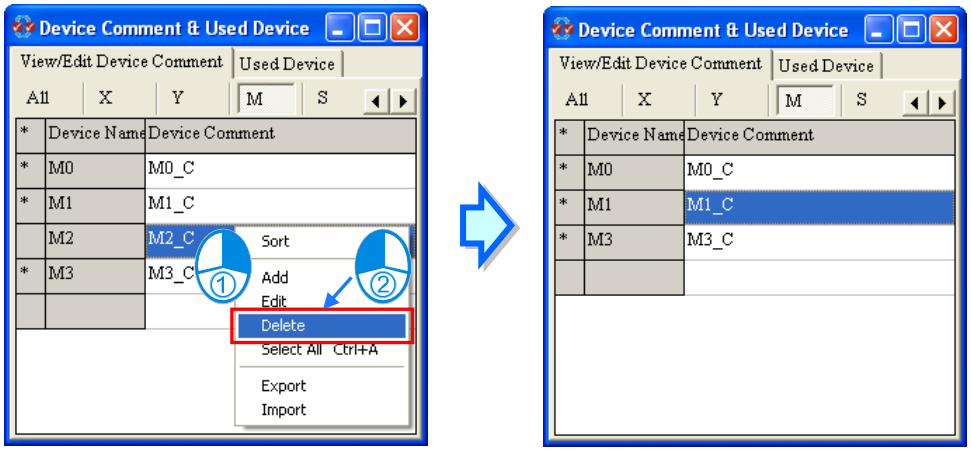
If the users want to delete the comment on a device on a **View/Edit Device Comment** page, they can select the comment, right-click the comment, and click **Delete** on the context menu. If the users want to delete a range of comments on a **View/Edit Device Comment** page, they can select the

first comment and the last comment while they press Shift on the keyboard, right-click a comment selected, and click **Delete** on the context menu. If the users want to delete all the comments on a **View/Edit Device Comment** page, they can right-click the page, select **Select All** on the context menu, right-click a comment selected, and click **Delete** on the context menu.

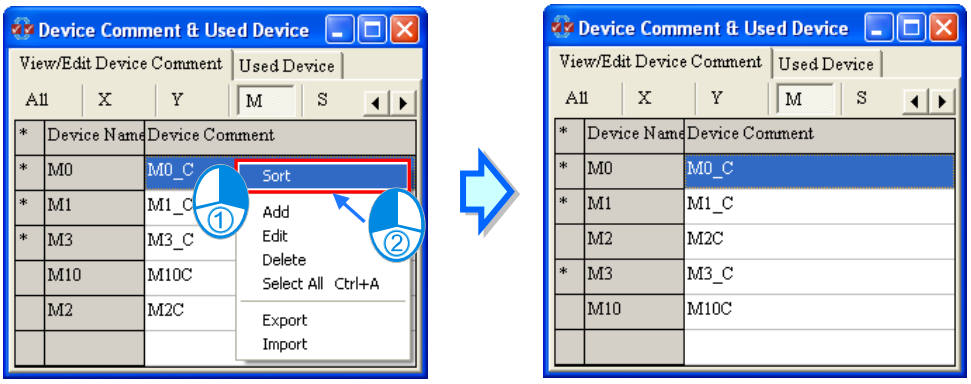


13

The users can also delete the device selected or the devices selected by pressing Delete on the keyboard. If the comments deleted include the comments on SM devices or the comments on SR devices, the comments on the SR devices or the comments on the SR devices will be recovered.



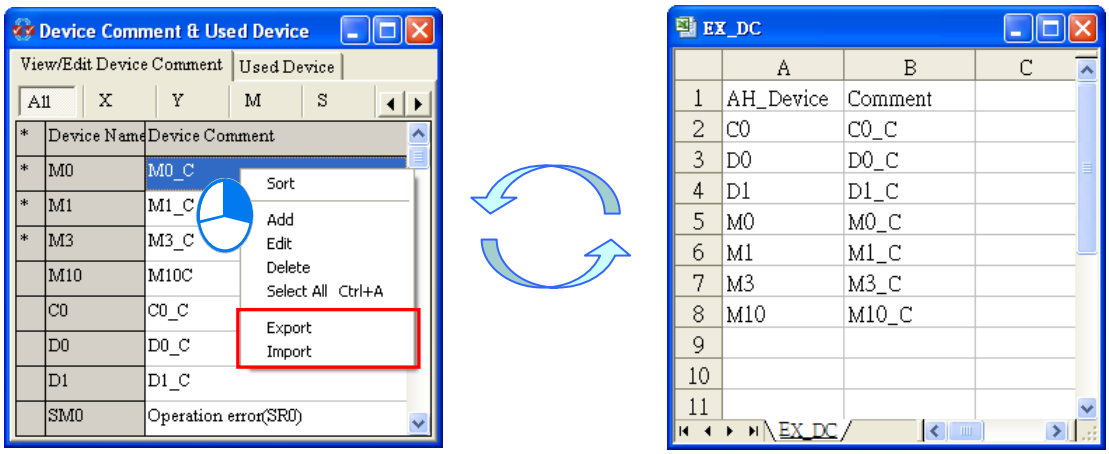
If the users want to rearrange the devices on a **View/Edit Device Comment** page, they can right-click the page, and click **Sort** on the context menu.



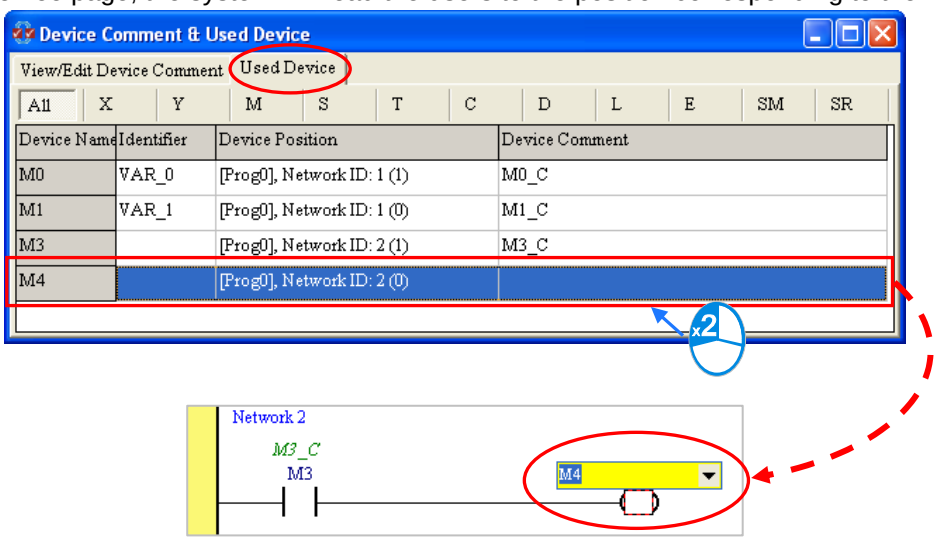
The users can export the comments on devices as a CSV file, edit the CSV file through Microsoft Excel, and import the new CSV file into ISPSoft. If the default comments on SM/SR devices are modified, the comments exported will include the new comments on theses SM/SR devices. The comments exported do not include the default comments on all the SM/SR devices. If the comments

imported include the comments on SM/SR devices, the default comments on these SM/SR devices will be modified after the comments are imported, and the default comments on the remaining SM/SR devices will not be changed after the comments are imported.

If the users want to export the comments on devices, they can right-click a **View/Edit Device Comment** page, click **Export** on the context menu, select a folder in the **Save in** drop-down list box in the **Export Device Comments** window, type a filename in the **File name** box, and click **Save**. If the users want to import a CSV file, they can right-click a **View/Edit Device Comment** page, click **Import** on the context menu, select the CSV file in the **Import Device Comments** window, and click **Open**.



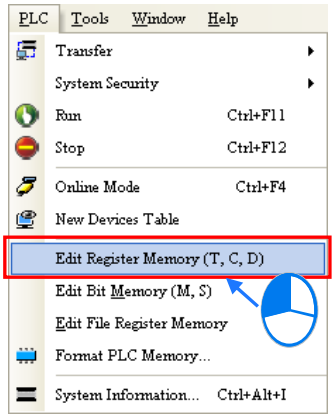
The devices used in the program and the information about the devices are on the **Used Device** pages. The tabs in the window indicate device types. Besides, after the users double-click a row on a **Used Device** page, the system will lead the users to the position corresponding to the row.



13.2.4 Editing the Values in the T/C/D Devices in a DVP Series PLC

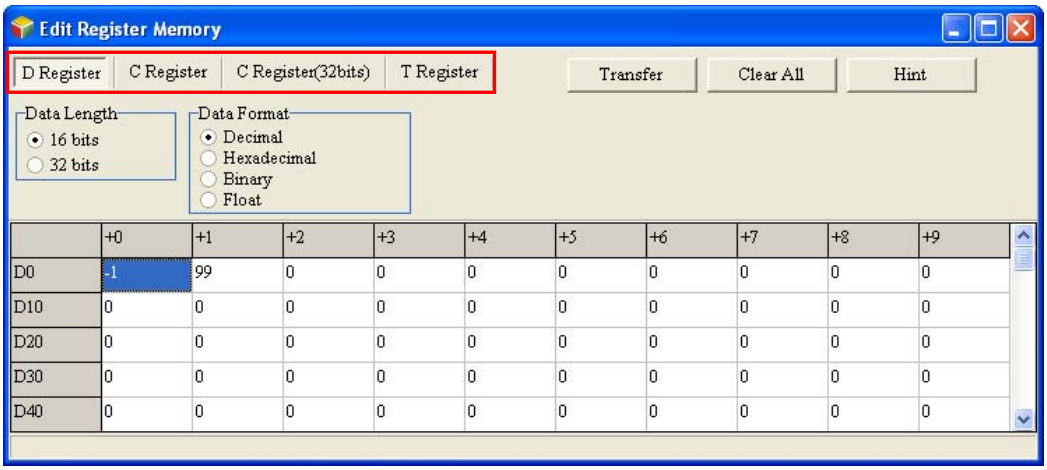
Users can edit the values in several T/C/D devices in a DVP series PLC at a time. The new values in these T/C/D devices can be saved and downloaded.

After users create a project for a DVP series PLC, they can click **Edit Register Memory (T, C, D)** on the **PLC** menu to open the **Edit Register Memory** window.



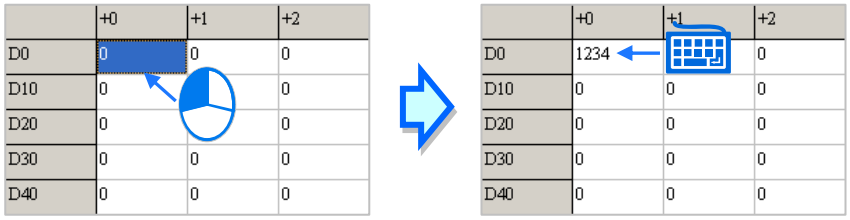
13

The values in the tables in the **Edit Register Memory** window are not the values uploaded form the PLC. They are the values saved last time. If the **Edit Register Memory** window in the project is opened for the first time, the default values in the tables are 0.
The tabs at the top of the **Edit Register Memory** window indicate device types.



	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9
D0	-1	99	0	0	0	0	0	0	0	0
D10	0	0	0	0	0	0	0	0	0	0
D20	0	0	0	0	0	0	0	0	0	0
D30	0	0	0	0	0	0	0	0	0	0
D40	0	0	0	0	0	0	0	0	0	0

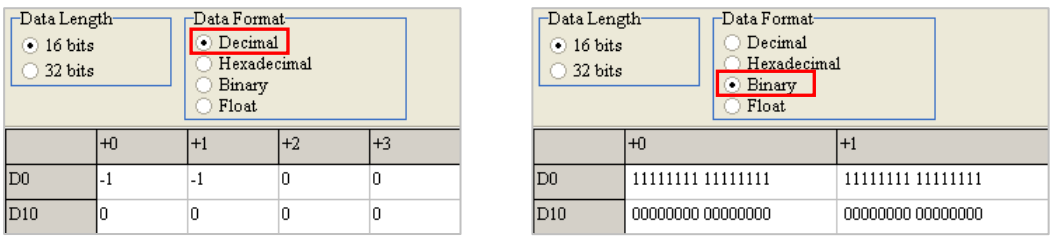
If the users want to edit the value in the cell for a device, they can click the cell for the device, and type a value.



	+0	+1	+2
D0	0	0	0
D10	0	0	0
D20	0	0	0
D30	0	0	0
D40	0	0	0

	+0	+1	+2
D0	1234		0
D10	0	0	0
D20	0	0	0
D30	0	0	0
D40	0	0	0

The users can select a data format in the **Data Format** section.

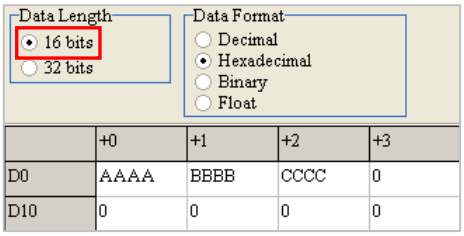


	+0	+1	+2	+3
D0	-1	-1	0	0
D10	0	0	0	0

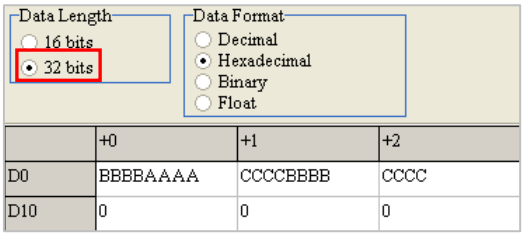
	+0	+1
D0	11111111 11111111	11111111 11111111
D10	00000000 00000000	00000000 00000000

The users can select the **16 bits** option button or the **32 bits** option button in the **Data Length** section. If the **32 bits** option button is selected, the value in the cell for a device includes the value in the next device.

Take the figures below for example. The hexadecimal value in D0 is AAAA, the hexadecimal value in D1 is BBBB, and the hexadecimal value in D2 is CCCC. If the **32 bits** option button in the **Data Length** section is selected, the value in the cell for D0 will include the value in D1, the value in the cell for D1 will include the value in D2, and the value in the cell for D2 will include the value in D3. In other words, if the **32 bits** option button is selected, the value in the cell for D0 will become BBBBAAAA, the value in the cell for D1 will become CCCCBBBB, and the value in the cell for D2 will become CCCC.

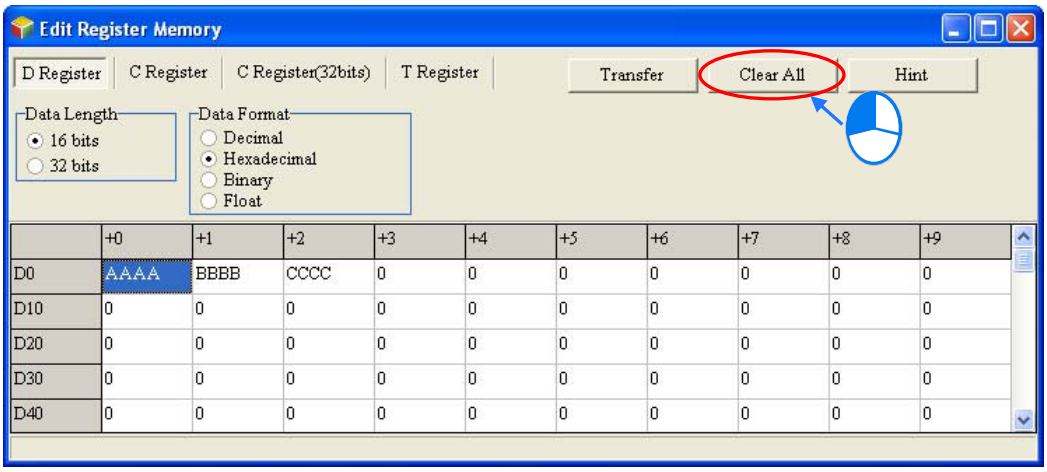


	+0	+1	+2	+3
D0	AAAA	BBBB	CCCC	0
D10	0	0	0	0

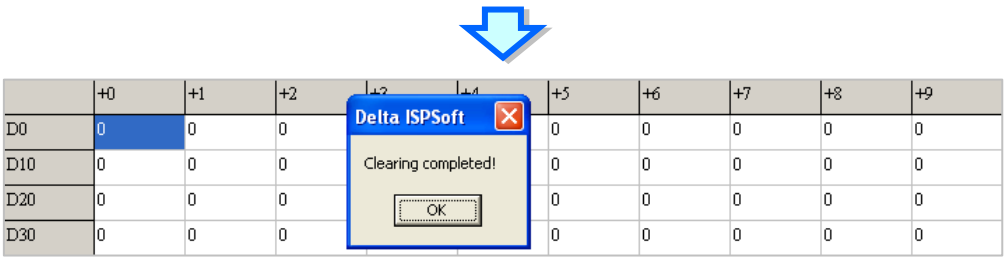


	+0	+1	+2
D0	BBBBAAAA	CCCCBBBB	CCCC
D10	0	0	0

If the users click **Clear All**, the values on the present page will be cleared to 0, and the values on the other pages will remain unchanged.

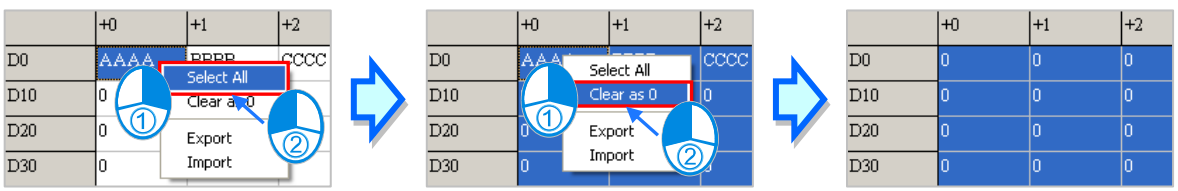


	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9
D0	AAAA	BBBB	CCCC	0	0	0	0	0	0	0
D10	0	0	0	0	0	0	0	0	0	0
D20	0	0	0	0	0	0	0	0	0	0
D30	0	0	0	0	0	0	0	0	0	0
D40	0	0	0	0	0	0	0	0	0	0

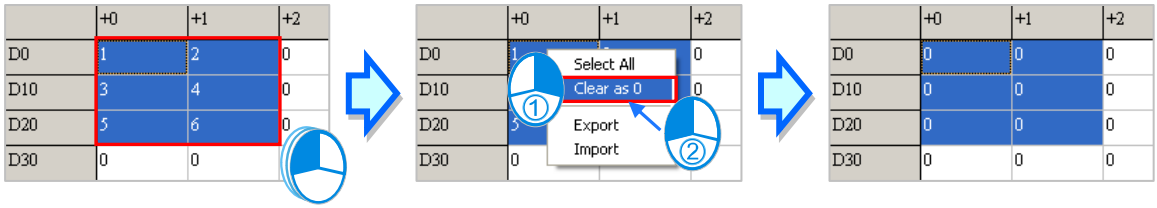


	+0	+1	+2	+5	+6	+7	+8	+9
D0	0	0	0	0	0	0	0	0
D10	0	0	0	0	0	0	0	0
D20	0	0	0	0	0	0	0	0
D30	0	0	0	0	0	0	0	0

Besides, the users can clear the values on the present page by right-clicking the present page, clicking **Select All** on the context menu, right-clicking the present page again, clicking **Clear as 0** on the context menu.

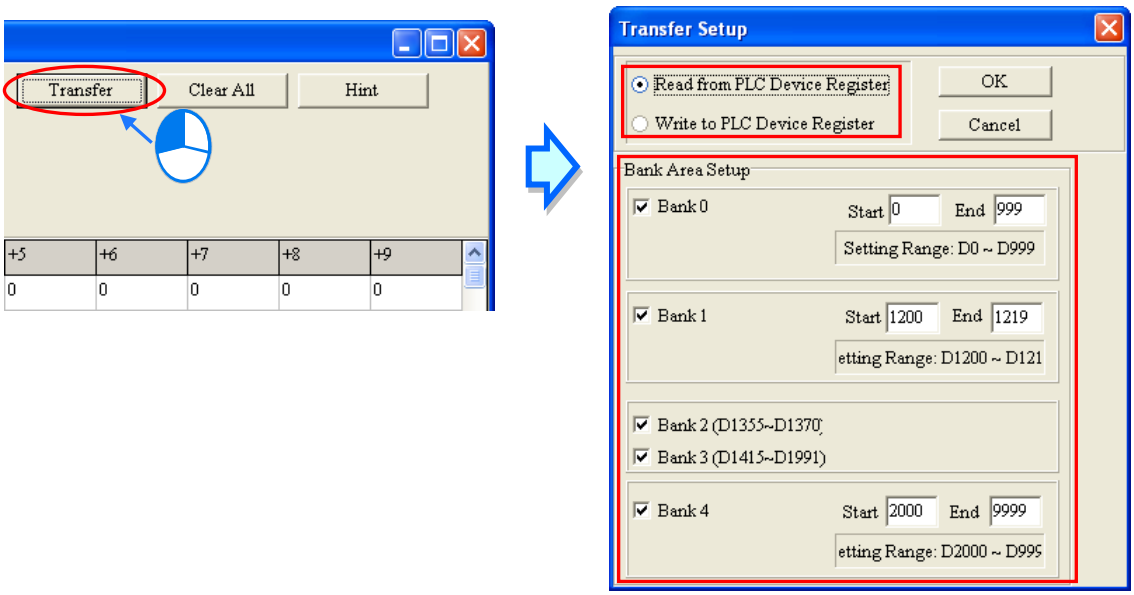


If the users want to clear the values in the cells for a range of devices on the present page, they can drag across the cells, right-click the present page, and click **Clear as 0** on the context menu.



After the users click **Transfer**, the **Transfer Setup** window will appear. The users can download the values in the tables to the PLC, or upload the values in the T/C/D devices from the PLC through the window. Before **Transfer** is clicked, the users have to make sure that ISPSoft is connected to the PLC normally. Please refer to section 2.4 for more information.

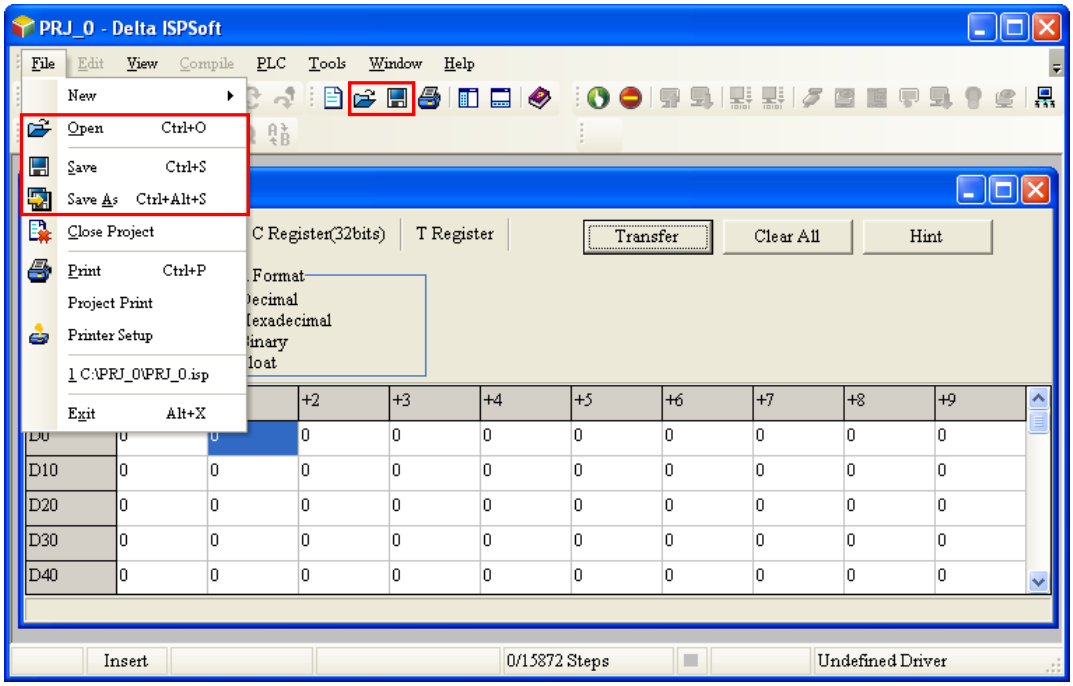
The users can select the **Read from PLC Device Register** option button or the **Write to PLC Device Register** option button. Besides, the users can select device ranges. The values in the devices which are not in the device ranges will remain unchanged after **OK** is clicked. After the setting is complete, the users can click **OK**. After **OK** is clicked, the values in the T devices, the values in the C devices, and the values in the D devices which are in the device ranges set will be transferred.



⚠ **Before the values are written into the registers, users have to make sure that the operation does not affect the system, or cause damage to the system and staff.**

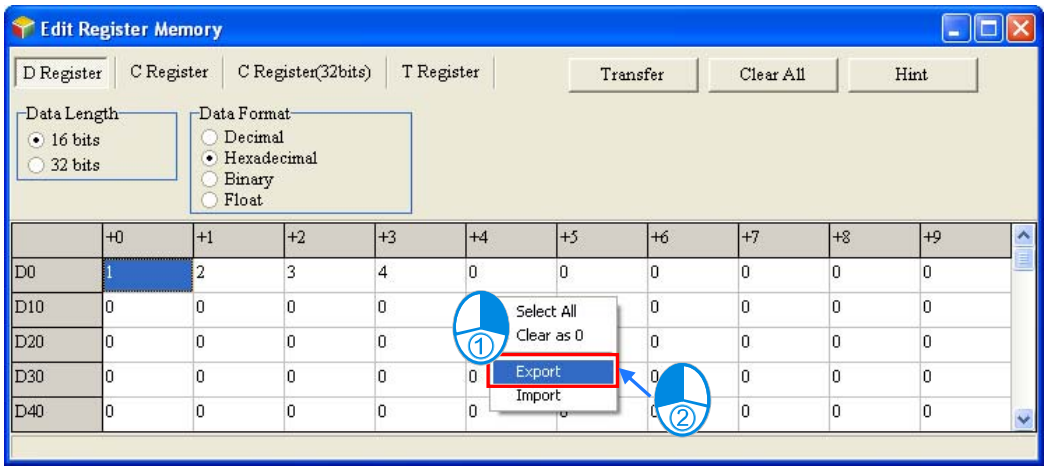
The values in the **Edit Register Memory** window can be saved with the project. If the present window is the **Edit Register Memory** window, the values in the **Edit Register Memory** window will be saved as a dvl file whose primary filename is the project name in the folder in which the project (*.isp) is saved after the users click **Save** on the **File** menu or the toolbar. If the users want to open the **Edit Register Memory** window in the project again, the system will open the dvl file in the folder in which the project (*.isp) is saved. If no dvl is in the folder in which the project is saved, the values in the **Edit Register Memory** window will be 0.

Besides, the users can save the values in the **Edit Register Memory** window as a dvl file in another folder after they click **Save As** on the **File** menu. If the users want to open the dvl file which was saved previously, they can click **Open** on the **File** menu or the toolbar.

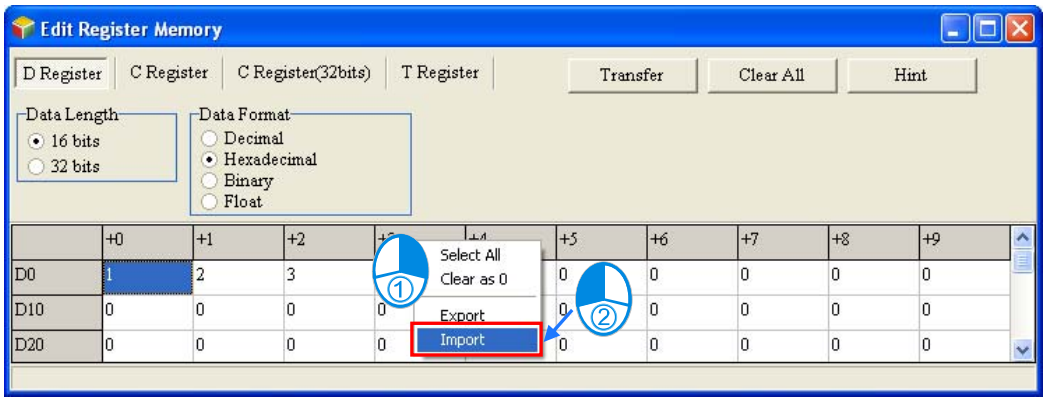


13

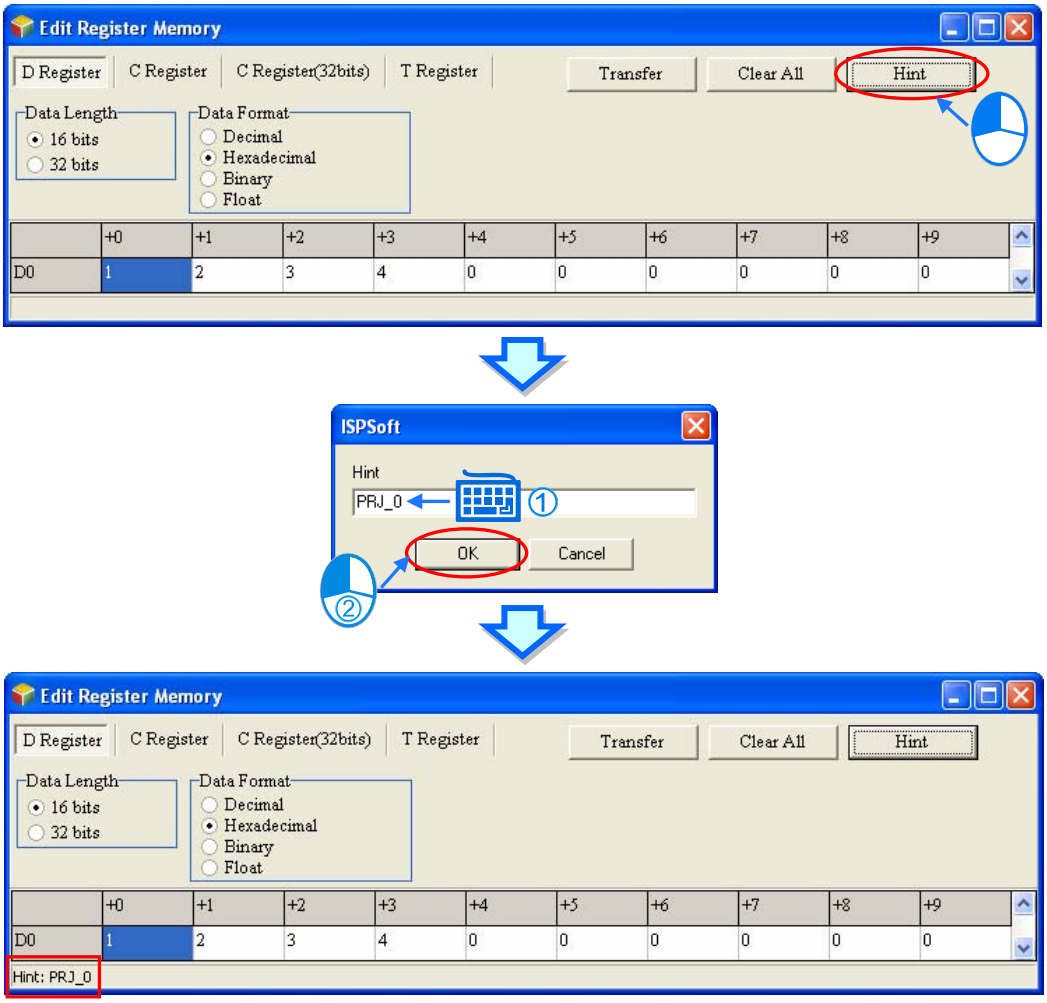
The users can export the values on the present page as a CSV file, and edit the CSV file through Microsoft Excel. If the users want to export the values on the present page, they can right-click the present page, click **Export** on the context menu, select a folder in the **Save in** drop-down list box in the **Save As** window, type a filename in the **File name** box, and click **Save**. The contents exported are the contents of the page selected.



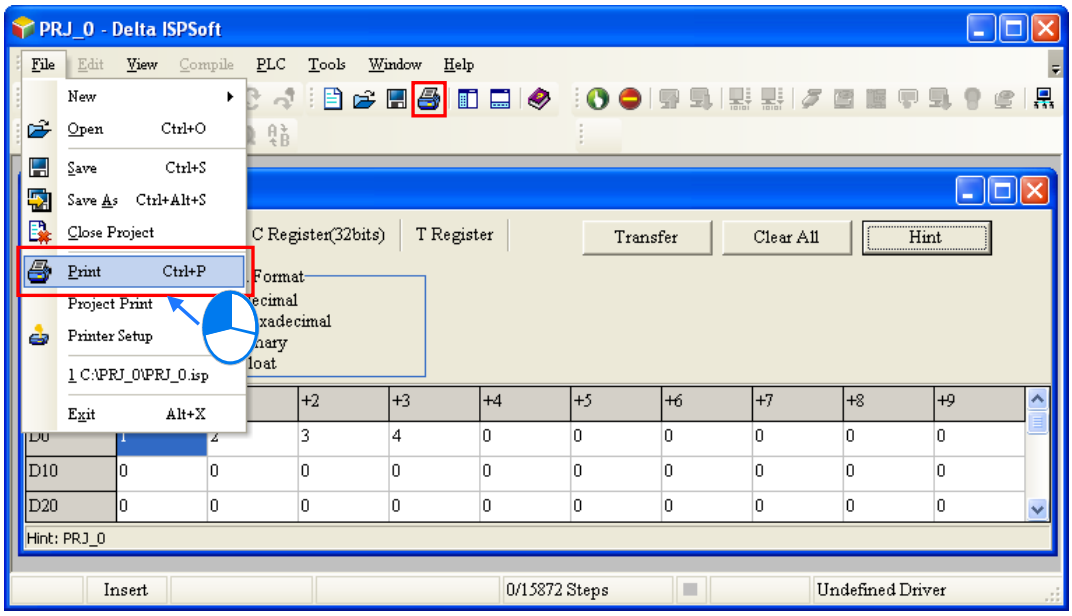
If the users want to import a CSV file into a page, they can right-click the page, click **Import** on the context menu, select the CSV file in the **Open** window, and click **Open**. The file imported must be matched with the device type clicked.



13 After the users click **Hint** in the **Edit Register Memory** window, the **ISPSoft** window will appear. After a hint is typed in the **ISPSoft** window, the hint will appear at the bottom of the **Edit Register Memory** window. The hint is saved with the values in the **Edit Register Memory** window. After the users open a dvl file, they can identify the tables in the file through the hint in the file.

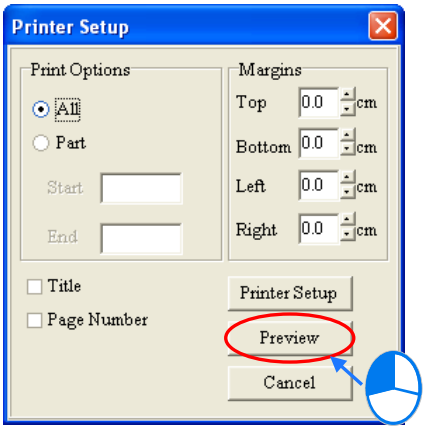


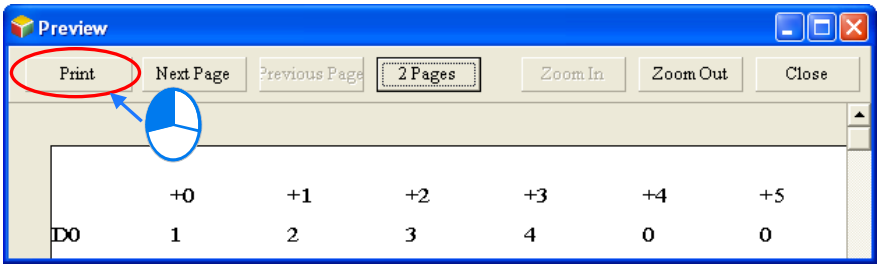
If the present window is the **Edit Register Memory** window, the users can print the present page in the **Edit Register Memory** window by clicking **Print** on the **File** menu or on the toolbar.



13

Select the **All** option button or the **Part** option button in the **Print Options** section, set margins, and select items which will be added. After the users click **Printer Setup**, they can select a printer, and set a print format. After the users click **Preview**, they can preview the document which will be printed. After the setting is complete, the users can click **Print** in the **Preview** window.

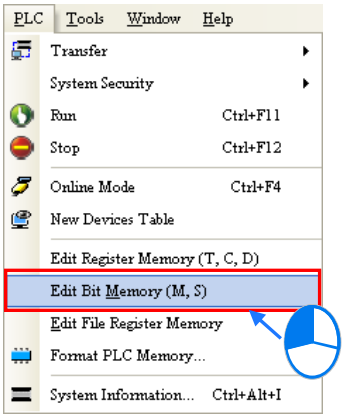




13.2.5 Editing the States of the M/S Devices in a DVP Series PLC

Users can edit the states of several M/S devices in a DVP series PLC at a time. The new states of these M/S devices can be saved and downloaded.

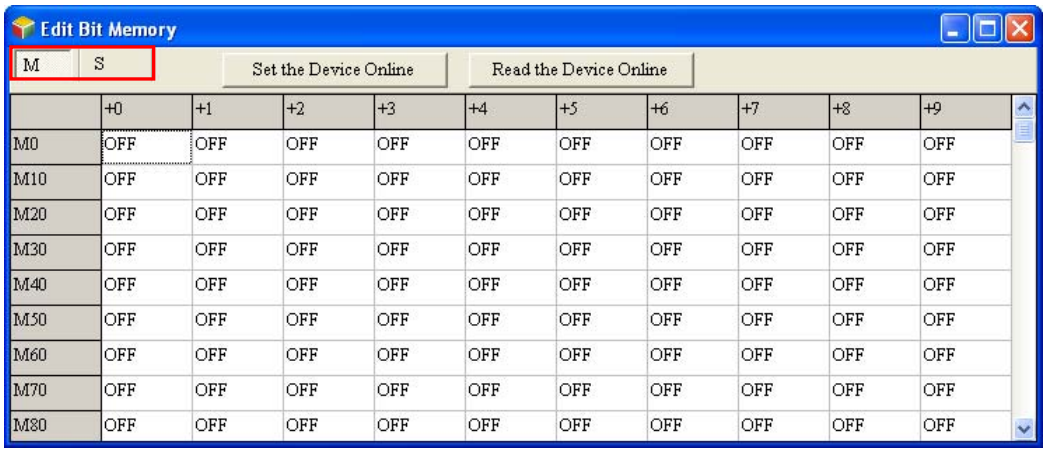
After users create a project for a DVP series PLC, they can click **Edit Bit Memory (M, S)** on the **PLC** menu to open the **Edit Bit Memory** window.



13

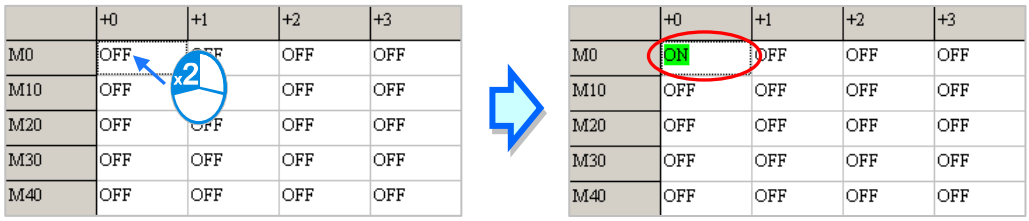
The states in the tables in the **Edit Bit Memory** window are not the states uploaded form the PLC. They are the states saved last time. If the **Edit Bit Memory** window in the project is opened for the first time, the default states in the tables are OFF.

The tabs at the top of the **Edit Bit Memory** window indicate device types.



	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9
M0	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF
M10	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF
M20	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF
M30	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF
M40	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF
M50	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF
M60	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF
M70	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF
M80	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF

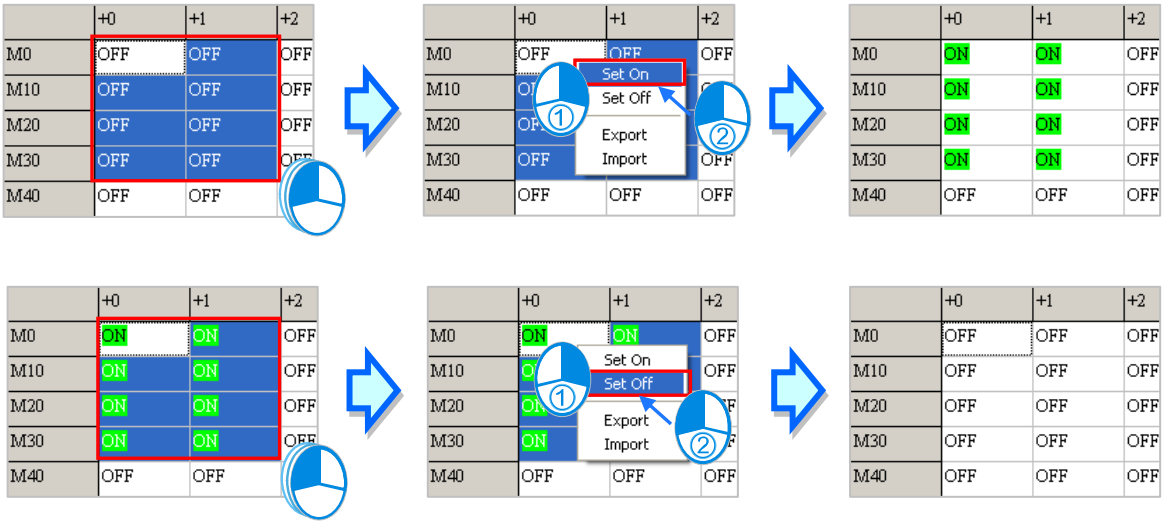
If the users want to change the state in the cell for a device, they can double-click the cell.



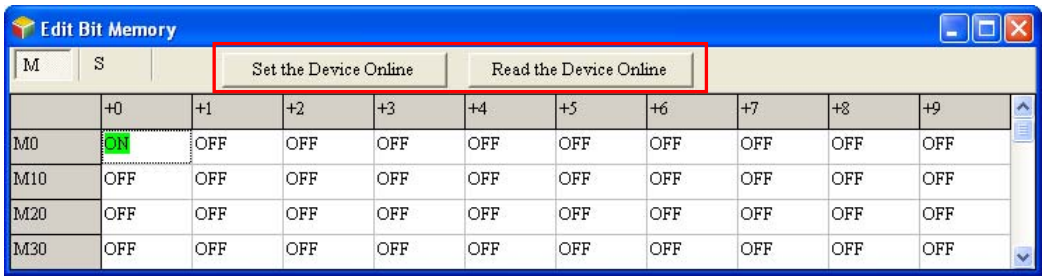
	+0	+1	+2	+3
M0	OFF	OFF	OFF	OFF
M10	OFF	OFF	OFF	OFF
M20	OFF	OFF	OFF	OFF
M30	OFF	OFF	OFF	OFF
M40	OFF	OFF	OFF	OFF

	+0	+1	+2	+3
M0	ON	OFF	OFF	OFF
M10	OFF	OFF	OFF	OFF
M20	OFF	OFF	OFF	OFF
M30	OFF	OFF	OFF	OFF
M40	OFF	OFF	OFF	OFF

If the users want to set the states in the cells for a range of devices on the present page simultaneously, they can drag across the cells, right-click the present page, and click **Set On** or **Set Off** on the context menu.

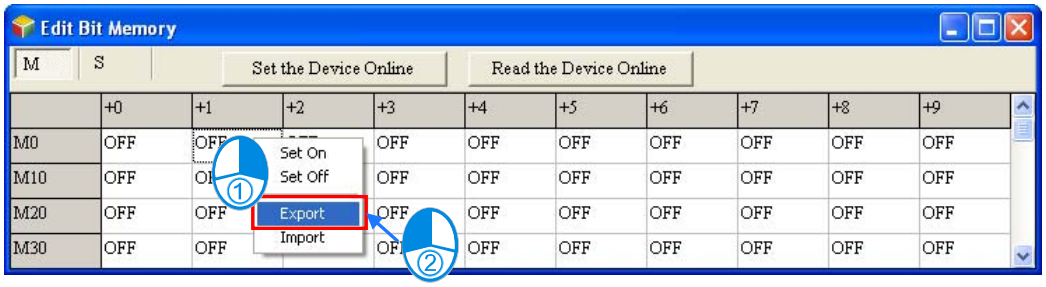


If the users click **Set the Device Online**, the states in the **Edit Bit Memory** window will be written into the PLC. If the users click **Read the Device Online**, the states of the M devices, and the states of the S devices will be read from the PLC. Before **Set the Device Online**/**Read the Device Online** is clicked, the users have to make sure that ISPSoft is connected to the PLC normally. Please refer to section 2.4 for more information.

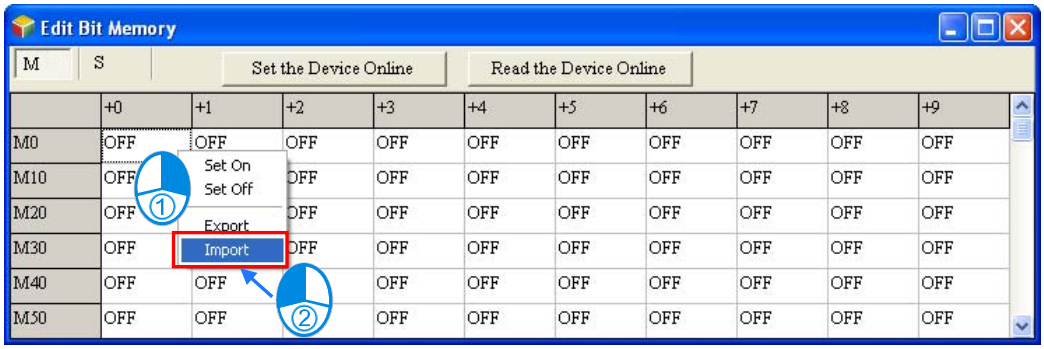


Before the states are written into the PLC, users have to make sure that the operation does not affect the system, or cause damage to the system and staff.

The users can export the states on the present page as a CSV file, and edit the CSV file through Microsoft Excel. If the users want to export the states on the present page, they can right-click the present page, click **Export** on the context menu, select a folder in the **Save in** drop-down list box in the **Save As** window, type a filename in the **File name** box, and click **Save**. The contents exported are the contents of the page selected.

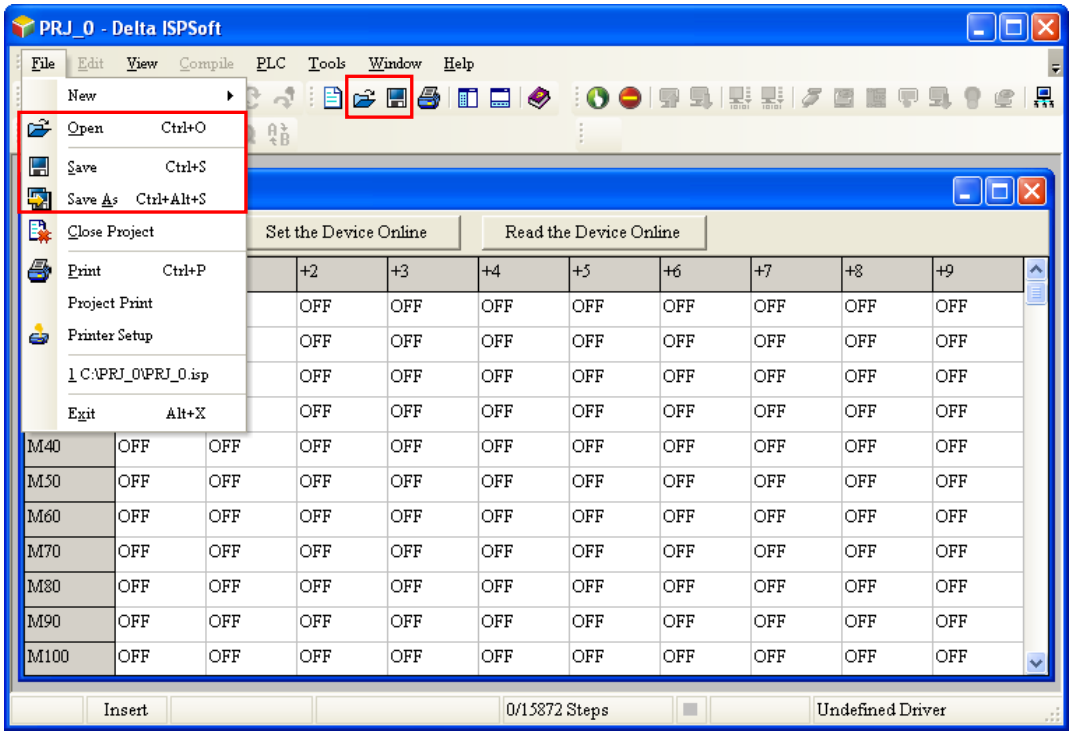


If the users want to import a CSV file into a page, they can right-click the page, click **Import** on the context menu, select the CSV file in the **Open** window, and click **Open**. The file imported must be matched with the device type clicked.

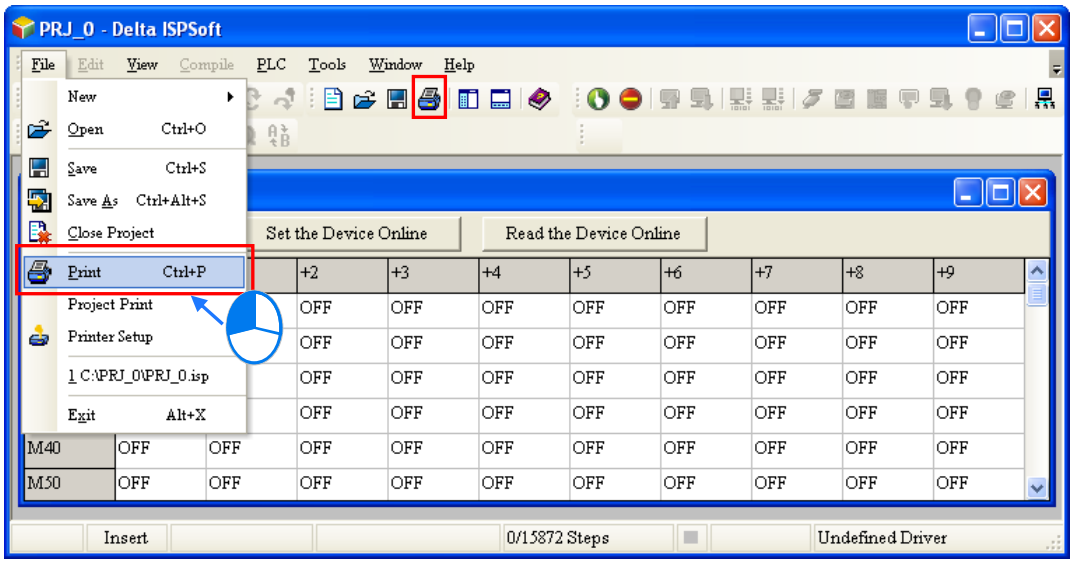


The states in the **Edit Bit Memory** window can be saved with the project. If the present window is the **Edit Bit Memory** window, the states in the **Edit Bit Memory** window will be saved as a dvb file whose primary filename is the project name in the folder in which the project (*.isp) is saved after the users click **Save** on the **File** menu or the toolbar. If the users want to open the **Edit bit Memory** window in the project again, the system will open the dvb file in the folder in which the project (*.isp) is saved. If no dvb is in the folder in which the project is saved, the states of the devices in the **Edit Bit Memory** window will be OFF

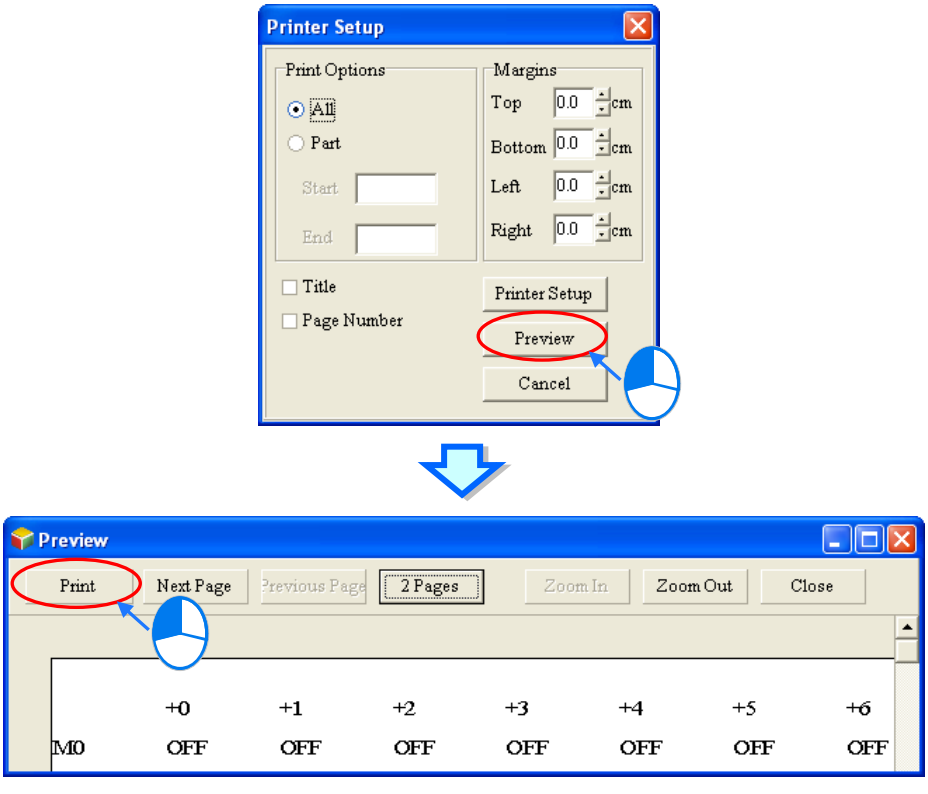
Besides, the users can save the states in the **Edit Bit Memory** window as a dvb file in another folder after they click **Save As** on the **File** menu. If the users want to open the dvb file which was saved previously, they can click **Open** on the **File** menu or the toolbar.



If the present window is the **Edit Bit Memory** window, the users can print the present page in the **Edit Bit Memory** window by clicking **Print** on the **File** menu or on the toolbar.



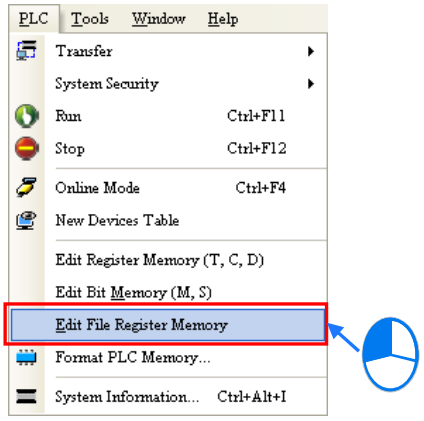
Select the **All** option button or the **Part** option button in the **Print Options** section, set margins, and select items which will be added. After the users click **Printer Setup**, they can select a printer, and set a print format. After the users click **Preview**, they can preview the document which will be printed. After the setting is complete, the users can click **Print** in the **Preview** window.



13.2.6 Editing the Values in the File Registers in a DVP Series PLC

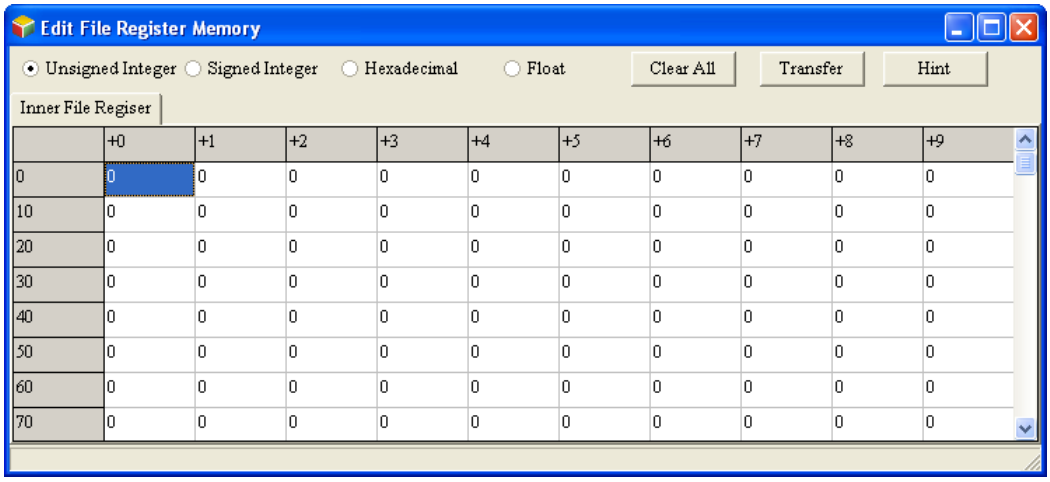
Users can edit the values in several file registers in a DVP series PLC at a time. The new values in these file registers can be saved and downloaded.

After users create a project for a DVP series PLC, they can click **Edit File Register Memory** on the **PLC** menu to open the **Edit File Register Memory** window.



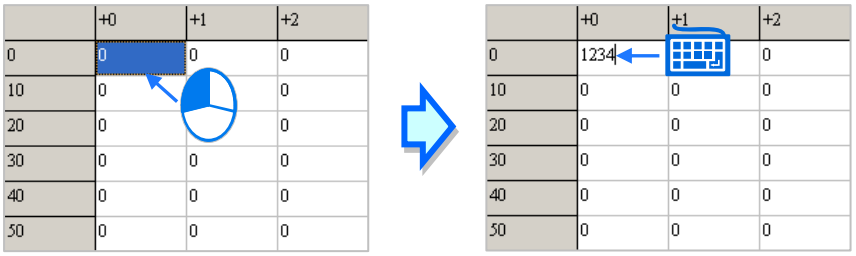
13

The values in the table in the **Edit File Register Memory** window are not the values uploaded form the PLC. They are the values saved last time. If the **Edit File Register Memory** window in the project is opened for the first time, the default values in the table are 0.



	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9
0	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0
20	0	0	0	0	0	0	0	0	0	0
30	0	0	0	0	0	0	0	0	0	0
40	0	0	0	0	0	0	0	0	0	0
50	0	0	0	0	0	0	0	0	0	0
60	0	0	0	0	0	0	0	0	0	0
70	0	0	0	0	0	0	0	0	0	0

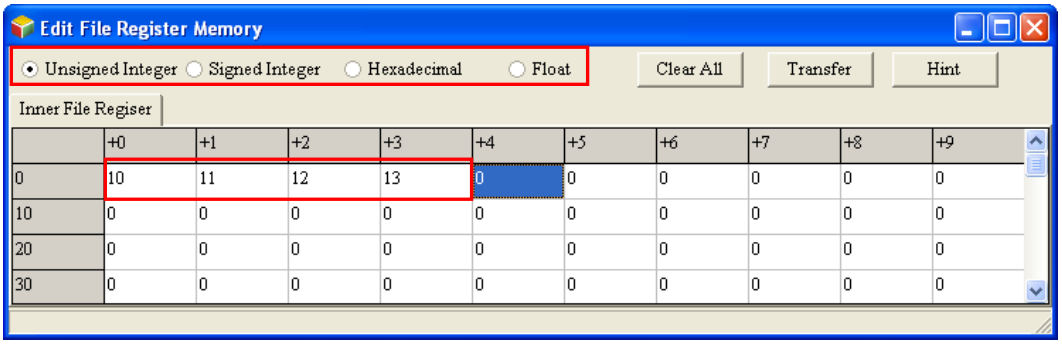
If the users want to edit the value in the cell for a file register, they can click the cell for the file register, and type a value.



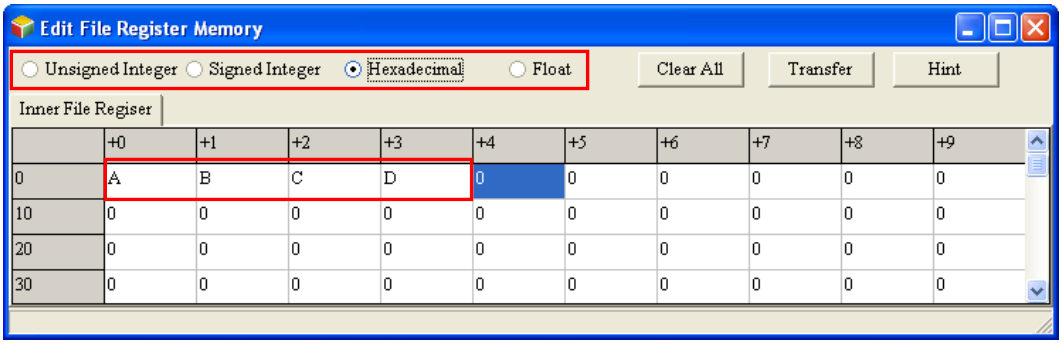
	+0	+1	+2
0	0	0	0
10	0	0	0
20	0	0	0
30	0	0	0
40	0	0	0
50	0	0	0

	+0	+1	+2
0	1234	0	0
10	0	0	0
20	0	0	0
30	0	0	0
40	0	0	0
50	0	0	0

The users can select a data format at the top of the **Edit File Register Memory** window.

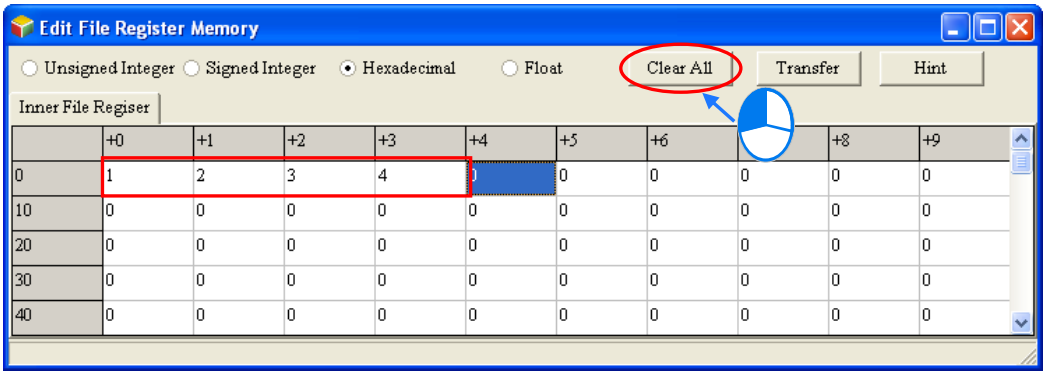


	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9
0	10	11	12	13	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0
20	0	0	0	0	0	0	0	0	0	0
30	0	0	0	0	0	0	0	0	0	0

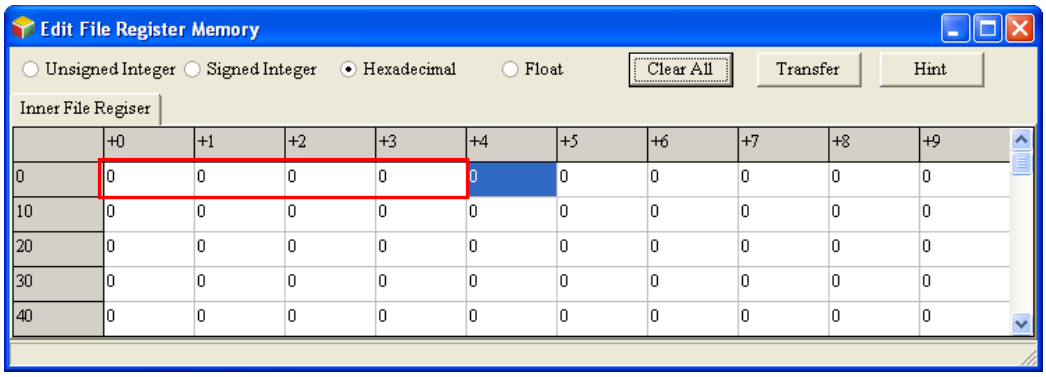


	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9
0	A	B	C	D	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0
20	0	0	0	0	0	0	0	0	0	0
30	0	0	0	0	0	0	0	0	0	0

If the users click **Clear All**, the values in the **Edit File Register Memory** window will be cleared to 0.



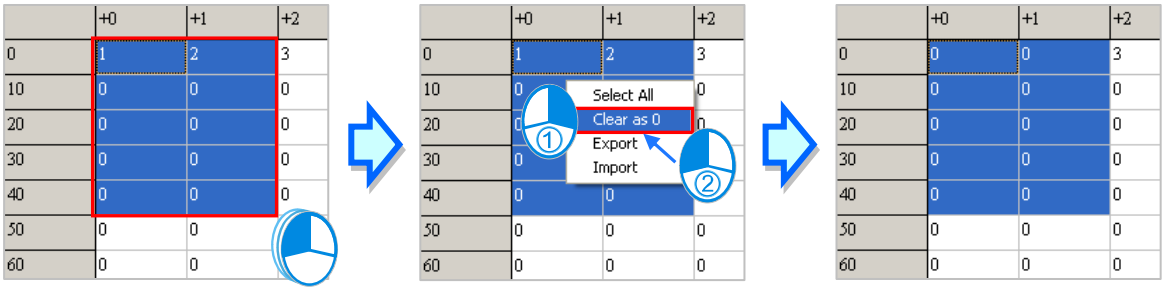
	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9
0	1	2	3	4	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0
20	0	0	0	0	0	0	0	0	0	0
30	0	0	0	0	0	0	0	0	0	0
40	0	0	0	0	0	0	0	0	0	0



	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9
0	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0
20	0	0	0	0	0	0	0	0	0	0
30	0	0	0	0	0	0	0	0	0	0
40	0	0	0	0	0	0	0	0	0	0

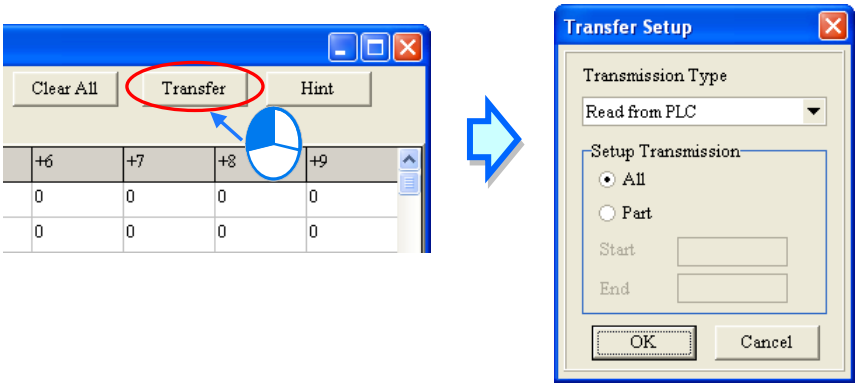
Besides, the users can clear the values on the present page by right-clicking the present page, clicking **Select All** on the context menu, right-clicking the present page again, clicking **Clear as 0** on the context menu.

If the users want to clear the values in the cells for a range of file registers on the present page, they can drag across the cells, right-click the present page, and click **Clear as 0** on the context menu.



After the users click **Transfer**, the **Transfer Setup** window will appear. The users can download the values in the table to the PLC, or upload the values in the file registers from the PLC through the window. Before **Transfer** is clicked, the users have to make sure that ISPSoft is connected to the PLC normally. Please refer to section 2.4 for more information.

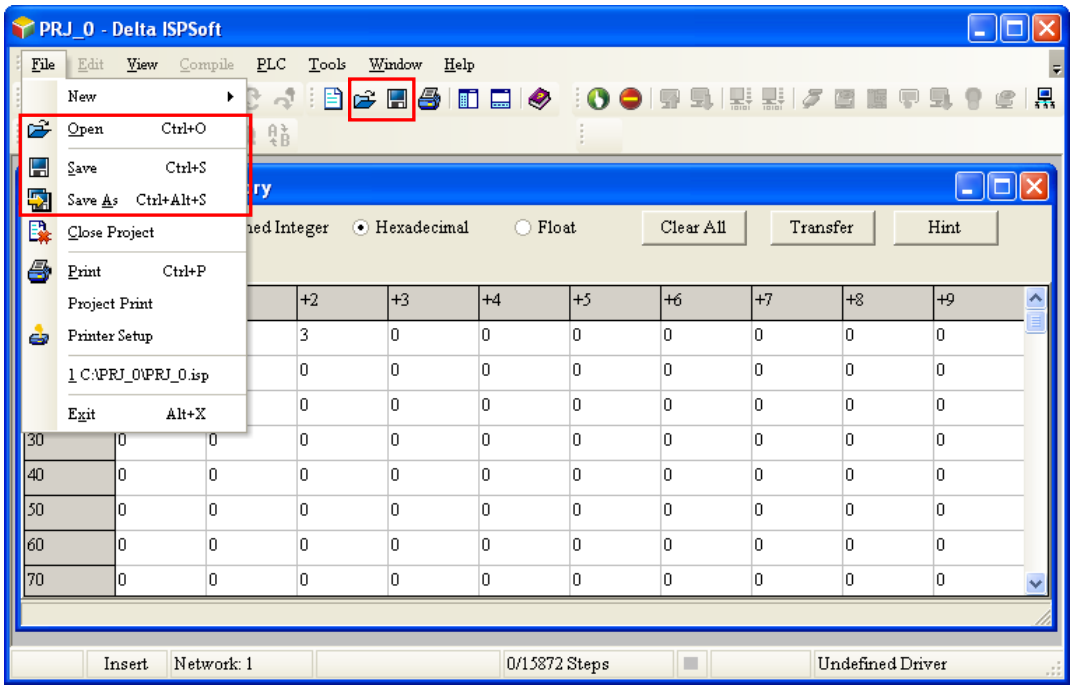
Select **Read form PLC** or **Write to PLC** in the **Transmission Type** drop-down list box, and then select the **All** option button or the **Part** option button. After the setting is complete, the users can click **OK**.



Before the values are written into the registers, users have to make sure that the operation does not affect the system, or cause damage to the system and staff.

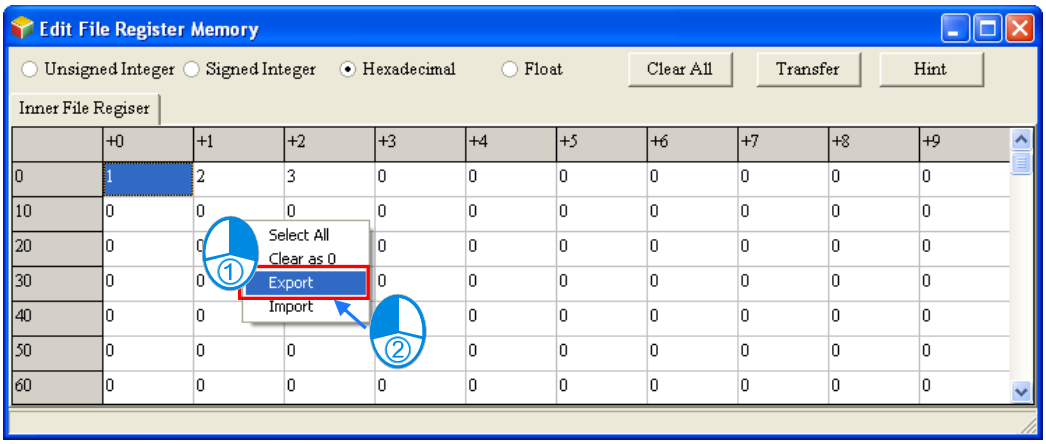
The values in the **Edit File Register Memory** window can be saved with the project. If the present window is the **Edit File Register Memory** window, the values in the **Edit Register Memory** window will be saved as a wft file whose primary filename is the project name in the folder in which the project (*.isp) is saved after the users click **Save** on the **File** menu or the toolbar. If the users want to open the **Edit File Register Memory** window in the project again, the system will open the wft file in the folder in which the project (*.isp) is saved. If no wft is in the folder in which the project is saved, the values in the **Edit File Register Memory** window will be 0.

Besides, the users can save the values in the **Edit File Register Memory** window as a wft file in another folder after they click **Save As** on the **File** menu. If the users want to open the wft file which was saved previously, they can click **Open** on the **File** menu or the toolbar.

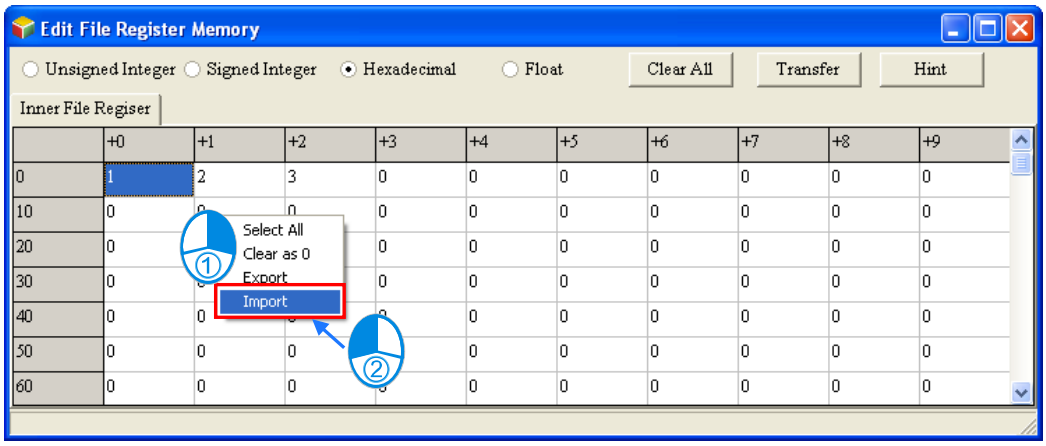


13

The users can export the values on the present page as a CSV file, and edit the CSV file through Microsoft Excel. If the users want to export the values on the present page, they can right-click the present page, click **Export** on the context menu, select a folder in the **Save in** drop-down list box in the **Save As** window, type a filename in the **File name** box, and click **Save**.

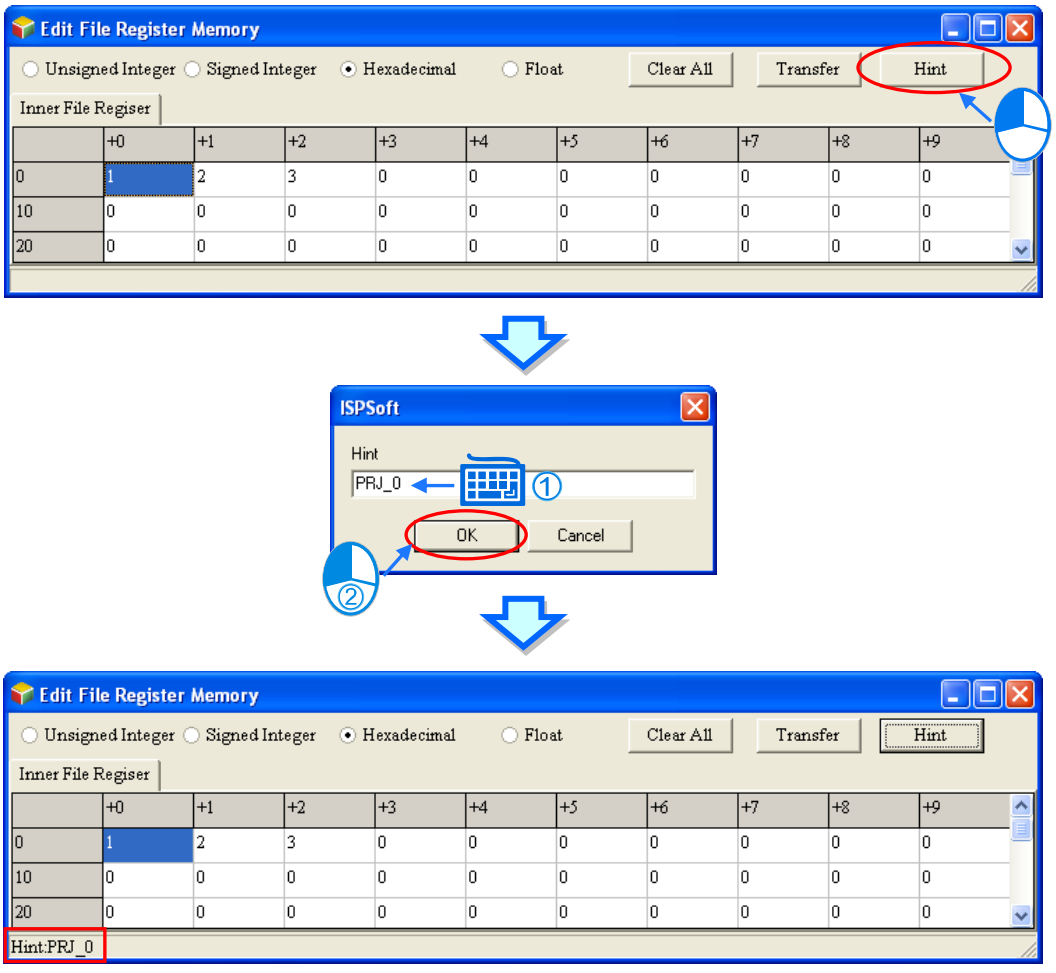


If the users want to import a CSV file into the page, they can right-click the page, click **Import** on the context menu, select the CSV file in the **Open** window, and click **Open**.

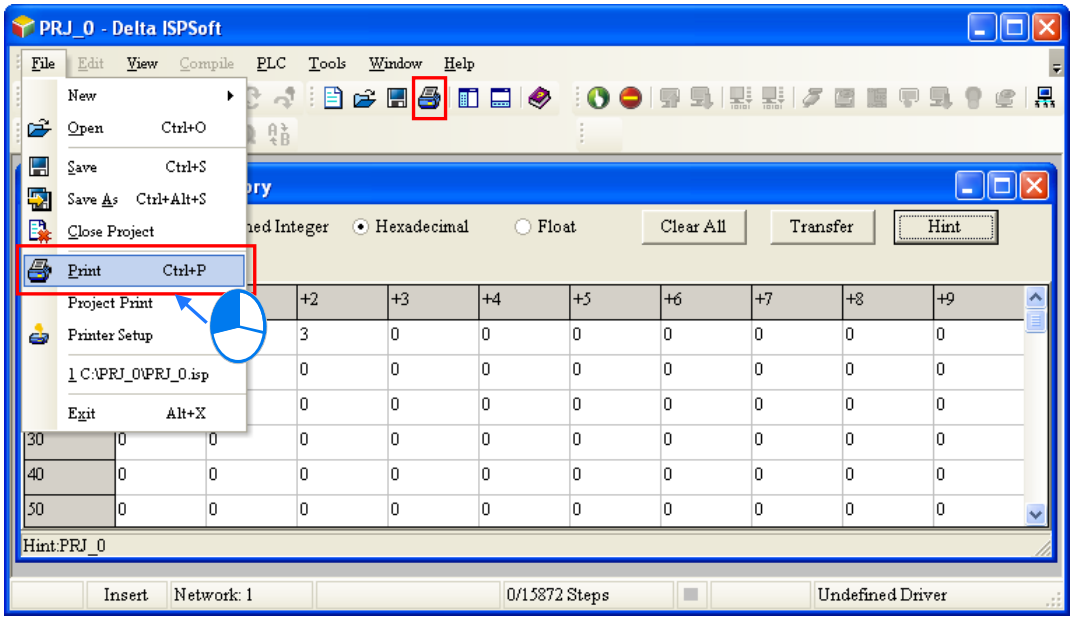


13

After the users click **Hint** in the **Edit File Register Memory** window, the **ISPSoft** window will appear. After a hint is typed in the **ISPSoft** window, the hint will appear at the bottom of the **Edit File Register Memory** window. The hint is saved with the values in the **Edit File Register Memory** window. After the users open a wft file, they can identify the table in the file through the hint in the file.

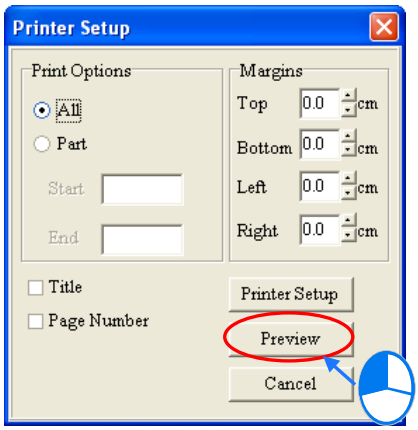


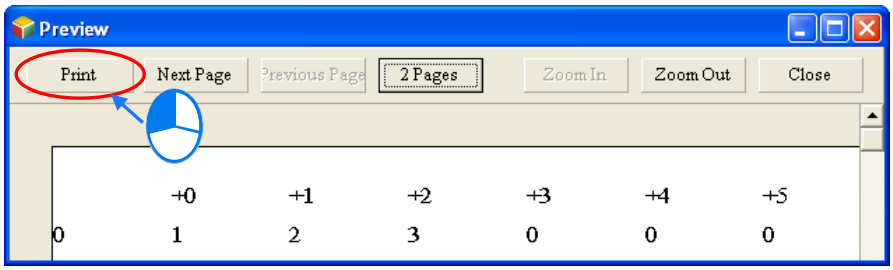
If the present window is the **Edit File Register Memory** window, the users can print the present page in the **Edit File Register Memory** window by clicking **Print** on the **File** menu or on the toolbar.



13

Select the **All** option button or the **Part** option button in the **Print Options** section, set margins, and select items which will be added. After the users click **Printer Setup**, they can select a printer, and set a print format. After the users click **Preview**, they can preview the document which will be printed. After the setting is complete, the users can click **Print** in the **Preview** window.





MEMO

Chapter 14 Test Tools and Debugging Tools

14

Table of Contents

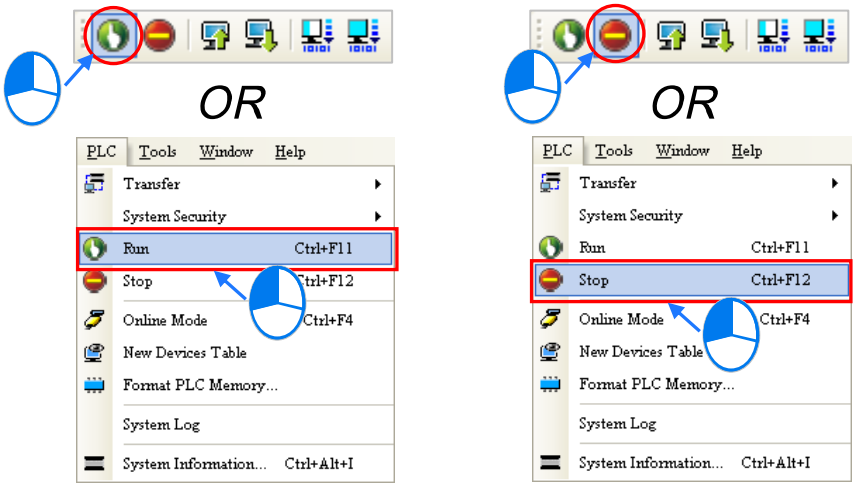
14.1 Online Monitoring Function 14-2
14.1.1 Changing the Operating Status of a PLC 14-2
14.1.2 Online Monitoring Function and Environment 14-2
14.1.3 Changing the States of the X Devices in the Online Mode 14-6
14.1.4 Monitoring a Program Online 14-7
14.1.5 Device Monitoring Table 14-11
14.1.6 Online Editing function and Online Update Function 14-17
14.2 DVP Series Simulator 14-18
14.2.1 Creating a DVP Simulator 14-18
14.2.2 Starting the DVP Simulator 14-19
14.3 Debugging Mode for a DVP Series PLC 14-20
14.3.1 Enabling the Debugging Mode for a DVP Series PLC 14-20
14.3.2 Adding and Clearing Breakpoints 14-21
14.3.3 Execution of the Program in the Debugging Mode 14-22
14.4 AH500 Series Simulator 14-23
14.4.1 Creating and Starting an AH Simulator 14-23
14.4.2 Operation Panel of an AH Simulator 14-25
14.5 Debugging Mode for an AH500 Series CPU Module 14-30
14.5.1 Enabling the Debugging Mode for an AH500 Series CPU Module 14-30
14.5.2 Adding and Clearing Breakpoints 14-31
14.5.3 Continuous Execution 14-32
14.5.4 Single-step Execution 14-32
14.6 Checking the Status of a PLC 14-33
14.6.1 System Information 14-33
14.6.2 System Log (Only for AH500 Series CPU Modules) 14-34

14.1 Online Monitoring Function

14.1.1 Changing the Operating Status of a PLC

Users need to constantly change the operating status of a PLC when they test and debug the program in the PLC. They can easily change the status of the PLC by means of ISPSoft. Before the operation status of a PLC is changed by means of ISPSoft, users have to make sure that ISPSoft is connected to the PLC normally. Please refer to section 2.4 for more information.

After users click **Run** on the **PLC** menu or on the toolbar, the PLC will begin to run. The PLC will stop running after **Stop** on the **PLC** menu or on the toolbar is clicked.



If the operating status of a PLC is changed by means of ISPSoft, users do not have to consider the state of the RUN/STOP switch on the PLC. After the operating status of a PLC is changed by means of ISPSoft, users still can change the operating status of the PLC again by means of the RUN/STOP switch on the PLC.

14.1.2 Online Monitoring Function and Environment

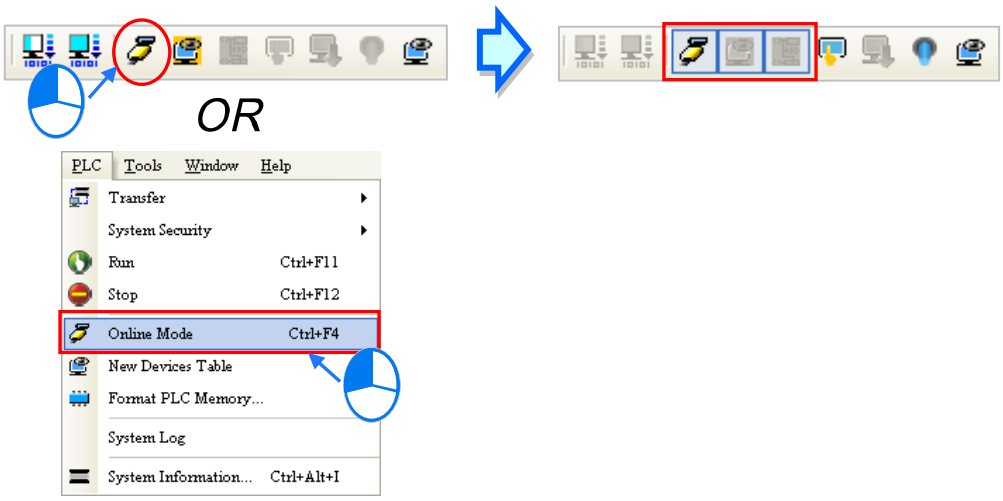
If ISPSoft is connected to the PLC normally, users can monitor the execution status of the PLC through the online monitoring modes. Please refer to section 2.4 for more information about setting the connection between ISPSoft and a PLC.

In ISPSoft, there are two monitoring modes. One is the device monitoring mode, and the other is the program monitoring mode.

Monitoring mode	Description
Device monitoring mode	Users can monitor the statuses of the devices in the PLC through the monitoring table. In this mode, ISPSoft only needs to update the statuses of the devices. The present program in ISPSoft does not need to be the same as the program in the PLC.
Program monitoring mode	In this mode, the operating status of the program is displayed in the program editing window. As a result, the present program in ISPSoft must be the same as the program in the PLC.

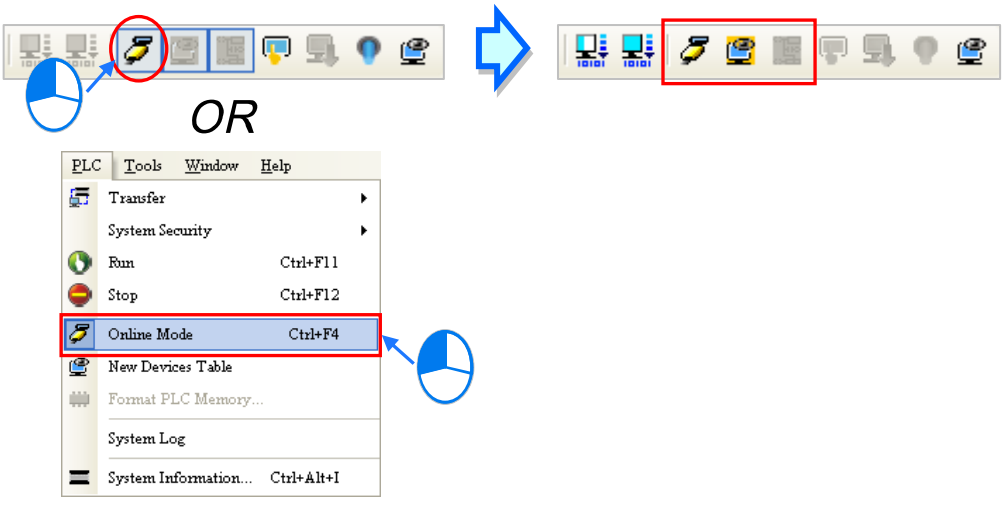
***. The device monitoring function can be enabled independently. However, if the program monitoring function is enabled, the device monitoring function is also enabled.**

After users click **Online Mode** on the **PLC** menu, or on the toolbar, the online monitoring function will be enabled. The system will also enable the device monitoring mode and the program monitoring mode.

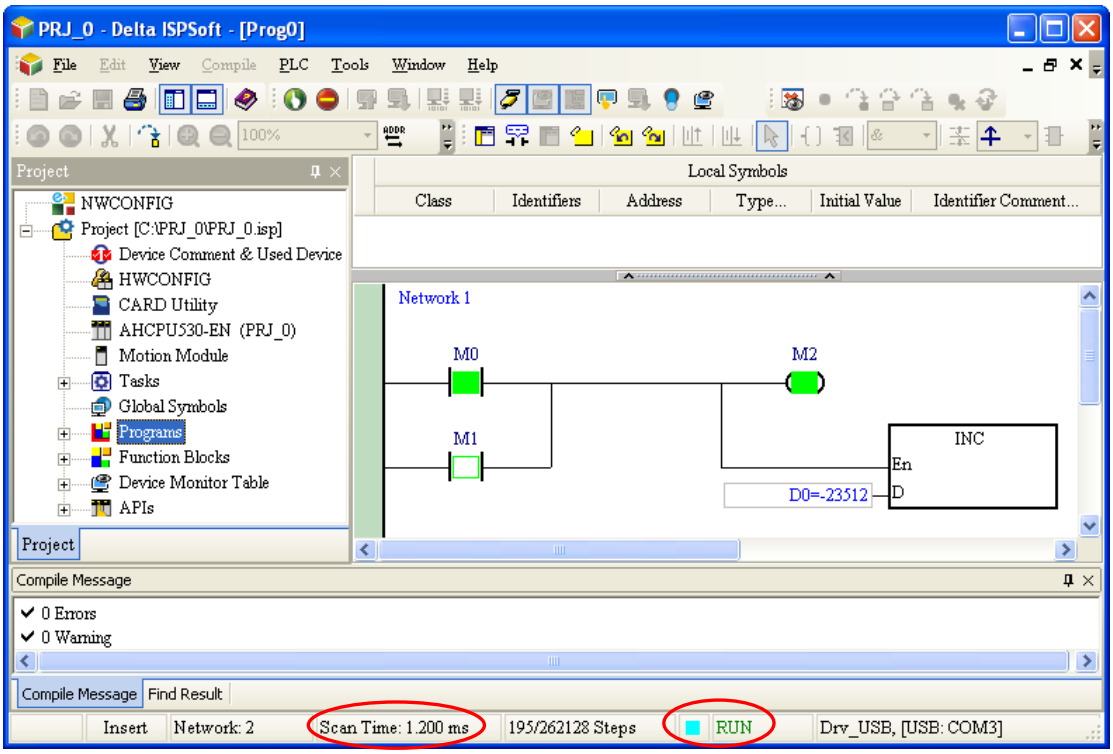


If the users only want to monitor the statuses of the devices in the PLC, they can click on the toolbar. When on the toolbar is clicked, on the toolbar is also clicked. If the users want to monitor the program, they can click on the toolbar.

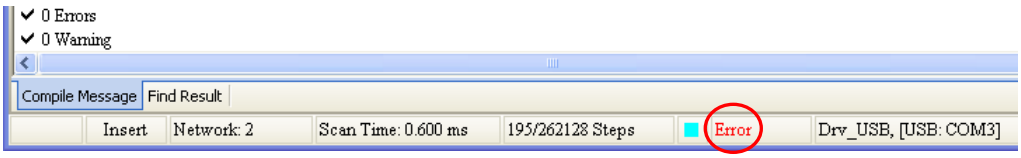
If the users want to disable the online monitoring function, they can click **Online Mode** on the **PLC** menu, or on the toolbar.



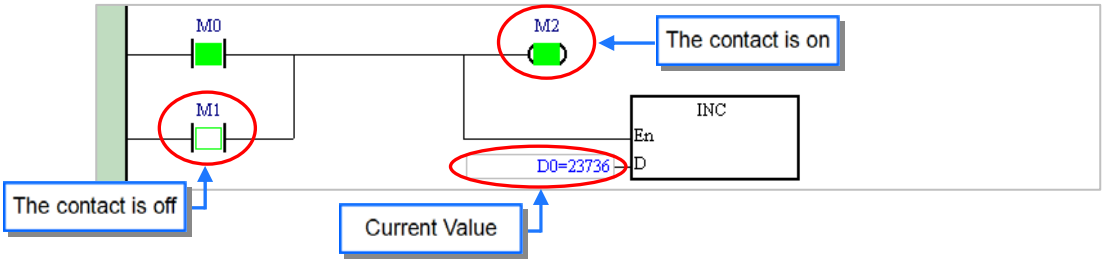
In the online monitoring mode, users can view the present scan time and the status of the PLC in the status bar in ISPSoft. The light at the left side of the status of the PLC indicates the communication status. When ISPSoft communicates with the PLC, the light blinks.



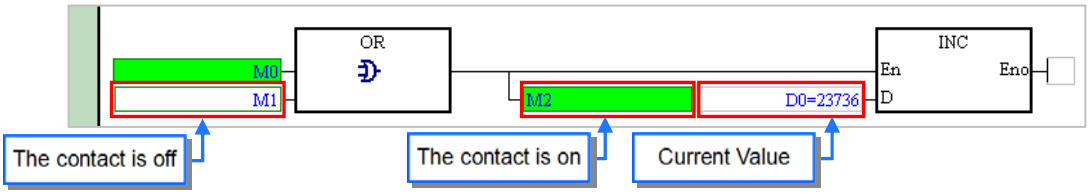
If an error occurs in the PLC, users may not be able to operate the PLC, e.g. the PLC can not run. Please refer to the manual for the model used for more information about troubleshooting.



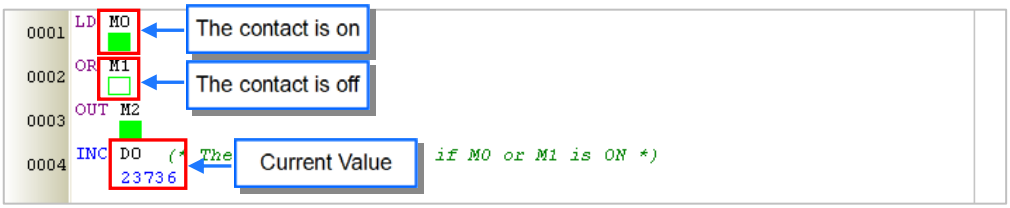
The present statuses of the devices will be displayed in the program editing window after the program monitoring function is enabled. Users can debug and test the program in the program monitoring mode. The way in which the statuses of the devices in a program are displayed depends on the programming language used to create the program.
A ladder diagram in the program monitoring mode is shown below.



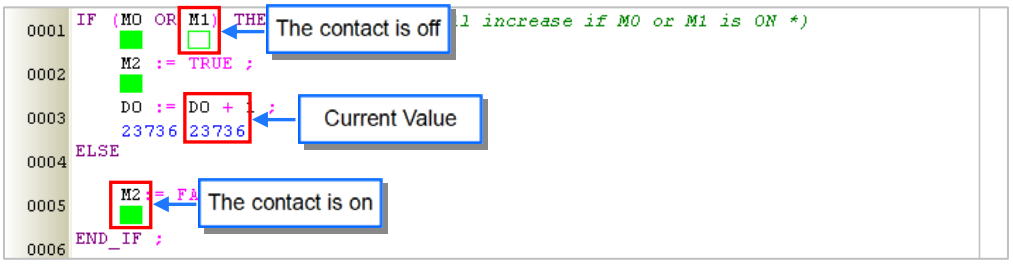
A function block diagram in the program monitoring mode is shown below.



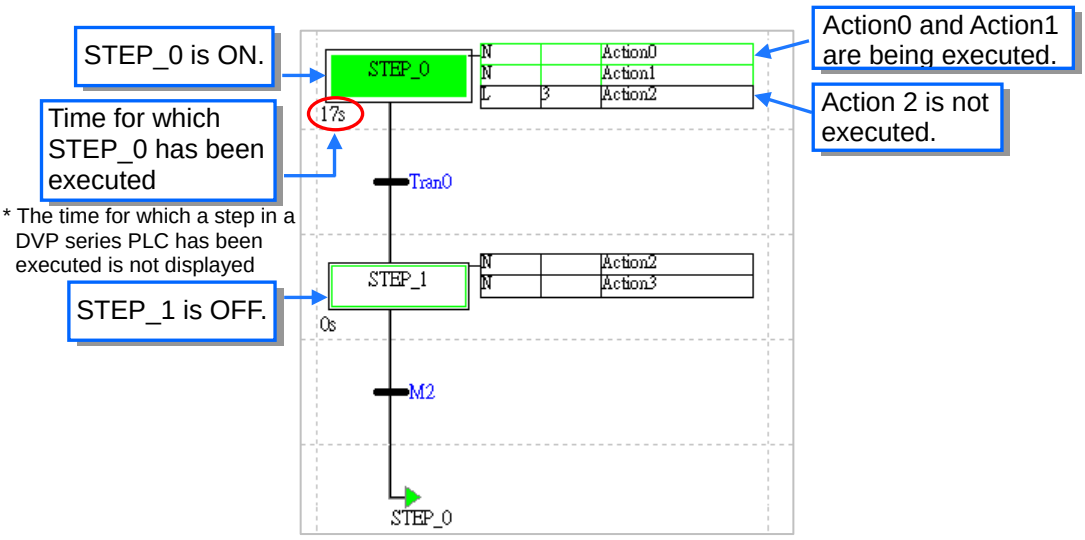
An instruction list in the program monitoring mode is shown below.



A structured text in the program monitoring mode is shown below.

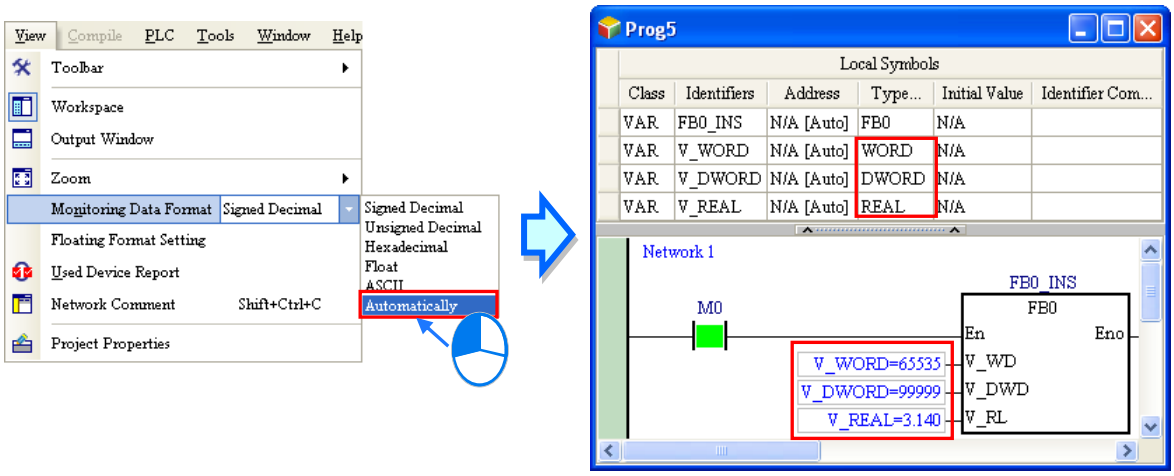


A sequential function chart in the program monitoring mode is shown below. Not only the sequential function chart itself is monitored, but also the actions and the transitions are monitored.

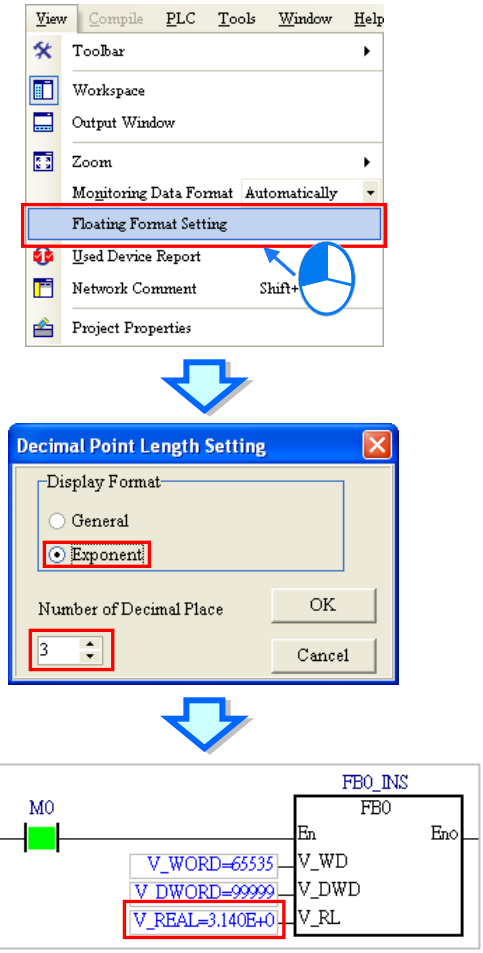


In the program monitoring mode, users can monitor not only the states of the contacts, but also the values in the registers. Besides, users can set the formats of the values monitored.

Click **Monitoring Data Format** on the **View** menu, and then select a data format on the drop-down list. If **Automatically** is selected, the formats of the values monitored depend on the data types of the symbols declared.

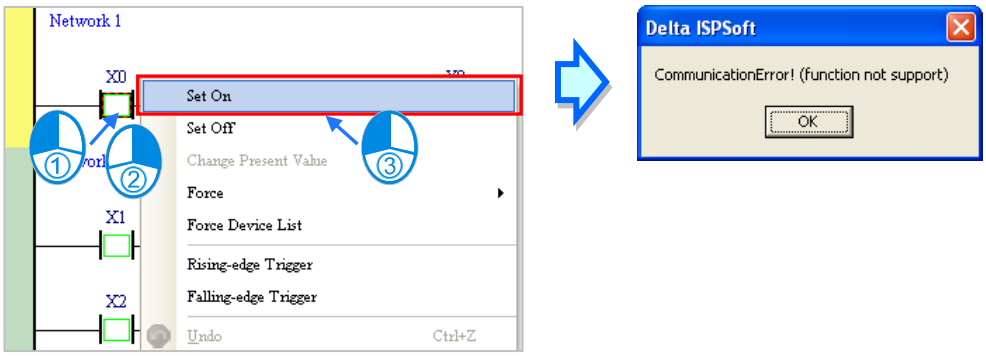


After users click **Floating Format Setting** on the **View** menu, they can set the formats of the floating-point numbers.

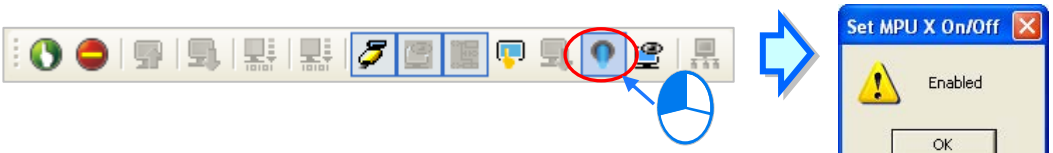


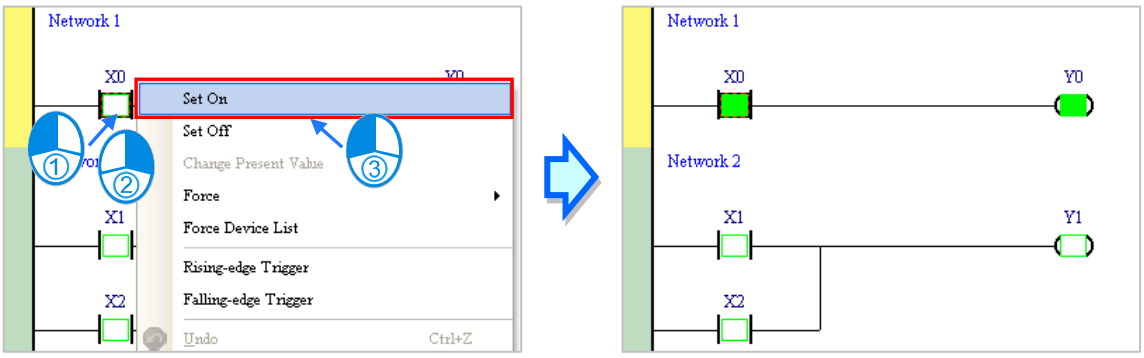
14.1.3 Changing the States of the X Devices in the Online Mode

In the online mode, users can change the states of the contacts in the program editing window. Users may not be allowed to change the states of the X devices in some PLCs in the online modes owing to the design of these PLCs. If users want to change the state of an X device, an error message will appear. Please refer to the manual for the model selected for more information.



If an X device needs to be operated when the program is tested, users can click on the toolbar to enable the function of setting the X device. However, the state of the X device will not be updated after the function is enabled.



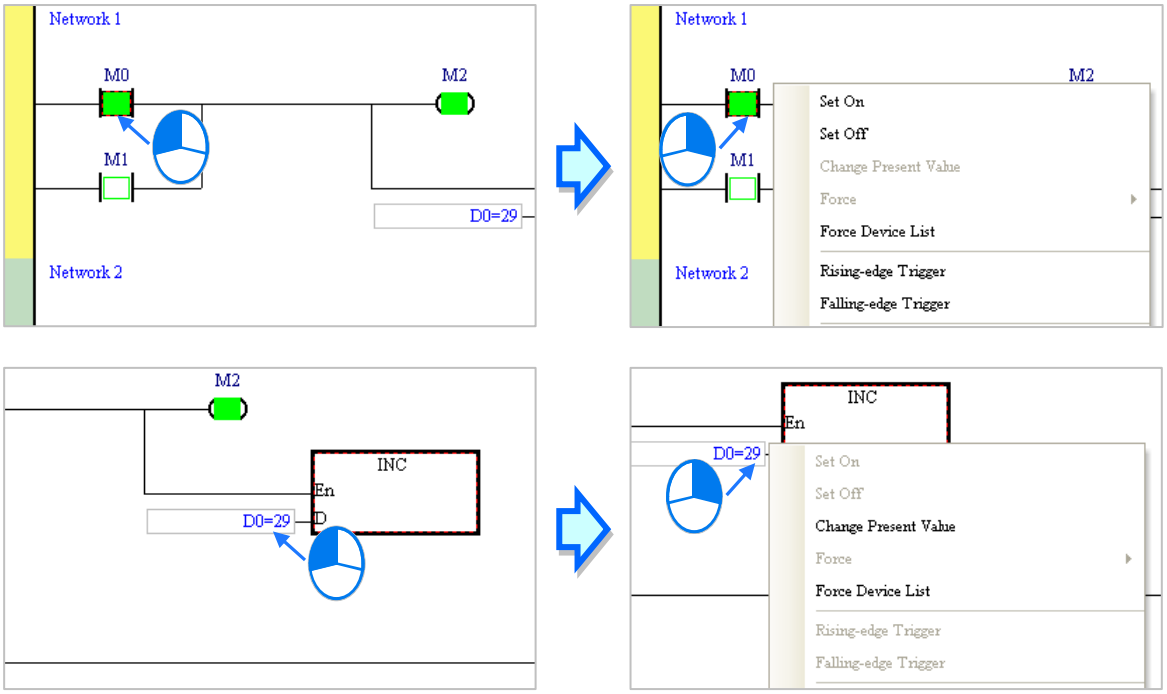


After the online monitoring function is disabled, the function of setting the X device will not be disabled automatically. As a result, be sure to disable the function of setting the X device before the online monitoring function is disabled lest the state of the X device should not be updated during the operation of the PLC.

Before the state of a contact is changed, users have to make sure that the operation performed does not affect the operation of the system, or cause damage to the system or the staff.

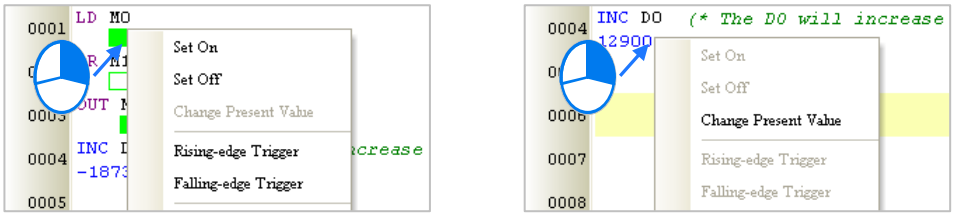
14.1.4 Monitoring a Program Online

In the program monitoring mode, users can change the state of a contact or the value in a register in the program editing window. By means of changing the state of a contact or the value in a register, users can debug or test the program. If users want to manipulate a device or an operand in a ladder diagram or a function block diagram, they can click the device or the operand, right-click the device or the operand, and select an item on the context menu. The items on the context menu vary with the programming language used.

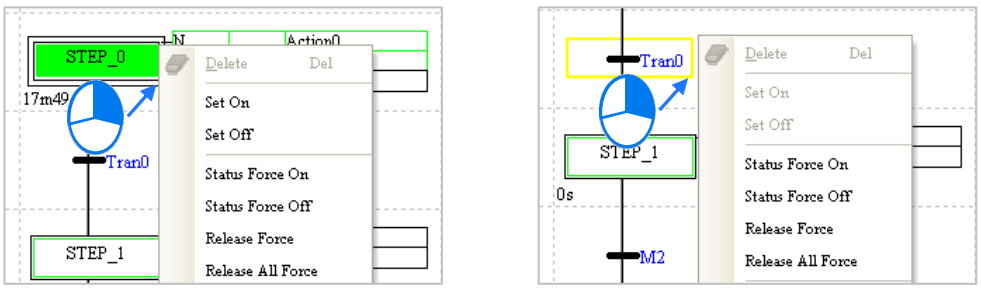


14

If users want to manipulate an operand on an instruction list or in a structured text, they can right-click the operand, and select an item on the context menu.



If users want to manipulate a step or a transition in a sequential function chart, they can right-click the step or the transition, and select an item on the context menu.



The items on the context menu are described below.

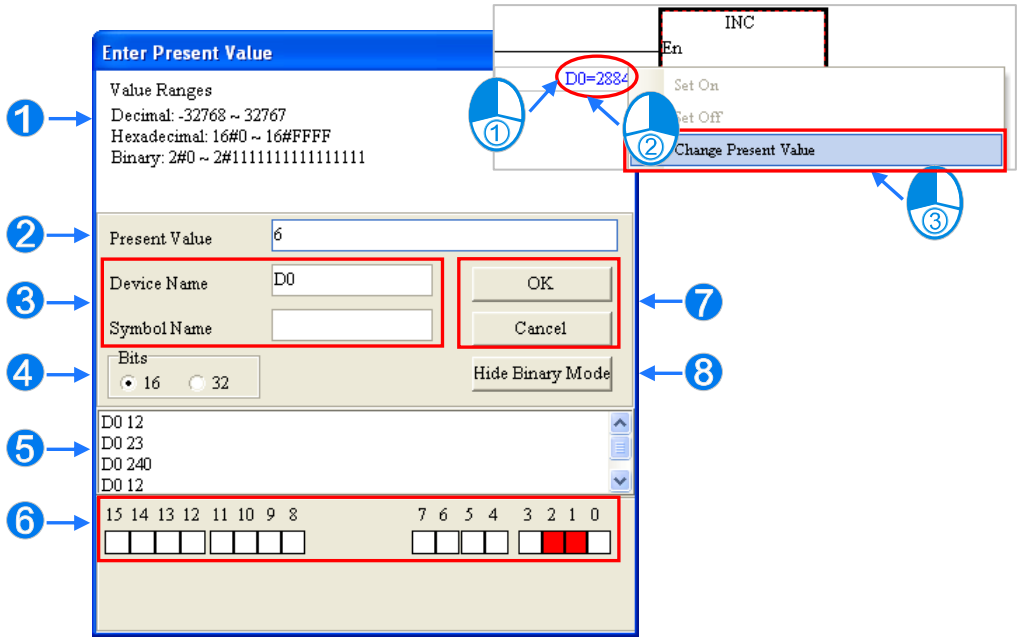
Item	Description
Set On	Setting the contact selected to ON
Set Off	Setting the contact selected to OFF
Rising-edge Trigger	No matter what the state of the contact selected is, the system set the contact to OFF, and then set it to ON.
Falling-edge Trigger	No matter what the state of the contact selected is, the system set the contact to ON, and then set it to OFF.

Item	Description
Change Present Value	Users can set the value of the operand selected.
Force	Forcing an input contact or output contact ON or OFF
Force Device List	Forcing several input contacts or output contacts in the tables ON or OFF

⚠ **Before the status of a device is changed, users have to make sure that the operation does not affect the operation of the system, or cause damage to the system or the staff.**

- **Change Present Value**

  First, select an operand, and right-click the operand. Next, select **Change Present Value** on the context menu. Finally, set the value of the operand in the **Enter Present Value** window.



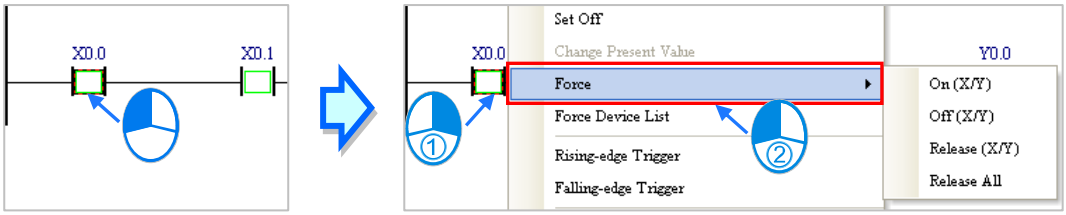
1. Message
2. Users can type a value in the **Present Value** box.
3. Name of a device or a symbol whose present value will be changed
4. Users can type a 16-bit value or a 32-bit value.
5. Value change history (Format: Device name Value)
6. In the binary mode, users can set the states of the bits through the mouse.
7. The setting values will be applied after **OK** is clicked. The window will be closed after **Cancel** is clicked.
8. Users can display or hide the binary mode.

- **Force**

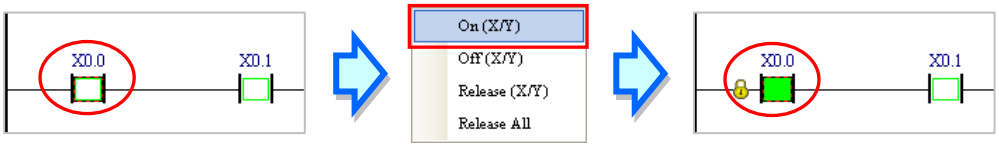
  Only X contact and Y contacts and be forced ON or OFF. If an X contact is forced ON or OFF, the state of the contact will not be affected by the actual input. Likewise, if a Y contact is forced ON or OFF, the state of the contact will be affected by the program execution result.

  Y0.0 in the figure below is forced ON. It will be ON even if M0 is OFF. Besides, if a contact is forced ON or OFF, a lock symbol will appear at the left side of the contact.

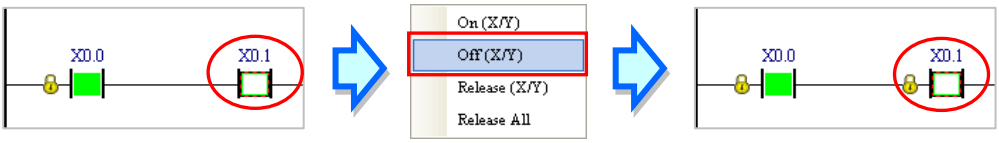
Click an X contact or a Y contact which will be manipulated, right-click the X contact or the Y contact, point to Force on the context menu, and select **On (X/Y)**, **Off (X/Y)**, **Release (X/Y)**, or **Release All**.



➢ **On (X/Y)**: Forcing the X contact or the Y contact selected ON

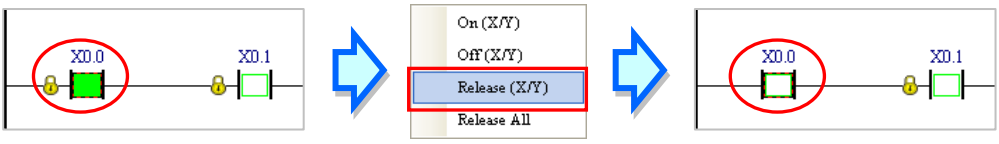


➢ **Off (X/Y)**: Forcing the X contact or the Y contact selected OFF

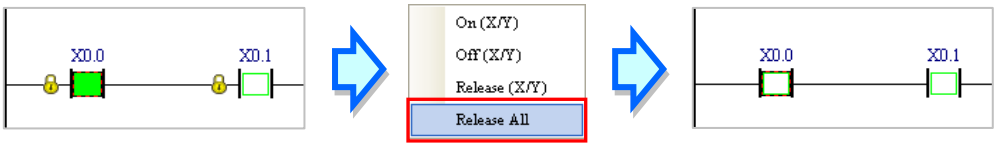


14

➢ **Release (X/Y)**: Releasing the contact from the locked state



➢ **Release All**: Releasing all the contacts from the locked states

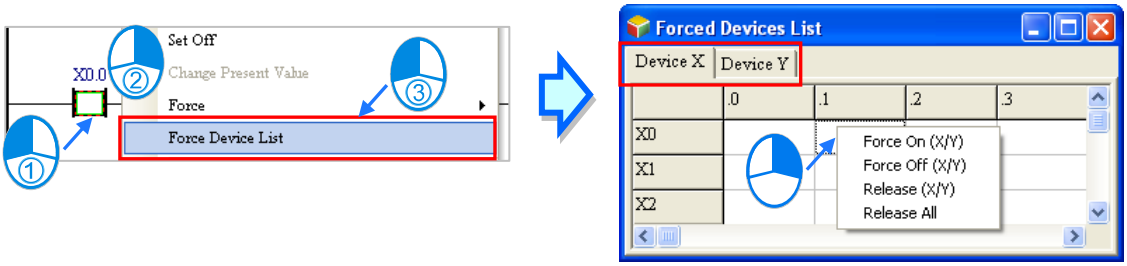


***1. The mechanism for forcing contacts ON/OFF, and the limit on the number of contacts which can be forced ON/OFF vary from model to model. Please refer to the manual for the model selected for more information.**

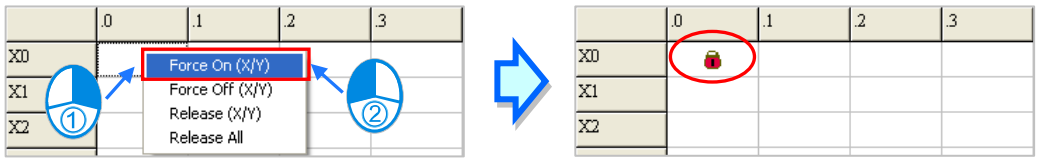
***2. If the online monitoring function is disabled, the contacts will not be automatically released from the locked states. Please check whether the present state will cause any problem.**

- **Force Device List**

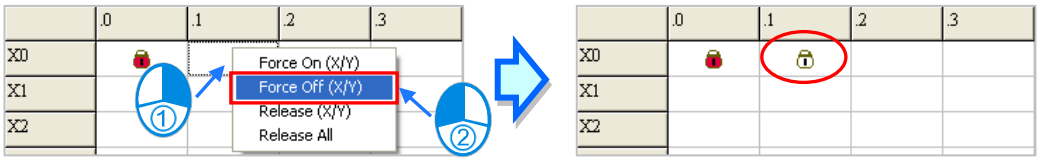
Right-click the program editing window, and then select **Force Device List** on the context menu. Click the **Device X** tab or the **Device Y** tab in the **Forced Devices List** window. Select a contact in the table, right click the contact, and select an item on the context menu.



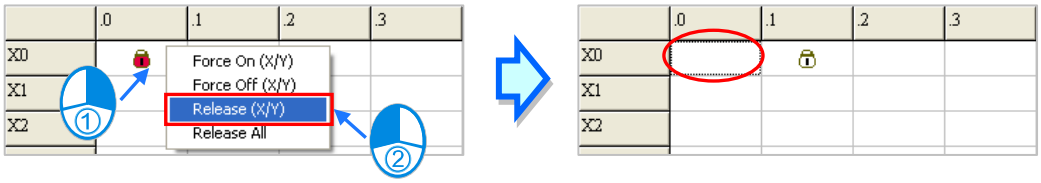
➢ **Force On (X/Y)**: Setting the contact selected to ON



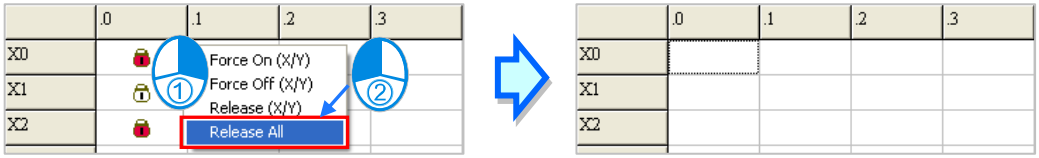
➢ **Force Off (X/Y)**: Setting the contact selected to OFF



➢ **Release (X/Y)**: Releasing the contact from the locked state



➢ **Release All**: Releasing all the contacts from the locked states



***1. The mechanism for forcing contacts ON/OFF, and the limit on the number of contacts which can be forced ON/OFF vary from model to model. Please refer to the manual for the model selected for more information.**

***2. If the online monitoring function is disabled, the contacts will not be automatically released from the locked states. Please check whether the present state will cause any problem.**

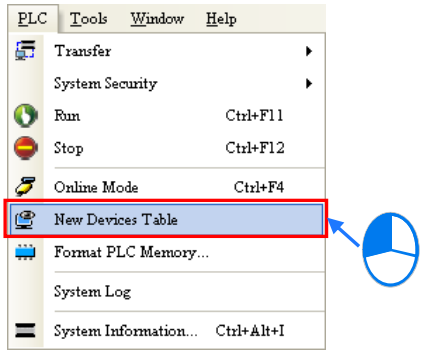
14.1.5 Device Monitoring Table

After users create monitoring tables, they can monitor the devices or the symbols in the tables online. Users are allowed to create more than one device monitoring table in a project in ISPSoft. 100 items at most can be added to a device monitoring table. Users can classify and manage the items which are monitored by themselves. Besides, device monitoring tables can be created online or offline. The device monitoring tables which are created in a project will be saved with the project, and users

There are three ways to create a device monitoring table.

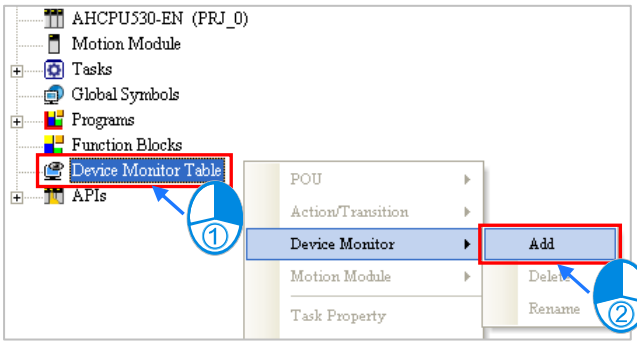
- **Method 1**

  Click **New Devices Table** on the **PLC** menu.



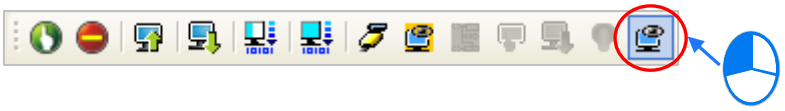
- **Method 2**

  Right-click **Device Monitoring Table** in the project management area, point to **Device Monitor** on the context menu, and click **Add**.

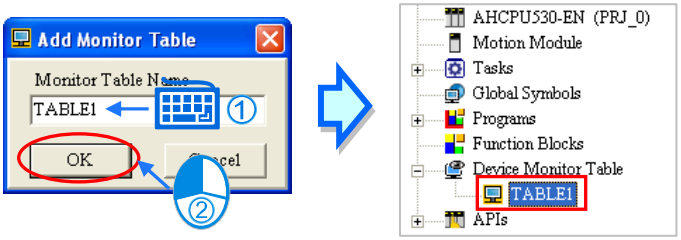


- **Method 3**

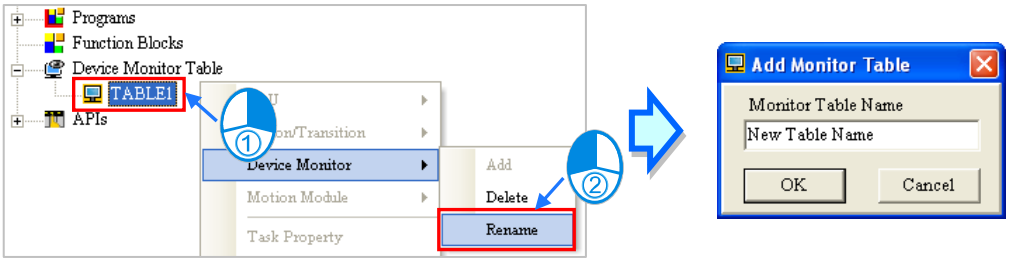
  Click on the toolbar.



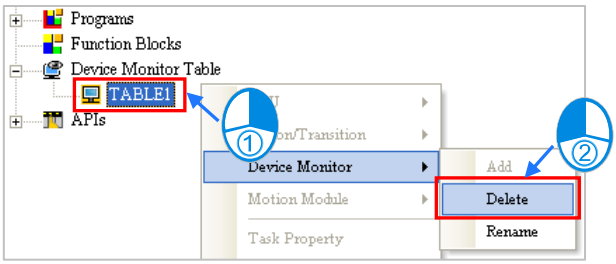
Type a table name in the **Add Monitor Table** window, and then click **OK**. An item will be under **Device Monitor Table** in the project management area, and a device monitoring window will appear in the working area. Users can manage the items in the window.



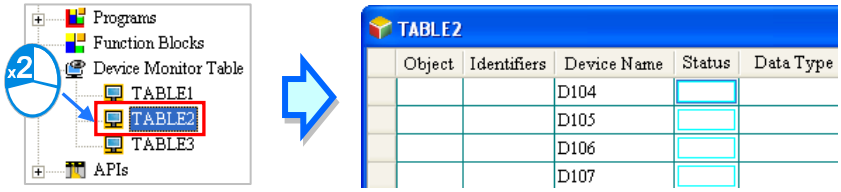
If users want to rename a device monitoring table which has been created, they can click the device monitoring table in the project management area, right-click the device monitoring table, point to **Device Monitor** on the context menu, click **Rename**, and type a new table name in the **Add Monitor Table** window.



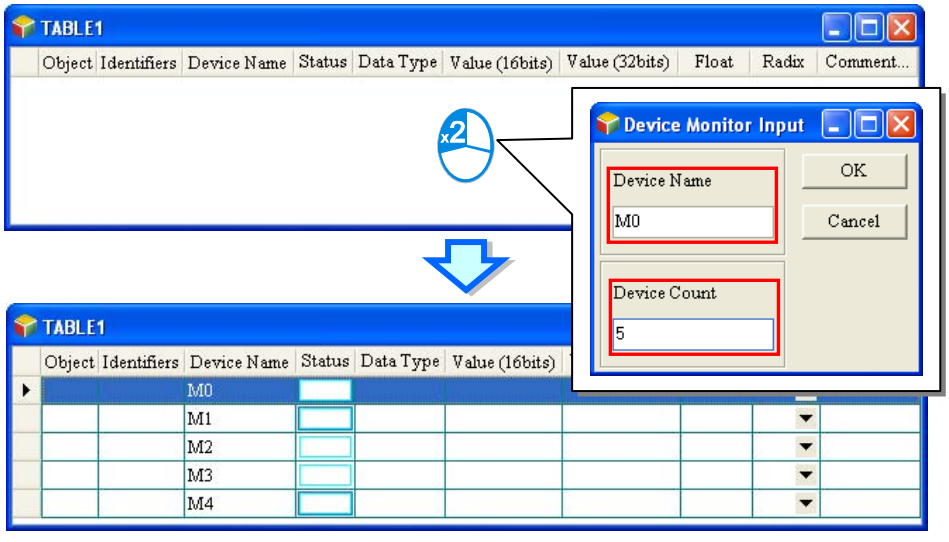
If users want to delete a device monitoring table, they can click the device monitoring table, right-click the device monitoring table, point to **Device Monitor** on the context menu, and click **Delete**.



Users are allowed to open several device monitoring windows in ISPSoft. If users want to open a device monitoring window, they can double-click the device monitoring table in the project management area.

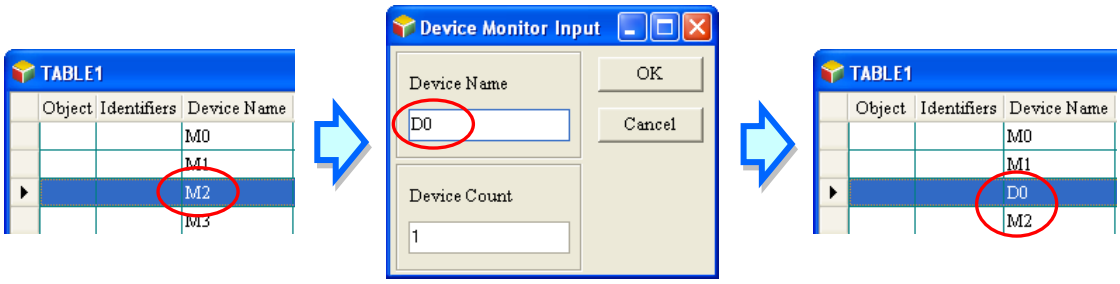


If users want to add an item to a device monitoring table, they have to double-click the blank in the device monitoring table, or type a start address and the number of devices which be monitored in the **Device Monitor Input** window. Please notice that 100 items at most can be added to a monitoring table.

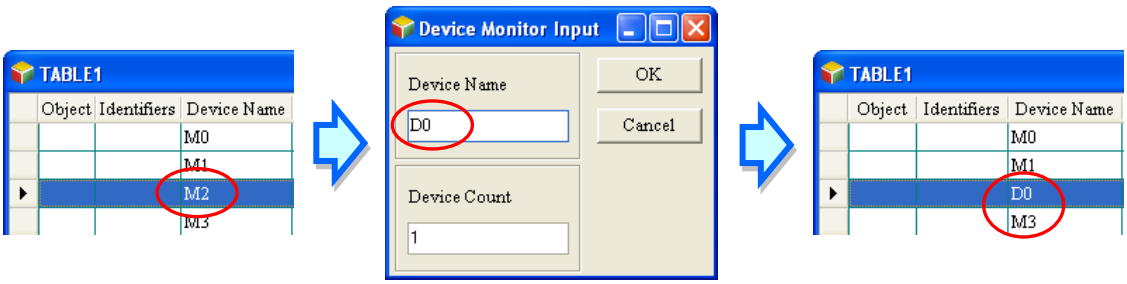


Users can press Insert on the keyboard to switch between inserting an item in the present device monitoring table and replacing an item in the present device monitoring table. The mode which is selected is displayed in the status bar in ISPSoft.

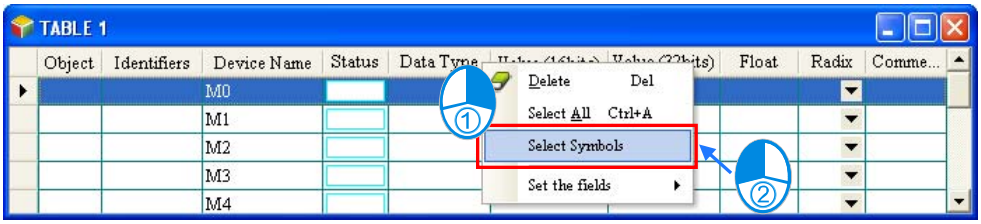
If the insertion mode is selected, the item added will be inserted above the item selected in the device monitoring table.



If the replacement mode is selected, the item added will overwrite the item selected in the device monitoring table.

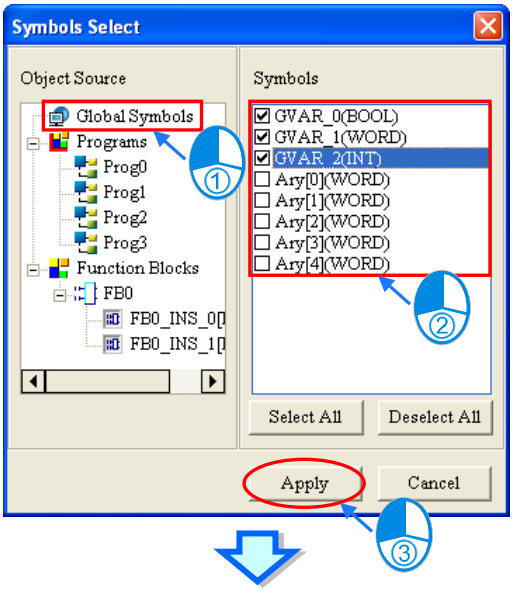


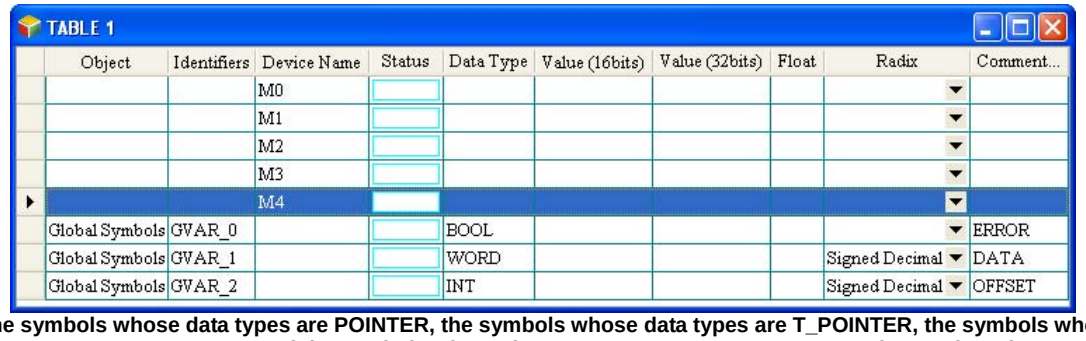
Users can add symbols as well as devices to a device monitoring table. After they right-click a device monitoring table, and select **Select Symbols** on the context menu, they can add symbols to the device monitoring table.



The users can select global symbols or local symbols in the POUs in the **Symbols Select** window. Before they select local symbols in a function block, the program must be compiled. After the function block instances declared in the superior POU are produced, they will be assigned substantial memory blocks. The users can monitor the local symbols in the function block instances. Different function block instances are assigned different memory blocks. If users want to monitor local symbols in a function block, they must select the local symbols in the instances of the function block.

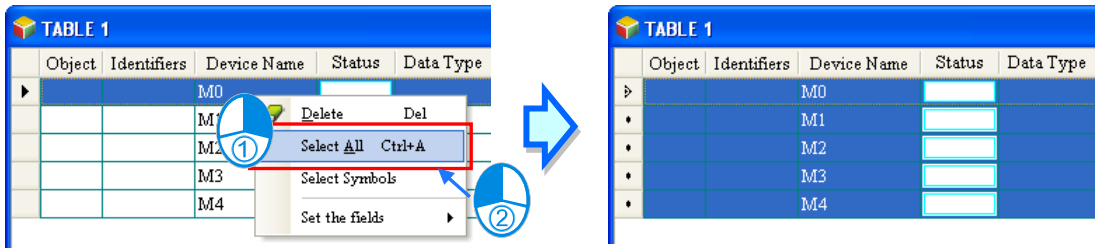
After the users select an item in the **Object Source** section, they can select symbols in the **Symbols** section. If a symbol whose data type is ARRAY is declared, the elements will be listed in the **Symbols** section. After symbols in the **Symbols** section are selected, the user can click **Apply** to add the symbols to the device monitoring table. The users can continue to select other symbols. After the users finish adding symbols to the device monitoring table, they can click **Cancel** to close the window.

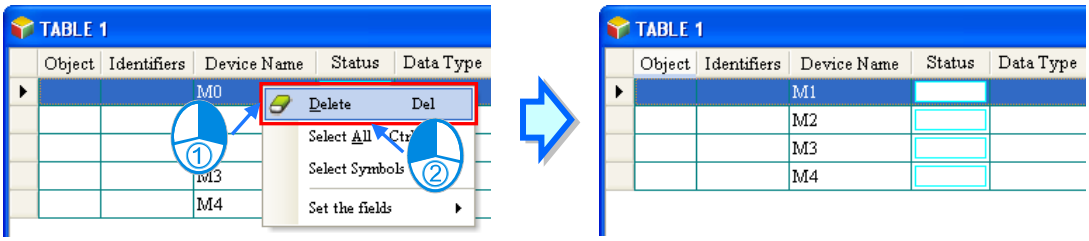




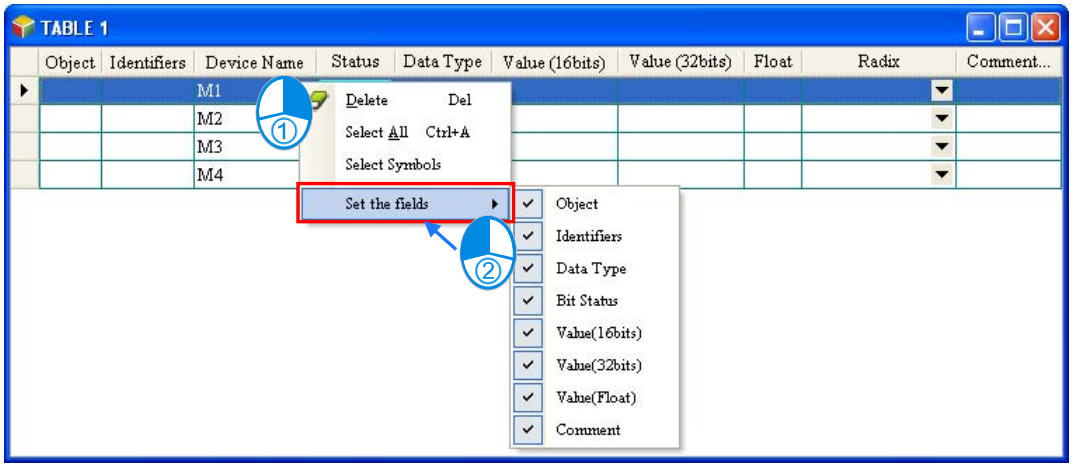
***. The symbols whose data types are POINTER, the symbols whose data types are T_POINTER, the symbols whose data types are C_POINTER, and the symbols whose data types are HC_POINTER can not be monitored.**

If the users want to delete an item in a device monitoring table, they can select the item, right-click the item, and click **Delete** on the context menu. If the users want to delete several items in a device monitoring table, they can select the items while they press Ctrl on the keyboard, right-click an item selected, and click **Delete** on the context menu. If the users want to delete a range of items in a device monitoring table, they can select the first item and the last item while they press Shift on the keyboard, right-click an item selected, and click **Delete** on the context menu. If the users want to delete all the items in a device monitoring table, they can right-click the device monitoring table, select **Select All** on the context menu, right-click an item selected, and select **Delete** on the context menu. The users can also delete the item selected or the items selected by pressing Delete on the keyboard.





The users can hide some columns in a device monitoring table. If the users want to hide some columns in a device monitoring table, thy can right-click the device monitoring table, point to **Set the Fields**, and unselect items which correspond to the columns they want to hide. Please refer to the description of the columns below for more information.

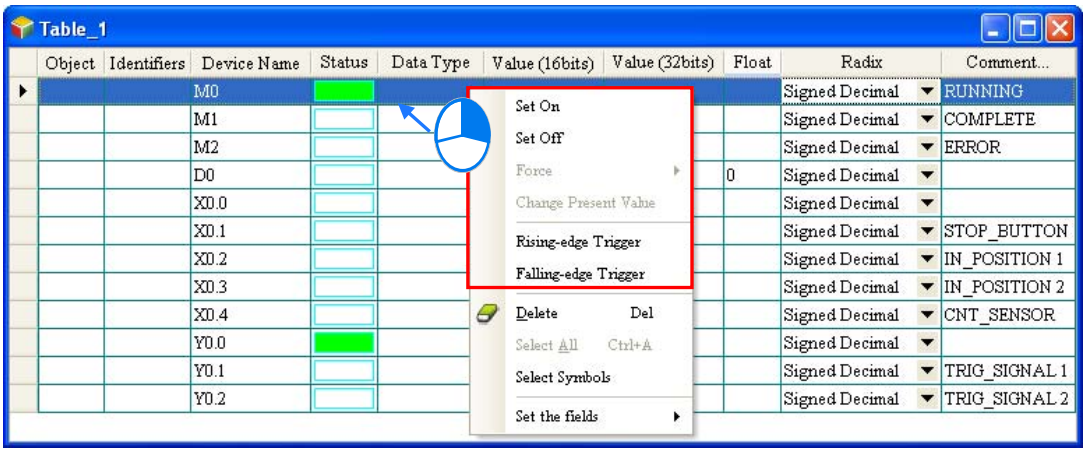


14

The description of the columns in a device monitoring table is as follows.

Column	Description
Object	The sources of the symbols are displayed.
Identifier	The identifiers of the symbols are displayed.
Device Name	The names of the devices monitored are displayed.
Status	If the objects monitored are bit devices or contacts, the states of the bit devices or the states of the contacts will be displayed in this column.
Data Type	If the objects monitored are symbols, the data types of the symbols will be displayed in this column.
Value (16 bits)	When the devices and the symbols are monitored, the 16-bit values are displayed in this column.
Value (32 bits)	When the devices and the symbols are monitored, the 32-bit values are displayed in this column.
Float	When the devices and the symbols are monitored, the 32-bit floating-point numbers are displayed in this column.
Radix	Users can select the formats of the values monitored.
Comment…	The comments on the devices monitored, and the comments on the symbols monitored are displayed in this column.

After a device monitoring table is created, the users can monitor the items in the device monitoring table online. After the users right-click an item in a device monitoring table in the online mode, they can select an item on the context menu to change the state of the item monitored, or the value of the item monitored. Please refer to section 14.1.4 for more information.



Additional remark

If the comment on a device which is monitored is changed, users have to close the device monitoring table in which the device is added, and open the window again to get the new comment on the device. If the properties of a symbol in a device monitoring table are changed, or a symbol in a device monitoring table is removed from the symbol table, the symbol in the device monitoring table will not be updated, and will not have any effect in the online mode. Please delete the symbol from the device monitoring table, and add it again.

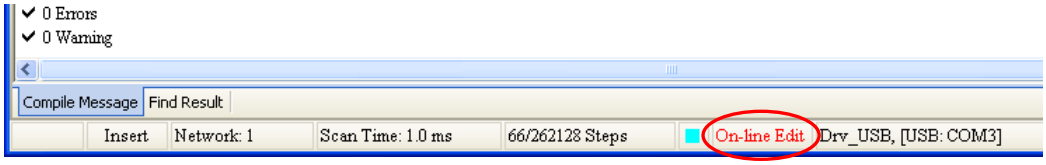
14.1.6 Online Editing Function and Online Update Function

When the system is in the online mode and the PLC is running, users can modify the program monitored through the online editing function and the online update function. Please refer to the following table for more information about the limits set on the editing of programs.

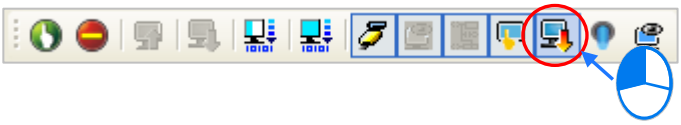
Programming language	Description
Ladder diagram	One network is edited at a time. Users are allowed to add new networks. After a new network is added, the network which can be edited will be the new network, that is, the previous networks can not be edited.
Function block diagram	One network is edited at a time. Users are allowed to add new networks. After a new network is added, the network which can be edited will be the new network, that is, the previous networks can not be edited.
Instruction list	There is no limit on the number of lines which can be edited.
Structured text	There is no limit on the number of lines which can be edited.
Sequential function chart	A sequential function chart can not be modified online, but actions and transitions can be modified. The limit set on the editing of an action or a transition depends on the language used to create the action or the transition.
Common limit	After the program is edited, users can only edit one POU. In the online editing mode, users can not modify or add a symbol.

If users want to edit the program online, they must enable the program monitoring function, make sure that the PLC runs, and click on the toolbar.

If the online editing function is enabled, **On-line Edit** will be displayed in the status bar in ISPSoft.



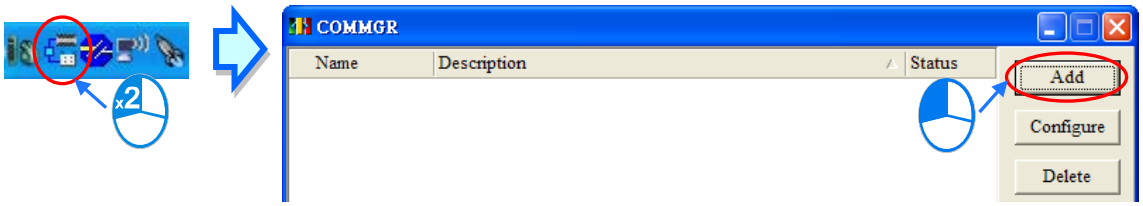
If the editing of the program is complete and the program is compiled successfully, the program will be downloaded to the PLC after users click on the toolbar. If the program is not compiled, the system will ask users to compile the program.



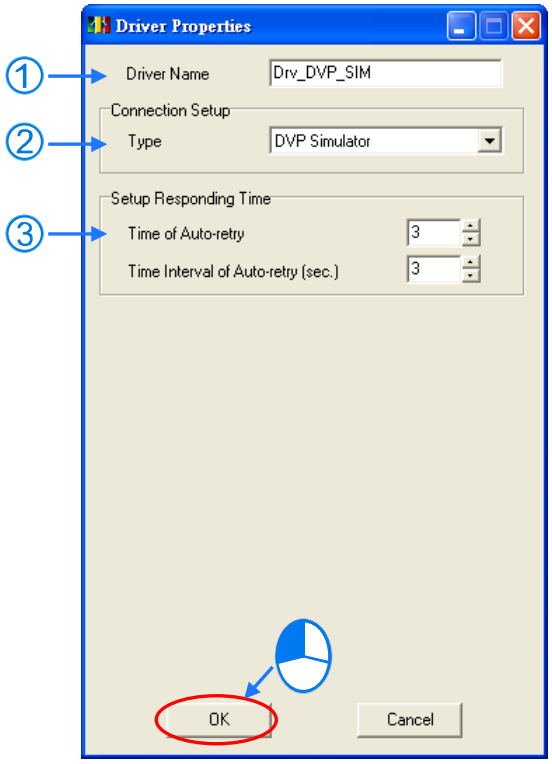
14.2 DVP Series Simulator

14.2.1 Creating a DVP Simulator

Start COMMGR, and then click **Add** in the **COMMGR** window to create a driver. Please refer to section 2.4 for more information about COMMGR.



Type a driver name in the **Driver Name** box, select **DVP Simulator** in the **Type** drop-down list box, select the number of times the sending of a command is retried in the **Time of Auto-retry** box, and select an interval of retrying the sending of a command in the **Time Interval of Auto-retry** box. After the setting is complete, users can click **OK** to create a DVP simulator.

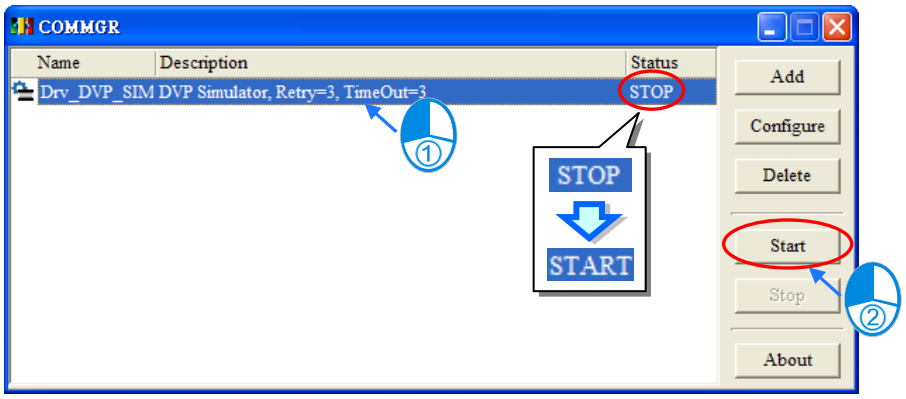


(1) Type a driver name in the **Driver Name** box.
(2) Select **DVP Simulator** in the **Type** drop-down list box in the **Connection Setup** section.
(3) Users can select the number of times the sending of a command is retried if a connection error occurs in the **Time of Auto-retry** box, and select an interval of retrying the sending of a command in the **Time Interval of Auto-retry** box.

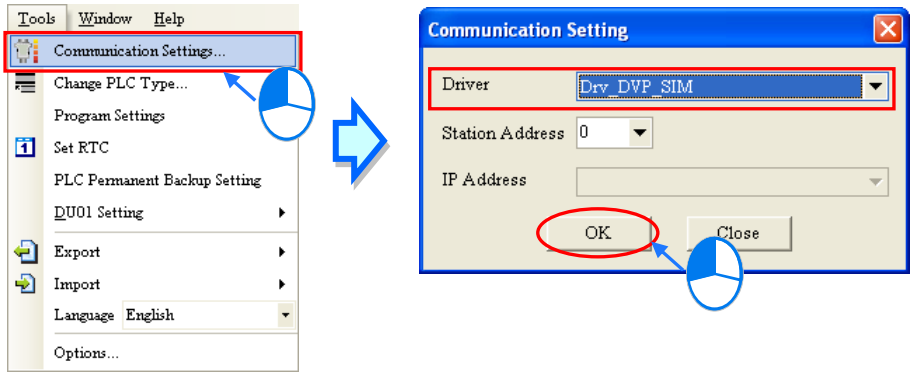
***. Users can only create a DVP simulator in COMMGR. If users want to create the second DVP simulator, a warning message will appear.**

14.2.2 Starting the DVP Simulator

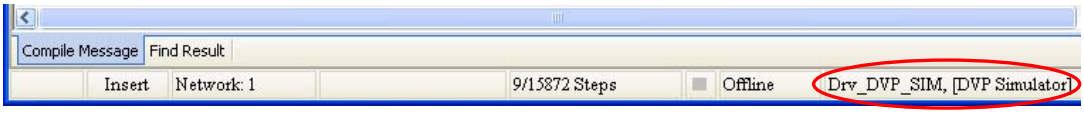
Open the **COMMGR** window, and then start the DVP simulator in COMMGR.



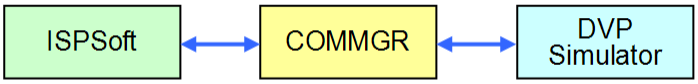
Click **Communication Settings…** on the **Tools** menu in ISPSoft, select the DVP simulator in the **Driver** box in the **Communication Setting** window, and click **OK**.



After the communication setting is complete, the connection type displayed in the status bar in ISPSoft will be a DVP simulator.



COMMGR will transmit data between ISPSoft and the DVP simulator once related connection operation is performed. During the connection, users can only download the program in ISPSoft to the DVP simulator, upload the program in the DVP simulator to ISPSoft, enable the online monitoring function, edit devices, and registers, and force X devices ON/OFF.



14.3 Debugging Mode for a DVP Series PLC

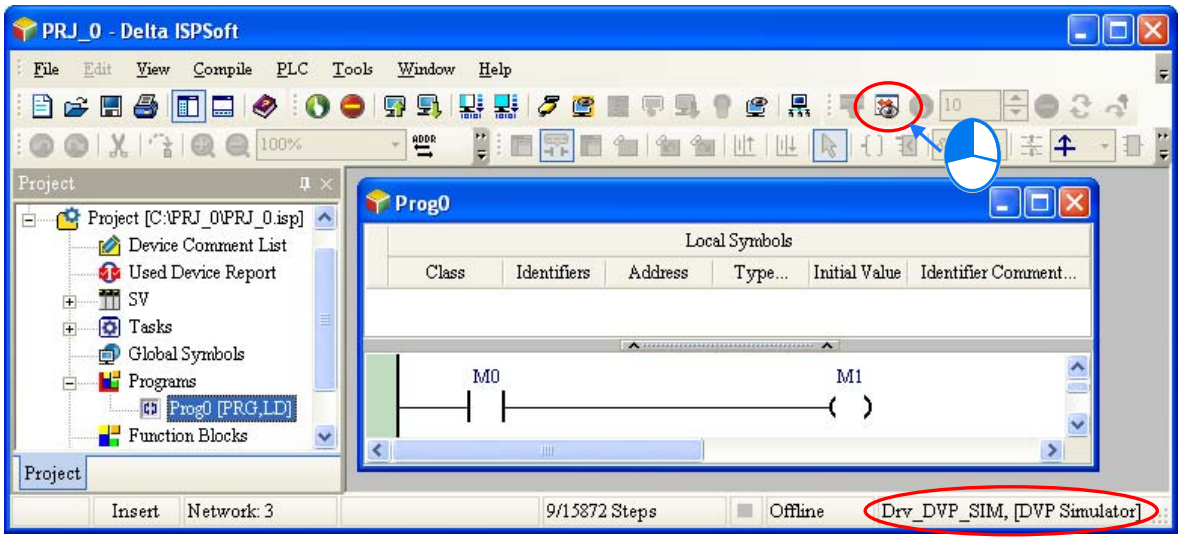
ISPSoft provides debugging functions for DVP series PLCs. Users can debug and test the program in a DVP series PLC by means of the debugging tools. Before the debugging mode is enabled, users have to pay attention to two points below.

(a) The debugging mode for a DVP series PLC can be executed only if the connection type is a simulator.

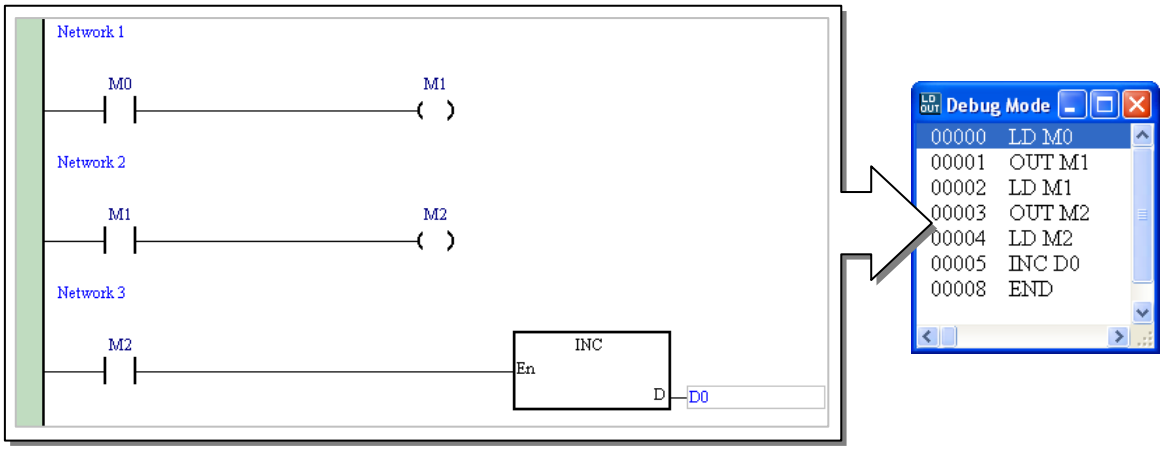
(b) If the online mode is enabled, the debugging mode can not be enabled. However, the device monitoring function can be enabled after the debugging mode is enabled.

14.3.1 Enabling the Debugging Mode for a DVP Series PLC

The connection type must be a DVP simulator. Please refer to section 14.2 for more information. After users click on the toolbar, a window will appear. If the users want to disable the debugging mode, they can click again.

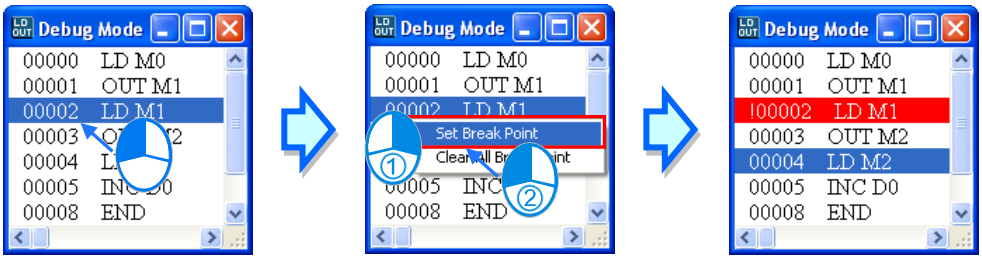


After the program in a DVP series PLC is translated into an instruction list, the instruction list will appear in the debugging window for the DVP series PLC. The part of the instruction list which is on a blue ground in the debugging window is the position where the execution of the program stops. The position where the execution of the program stops is the part of the program which has not been executed.

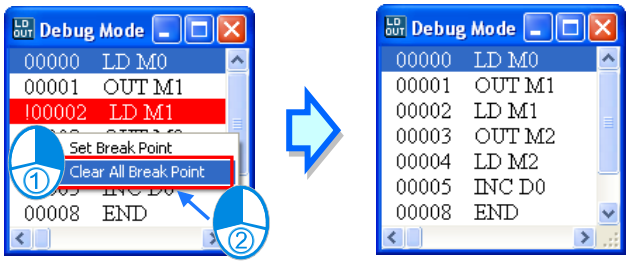


14.3.2 Adding and Clearing Breakpoints

After users right-click a line in the debugging window and select **Set Break Point** on the context menu, a breakpoint will be added to the line. If the users right-click the line again and select **Set Break Point** on the context menu, the breakpoint will be canceled. In the debugging window, an exclamation appears at the beginning of the line to which a breakpoint is added, and the line is on a red ground.



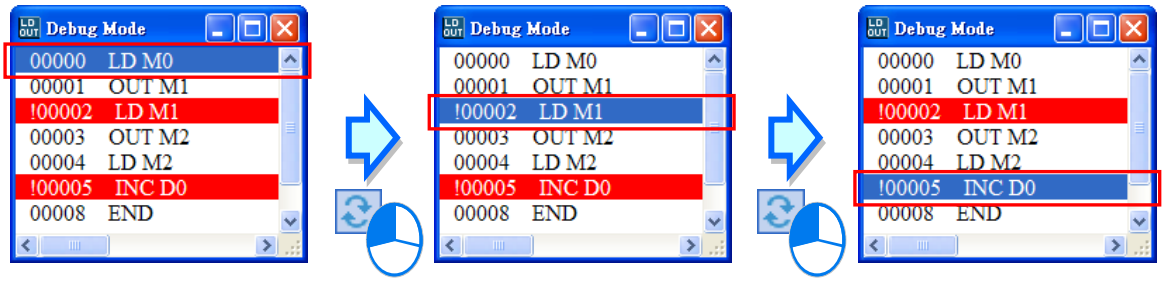
If users want to clear the breakpoints in the debugging window, they can right-click a line, and select **Clear All Break Points** on the context menu.



14.3.3 Execution of the Program in the Debugging Mode

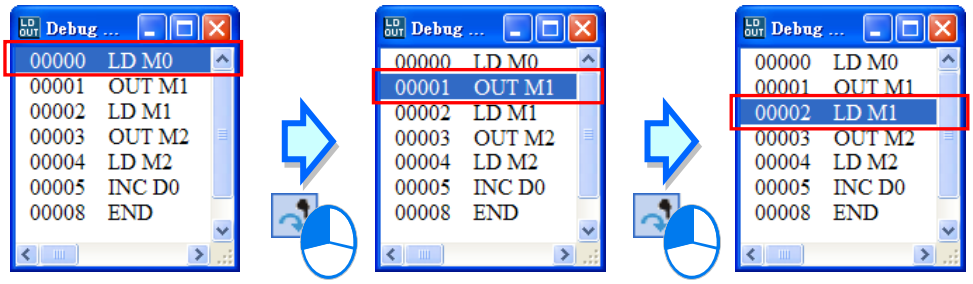
- **Continuous execution**

  After on the toolbar is clicked, the execution of the program will stop at a breakpoint. If users click again, the execution of the program will continue, and will stop at the next breakpoint. If there are no breakpoints, the program will be scanned repeatedly after clicked.



- **Single-step execution**

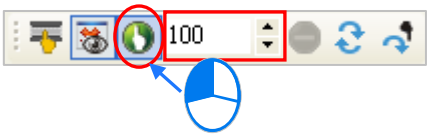
  If the execution of the program stops, single-step execution will be performed after on the toolbar is clicked.



- **Setting the number of times the program is executed**

  If the number of times the program is scanned reaches the number specified, the scan will stop.

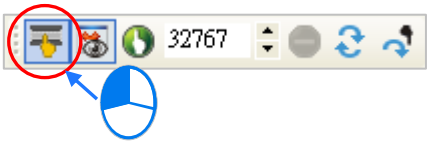
Besides, the scan of the program is not affected by the breakpoints in the program. Users have to set a number in 100 on the toolbar. The maximum number which can be set is 32767. Click on the toolbar after the setting is complete.



- **Stopping the execution of the program and resetting the execution order**

  After users click on the toolbar, the execution of the program stops.

After on the toolbar is clicked, the execution order and the values will be reset. The execution of the program will start form the beginning of the program next time the program is executed.

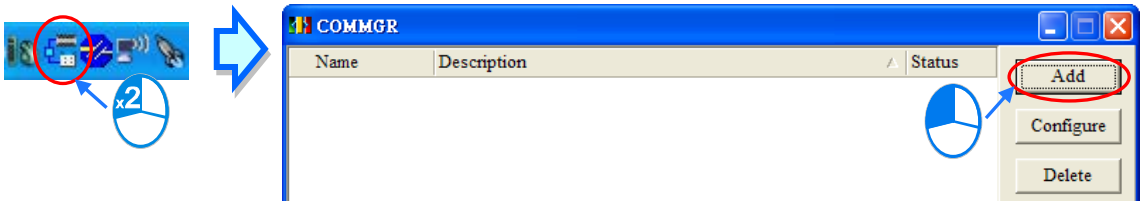


***. After the debugging mode for a DVP series PLC is enabled, the devices in the device monitoring table still can be monitored.**

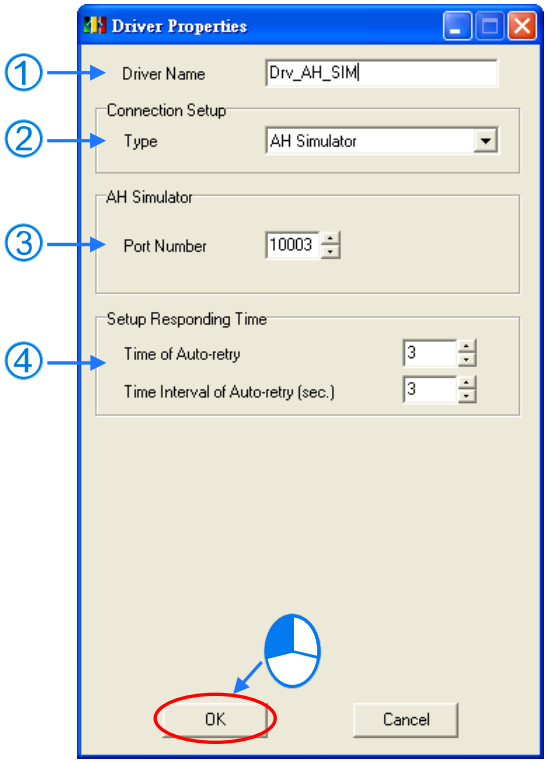
14.4 AH500 Series Simulator

14.4.1 Creating and Starting an AH Simulator

Start COMMGR, and then click **Add** in the **COMMGR** window to create a driver. Please refer to section 2.4 for more information about COMMGR.



Type a driver name in the **Driver Name** box, select **AH Simulator** in the **Type** drop-down list box, type a communication port number in the **Port Number** box, select the number of times the sending of a command is retried in the **Time of Auto-retry** box, and select an interval of retrying the sending of a command in the **Time Interval of Auto-retry** box. After the setting is complete, users can click **OK** to create an AH simulator.



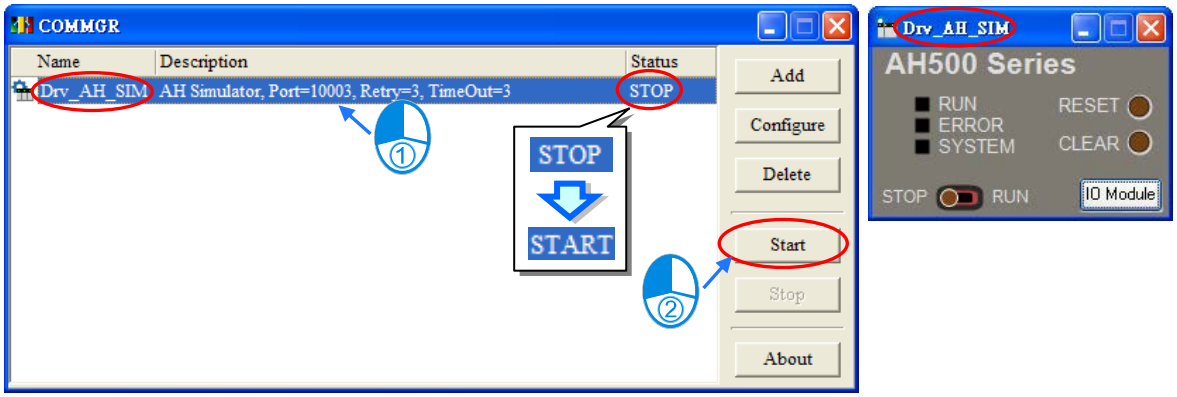
(1) Type a driver name in the **Driver Name** box.

(2) Select **AH Simulator** in the **Type** drop-down list box in the **Connection Setup** section.

(3) Users can type a communication port number in the **Port Number** box. If it is not necessary to specify a communication port number, please use the default communication port number. Besides, if several AH simulators are created, the communication port numbers assigned to the AH simulators can not be the same, and the system skips the port numbers which have been specified.

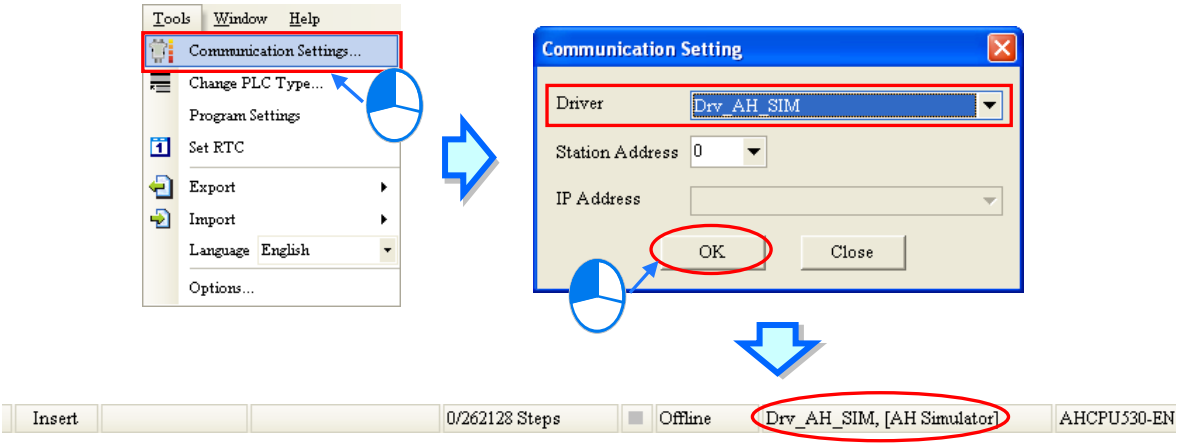
(4) Users can select the number of times the sending of a command is retried if a connection error occurs in the **Time of Auto-retry** box, and select an interval of retrying the sending of a command in the **Time Interval of Auto-retry** box.

Open the **COMMGR** window, and then start an AH simulator in COMMGR. After the AH simulator is started, a window will appear. The title of the window is the driver name of the AH simulator. Besides, users are allowed to create several AH simulators, but eight AH simulators at most can be started simultaneously.



Click **Communication Settings…** on the **Tools** menu in ISPSoft, select the AH simulator in the **Driver** box in the **Communication Setting** window, and click **OK**. After the communication setting is

complete, the connection type displayed in the status bar in ISPSoft will be an AH simulator.



COMMGR will transmit data between ISPSoft and the AH simulator once related connection operation is performed.

The software functions, parameters, and instructions which are not supported by an AH simulator are as follows.

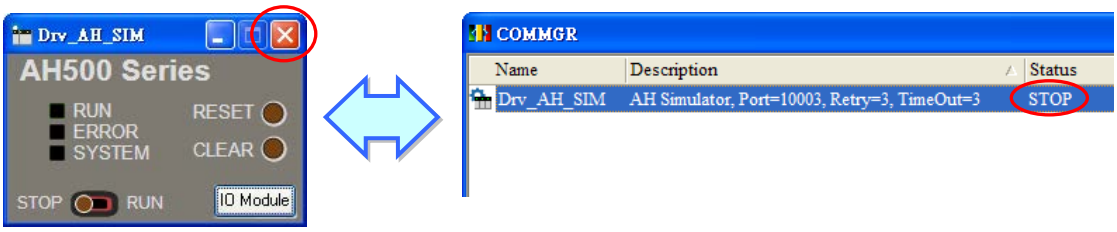
Software function	NWCONFIG, CARD Utility, the I/O scan provided by HWCONFIG, and the online mode provided by HWCONFI
Parameter	Module parameters, communication port parameters, Ethernet parameters, and parameters related to a memory card
Instruction	RS, FWD, REV, STOP, RDST, RSTEF, MODRW, FROM, TO, READ, WRITE, RPASS, SRCVD, SSEND, SOPEN, SCLOSE, MSEND, EMDRW, MWRIT, MREAD, MTWRIT, and REF are not supported. Accessing X devices and Y devices directly is also not supported.

14.4.2 Operation Panel of an AH Simulator

- **Introduction of the operation panel of an AH simulator**

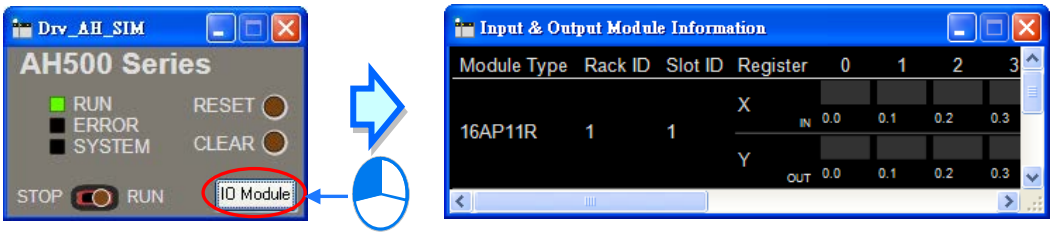
  After an AH simulator in COMMGR is started, the operation panel of the AH simulator will appear. The lights and buttons on the operation panel correspond to the lights and buttons on an AH500 CPU module. Users can simulate the operation of an actual AH500 CPU module by means of the operation panel. Please refer to the manuals related to AH500 CPU modules for more information about the lights and the buttons on the operation panel.

  After the operation panel is closed, the status of the AH simulator displayed in COMMGR will be **STOP**. Likewise, if the status of the AH simulator displayed in COMMGR becomes **STOP**, the operation panel will be closed. After the operation panel is closed, the program and the parameters which have been downloaded will be reserved, and will be loaded into the AH simulator next time the AH simulator is started. Besides, if the computer is restarted, the system will start COMMGR automatically, and the statuses of the AH simulators displayed in COMMGR will become **STOP**.



A hardware configuration can be simulated in the AH simulator. As a result, after the parameters in HWCONFIG is downloaded to the AH simulator, the parameters will be memorized in the AH simulator.

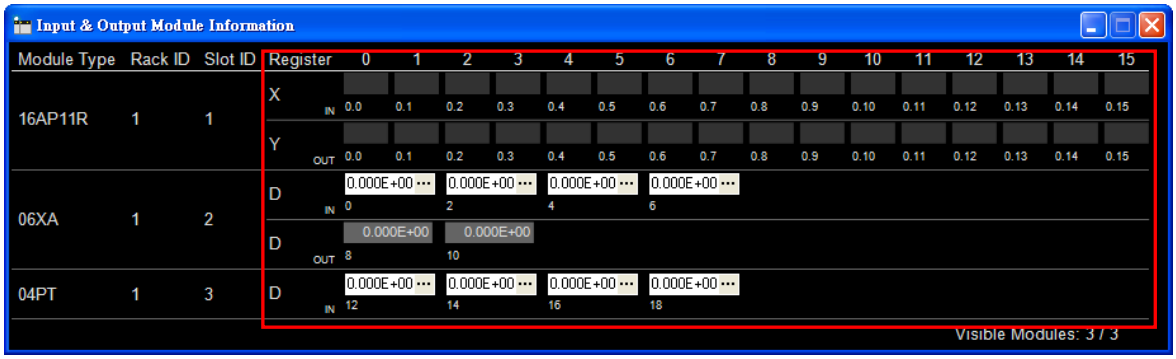
If the modules which are configured include a digital I/O module, an analog I/O module, and a temperature measurement module, an operation panel for these modules will appear after users click **IO Module** on the operation panel of the AH simulator. The users can simulate input statuses and output statuses through the panel. However, if the users want to simulate input statuses and output statuses, the AH simulator must run.



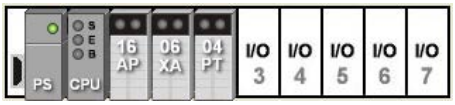
The AH simulator simulates the functions of an AH500 series CPU module. Although the parameters in the I/O modules can be downloaded to the AH simulator, the processing procedures in the actual I/O modules can not be simulated. The users can only debug or test the program by means of the states of the X devices, the states of the Y devices, and the values in the D registers in the modules.

- **Introduction of the operation panel for I/O modules**

  The operation panel for AH16AP11R-5A, AH06XA-5A, and AH04PT-5A are shown below. The names of the I/O modules, the rack that the modules occupy, the slot that the modules occupy, the input devices, and the output devices are displayed on the operation panel.



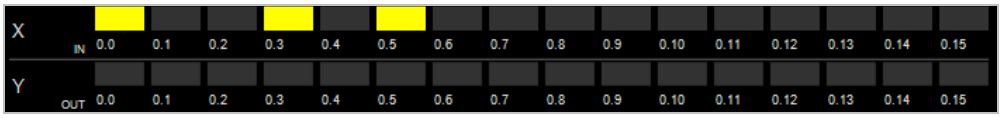
The input devices and the output devices in the red frame in HWCONFIG below are the same as the devices in the red frame in the **Input & Output Module Information** window above. As a result, users can simulate input statuses and output statuses by means of the operation panel.



Information: Rack 1

Slot No.	Label	Firmware...	Description	Input Device ...	Output Device ...	Comment
-	AHPS05-5A	-	AH Power Supply Module	None	None	
-	AHCPU530-EN	1.00	Basic CPU module building	None	None	
0	AH16AP11R-5A	-	8 x DI VDC, 8 x DO VAC/VI	X0.0 ~ X0.15	Y0.0 ~ Y0.15	
1	AH06XA-5A	1.00	4 x 16bit AI, 2 x 16bit AO	D0 ~ D7	D8 ~ D11	
2	AH04PT-5A	1.00	4 x 3/4 wires RTD input 0.1	D12 ~ D19		

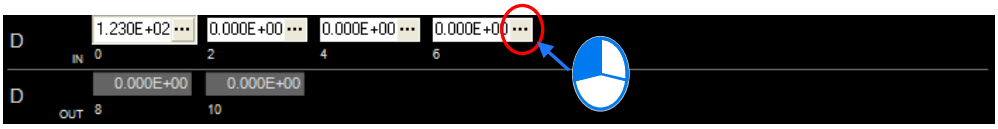
The input devices and the output devices assigned to AH16AP11R-5A are shown below. Users can view the states of the Y devices, but they can not manipulate the Y devices. Besides, users can manipulate the states of the X devices by clicking the X devices.



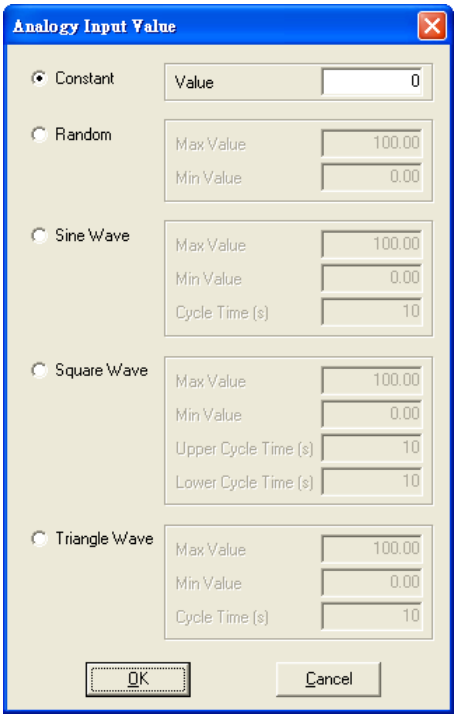
The input devices and the output devices assigned to AH06XA-5A are shown below. The values in the boxes are 32-bit floating-point numbers. Users can view the values in the boxes above D8 and D10, but they can not set values in the boxes. Besides, users can simulate input signals by setting values in the boxes above D0, D2, D4, and D6. The values in the boxes are the values in the D devices, and are not analog values.

- **Setting simulated analog input signals**

If users want to set an analog input value in the box above a device, they have to click ··· at the right side of the box to open the **Analog Input Value** window.

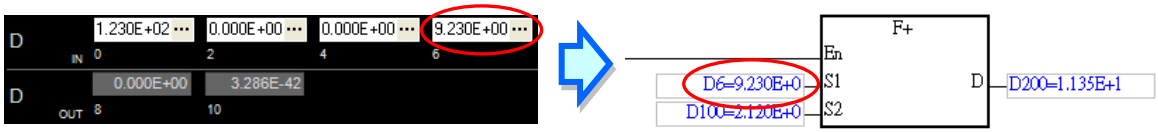


The **Analog Input Value** window is shown below. Select a signal type, and then type values at the right side of the signal type selected. Please refer to the following table for more information about the signal types in the **Analog Input Value** window.



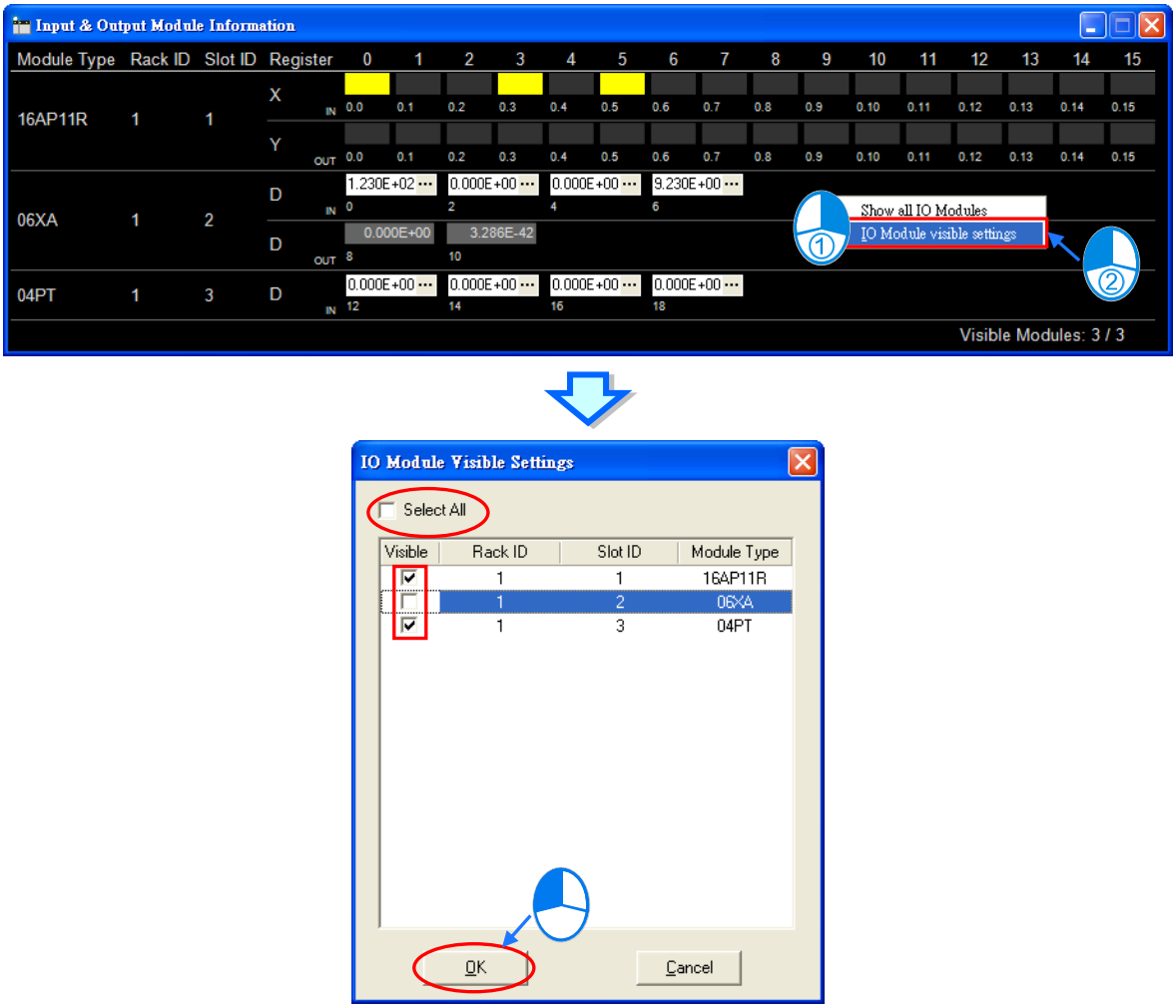
Signal type	Description
Constant	Users can specify a constant.
Random	Users can specify a maximum value and a minimum value.
Sine Wave	Users can specify a maximum value, a minimum value, and specify a cycle.
Square Wave	Users can specify a maximum value, a minimum value, a high period, and a low period.
Triangle Wave	Users can specify a maximum value, a minimum value, and a cycle.

After the setting is complete, the signals sent to COMMGR will depend on the signal types selected. ISPSoft can get the input signals received by the AH simulator by means of COMMGR.

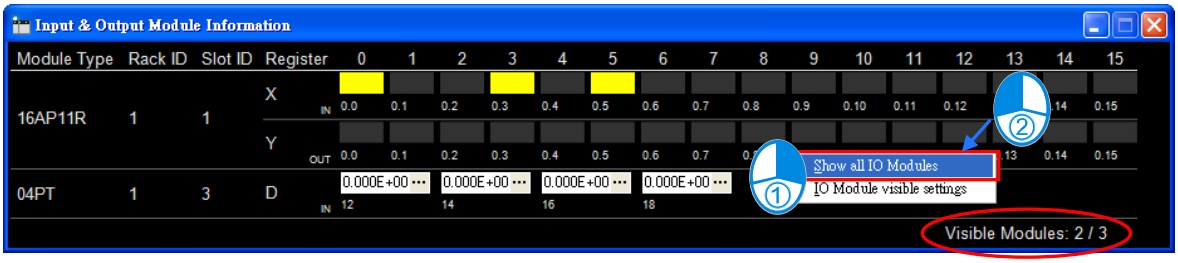


- **Hiding the modules displayed on the operation panel for I/O modules**

  Right-click the operation panel for the I/O modules, and then click **IO Module Visible Settings** on the context menu. If users select a module in the **IO Module Visible Setting** window, the module will be displayed. If users unselect a module in the **IO Module Visible Setting** window, the module will be hid. If users select or unselect the **Select All** checkbox in the **IO Module Visible Setting** window, all the modules in the **IO Module Visible Setting** window will be selected or unselected. Finally, click **OK**.



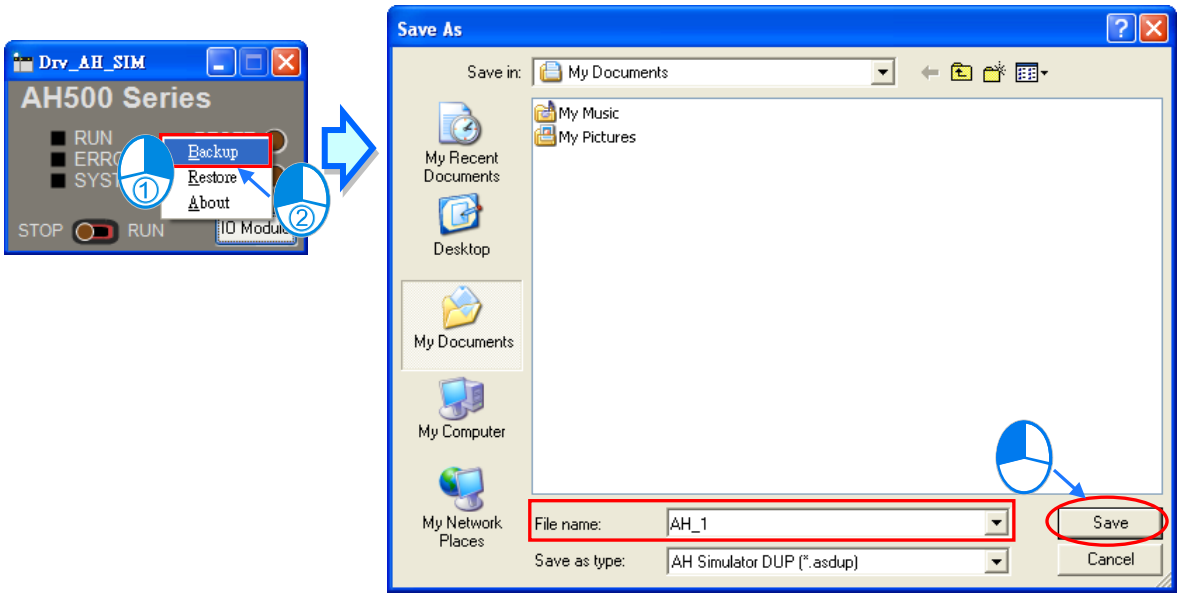
The numbers in the lower right corner indicates the number of modules which are displayed and the total number of modules. The format is **Number of modules which are displayed/Total number of modules**. If users want to display all the modules, they can right-click the operation panel, and select **Show All IO Modules** on the context menu.



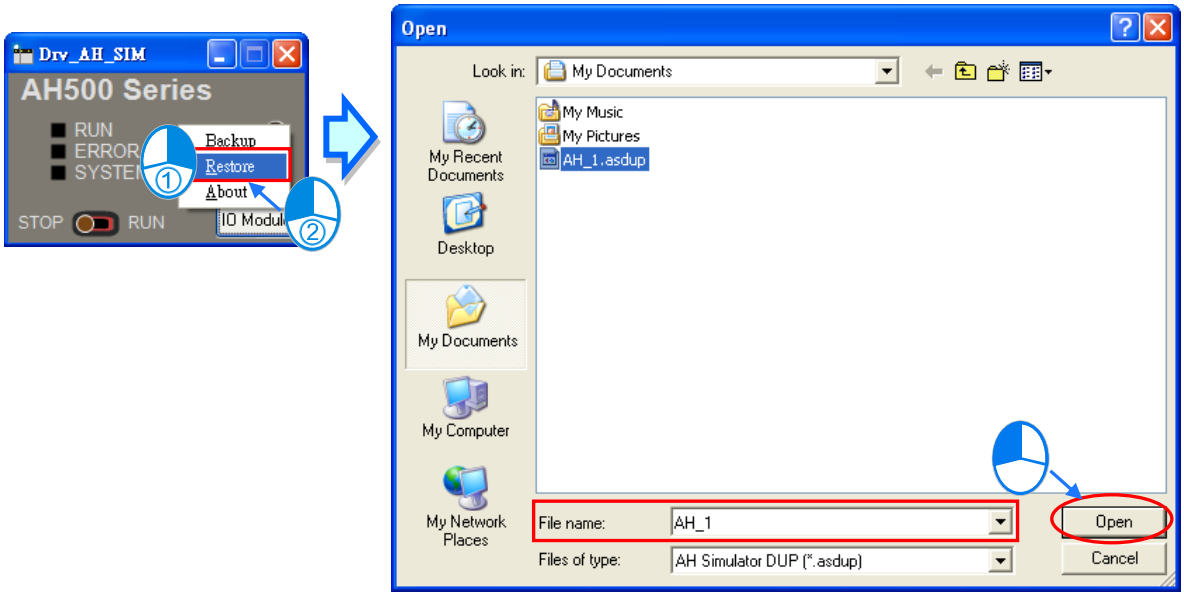
- **Backing up the data in an AH simulator and restoring the data to the AH simulator**

  The parameters and the program downloaded to the AH simulator can be backed up onto the computer. The backup file can be restored to the AH simulator, or copied into other AH simulators.

  If users want to backing up the data in the AH simulator, they have to right-click the operation panel of the AH simulator, click **Backup** on the context menu, specify a path in the **Save As** window, type a file name in the **File name** box, and click **Save**.



14 If the users want to restore the data, they have to right-click the operation panel of the AH simulator, click **Restore** on the context menu, click the backup file in the **Open** window, and click **Open**.



14.5 Debugging Mode for an AH500 Series CPU Module

14.5.1 Enabling the Debugging Mode for an AH500 Series CPU Module

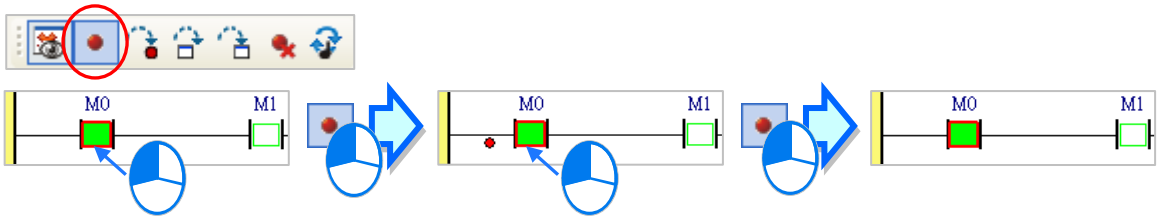
The debugging mode which ISPSoft provides for AH500 series CPU modules supports all the programming languages. However, the debugging functions can only be executed in the actions in a sequential function chart, and the steps to which the actions belong must be ON. Besides, if the state of a transition in a sequential function chart does not make the transition from one step to another step, users must change the state of the transition.

Click on the toolbar, click on the toolbar, and click on the toolbar.

If the debugging mode is not enabled, but the CPU module is in the debugging mode, the debugging mode can not be enabled. (When the CPU module is in the debugging mode, **Debug** is displayed in the status bar in ISPSoft.) The reason is that the debugging mode was not disabled normally last time or another user on the network is debugging the program in the CPU module. If users want to reset the debugging mode, they must stop the CPU module, start the CPU module again, and enable the debugging mode.

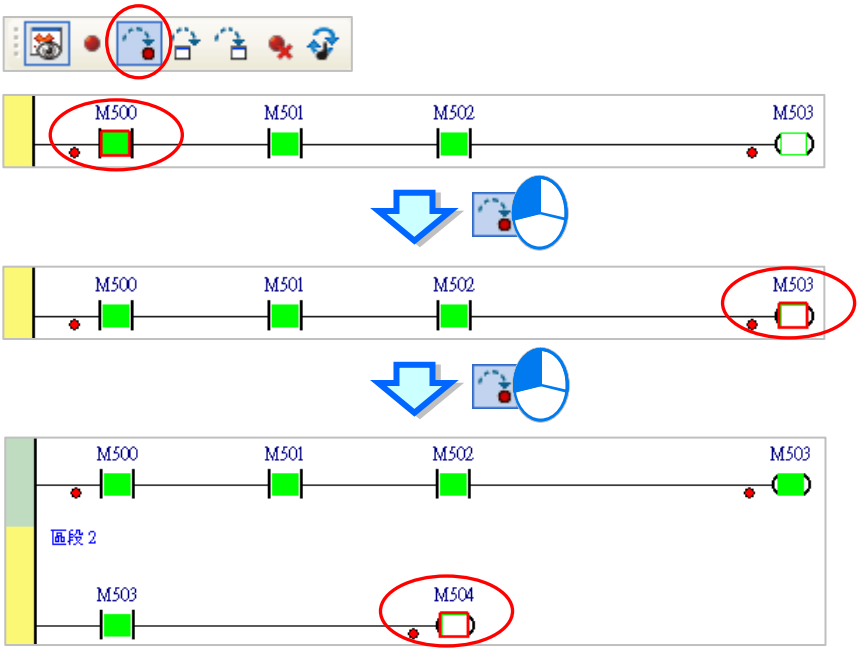
14.5.2 Adding and Clearing Breakpoints

Ten breakpoints at most can be added to the program in a project for an AH500 series CPU module. After users click a position in a program editing window and click on the toolbar, a breakpoint will be added to the position. If the users click the position again and click on the toolbar, the breakpoint will be canceled.



If users want to clear all the breakpoints in the program in an AH500 series CPU module, they can click on the toolbar.

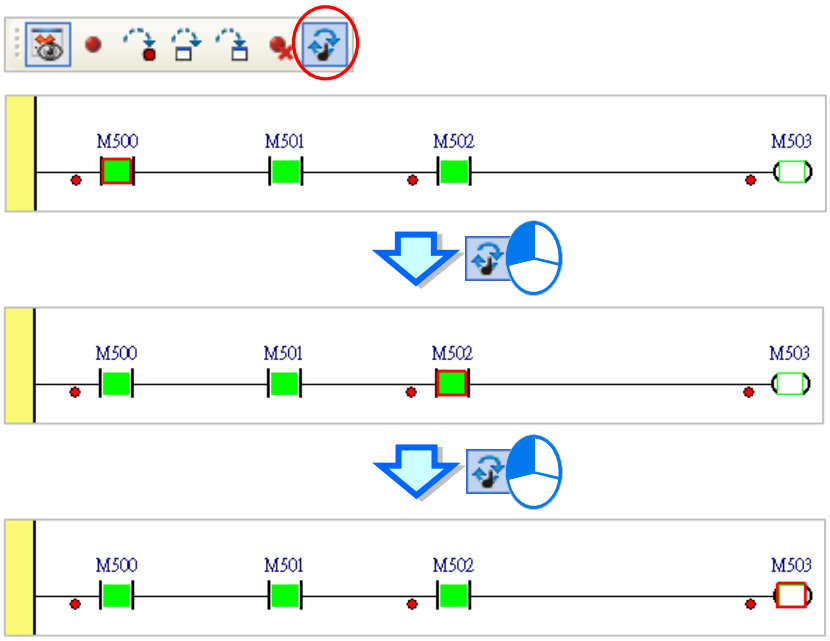
After on the toolbar is clicked, the execution of the program will stop at a breakpoint. If users click again, the execution of the program will continue, and will stop at the next breakpoint. The position where the execution of the program stops is the part of the program which has not been executed.



14

14.5.3 Continuous Execution

If on the toolbar is clicked, the execution of the program will stop at a breakpoint. If users click again, the execution of the program will continue, and will stop at the next breakpoint. If there are no breakpoints, the program will be scanned repeatedly after is clicked.

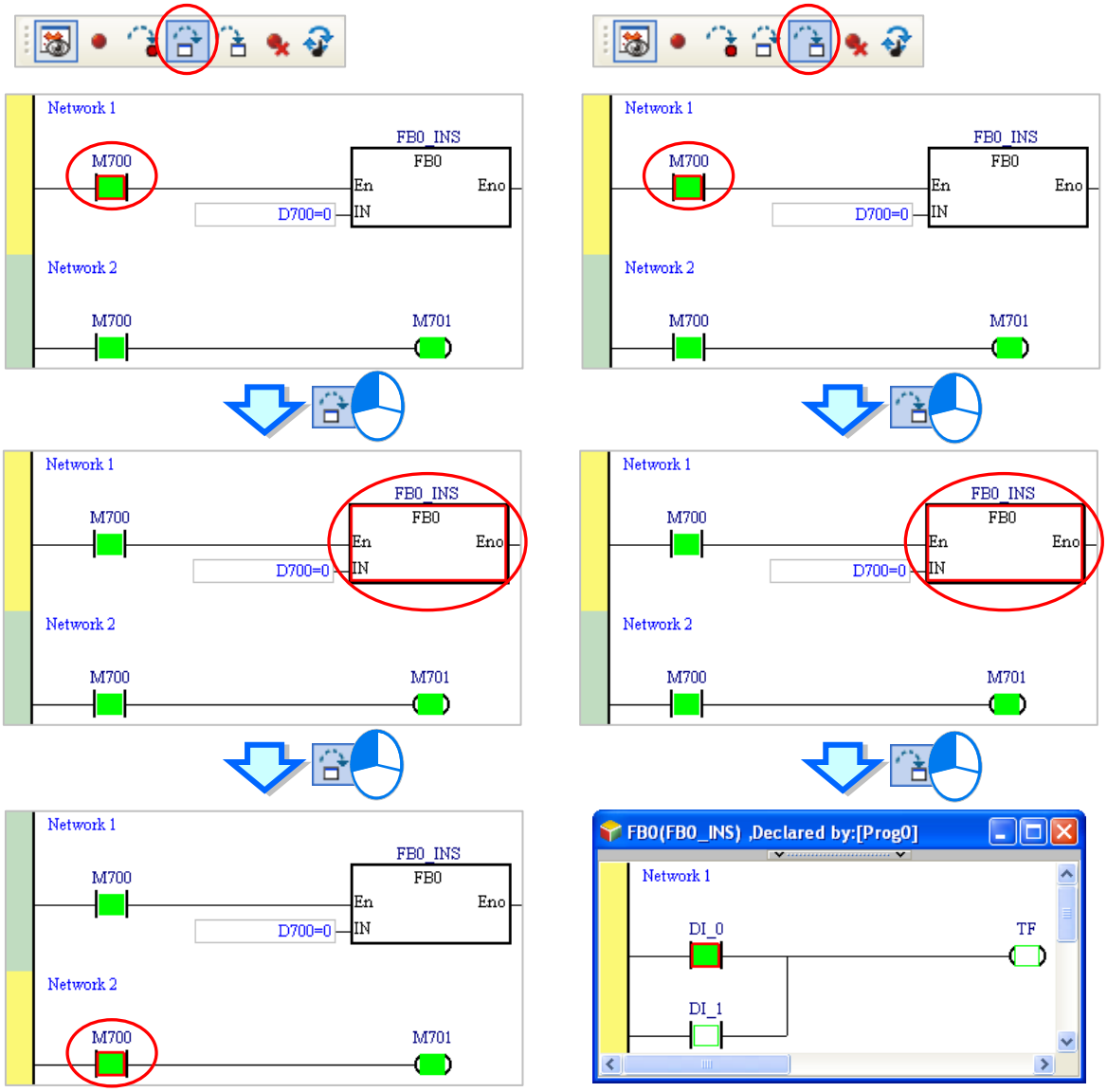


14.5.4 Single-step Execution

After or on the toolbar is clicked, the part of the program selected will be executed. If a function block is selected, the function block will be executed after is clicked, and the first step in the function block will be executed after is clicked. However, if the function block selected is protected with a password, or the state of the En pin of the function block selected is OFF, the first

step in the function block will not be executed whether or is clicked. Besides, if users want to perform single-step execution in a structured text, they must add breakpoints to the structured text. If users want to perform single-step execution in the POUs following a sequential function chart, they must add breakpoints to the POUs.

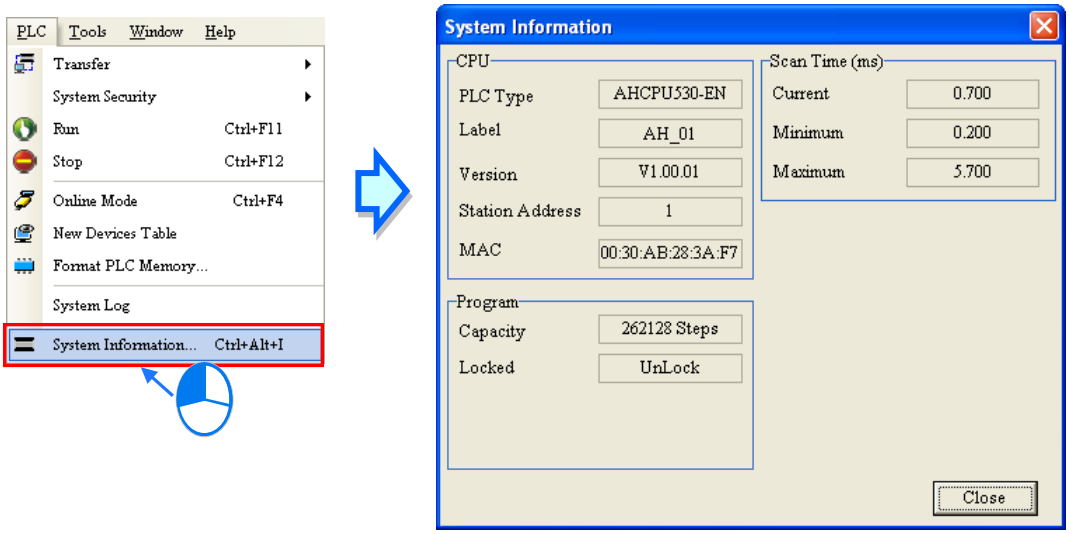
The different between and is shown below.



14.6 Checking the Status of a PLC

14.6.1 System Information

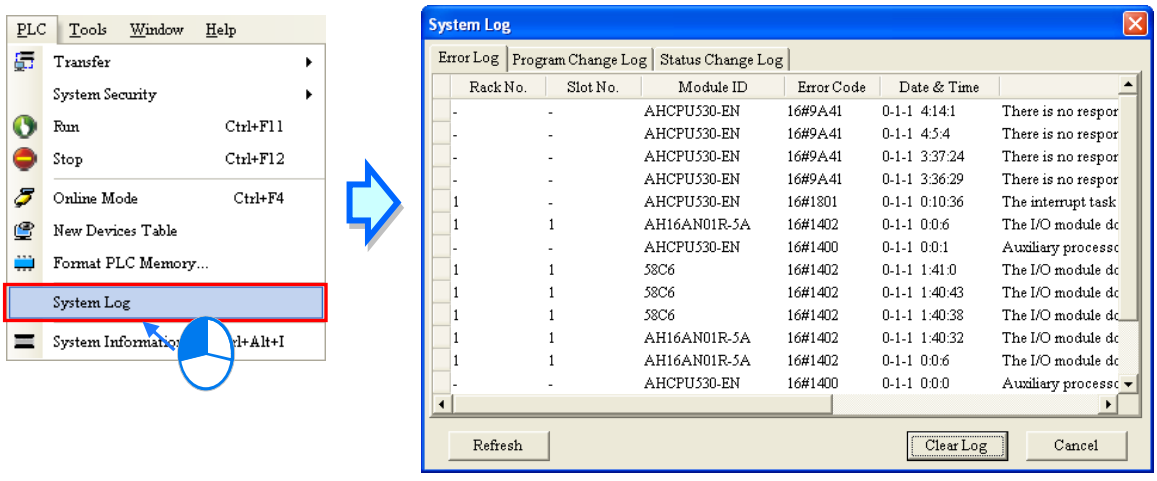
If ISPSoft is connected to the PLC normally, the system will retrieve information from the PLC after users click **System Information…** on the **PLC** menu.



14.6.2 System Log (Only for AH500 Series CPU Modules)

The actions and the errors which occur in an AH500 series CPU module are stored in special data registers in the AH500 series CPU module. If ISPSoft is connected to an AH500 series CPU module normally, users can view the actions and the errors which occur in the AH500 series CPU module after they click **System Log** on the **PLC** menu.

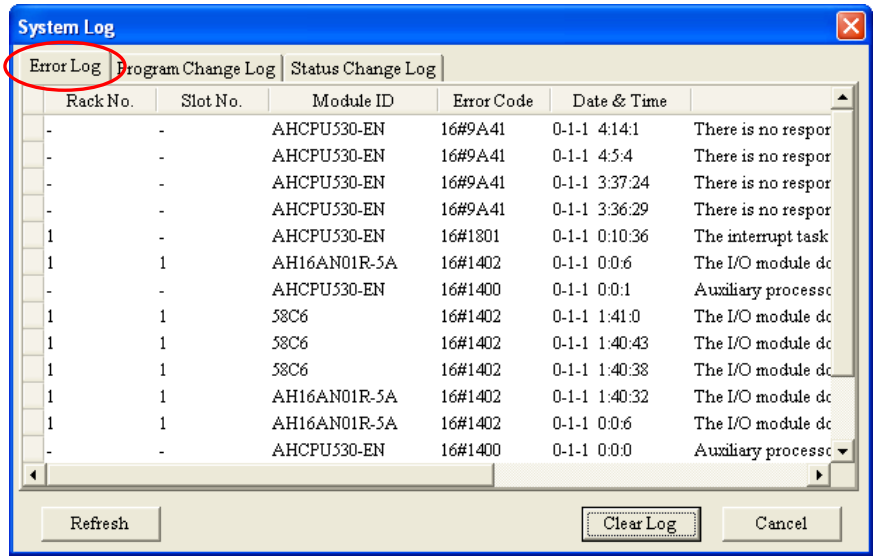
After users click **System Log** on the **PLC** menu, the **System Log** window will appear. There are three tabs in the **System Log** window. The data displayed in the window is the data retrieved from the CPU module. The data displayed in the window is not synchronized with the data in the CPU module. After users click **Refresh** in the window, the data displayed in the window will be updated.



- **Error Log**

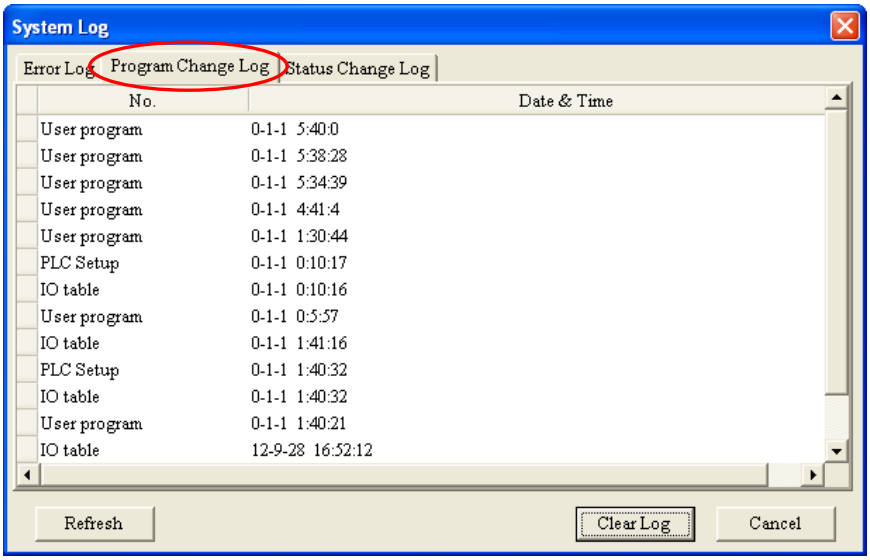
  After users click the **Error Log** tab, they can view the errors occurring in the CPU module and the I/O modules. The error codes which correspond to the errors occurring in the CPU module and the I/O modules, the points when errors occur in the CPU module and the I/O modules, the descriptions of the errors occurring in CPU module and the I/O modules, the modules where errors occur, the racks which are occupied by the modules where errors occur, and the slots which are occupied by the modules where errors occur are displayed in the window.

  After users click **Refresh**, the system will retrieve data from the CPU module, and the data displayed in the window will be updated. After users click **Clear Log**, the error log in the window and the error log in the CPU module will be cleared, and the CPU module will be reset. If the reasons for the errors still exist, the errors will still occur in the CPU module, and the new error log related to the errors will be produced.



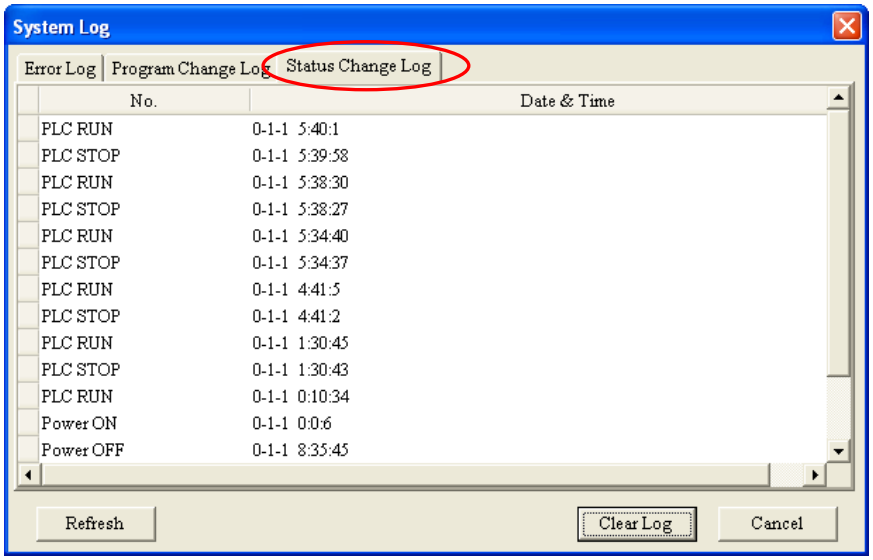
- **Program Change Log**

  After users click the **Program Change Log** tab, they can view the log related to the downloading/uploading of programs and the downloading/uploading of parameters. After users click **Refresh**, the system will retrieve data from the CPU module, and the data displayed in the window will be updated. After users click **Clear Log**, the program change log in the window and the program change log in the CPU module will be cleared.



- **Status Change Log**

  After users click the Status Change Log tab, they can view the log related to the change of the status of the CPU module. After users click **Refresh**, the system will retrieve data from the CPU module, and the data displayed in the window will be updated. After users click **Clear Log**, the status change log in the window and the status change log in the CPU module will be cleared.


System Log
Error Log
Program Change Log
Status Change Log
No.
Date & Time
PLC RUN 0-1-1 5:40:1
PLC STOP 0-1-1 5:39:58
PLC RUN 0-1-1 5:38:30
PLC STOP 0-1-1 5:38:27
PLC RUN 0-1-1 5:34:40
PLC STOP 0-1-1 5:34:37
PLC RUN 0-1-1 4:41:5
PLC STOP 0-1-1 4:41:2
PLC RUN 0-1-1 1:30:45
PLC STOP 0-1-1 1:30:43
PLC RUN 0-1-1 0:10:34
Power ON 0-1-1 0:0:6
Power OFF 0-1-1 8:35:45
Refresh
Clear Log
Cancel

14

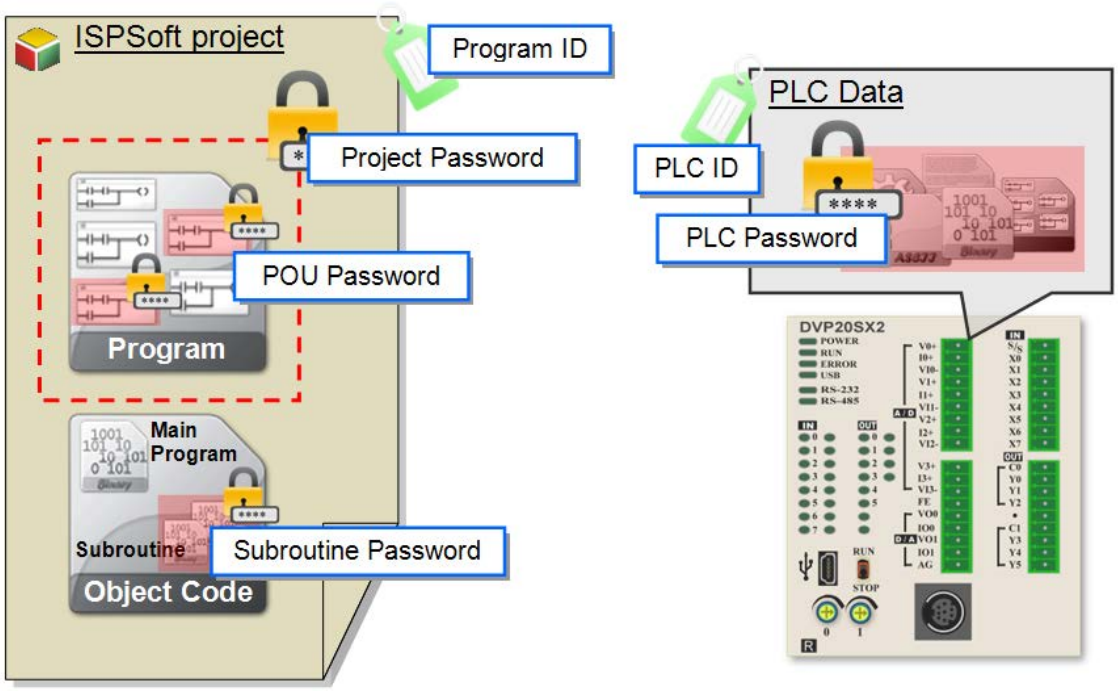
Chapter 15 Password Management and Data Protection

Table of Contents

15.1 Password Protection Mechanisms Provided by ISPSoft 15-2
15.2 Program ID and PLC ID 15-3
15.2.1 Setting and Unlocking a Program ID 15-3
15.2.2 Setting and Unlocking a PLC ID 15-3
15.3 Project Password and PLC Password 15-4
15.3.1 Setting and Unlocking a Project Password 15-4
15.3.2 Setting and Unlocking a PLC Password 15-6
15.3.3 Making a Project Password and a PLC Password Become the Same 15-7
15.4 POU Password 15-8
15.5 Subroutine Password 15-9
15.5.1 Introduction of a Subroutine Passwords 15-9
15.5.2 Setting and Unlocking a Subroutine Password 15-10
15.6 Other Password Protection functions and Data Protection Functions 15-11
15.6.1 Inactivating the Uploading of the Program from a DVP Series PLC 15-11
15.6.2 Setting Read-only Device Ranges in a DVP Series PLC 15-12
15.6.3 Setting a TC-01 Password Key (for a DVP Series PLC) 15-12

15.1 Password Protection Mechanisms Provided by ISPSoft

To protect the programs or the core techniques developed by users, ISPSoft provides multiple protection mechanisms for the projects developed by the users. The password protection mechanisms provided by ISPSoft are shown below. Users can set or unlock the passwords, and adopt different data protection mechanisms according to their needs.



15

Password type		Description
ISPSoft project	Program ID	This is the first authentication information which needs to be checked on. If users want to download a project to a PLC, the program ID set in the project must be the same as the PLC ID set in the PLC. A specific project can only be downloaded to a specific PLC.
	Project password	The project password set in a project is used to protect the program in the project. If users want to open the POUs in a project protected by a project password, the system will require the users to type the correct password.
	POU password	The POU password set in a POU is used to protect the POU. Users can set a password in a POU to protect the source code or the core technique in the POU.
	Subroutine password	A subroutine password set in a subroutine is used to protect the subroutine. It prevents the programs in the function blocks from being analyzed after the programs are uploaded. This function only supports DVP series PLCs. AH500 series CPU modules adopts another mechanism.
PLC	PLC ID	This is the first authentication information which needs to be checked on. If users want to download a project to a PLC, the program ID set in the project must be the same as the PLC ID set in the PLC. A specific project can only be downloaded to a specific PLC.
	PLC password	The PLC password set in a PLC is used to protect the data in the PLC. If users want to download the data protected by a PLC password to a PLC, or upload the data protected by a PLC password from a PLC, they have to type the correct password.

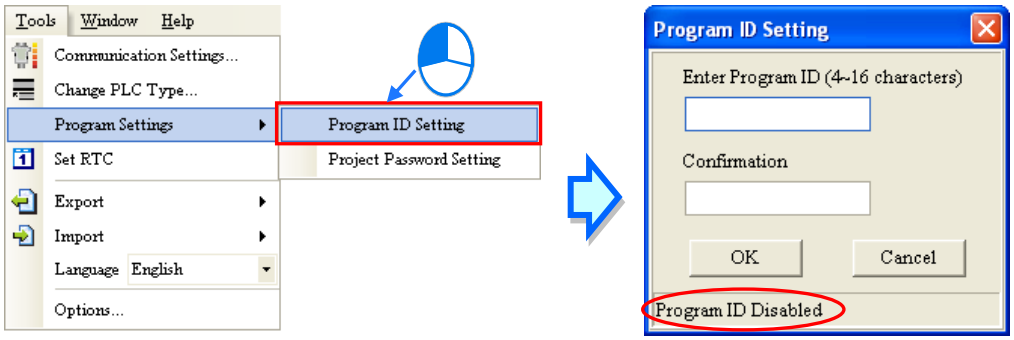
15.2 Program ID and PLC ID

An ID is the first authentication information which needs to be checked on. If users want to download a project to a PLC, the program ID set in the project must be the same as the PLC ID set in the PLC. The project which is protected by a program ID can only be downloaded to a specific PLC, and the PLC ID set in the PLC must be the same as the program ID.A specific project can only be downloaded to a specific PLC.

If users want to upload a project to a PLC protected by a PLC ID, the system will ask the users to type the correct ID before the project is uploaded. If users want to open a project protected by a program ID, the system will ask the users to type the correct ID before the project is opened. As a result, an ID is used for access approval to gain access to data.

15.2.1 Setting and Unlocking a Program ID

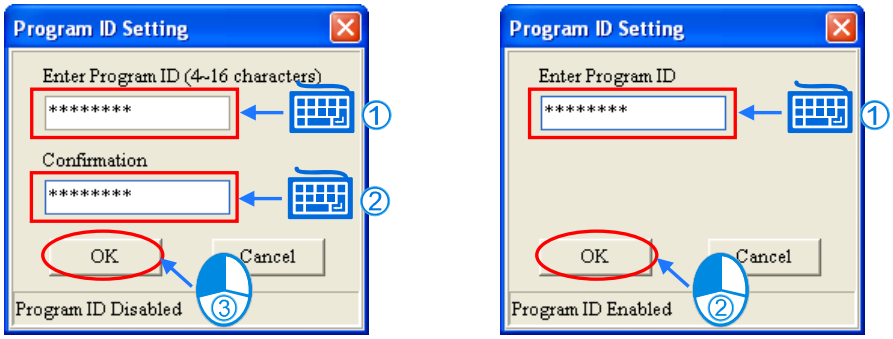
After users click the **Tools** menu, point to **Program Settings**, and click **Program ID Setting**, the **Program ID Setting** window will appear. The message at the bottom of the window indicates whether the project is protected by a program ID.



If the project is not protected by a program ID, there will be two boxes used to set a program ID in the **Program ID Setting** window. If the users want to set a program ID, they have to type the same password in the two boxes, and click **OK**.

If the project is protected by a program ID, there will be one box used to unlock the program ID in the **Program ID Setting** window. If the users want to unlock the program ID, they can type the correct ID in the box, and click **OK**.

If the users want to set a program ID, they can only type English letters, numerical digits, and spaces in the window. A program ID is case-sensitive. Besides, the limit on the number of characters which can be typed depends on the model selected. The users have to type a program ID according to the hint in the window. After the users set a program ID, the project will be protected by the program ID. The window will be closed after the users click **Cancel**.

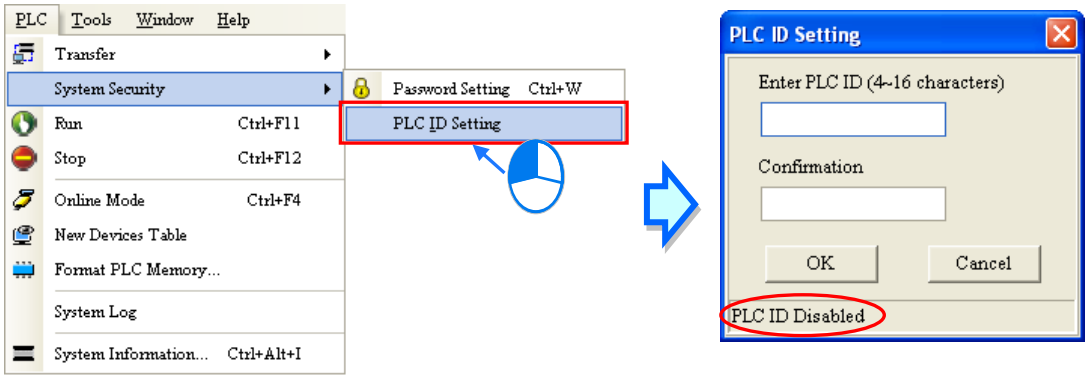


15.2.2 Setting and Unlocking a PLC ID

Before users set a PLC ID, they have to make sure that ISPSoft is connected to the PLC normally. Please refer to section 2.4 for more information.

After users click the **PLC** menu, point to **System Security**, and click **PLC ID Setting**, ISPSoft will connect to the PLC, and read the information from the PLC. If the reading of the information from the

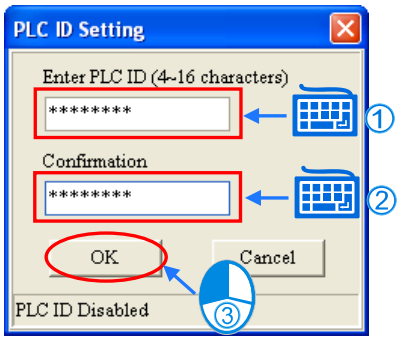
PLC is successful, ISPSoft will open the **PLC ID Setting** window. The message at the bottom of the window indicates whether the PLC is protected by a program ID.

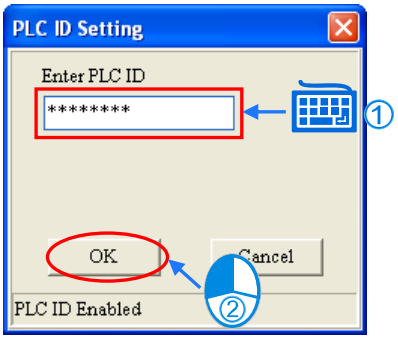


If the PLC is not protected by a PLC ID, there will be two boxes used to set a PLC ID in the **PLC ID Setting** window. If the users want to set a PLC ID, they have to type the same password in the two boxes, and click **OK**.

If the PLC is protected by a PLC ID, there will be one box used to unlock the PLC ID in the **PLC ID Setting** window. If the users want to unlock the PLC ID, they can type the correct ID in the box, and click **OK**.

If the users want to set a PLC ID, they can only type English letters, numerical digits, and spaces in the window. A PLC ID is case-sensitive. Besides, the limit on the number of characters which can be typed depends on the model selected. The users have to type a PLC ID according to the hint in the window. After the users set a PLC ID, the PLC will be protected by the PLC ID. The window will be closed after the users click **Cancel**.





15.3 Project Password and PLC Password

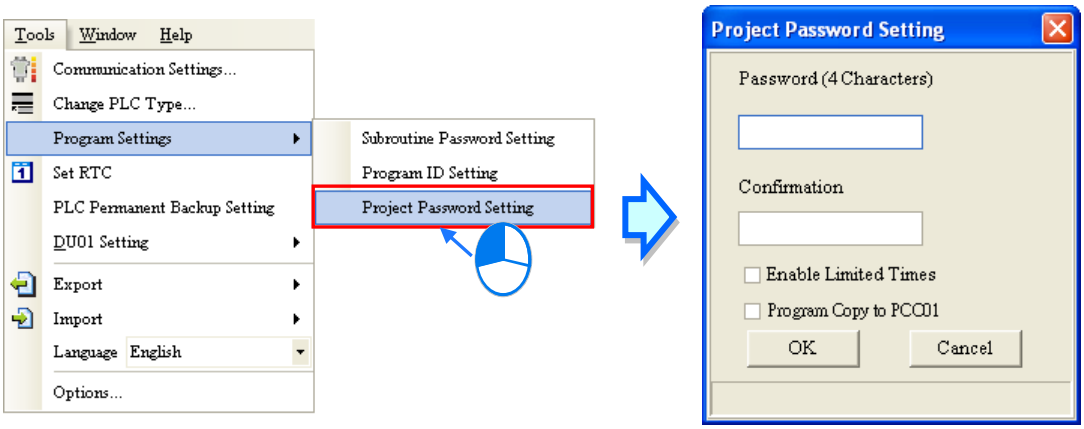
The project password set in a project is used to protect the program in the project. If users want to open the POUs in a project protected by a project password, the system will require the users to type the correct password. Besides, the users can limit the total number of password guesses that can be made. If the number of password guesses that is made is larger than the total number of password guesses that can be made, the project will be closed automatically.

The PLC password set in a PLC is used to protect the data in the PLC. If users want to download or upload a project, the system will ask the users to type the correct password. Besides, the users can limit the total number of password guesses that can be made. If the number of password guesses that is made is larger than the total number of password guesses that can be made, the PLC will be locked, and the data can not be downloaded/uploaded. To unlock the PLC, the users have to restore the PLC to the default setting.

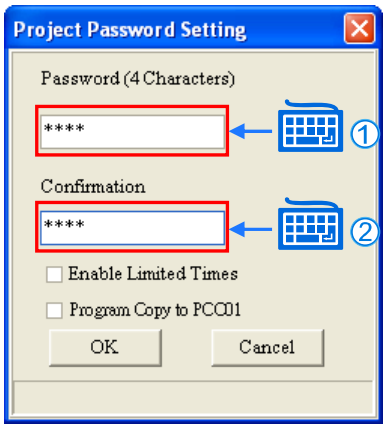
15.3.1 Setting and Unlocking a Project Password

After users click the **Tools** menu, point to **Program Settings**, and click **Project Password Setting**, the project password setting window will appear. The message at the bottom of the window indicates whether the project is protected by a project password. If the project opened is a project for

a DVP series PLC, the **Program Copy to PCC01** checkbox will appear in the **Project Password Setting** window.

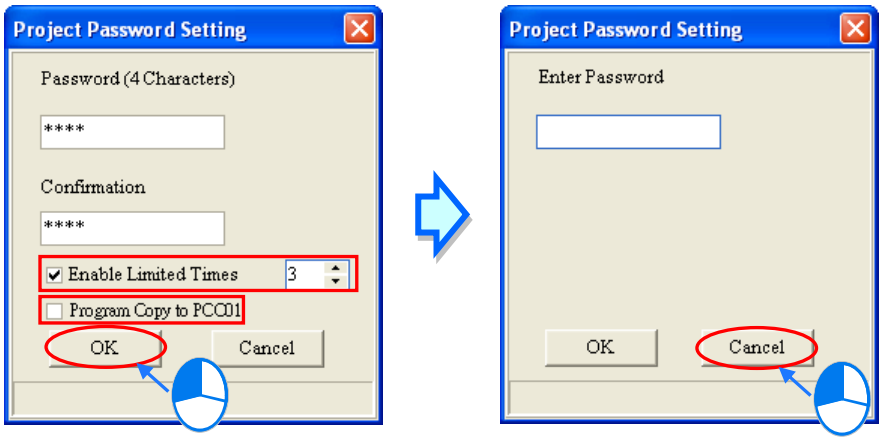


If the project is not protected by a project password, there will be two boxes used to set a project password in the **Project Password Setting** window. If the users want to set a project password, they have to type the same password in the two boxes, and click **OK**. The users can only type English letters, numerical digits, and spaces in the window. A project password is case-sensitive. Besides, the limit on the number of characters which can be typed depends on the model selected. The users have to type a project password according to the hint in the window.

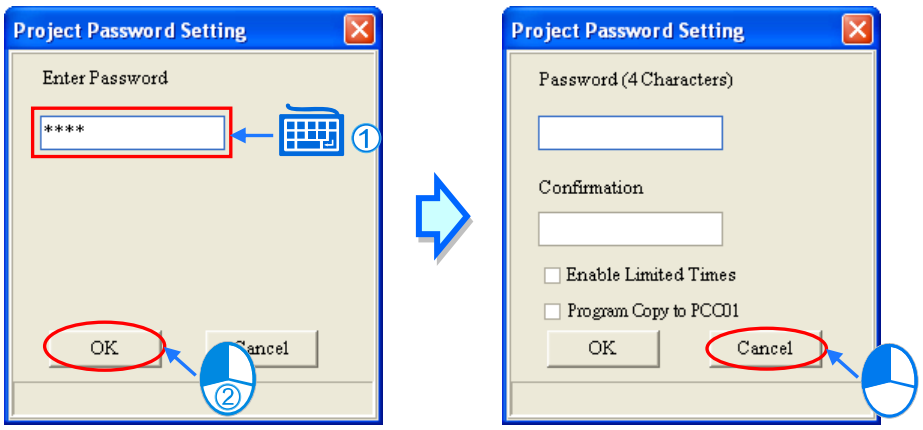


If the users want to limit the total number of password guesses that can be made, they can select the **Enable Limited Times** checkbox, and select a number in the box at the right side of the checkbox. If users want to copy the program into DVPPCC01, they can select the **Program Copy to PCC01** checkbox. (Please refer the additional remark below for more information.)

After the users set a project password, the project will be protected by the project password. The window will be closed after the users click **Cancel**.



If the project is protected by a project password, there will be one box used to unlock the project password in the **Project Password Setting** window. If the users want to unlock the project password, they can type the correct password in the box, and click **OK**. After the project password is unlocked successfully, the project will not protected by the project password. The window will be closed after the users click **Cancel**.



15

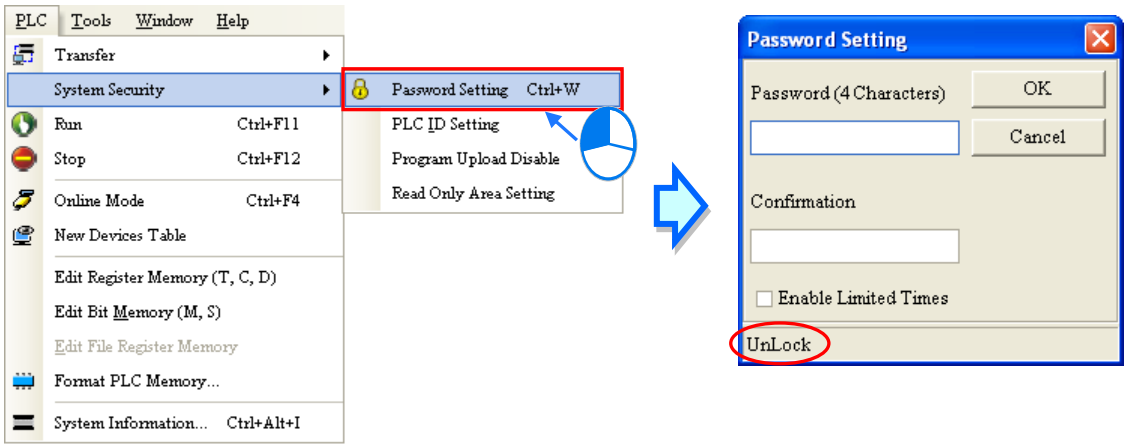
Additional remark

DVPPCC01 is a data backup memory card for DVP series PLCs. If the **Program Copy to PCC01** checkbox in the **Project Password Setting** window is selected, the PLC connected to ISPSoft will be in a status after the project is downloaded to the PLC. If DVPPCC01 is inserted into the PLC when the PLC is in that status, the project downloaded to the PLC will be backed up on DVPPCC01. However, if the PLC receives another communication command before DVPPCC01 is inserted, the program will not be copied into DVPPCC01. This function supports DVP-ES2 series PLCs. Please refer to the related user manuals or technical documents for more information about the firmware versions which support this function and the usage of DVPPCC01.

15.3.2 Setting and Unlocking a PLC Password

Before users set a PLC password, they have to make sure that ISPSoft is connected to the PLC normally. Please refer to section 2.4 for more information.

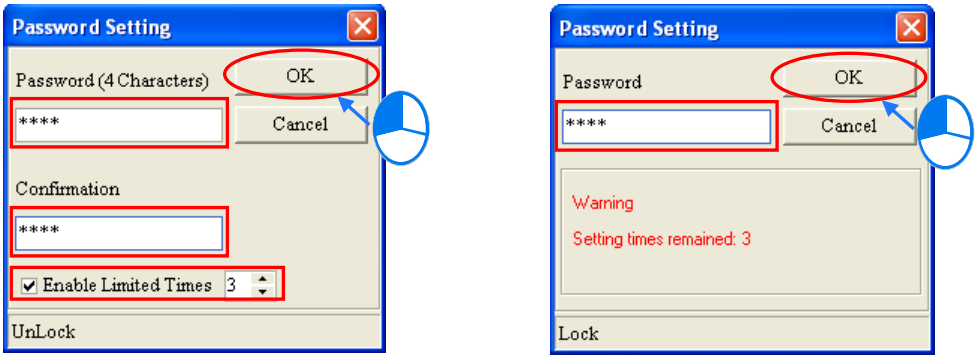
After users click the **PLC** menu, point to **System Security**, and click **Password Setting**, ISPSoft will connect to the PLC, and read the information from the PLC. If the reading of the information from the PLC is successful, ISPSoft will open the **Password Setting** window. The message at the bottom of the window indicates whether the data in the PLC is protected by a PLC password.



If the data in the PLC is not protected by a PLC password, there will be two boxes used to set a PLC password in the **Password Setting** window. If the users want to set a PLC password, they have to type the same password in the two boxes, and click **OK**. If the users want to limit the total number of password guesses that can be made, they can select the **Enable Limited Times** checkbox, select a number in the box at the right side of the checkbox, and click OK.

If the data in the PLC is protected by a PLC password, there will be one box used to unlock the PLC password in the **Password Setting** window. If the users want to unlock the PLC password, they can type the correct password in the box, and click **OK**.

The users can only type English letters, numerical digits, and spaces in the window. A project password is case-sensitive. Besides, the limit on the number of characters which can be typed depends on the model selected. The users have to type a project password according to the hint in the window. After the users set a PLC password, the data in the PLC will be protected by the PLC password. The window will be closed after the users click **Cancel**.



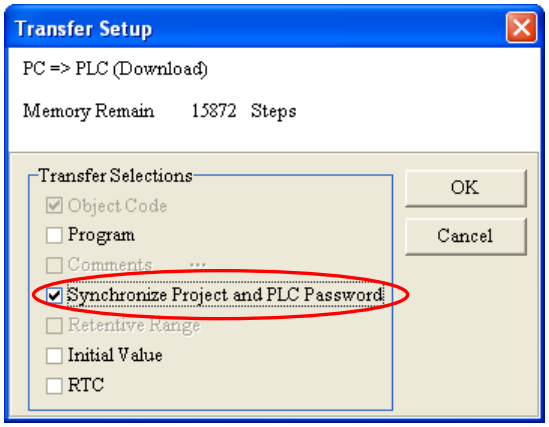
15.3.3 Making a Project Password and a PLC Password Become the Same

ISPSoft provides a convenient function for users. During the downloading/uploading of a project, users can make the project password and the PLC password become the same.

- **Downloading a project**

  If users select the **Synchronize Project and PLC Password** checkbox in the **Transfer Setup** window, the PLC password will become the same as the project password after the project is downloaded.

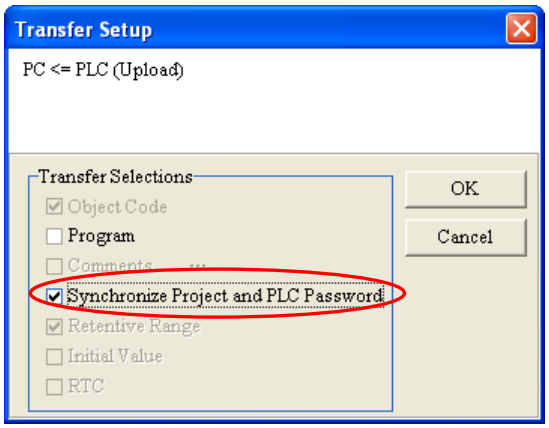
  If the users do not set a project password, the system will ask the users to set a project password after the users select the **Synchronize Project and PLC Password** checkbox in the **Transfer Setup** window. Besides, if the users open an old project which is protected by a project password, the **Synchronize Project and PLC Password** checkbox in the **Transfer Setup** window will be selected.



- **Uploading a project**

  If users select the **Synchronize Project and PLC Password** checkbox in the **Transfer Setup** window, the project password will become the same as the PLC password after the object code is uploaded. However, if the users do not set a PLC password in the PLC, no project password will be uploaded. Besides, the object code in AH500 series CPU modules can not be uploaded independently, and therefore this function has little significance to the AH500 series CPU modules.

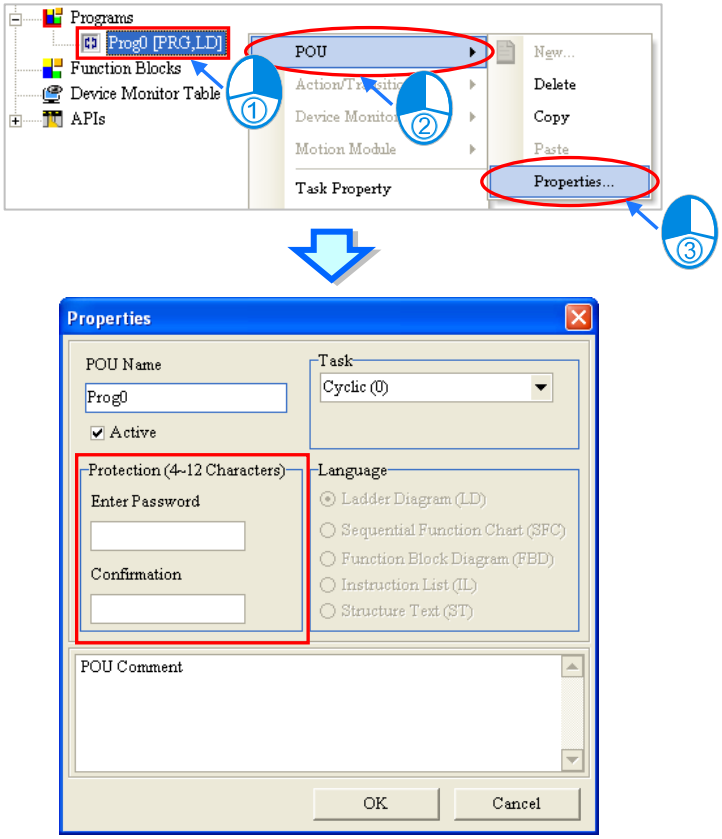
  If the **Program** checkbox in the **Transfer Setup** window is selected, the **Synchronize Project and PLC Password** checkbox can not be selected, and the project password uploaded will be the same as the project password which protects the original program (source code.)



15.4 POU Password

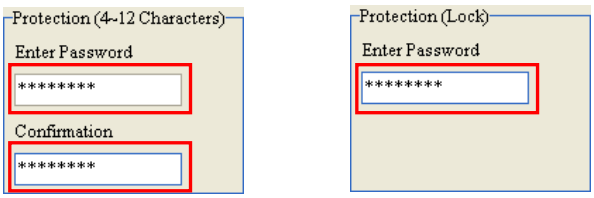
ISPSoft allows users to set a password in a POU. If users want to open a POU protected by a password, the system will ask the users to type the correct password. The password set in a POU protects the source code or the core technique in the POU. The setting of a POU password and the unlocking of a POU password are described below.

Right-click a POU in the project management area, point to **POU** on the context menu, and click **Properties…**. The **Protection** section in the **Properties** window indicates whether the POU is protected by a POU password.



If the POU is not protected by a POU password, there will be two boxes used to set a POU password in the **Protection** section. If the users want to set a POU password, they have to type the same password in the two boxes. The users can only type English letters, numerical digits, and marks in the window. A POU password is case-sensitive. After the setting of a POU password is complete, the users can click **O**K.

If the POU is protected by a POU password, there will be one box used to unlock the POU password in the **Protection** section. If the users want to unlock the POU password, they can type the correct password in the box, and click **OK**.



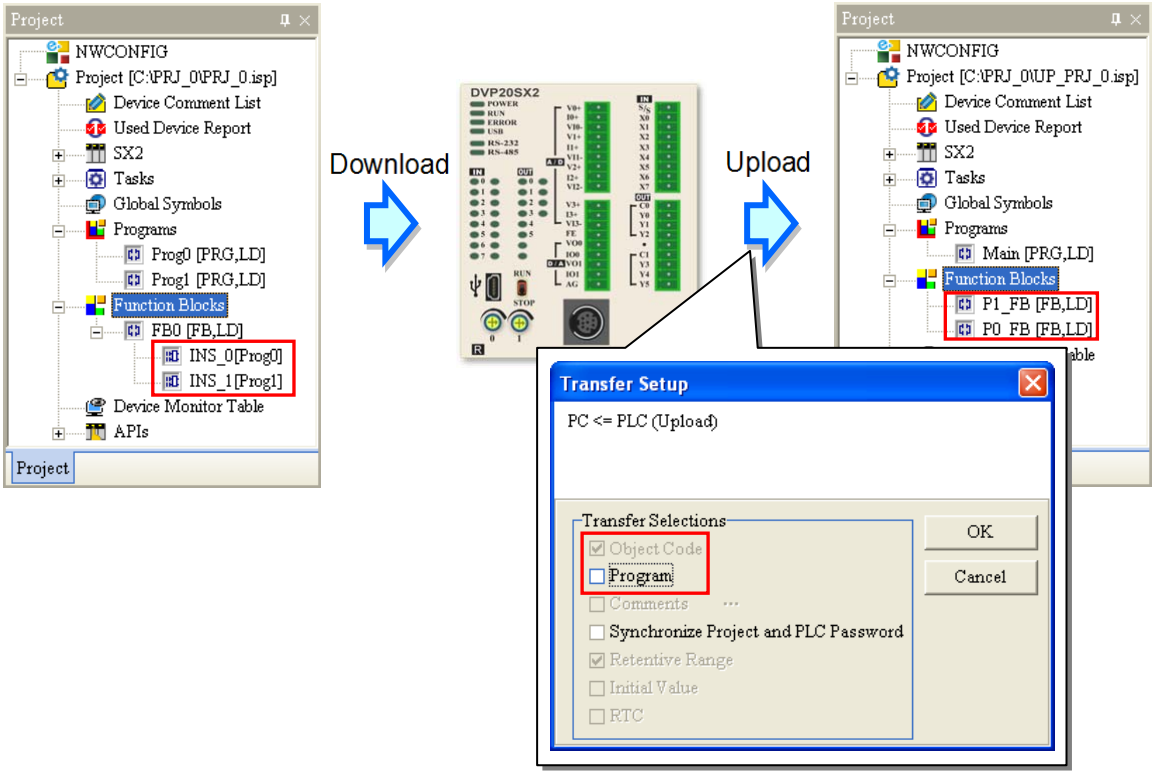
*** Users can set a POU password when they create a POU. Please refer to section 5.4.1 for more information.**

15.5 Subroutine Password

15.5.1 Introduction of a Subroutine Passwords

After the program in a project for a DVP series PLC is compiled, the object code produced by the system may include a main program, a general subroutine, and an interrupt subroutine. The subroutine is transformed from the source code in the POUs of the function block type in the project. If the **Program** checkbox in the **Transfer Setup** window is not selected, and the **Object Code** checkbox in the **Transfer Setup** window is selected, the general subroutine which is a part of the object code will be transformed into POUs of the function blocks type in ISPSoft after the object

code is uploaded. Owing to the fact that a POU password protects the source code in a POU, the POUs transformed from the object code are not protected by POU passwords. Users can open the POUs of the function block type at will, and analyze the contents of the POUs.



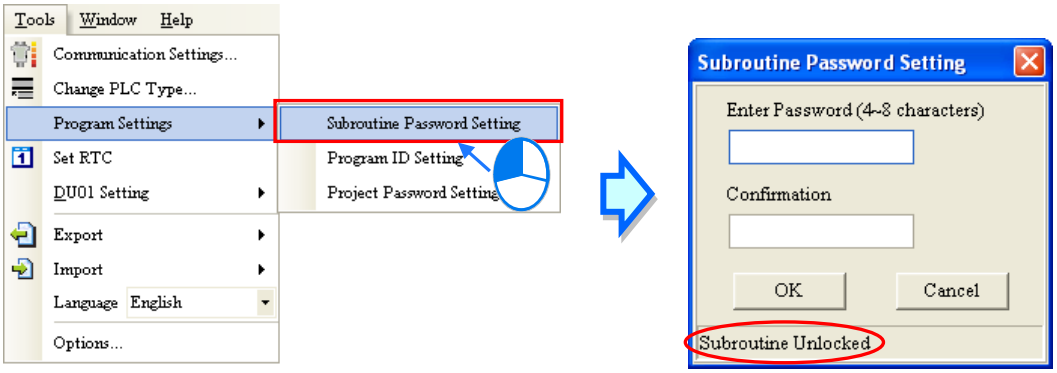
15

Generally speaking, developers pack core techniques or low-level programs in function blocks. If object code is gotten in the way described above, all the techniques may be analyzed. To prevent this situation, the users can set a subroutine password before the project is downloaded. If the users set a subroutine password, the subroutine password will become the POU passwords which protect the POUs of the function block type transformed from the object code which is uploaded. As a result, if the users want to open the POUs of the function block type, the system will ask the users to type the correct passwords.

***. The object code in an AH500 series CPU module can not be uploaded independently, and therefore there is no problem of information leakage, and users do not need to set a subroutine password.**

15.5.2 Setting and Unlocking a Subroutine Password

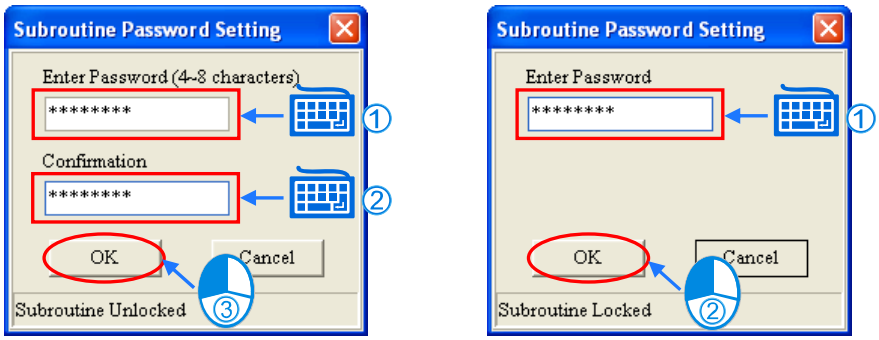
After users click the **Tools** menu, point to **Program Settings**, and click **Subroutine Password Setting**, the **Subroutine Password Setting** window will appear. The message at the bottom of the window indicates whether the subroutine is protected by a subroutine password.



If the subroutine is not protected by a subroutine password, there will be two boxes used to set a subroutine password in the **Subroutine Password Setting** window. If the users want to set a subroutine password, they have to type the same password in the two boxes, and click **OK**.

If the subroutine is protected by a subroutine password, there will be one box used to unlock the subroutine password in the **Subroutine Password Setting** window. If the users want to unlock the subroutine password, they can type the correct password in the box, and click **OK**.

The users can only type English letters, numerical digits, and spaces in the window. A POU password is case-sensitive. After the users set a subroutine password, the subroutine will be protected by the subroutine password. The window will be closed after the users click **Cancel**.



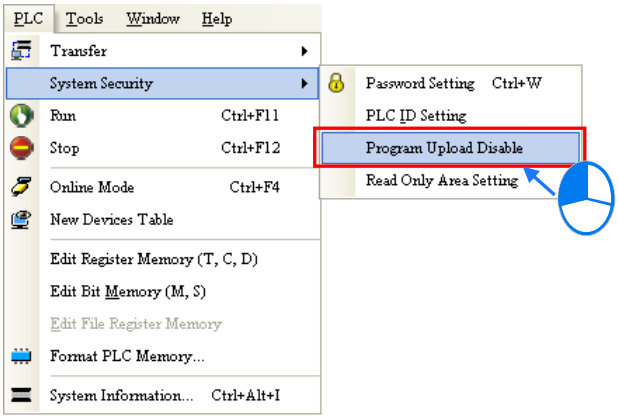
15

15.6 Other Password Protection functions and Data Protection Functions

15.6.1 Inactivating the Uploading of the Program from a DVP Series PLC

Users can inactivate the uploading of the program from a DVP series PLC. If this function is used, the uploading of the program from the DVP series PLC will be prohibited. The uploading of the program from the DVP series PLC will be allowed only after the PLC is restored to the default setting. Besides, if a PLC password is set in the PLC, it has to be unlocked before this function is used. The data in the PLC do not need to be protected by a PLC password after the uploading of the program from the PLC is inactivated, and therefore the users can not set a PLC password.

If the users want to inactivate the uploading of the program from the DVP series PLC, they can click the **PLC** menu, point to **System Security**, and click **Program Upload Disable**. After the uploading of the program from the PLC is inactivated, the data in the PLC can not be uploaded. As a result, the users have to decide whether they need to back the data in the PLC up in advance.

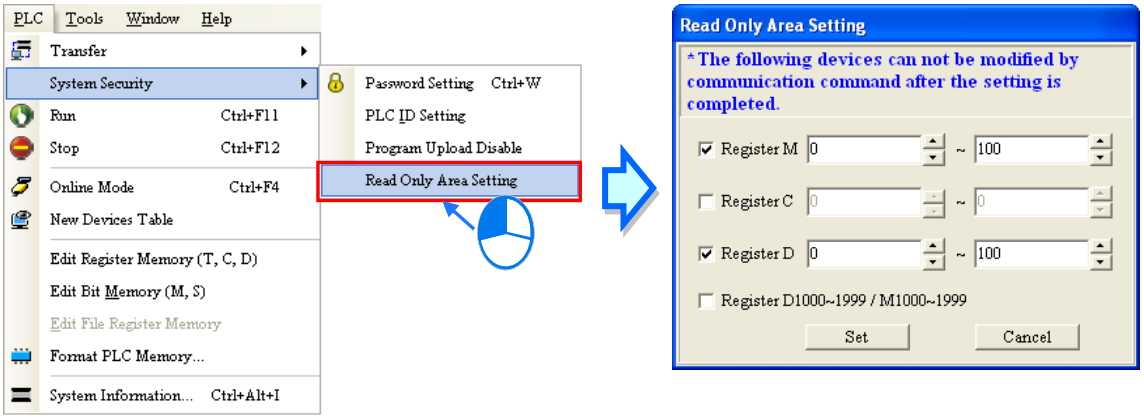


***. Before the function is used, users have to check whether the model selected and the firmware version of the model support this function. Please refer to operation manuals or technical documents for more information.**

15.6.2 Setting Read-only Device Ranges in a DVP Series PLC

The values stored in the read-only devices in a PLC can not be altered at will. After users set read-only device ranges in a DVP series PLC, they can not alter the values in the devices in these ranges through communication. If the users want to alter the values in the devices in these ranges, they have to restore the PLC to the default setting.

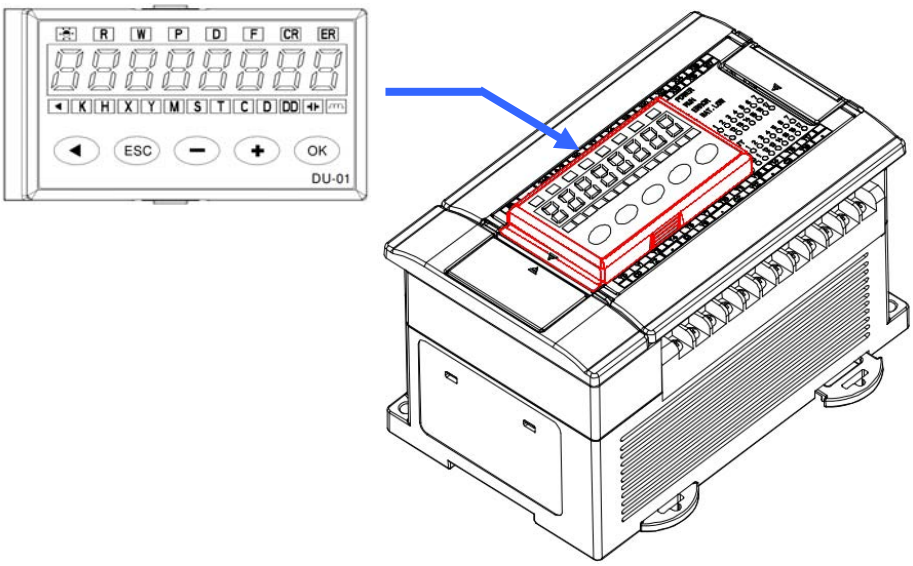
If the users want to set read-only device ranges in the DVP series PLC, they can click the **PLC** menu, point to **System Security**, click **Read Only Area Setting**. In the **Read Only Area Setting** window, the users have to select device types, and set read-only device ranges. After the setting is complete, the users can click **OK** in the **Read Only Area Setting** window.



***. Before the function is used, users have to check whether the model selected and the firmware version of the model support this function. Please refer to operation manuals or technical documents for more information.**

15.6.3 Setting a TC-01 Password Key (for a DVP Series PLC)

Users can set a TC-01 password key in DVPDU-01, a digital setup display. A drawing of DVPDU-01 is shown below. DVPDU-01 provides a simple operation interface for a DVP series PLC.

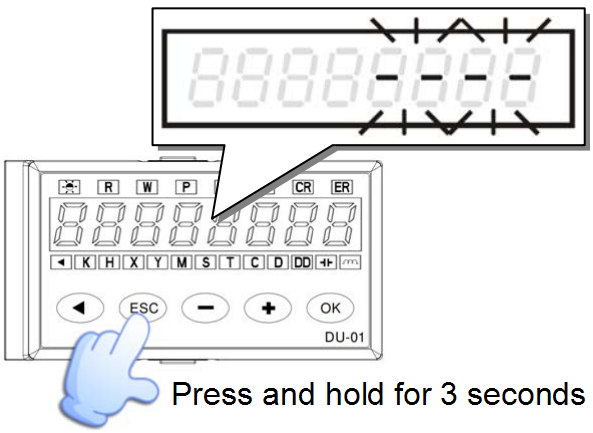


If users want to monitor the devices in a DVP series PLC through DVPDU-01, they can select the TS-01 mode in DVPDU-01. If users want to read data from a DVP series PLC through DVPDU-01, or write data into a DVP series PLC through DVPDU-01, they can select the TC-01 mode in DVPDU-01. If users want to read data from a DVP series PLC in which a PLC password is set through DVPDU-01, or write data into a DVP series PLC in which a PLC password is set through DVPDU-01, they can set a TC-01 password key which is the same as the PLC password in DVPDU-01. After the system makes sure that the TC-01 password key set in DVPDU-01 is the same as the PLC password, the data will be read from the DVP series PLC, or the data will be written into the DVP series PLC. Besides, when users read data from a DVP series PLC in which no PLC password is set through DVPDU-01, or write data into a DVP series PLC in which no PLC password is set through DVPDU-01, the system sets a PLC password in the DVP series PLC. The PLC password set in the DVP series PLC is the same as the TC-01 password key set in DVPDU-01.

15

The setting of a TC-01 password key and the unlocking of a TC-01 password key are described below. Users have to make sure that DVPDU-01 is on standby before they set a TC-01 password. If an error occurs in DVPDU-01, or DVPDU-01 is in a process, the users have to eliminate the error, or complete the process. Please refer to DVPDU-01 User Manual for more information.

(1) After the users press **ESC** on DVPDU-01 for three seconds, the password setting screen will appear.

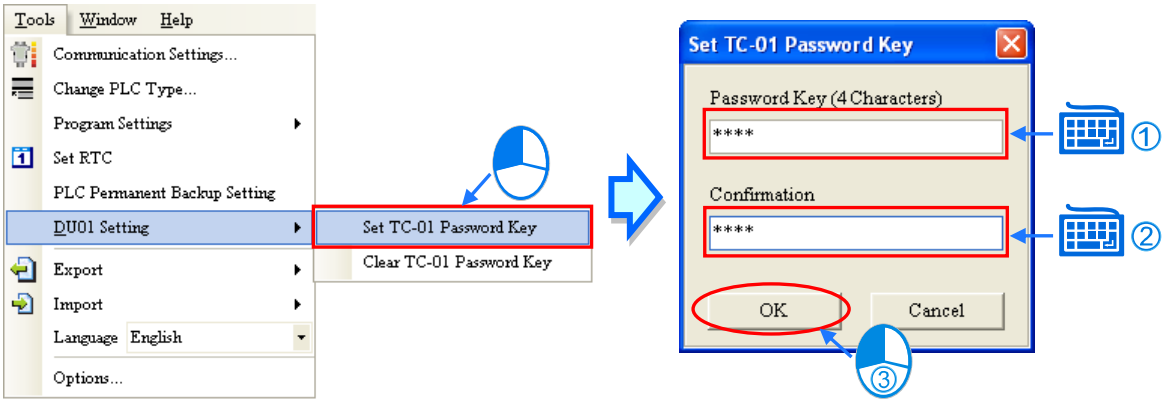


(2) Set a TC-01 password key by means of ISPSoft, or unlock the TC-01 password key by means of ISPSoft.

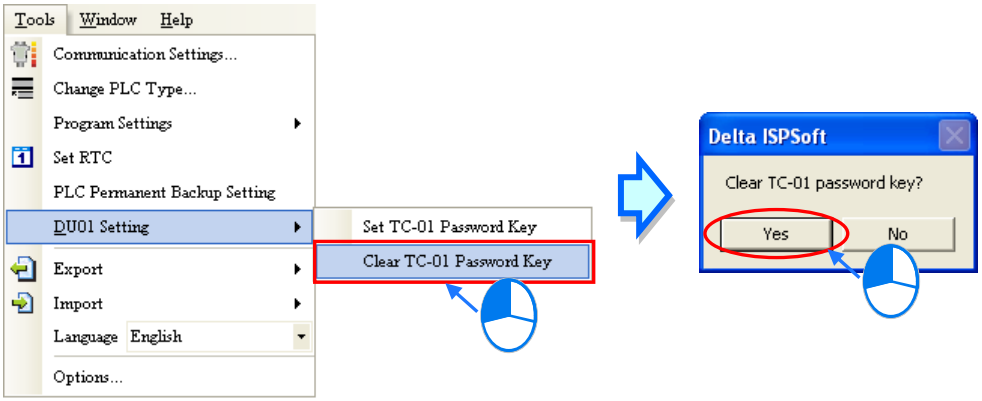
- **Setting a TC-01 password key**

  After users click the **Tools** menu, point to **DU01 Setting**, and click **Set TC-01 Password Key**, the **Set TC-01 Password Key** window will appear. The users have to type the same

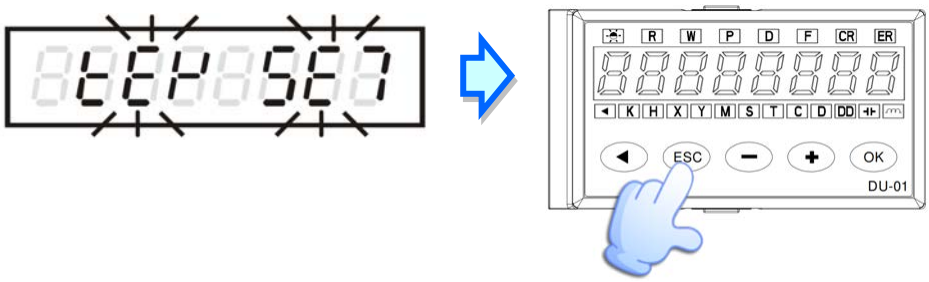
password in the two boxes in the window. After the setting of a TC-01 password key is complete, the users can click **OK** in the window.



- **Unlocking the TC-01 password key**
  Click the **Tools** menu, point to **DU01 Setting**, click **Clear TC-01 Password Key**, and click **Yes** in the **Delta ISPSoft** window.



(3) After the users set a TC-01 password key, or unlock the TC-01 password key, **KEY SET** will be displayed on DVPDU-01. Press **ESC** on DVPDU-01.



Chapter 16 Network Configuration and Data Exchange

Table of Contents

16.1 Network Configuration Tool—NWCONFIG 16-3
16.1.1 Introduction of NWCONFIG 16-3
16.1.2 Basic Knowledge 16-3
16.1.3 Communication Setting in NWCONFIG 16-5
16.1.3.1 Connection Mechanism in NWCONFIG 16-6
16.1.3.2 Setting Communication Parameters 16-7
16.1.4 Workflow 16-8
16.2 Creating a Network Architecture 16-12
16.2.1 Deploying Nodes 16-12
16.2.2 Connecting to a Network 16-16
16.2.3 Adjusting or Deleting Devices or Networks 16-20
16.2.4 Setting the Attributes of a Node/Network 16-23
16.2.5 Hiding/Displaying Devices or Networks 16-27
16.2.6 Correct Network Architecture 16-30
16.2.7 Downloading Routing Tables 16-33
16.2.8 Testing Routing 16-35
16.3 Constructing a PLC Link 16-37
16.3.1 Opening the PLC Link Table Editor Window 16-38
16.3.2 Designating a Port as a Master Station (Step 1) 16-39
16.3.3 Setting Communication Parameters (Step 2) 16-40
16.3.4 Creating a Data Exchange Table (Step 3) 16-42
16.3.4.1 Introduction of a Data Exchange Table 16-42
16.3.4.2 Setting a Data Exchange Group 16-44
16.3.4.3 Consistency Between a Data Exchange Table and the Network Created in NWCONFIG 16-47
16.3.4.4 Managing a Data Exchange Table 16-49
16.3.5 Monitoring a PLC Link 16-50
16.3.6 Importing Points About Constructing a PLC Link 16-55
16.4 Constructing an Ether Link 16-56
16.4.1 Introduction of an Ether Link 16-56
16.4.2 Opening the Ether Link Configuration Window 16-57
16.4.3 Creating and Managing a Data Exchange Table 16-59
16.4.4 Node List and Display Area 16-62
16.4.5 Start Mode of an Ether Link 16-64
16.4.6 Downloading the Parameters Related to an Ether Link 16-66
16.4.7 Uploading the Parameters Related to an Ether Link 16-68
16.4.8 Deleting Asynchronous Device 16-70
16.4.9 Enabling/Disabling the Online Monitoring Function 16-71
16.4.9.1 Enabling a Monitoring Function 16-71
16.4.9.2 Monitoring Statuses 16-73
16.4.9.3 Disabling a Monitoring Function 16-73
16.4.10 Starting/Stopping the Execution of an Ether Link Online 16-75

16.4.10.1 Starting the Execution of an Ether Link 16-75
16.4.10.2 Stopping the Execution of an Ether Link..... 16-76
16.4.11 Monitoring Table and Error Log 16-78
16.5 Managing and Applying NWCONFIG 16-79
16.5.1 Saving Parameters and Printing a Network Framework..... 16-79
16.5.2 Downloading Parameters 16-80
16.5.2.1 Introduction of Parameters..... 16-80
16.5.2.2 Description of Downloading Parameters 16-81
16.5.3 Using Routing in ISPSoft..... 16-82

16.1 Network Configuration Tool–NWCONFIG

16.1.1 Introduction of NWCONFIG

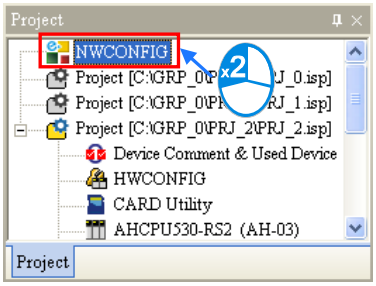
NWCONFIG is the network configuration tool provided by ISPSoft. Users can configure the network in a project and set up a mechanism for data exchange through NWCONGIF. The functions of NWCONFIG are listed below. They will be described in the following sections.

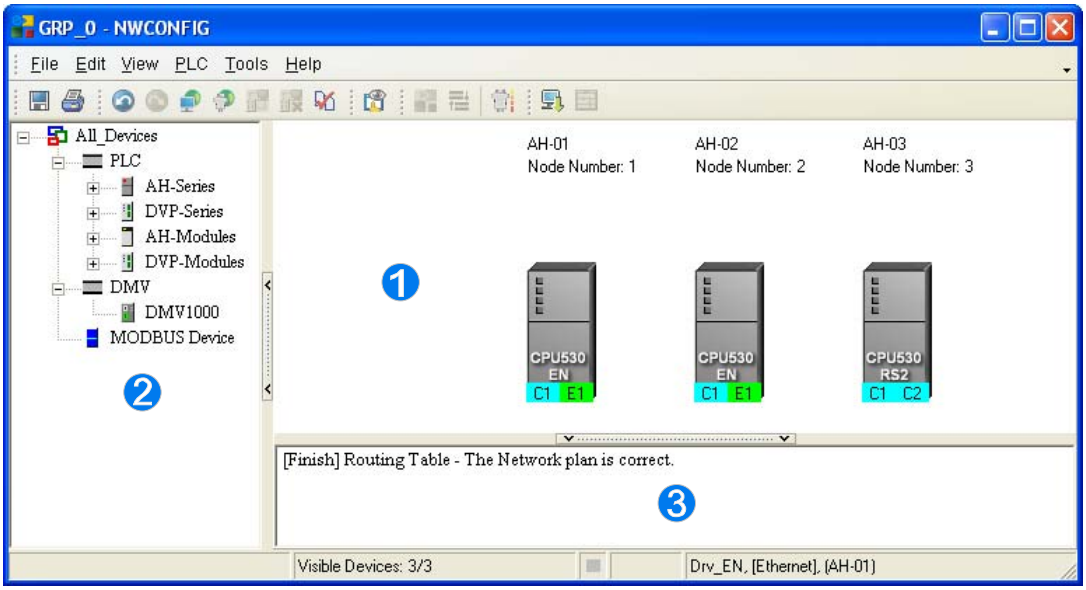
(a) Creating networks in a project, and selecting paths along which data is sent

(b) Performing data exchange through an RS-485 cable–PLC Link

(c) Performing data exchange through Ethernet–Ether Link

NWCONFIG is used to create a network framework for projects, and therefore it is at the top of the project management area. If users want to start NWCONGIF, they can double-click **NWCONFIG** in the project management area.





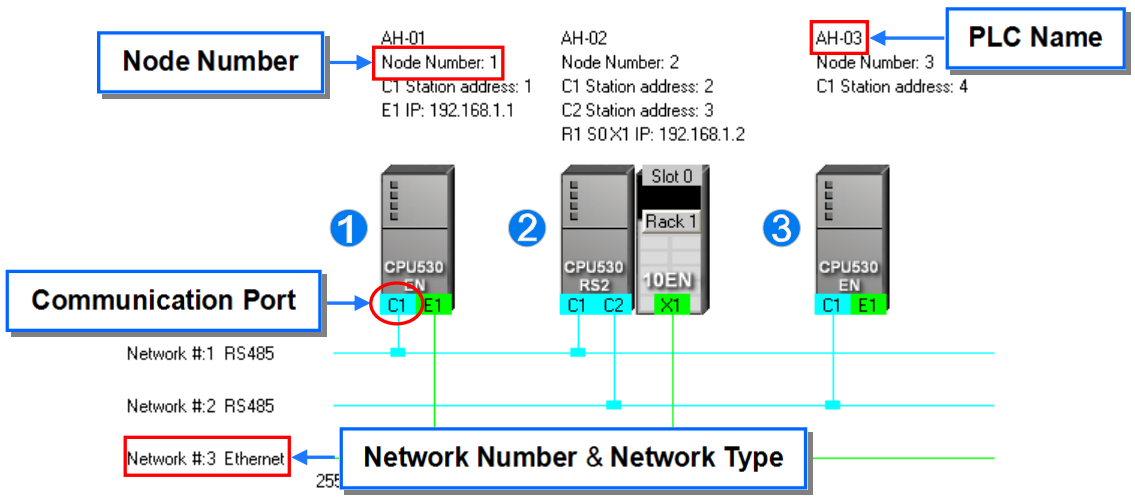
❶ **Working area**: It is a main working area. Users can create a network framework in this area.

❷ **Device list**: All the devices which can be used are listed in a catalog.

❸ **Message display area**: The messages related to operation are displayed in this area.

16.1.2 Basic Knowledge

Before creating networks, users need to have some basic knowledge. The basic knowledge is introduced in this section.



- **Device and network**

  A device is the most basic element in a network. It is a PLC, a module, or equipment defined by users. A network is a collection of devices which are interconnected. Every network is assigned a network number. There are RS-485 networks and Ethernet networks. Besides, a physical interface that a device uses to connect to a network is a port of the device. If there are more than two ports on a device, the device can connect to networks which are assigned different network numbers. Please refer to section 16.2.2 for more information about the marking of a port in NWCONFIG.

- **PLC name**

  "**AH-01**", "**AH-02**", and "**AH-03**" in the figure above are PLC names. The PLC name of an AH500 series CPU module depends on the setting in HWCONFIG. Users can identify a device in a network by means of the PLC name of the device. Please refer to section 3.3.2.1 for more information. However, the PLC name of a device which is not an AH500 series CPU module is like a comment on the device. It has little significance.

- **Node and node number**

  A node is a basic unit which can operate independently in a network. ❶ ~ ❸ in the figure above are nodes. ❷ consists of a CPU module and a network module. The network module can not operate by itself, and therefore the CPU module and the network module are regarded as one node.

  Besides, AH500 series CPU modules can forward packets and perform routing. For example, ❸ in the figure above can be monitored through ❶. Before routing is performed, users have to create paths along which data is sent, and assign node numbers to the nodes which forward the data along the paths. Only AH500 series CPU modules can be assigned node numbers, and the node number of a node in a network can not be the same as the node number of another node in the network. After the paths created are downloaded to the PLCs which forward the data along the paths, every PLC has its own routing table. The forwarding of the data is directed on the basis of the routing tables produced.

- **Station address**

  Users can identify a port in an RS-485 network by means of the station address of the port. The station address of a port in a network can not be the same as the station address of another port in the network. Besides, a port is assigned a station address. A port basically represents a station. If a node has several ports, the ports connected to networks must be assigned station addresses.

- **IP address and DHCP mode**

  A port in an Ethernet network is assigned an IP address. The IP address of a port in a network can not be the same as the IP address of another port in the network, and an IP address can not end with 0 or 255. If a node has several Ethernet ports, the Ethernet ports connected to an Ethernet network must be assigned IP addresses.

DHCP is a protocol for assigning dynamic IP addresses to ports in a network. If a server using DHCP assigns an IP address to a port, it assigns a dynamic IP address to the port. In NWCONFIG, the ports which are assigned dynamic IP addresses can not connect to any network.

- **Subnet mask**

  A subnet mask is a mask used to determine what subnet an IP address belongs to. The ports in a network are assigned the same subnet mask. Besides, if the devices in a network want to perform data exchange, they must be in the same domain.

- **PLC Link**

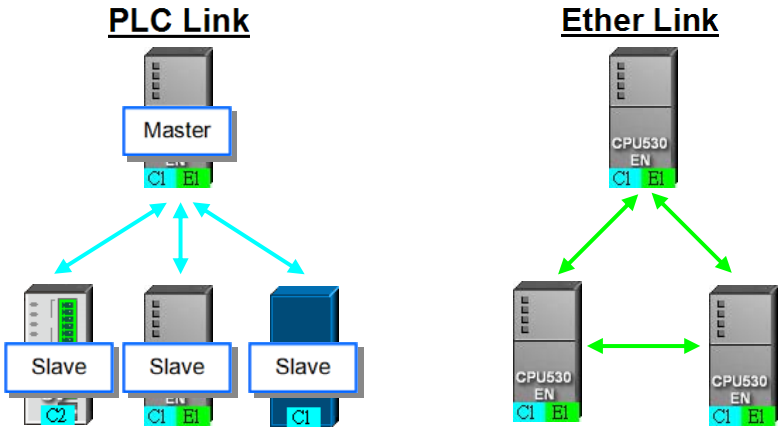
  A PLC Link is a network mechanism for data exchange performed through an RS-485 cable. If there are several nodes in an RS-485 network, users can create a mechanism for data exchange in the network. If the parameters which are set are downloaded to the PLC which functions as a master station, the system of the PLC will perform data exchange through special relays and special registers when the PLC runs.

  A PLC Link is a master/slave model. There is only one master station in an RS-485 network, and the other stations which are slave stations passively receive reading/writing commands from the master station. The slave stations can not exchange data. They have to exchange data through the master station.

- **Ether Link**

  An Ether Link is a network mechanism for data exchange performed through an Ethernet connection. If there are several nodes in an Ethernet network, users can create a mechanism for data exchange in the network, and select a start mode. If the parameters which are set are downloaded to the PLCs in the network, the systems of the PLCs perform data exchange according to the start mode selected when the PLCs run. Besides, only AH500 series CPU modules support Ether Links.

  An Ether Link is not a master/slave model. It allows a node to send reading commands which ask for data to other nodes. The nodes will send the data to the node after they receive the reading commands. Owing to the fact that a node can not send writing commands to other nodes, the use of an Ether Link is safer than the use of a PLC Link. Besides, the system automatically manages the transmission of packets through TCP/IP. Compared with a PLC Link, an Ether Link is more efficient.



***. Please refer to related books or technical documents for more information about RS-485 and Ethernet.**

16.1.3 Communication Setting in NWCONFIG

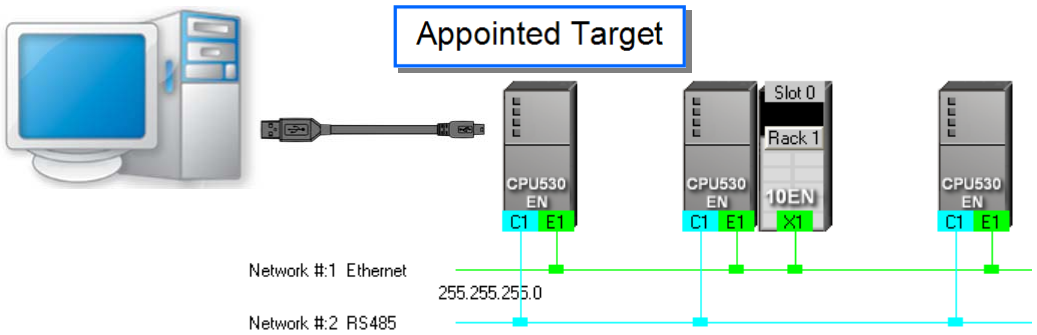
NWCONFIG is used to configure a network. When users configure a network, they have to download parameters to the nodes in the network, upload parameters from the nodes in the network, or monitor nodes in the network. The nodes in the networks created in NWCONFIG may include a device which is not the device for which the ISPSoft project is created, and therefore users have to set the communication parameters in the device. In order to help people select appropriate parameters, the communication mechanism in NWCONFIG is introduced before communication setting is described.

16.1.3.1 Connection Mechanism in NWCONFIG

In the networks created in NWCONFIG, users can download parameters to a single node or multiple nodes, upload parameters from a single node or multiple nodes, and monitor a single node or multiple nodes. Before users download parameters to a single node or multiple nodes, upload parameters from a single node or multiple nodes, or monitor a single node or multiple nodes, they have to select appropriate parameters.

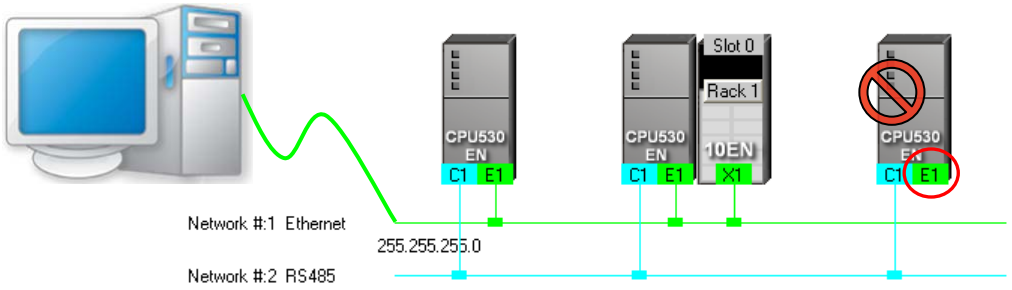
- **Single node**

  Users can download parameters to a single device, upload parameters from a single device, and monitor a single device. Before users download parameters to a single device, upload parameters from a single device, or monitor a single device, they have to make sure that the device specified is the same as the device which is actually connected to the computer.

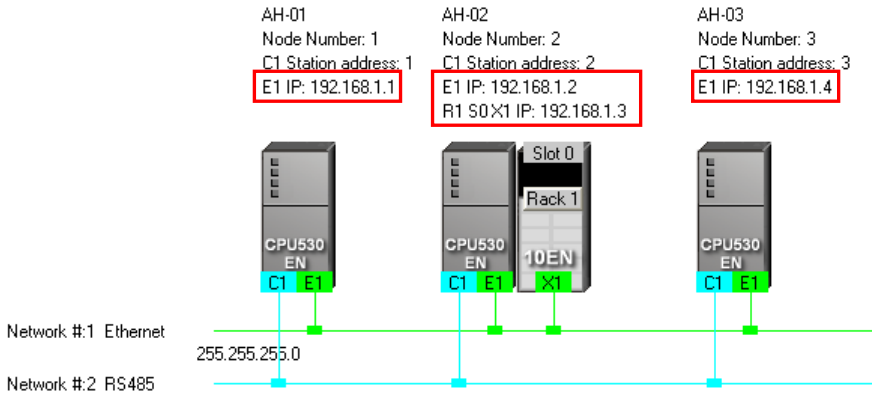


- **Multiple nodes**

  In a network, users can download parameters to multiple devices, upload parameters from multiple devices, and monitor multiple devices. Before users download parameters to multiple devices, upload parameters from multiple devices, or monitor multiple devices, they have to make sure that the devices are connected to an Ethernet network, the devices are assigned IP addresses, and the connection type that the driver uses is Ethernet.

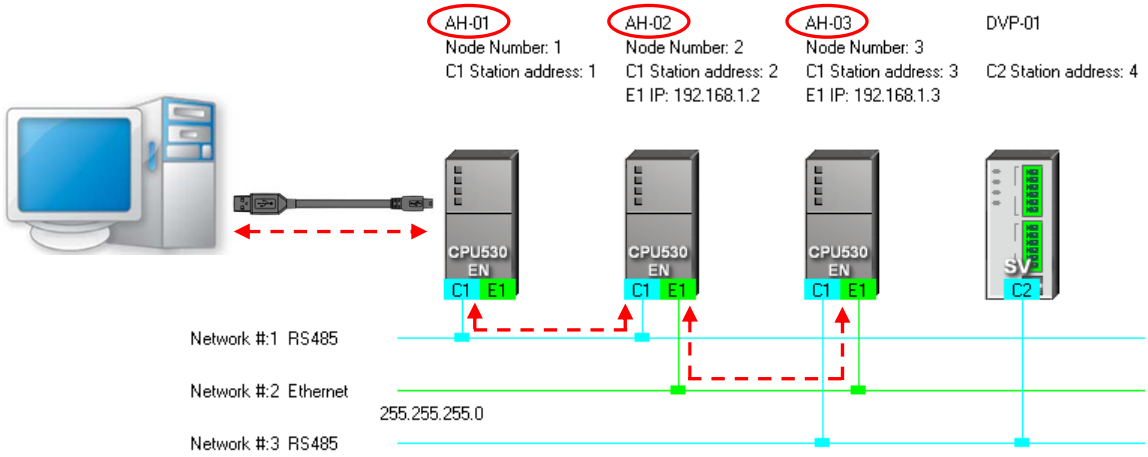


If the connection type that the driver selected uses is Ethernet, the system will carry out communication according to the IP addresses assigned to the devices in NWCONFIG. Before the communication is carried out, users have to make sure that the IP addresses actually assigned to the devices are the same as the IP addresses set in NWCONFIG, and the networks actually created are the same as the networks created in NWCONFIG. Otherwise, an error will occur if the communication is carried out.



NWCONFIG can also carries out communication through routing.

Routing is a function provided by AH500 series CPU modules. It directs packet forwarding. Packet forwarding is the relaying of packets from their source toward their destination through intermediate nodes. In the figure below, the device which actually connects to the computer is AH-01. If the computer wants to connect to AH-03, it can communicate with it through routing, and designates AH-01 as the first station. After the computer sends a command, the command is transmitted to AH-03 through AH-01 and AH-02.



The important points about routing are listed below.

(a) Users have to create networks in NWCONFIG, and download the routing tables produced to nodes in the networks. Please refer to section 16.2 for more information.

(b) AH500 series CPU modules support routing whereas DVP series PLCs and other devices do not support routing. Although DVP series PLCs and other devices can not function as intermediate nodes through which packets pass, they can function as destinations to which packets are transmitted.

16.1.3.2 Setting Communication Parameters

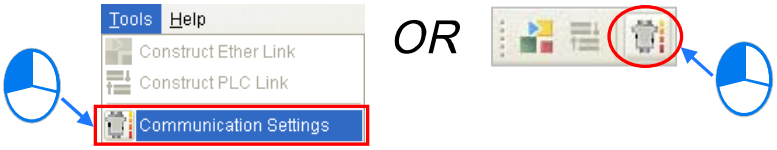
The steps of setting the communication parameters in NWCONFIG are as follows. Some prerequisites have to be considered. Please refer to section 2.4 for more information

(1) Start the communication manager COMMGR, and then create a driver in COMMGR.

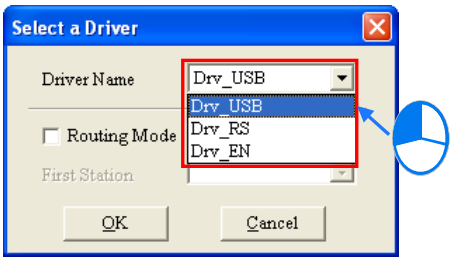
(2) If users want to download parameters to a single device, upload parameters from a single device, or monitor a single device, they have to make sure that the device specified is the same as the device which is actually connected to the computer. If the users want to download parameters to multiple devices, upload parameters from multiple devices, or monitor multiple devices, they have to make sure that the devices are connected to an Ethernet network, the devices are assigned IP addresses, the IP addresses actually assigned to the devices are the

same as the IP addresses set in NWCONFIG, and the networks actually created are the same as the networks created in NWCONFIG.

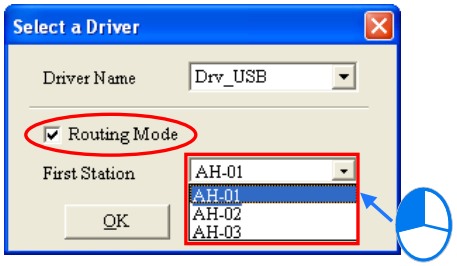
(3) Click **Communication Setting** on the **Tools** menu or on the toolbar in the NWCONFIG window. After the users complete the setting described below, they can click **OK** in the **Select a Driver** window.



(4) Select a driver in the **Driver Name** drop-down list box. If the users want to download parameters to a single device, upload parameters from a single device, or monitor a single device, they have to select a driver which can connect to the device specified. If the users want to download parameters to multiple devices, upload parameters from multiple devices, or monitor multiple devices, the connection type that the driver selected uses must be Ethernet. The users have to make sure that the driver selected is started.



(5) The users have to make sure that the routing tables produced are downloaded to nodes in the networks before they use routing. If the users want to use routing, they have to select the **Routing Mode** checkbox, and select a device in the **First Station** drop-down list box. Generally speaking, the device which actually connects to the computer is the first station. If the computer connects to several devices, or connect to devices through Ethernet, the users have to designate a device as the first station according to the network framework created in NWCONFIG. Besides, if the **Routing Mode** checkbox is selected, the driver selected in the **Driver Name** drop-down list box must be a driver which can connect to the first station.



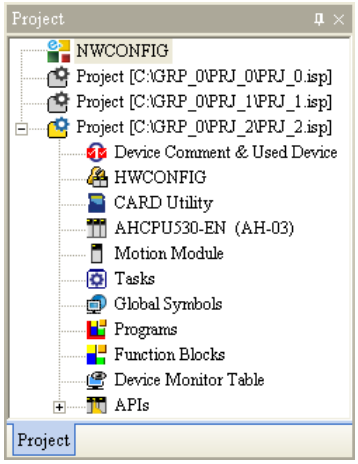
16.1.4 Workflow

The creation of networks involves the operation of a system, and therefore the workflow needed must consist of a sequence of connected steps. The workflow needed to create networks is introduced briefly in this section, and will be described in length in the following sections. The workflow introduced in this section is a method which can be used to efficiently complete work in a general condition. It is not necessarily applicable to all conditions. Users can adjust the workflow according to the actual situations or their habits.

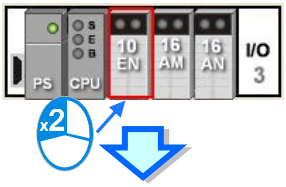
(1) Before users create a system by means of ISPSoft, they have to design networks. The users have to decide what PLCs or devices are used in the networks, whether a PLC needs to be connected to a network module, how the nodes in the networks are connected, what IP

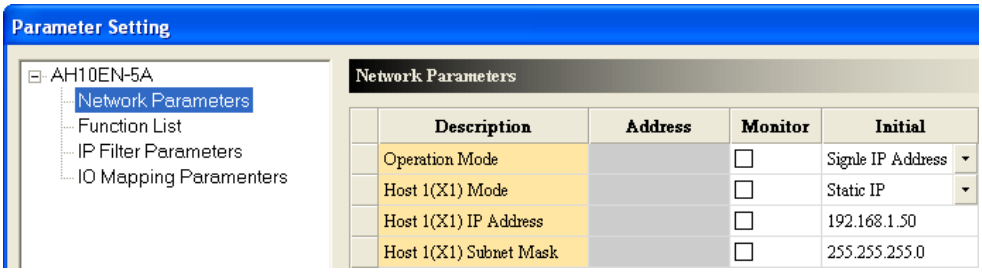
address or RS-485 station address are assigned to the ports connected, and what the values of RS-485 communication parameters are. Besides, the users have to decide what devices perform data exchange. The data exchange is related to the programs in the PLCs used in the networks. After the users design networks, they can create the networks in ISPSoft.

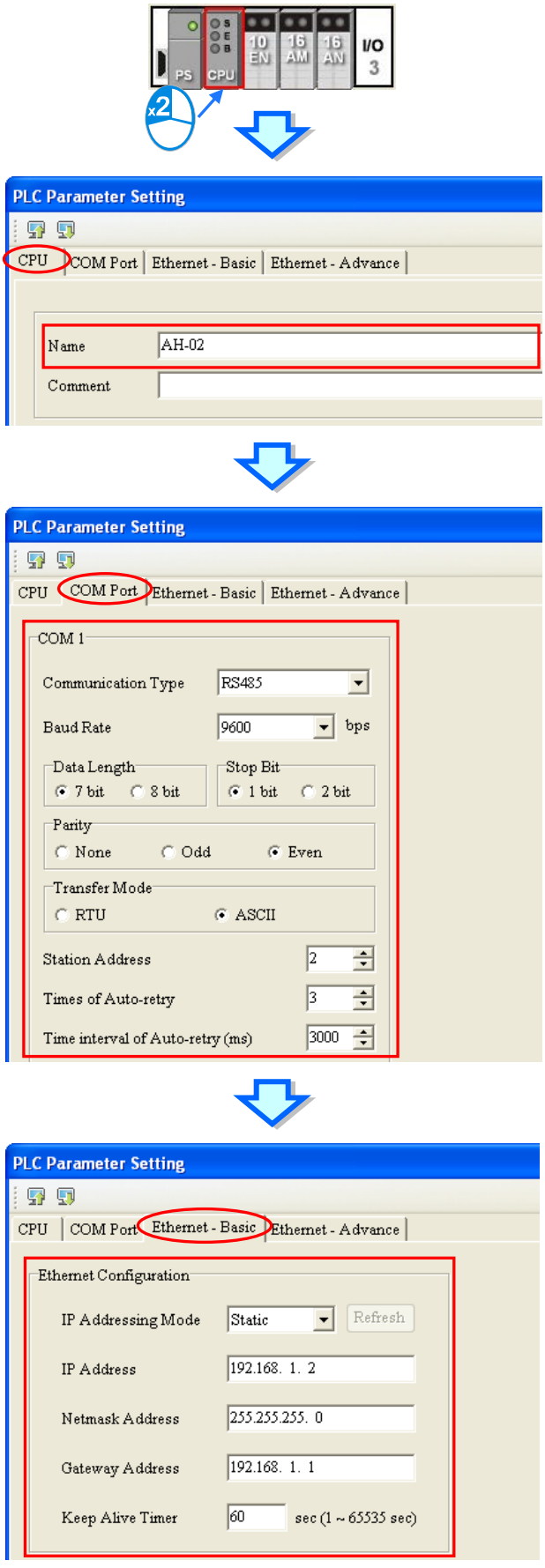
(2) Create a project in ISPSoft. If there are more than two Delta PLCs in a system, it is recommended that the users should create a group of projects in ISPSoft. Please refer to section 2.2 for more information.



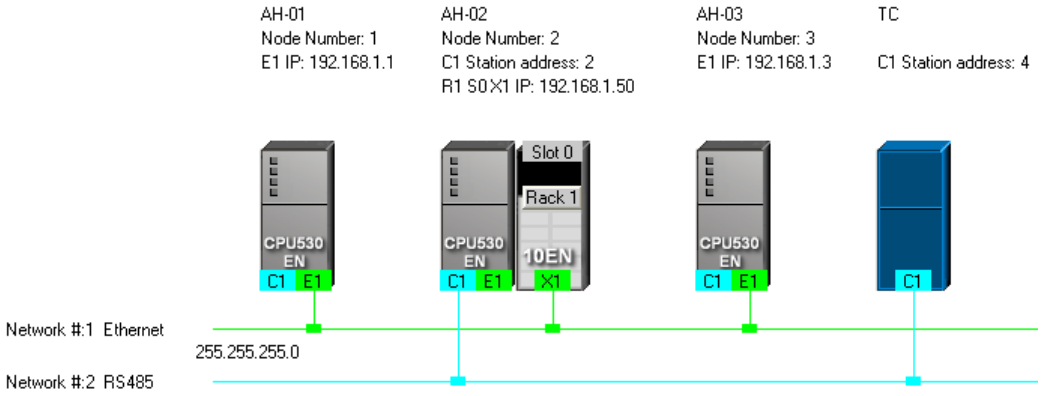
(3) If there are projects for AH500 series CPU modules, the users have to open the HWCONFIG windows in the projects, and complete hardware configurations. The users have to configure modules, set the parameters in network modules, gives names to the CPU modules, set ports, and set Ethernet ports. Please refer to chapter 3 for more information.





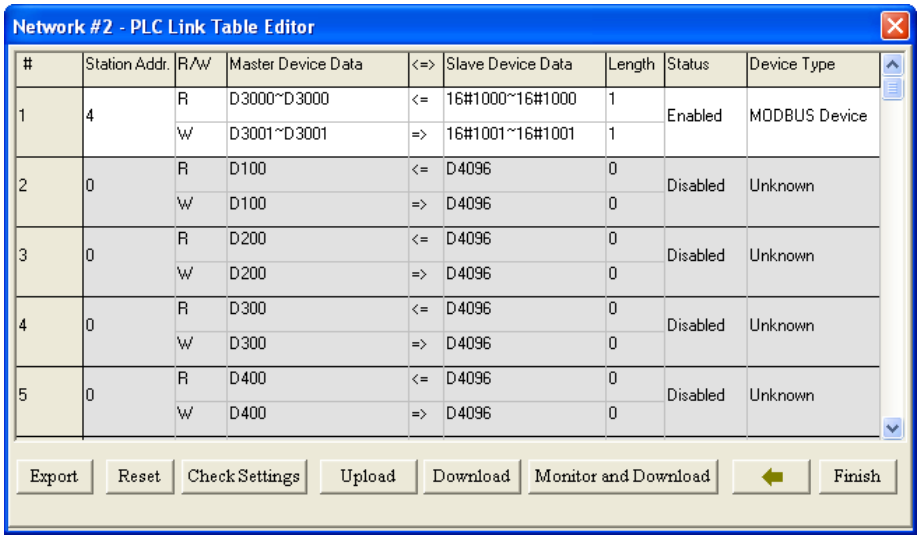

PS
CPU
10 EN
16 AM
16 AN
I/O 3
x2
PLC Parameter Setting
CPU
COM Port
Ethernet - Basic
Ethernet - Advance
Name
AH-02
Comment
PLC Parameter Setting
CPU
COM Port
Ethernet - Basic
Ethernet - Advance
COM 1
Communication Type
RS485
Baud Rate
9600
bps
Data Length
7 bit
8 bit
Stop Bit
1 bit
2 bit
Parity
None
Odd
Even
Transfer Mode
RTU
ASCII
Station Address
2
Times of Auto-retry
3
Time interval of Auto-retry (ms)
3000
PLC Parameter Setting
CPU
COM Port
Ethernet - Basic
Ethernet - Advance
Ethernet Configuration
IP Addressing Mode
Static
Refresh
IP Address
192.168. 1. 2
Netmask Address
255.255.255. 0
Gateway Address
192.168. 1. 1
Keep Alive Timer
60
sec (1 ~ 65535 sec)

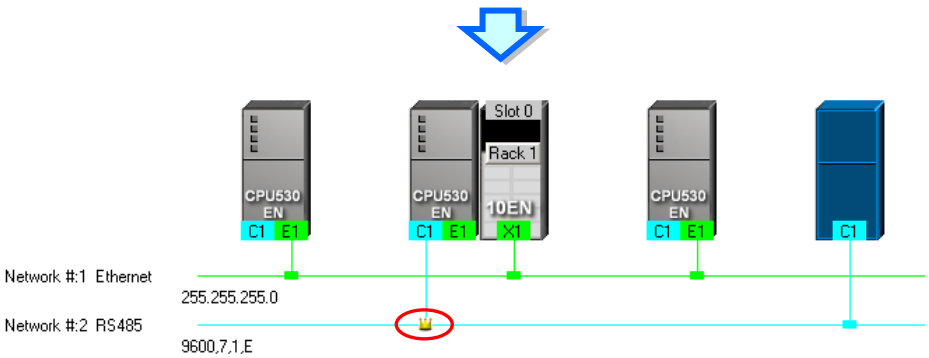
(4) Complete a network configuration in NWCONFIG.



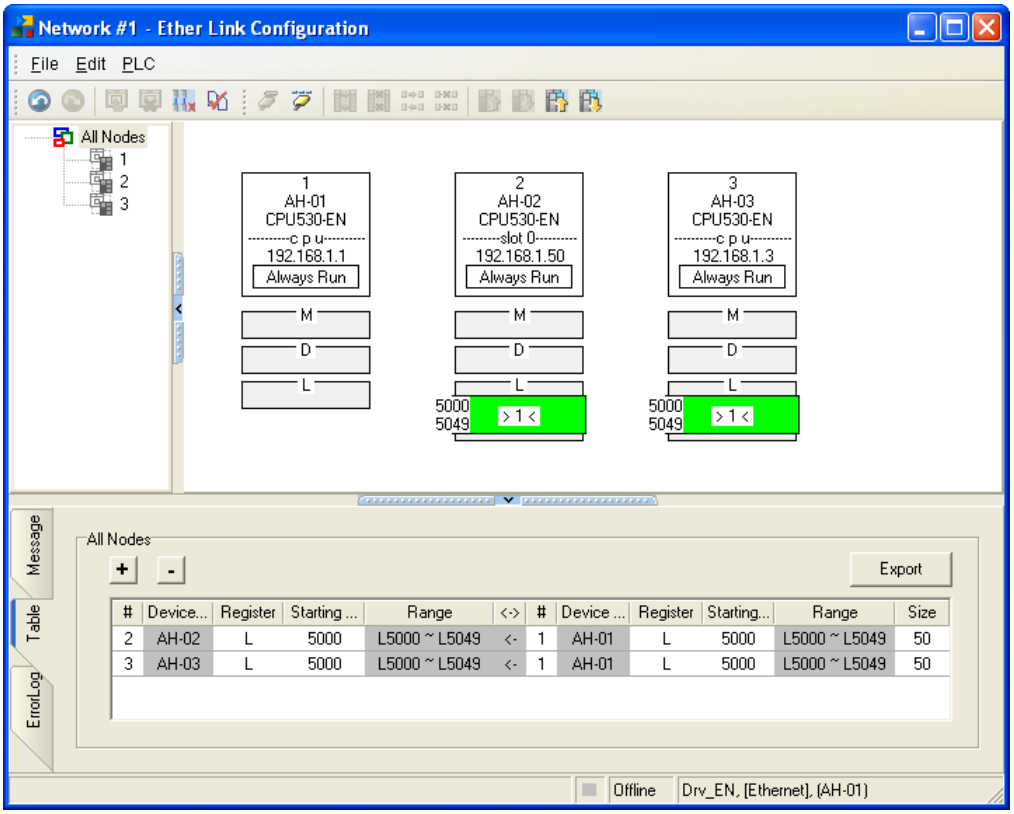
(5) Create a mechanism for data exchange performed by means of a PLC Link or an Ether Link. A PLC Link and an Ether Link operate independently. The users can create them in any order. The addresses involved in data exchange can not overlap, otherwise an error will occur after the data exchange is performed. It is recommended that the users should set addresses which are involved in data exchange according to the programs in the projects created.

The figure below is a table related to data exchange performed by means of a PLC Link. After the users complete the setting in the table, the master station in NWCONFIG will be marked.



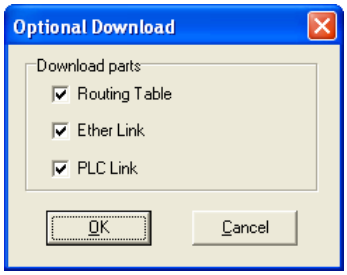


The figure below is a table related to data exchange performed by means of an Ether Link.



(6) Download the programs in the projects, the parameters set in HWCONIFG, and the parameters set in NWCONFIG to the PLCs. If the devices in the networks designed include DVP series PLCs or devices which are not AH500 series CPU modules, the users have to set the communication parameters in these DVP series PLCs or devices which are not AH500 series CPU modules.

The Optional Download window in NWCONFIG is shown below. The items which can be downloaded vary with the node selected.

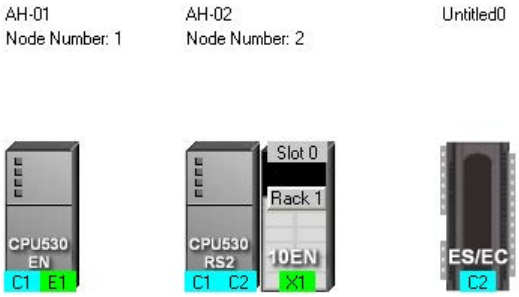


(7) Before the users start the system, they have to create actual networks according to the networks created in NWCONFIG.

16.2 Creating a Network Architecture

16.2.1 Deploying Nodes

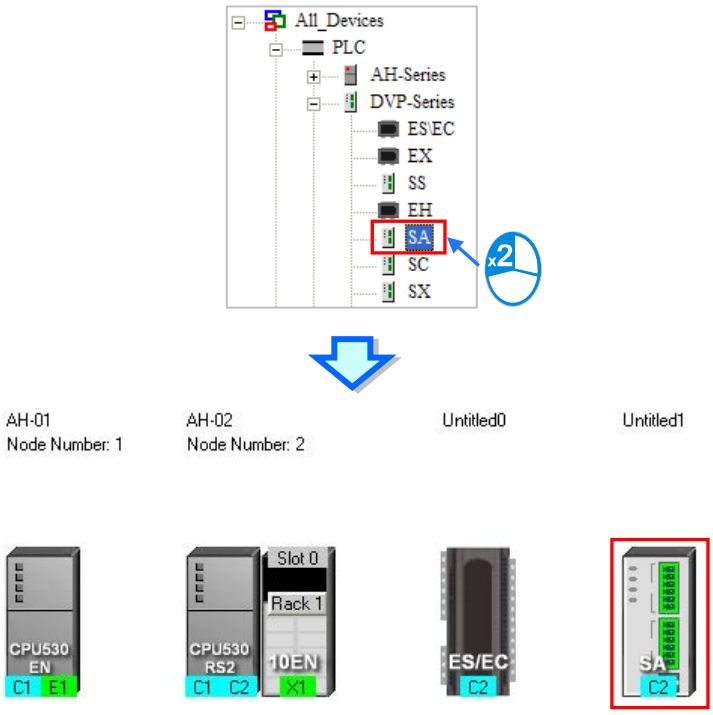
After users open the NWCONFIG window for projects for the first time, the devices for which the projects are created will be deployed in the working area in the NWCONFIG window. If the devices deployed in the working area include AH500 series CPU modules, the modules connected to the CPU modules, the parameters in the CPU modules, and the parameters in the modules connected to the CPU modules will be displayed according to the setting in HWCONFIG. The devices deployed in the working area can not be changed or deleted.



The users can add other PLCs or devices to the working area. There are two ways to add a new device to the working area.

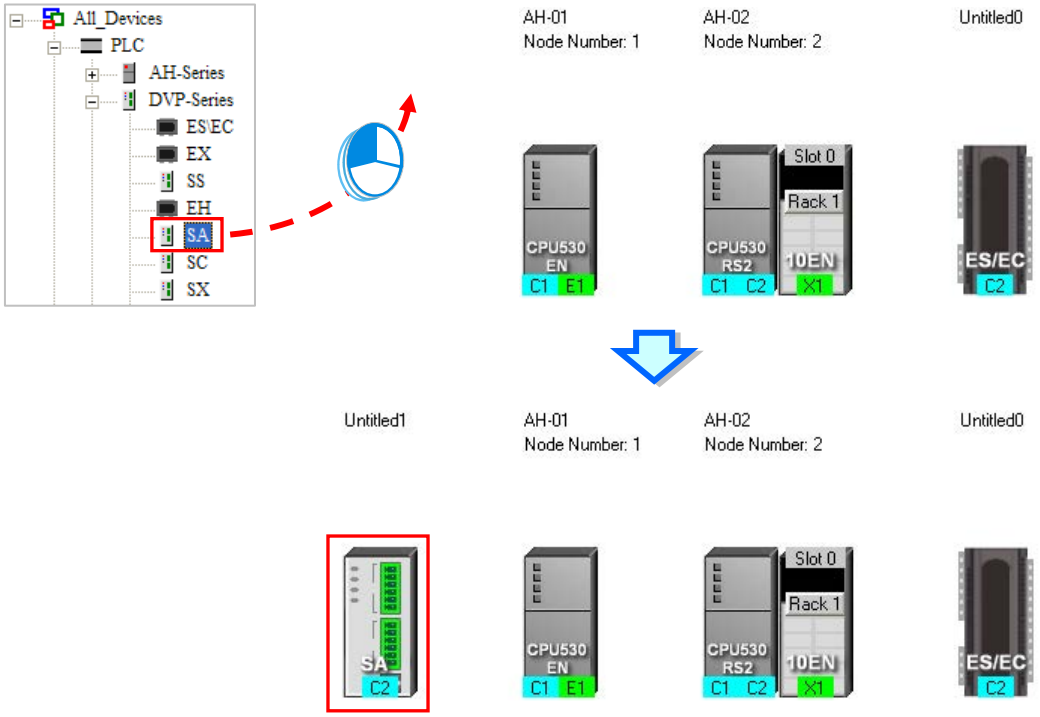
- **Method 1**

  Select a PLC or a device on the device list. After the users double-click the PLC or the device, the PLC or the device will be put at the right side of the rightmost device in the working area.

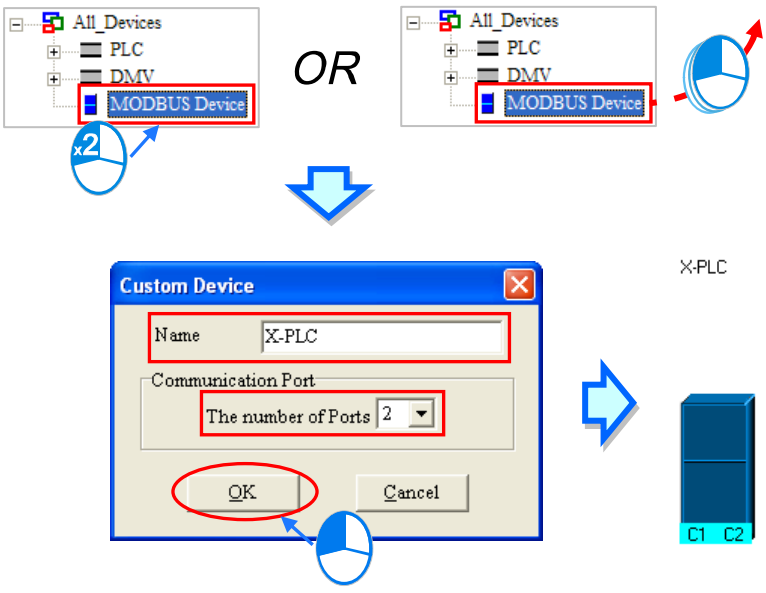


- **Method 2**

  Select a PLC or a device on the device list, and then drag it to a position in the working area.



The users can add a user-defined Modbus device to the working area. After the users select **MODBUS Device** on the device list, and add it to the working area in one of the two ways described above, the **Custom Device** window will appear. The users have to type a name in the **Name** box, select a number in the **The number of ports** drop-down list box, and click **OK**. (A user-defined Modbus device can have three ports at most.)

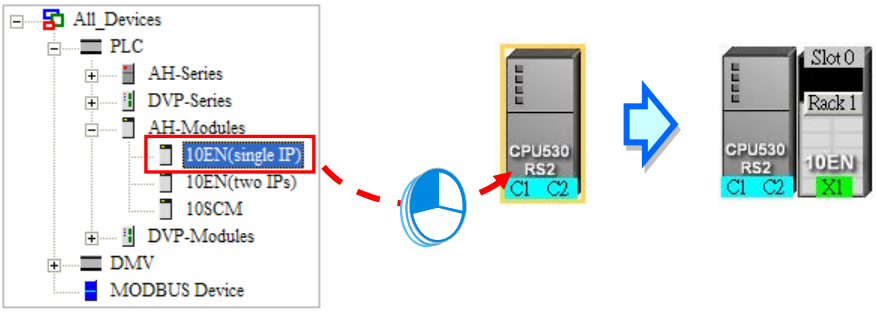


The users can add network modules to the working area. However, if the devices deployed in the working area include AH500 series CPU modules for which projects are created, the users can not connect network modules to the CPU modules. If the users want to connect network modules to the CPU modules, they have to close the NWCONFIG window, configure the network modules in HWCONFIG, and save the setting in HWCONFIG. After the users open the NWCONFIG window again, the system will update the network configuration in NWCONFIG. Besides, the users can directly connect network modules to the DVP series PLCs in the NWCONFIG window.

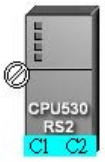
There are two ways to add a network module to the working area.

- **Method 1**

  Select a network module on the device list, and then drag it to a PLC in the working area.

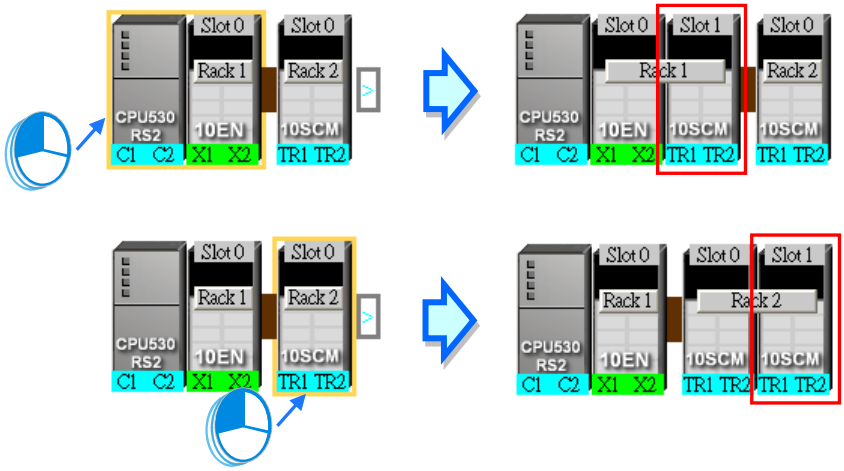


If the PLC does not support the network module selected, the mouse cursor becomes .

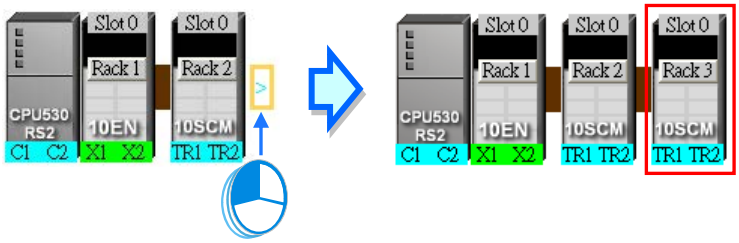


An AH10EN series module can only be put on a main backplane. Eight AH10EN series modules at most can be put on a main backplane. An AH10SCM series module can be put on an extension backplane. The number of AH10SCM series modules which can be put on a backplane depends on the number of slots on the backplane.

Users can choose a backplane to which an AH10SCM series module is dragged.

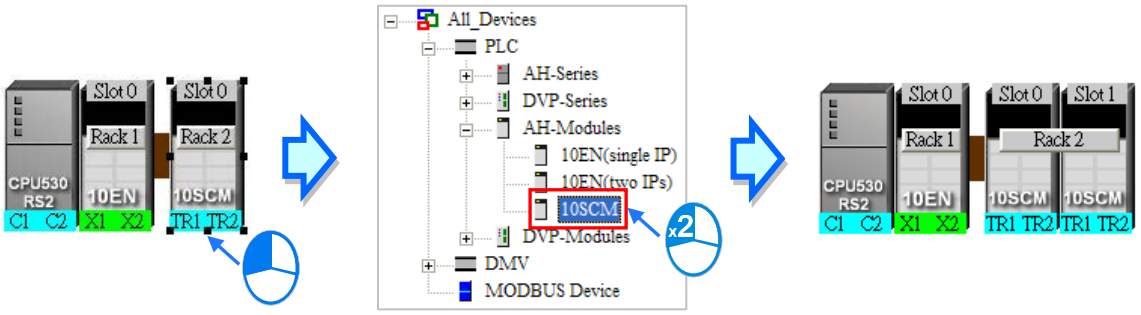


After the users drag an AH10SCM series module to > at the right side of a node, an extension rack will be added to the node.

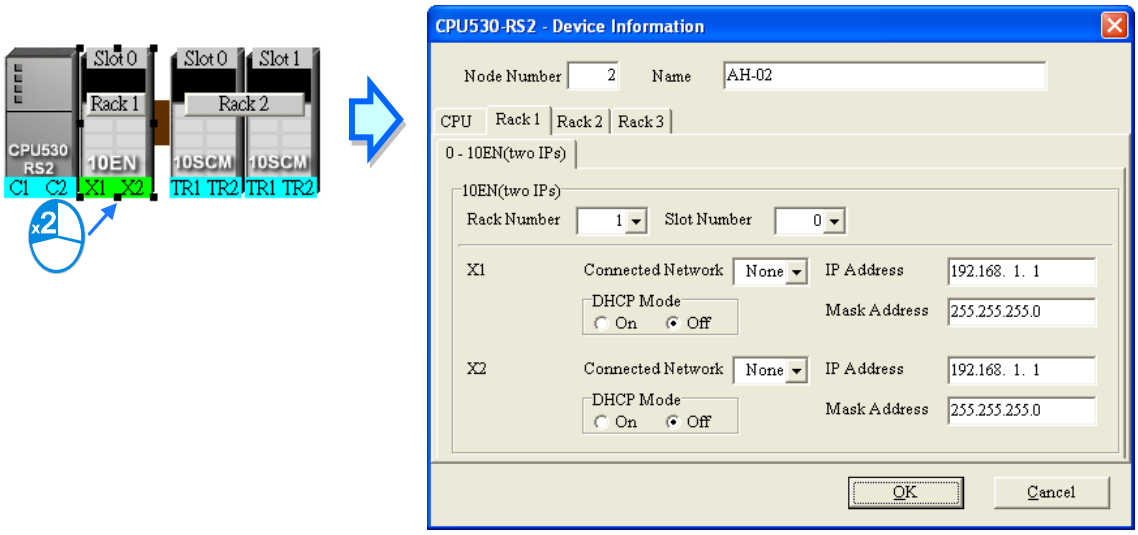


- **Method 2**

  Select a PLC or a rack in the working area, and then double-click a network module on the device list.



After the users add a network module to a node, the slot in which the network module is installed, and the backplane on which the network module is installed may be different from the actual slot in which the network module is installed, and the actual backplane on which the network module is installed. The users have to adjust the properties of the node. Please refer to section 16.2.4 for more information about setting properties of a node, and section 16.2.2 for more information about the marking of a port in NWCONFIG.



16.2.2 Connecting to a Network

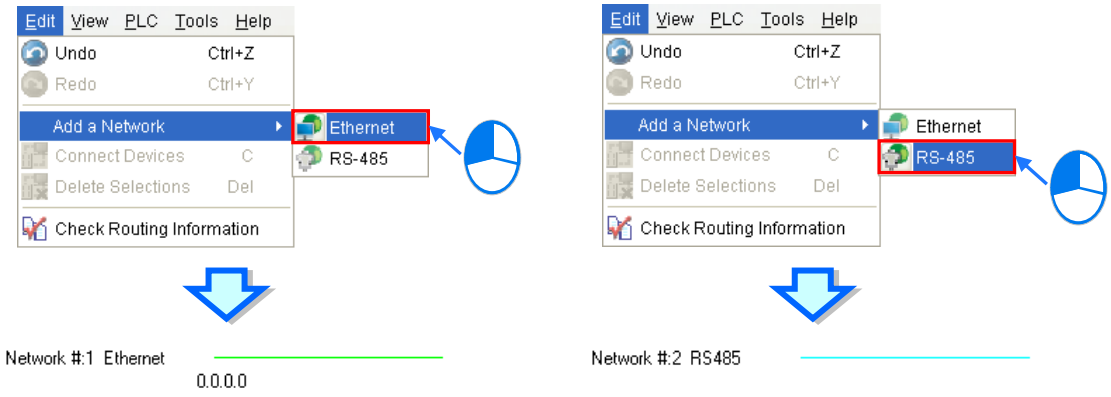
After users deploy the nodes in the NWCONFIG window, they can connect the nodes to the networks designed.

There are three ways to add a network to the working area. There are Ethernet networks and RS-485 networks.

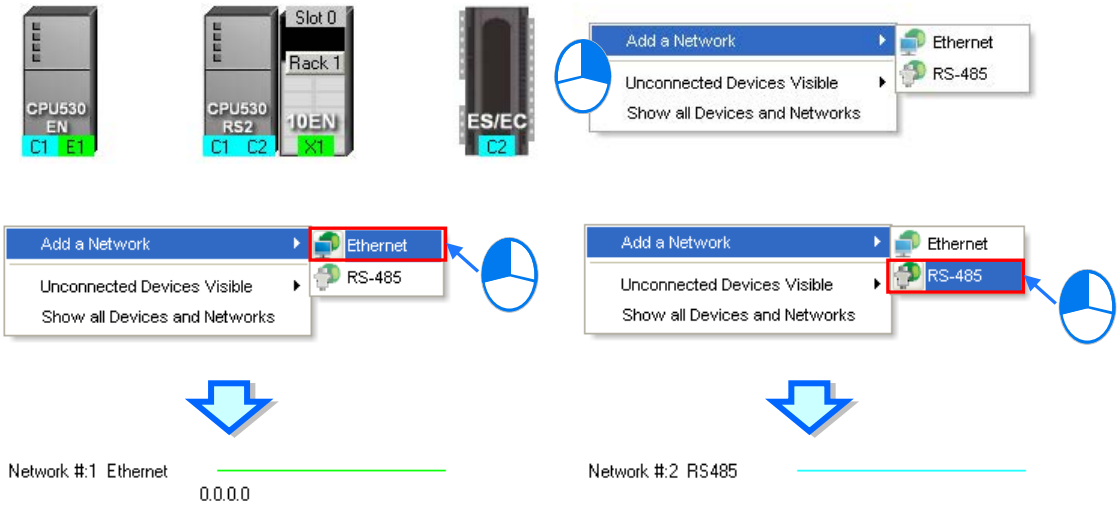
- **Method 1**

  After users click on the toolbar, an Ethernet network is added. After the users click on the toolbar, an RS-485 network is added.

- **Method 2**
  Click the **Edit** menu, point to **Add a Networ**k, and click **Ethernet** or **RS-485**.



- **Method 3**
  Right-click the blank in the working area, point to **Add a Network** on the context menu, and click **Ethernet** or **RS-485**.

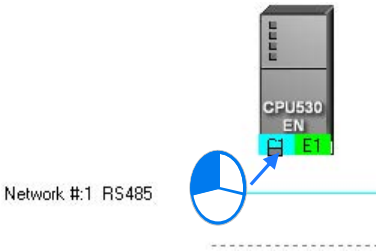


The ports of a device are displayed at the bottom of the device. If a port is blue, it is an RS-485 port. If a port is green, it is an Ethernet port. The port number assigned to a port of a device is consistent with the definition of the port. For example, E1 represents the first Ethernet port, C1 represents COM1, and C2 represents COM2. The ports of an AH10SCM series module are marked with TR1 and TR2, and the ports of an AH10EN series module are marked with X1 and X2. Besides, if the IP address assigned to an Ethernet port is a dynamic IP address, or a port of an AH10SCM series module is not a Modbus port, the Ethernet port or the port of the AH10SCN series module will be gray, and can not connect to any network.

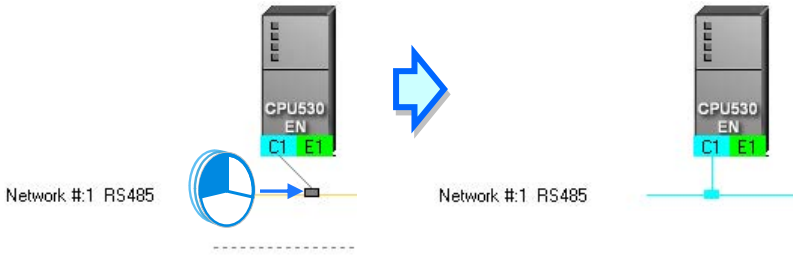
There are several ways to connect the nodes in the working area to networks.

- **Connecting a port to a network by means of dragging the port**
  The users press the left mouse button while the mouse cursor hovers over a port. A dotted line is

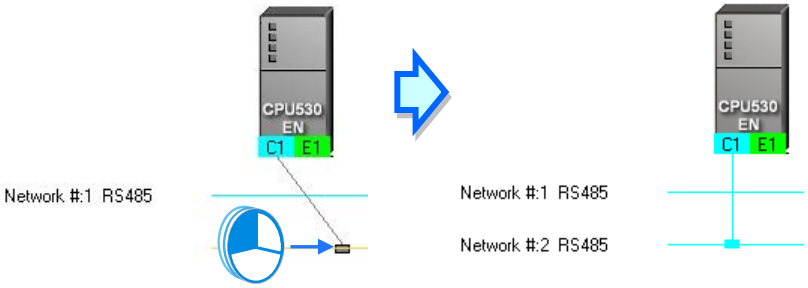
under the existing network.



The users move the mouse cursor to the existing network while holding the left mouse button down. If the network matches the port, the port will connect to the network after the users release the left mouse button.



If the users move the mouse cursor to the dotted line while holding the left mouse button down, the port will connect to a network which matches the port.



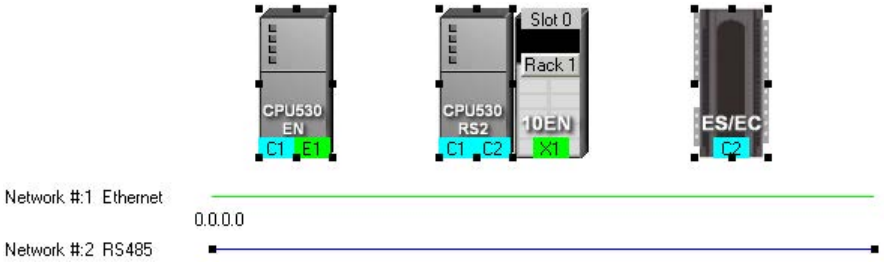
16

- **Connecting a single device or several devices to an existing network**

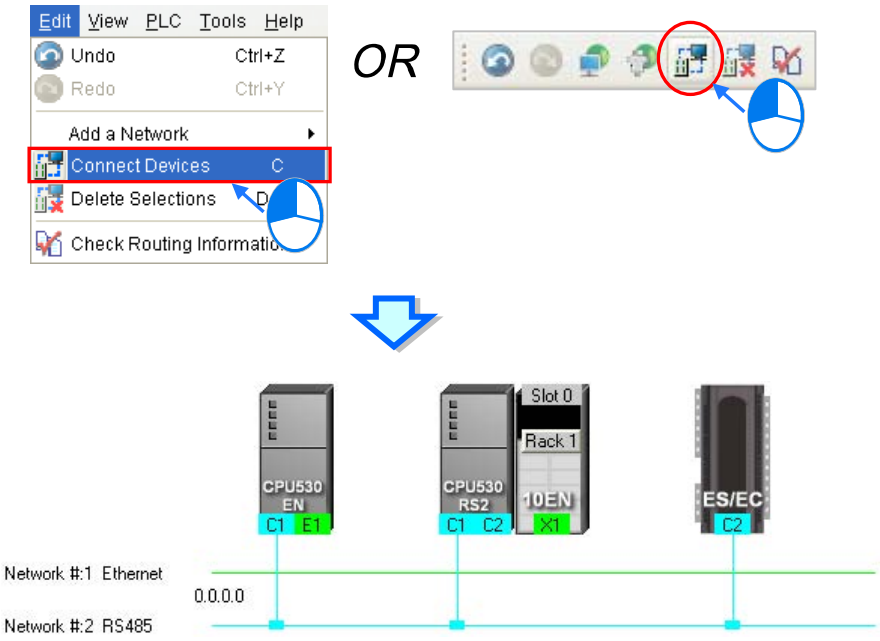
  (1) The users hold down Shift on the keyboard while they click devices and a network. They have to conform to the two principles below.

  (a) PLCs and modules are independent devices. A device that the users click must have at least one port which is not connected to any network, and matches the network clicked.

  (b) The users can click several devices, but they can only click one network.



(2) After the users click **Connect Devices** on the **Edit** menu, or on the toolbar, the system will connect the devices clicked to the network clicked.



Additional remark

(a) If the objects selected do not conform to the two principles listed above, the system will not connect the devices selected to the network selected, and a warning message will appear.

(b) If a device selected has more than one port which is not connected to any network and matches the network selected, the system will connect the port whose port number is smaller to the network selected.

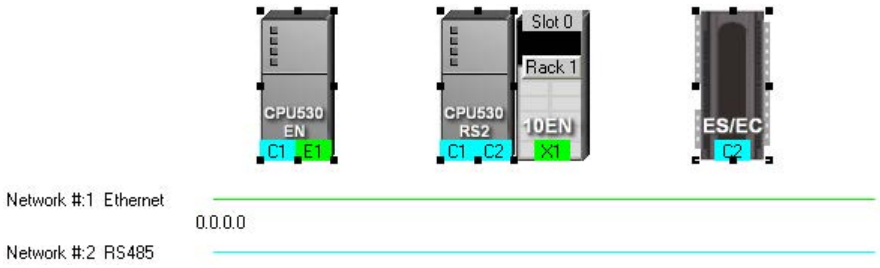
(c) Users can select multiple devices by dragging a selection net around them. If the users press Ctrl+A on the keyboard, all the devices and networks in the working area are selected.

(d) If users select a node which consists of a PLC and a module, and the PLC and the module conform to the principles listed above, the system will connect a port of the PLC and a port of the module to the network selected.

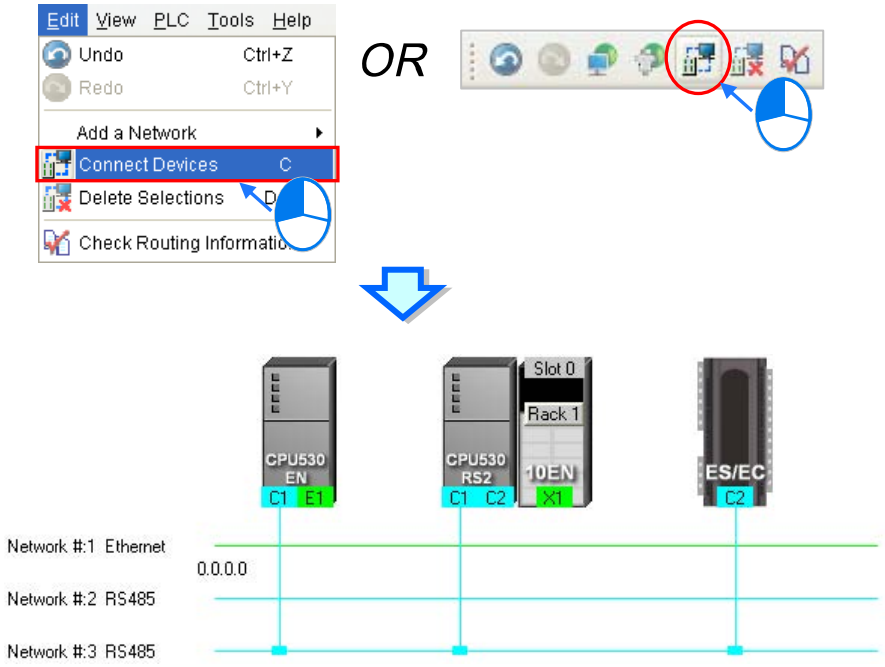
16

- **Automatically connecting a single device or several devices to a new network**

(1) The users hold down Shift on the keyboard while they click devices. PLCs and modules are independent devices. A device that the users click must have at least one port which is not connected to any network, and matches the new network added.



(2) After the users click **Connect Devices** on the **Edit** menu, or on the toolbar, the system will connect the devices clicked to the new network added.



Additional remark

(a) If the devices selected do not conform to the principle described in (1), the system will not connect the devices selected to a new network, and a warning message will appear.

(b) If a device selected has more than one port which is not connected to any network and matches the new network added, the system will connect the port whose port number is smaller to the new network added.

(c) Users can select multiple devices by dragging a selection net around them. If the users press Ctrl+A on the keyboard, all the devices and networks in the working area are selected.

(d) If users select a node which consists of a PLC and a module, and the PLC and the module conform to the principle described in (1), the system will connect a port of the PLC and a port of the module to a new network added.

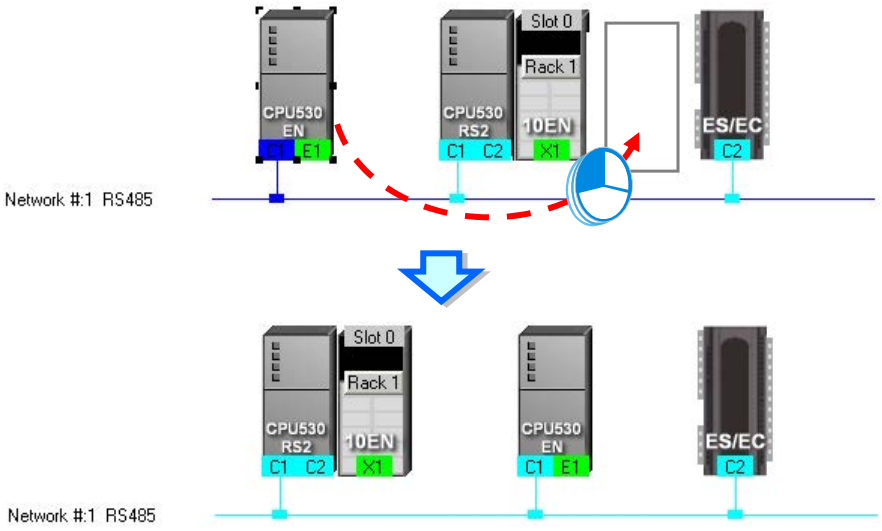
(e) If the devices that users select have ports which are not connected to any networks, and can be connected to an RS-485 network or an Ethernet network, the system will connect the ports to an Ethernet network.

16

16.2.3 Adjusting or Deleting Devices or Networks

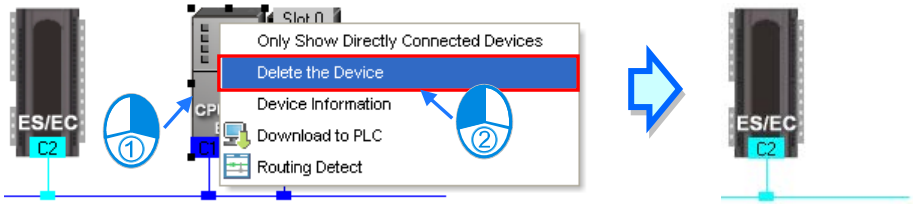
- **Adjusting the order in which the nodes in the working area are arranged**

  Users can change the order in which the nodes in the working area are arranged by dragging a node to a different position. The nodes in the working area can only be at the same level, and increase rightwards. The users can not drag a node to a position above or under another node.

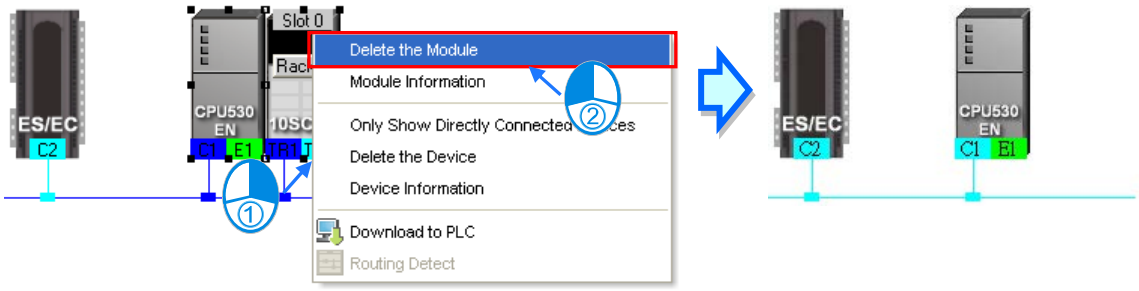


- **Deleting a single device by means of a context menu**

  After users right-click a PLC, and click **Delete the Device** on the context menu, the PLC and the modules connected to the PLC will be deleted. However, the PLC for which a project is created and the modules connected to the PLC can not be deleted.

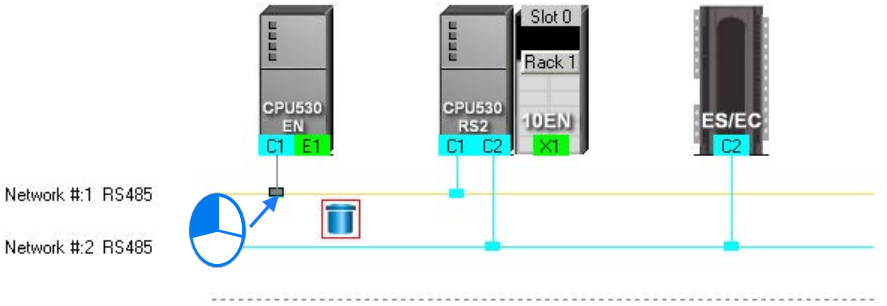


  After the users right-click a module, and click **Delete the Module** on the context menu, the module will be deleted.

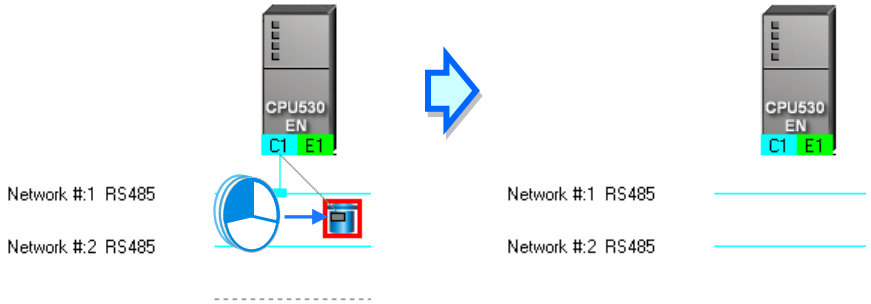


- **Adjusting a connection**

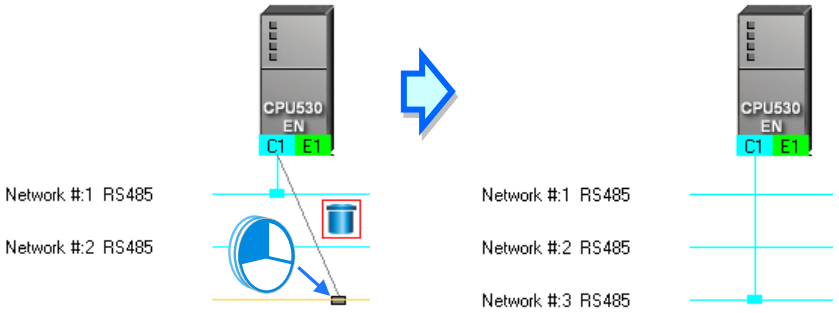
  If users press the left mouse button while the mouse cursor hovers over a connection point which connects a network and a port, a small picture representing a trash can and a dotted line will appear.



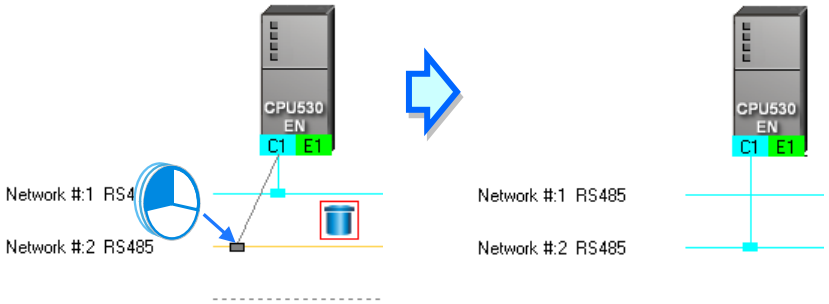
If the users release the left mouse button after they drag the connection point to the small picture representing a trash, the connection between the network and the port will be canceled.



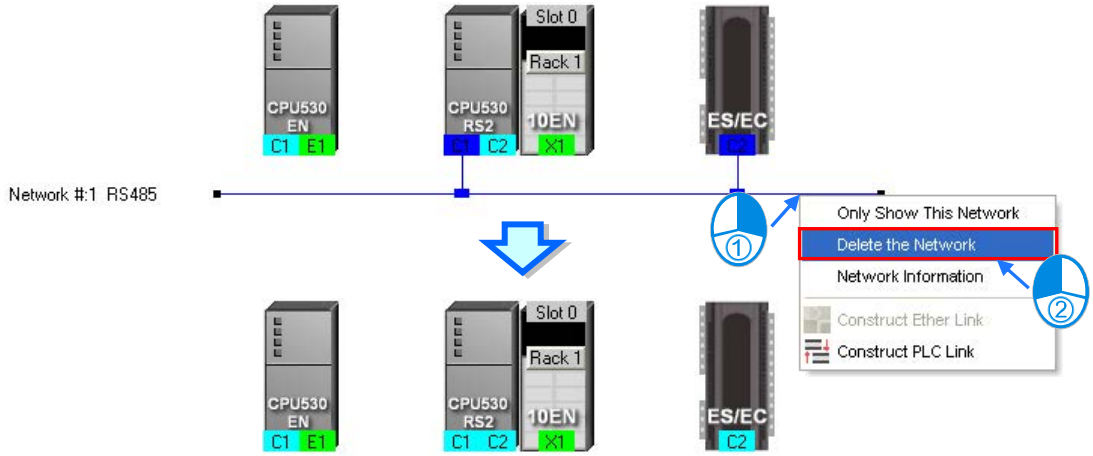
If the users drag the connection point to the dotted line, the system will connect the port to the new network added.



If the users release the left mouse button after they drag the connection point to another network which matches the port, the port will be connected to the network.

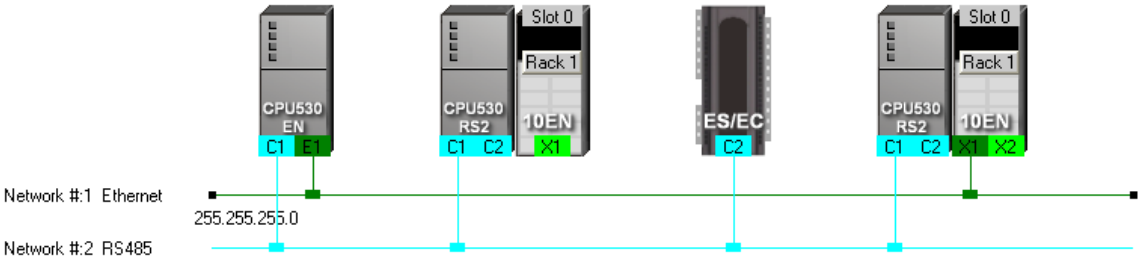


- **Deleting a single network by means of a context menu**
  After users right-click a network, and click **Delete the Network** on the context menu, the network and the lines connected to the network will disappear.

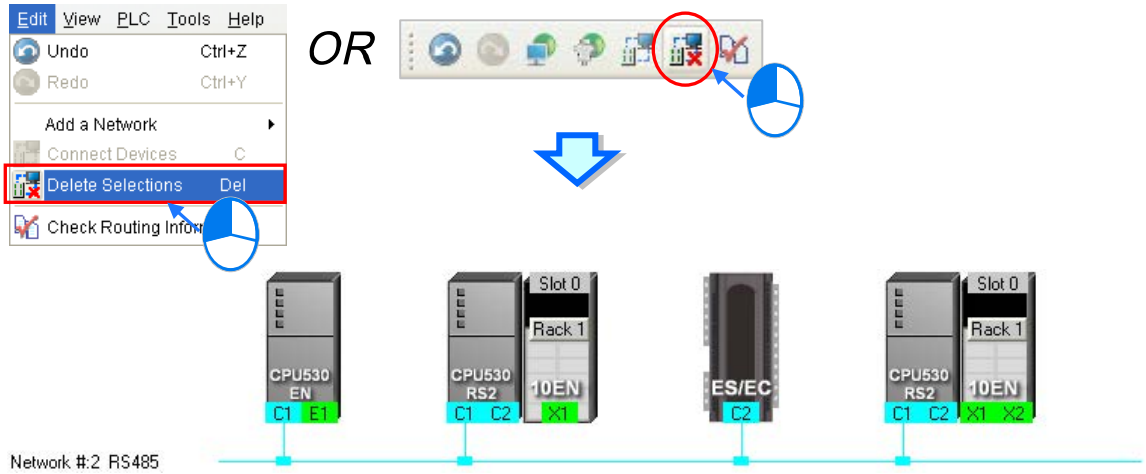


- **Deleting several devices or several networks**

  Users can select several objects by holding down Shift on the keyboard. Besides, the users can select multiple devices by dragging a selection net around them, or selecting all the objects in the working area by pressing Ctrl+A on the keyboard.



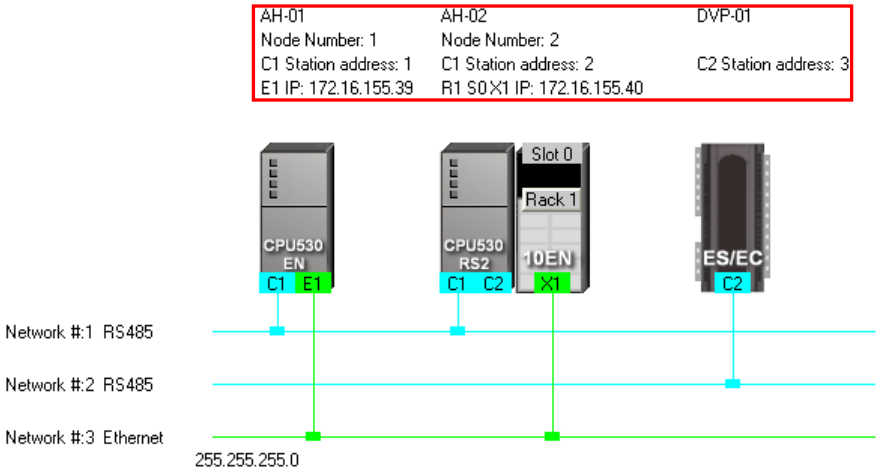
After the users click **Delete Selections** on the **Edit** menu, click on the toolbar, or press Delete on the keyboard, the objects selected will be deleted. However, the PLC for which a project is created and the modules connected to the PLC can not be deleted. Besides, if a PLC is deleted, the modules connected to the PLC will also be deleted.



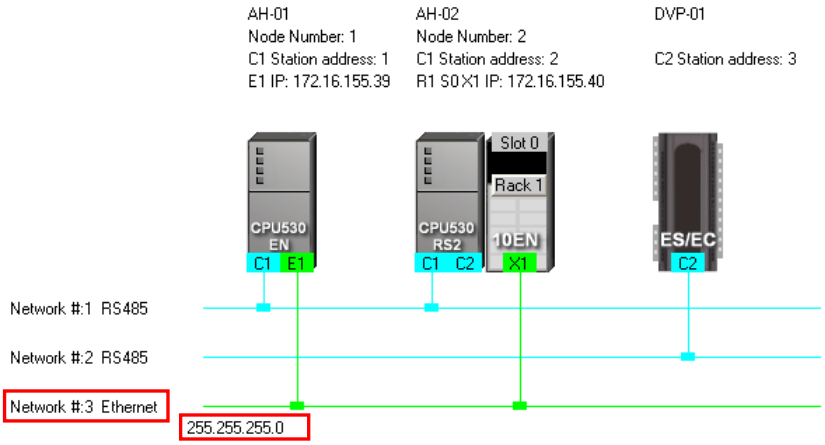
16.2.4 Setting the Attributes of a Node / Network

After users deploy the nodes in the NWCONFIG window, the information about the nodes will put above the nodes. The information includes PLC names and node numbers. The attributes of the ports connected to networks are also displayed. If a port is an RS-485 port, a station address will be

displayed. If a port is an Ethernet port, an IP address will be displayed. The information about a port will be shown if the port is connected to a network. If a port is not connected to any network, no information about the port will be shown. Besides, if a port of a network module is connected to a network, the information about the slot in which the network module is installed, and the information about the backplane on which the network module is installed will be shown.



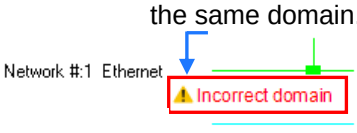
In addition to the information about the nodes, the network numbers assigned to the networks and the network types of the networks are shown. If a network is an Ethernet network, the subnet mask assigned to the ports connected to the network will be shown.



If the ports connected to an Ethernet network are not in the same domain, or are not assigned the same subnet mask, a warning sign will appear. If the ports connected to an Ethernet network are not assigned the same subnet mask, the strictest subnet mask will be shown.

The ports connected to the Ethernet network are not assigned the same subnet mask.

The ports connected to the Ethernet network are not in the same domain.

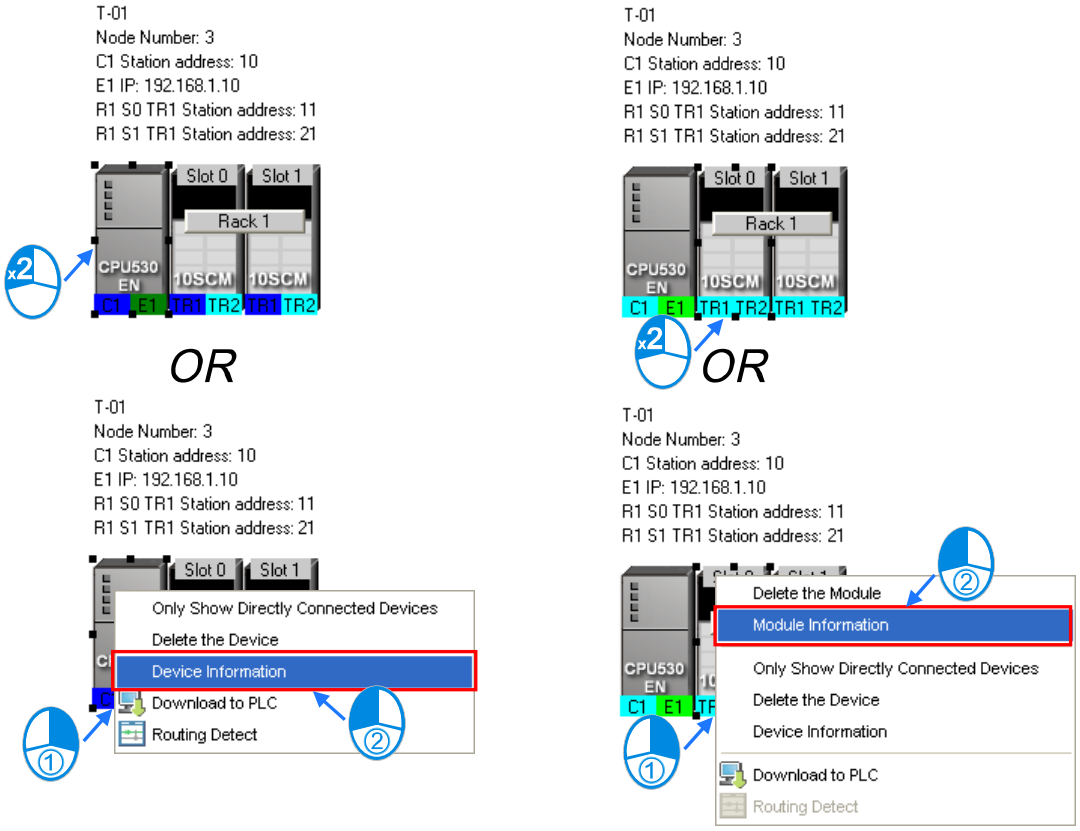


- **Setting the attributes of a node**

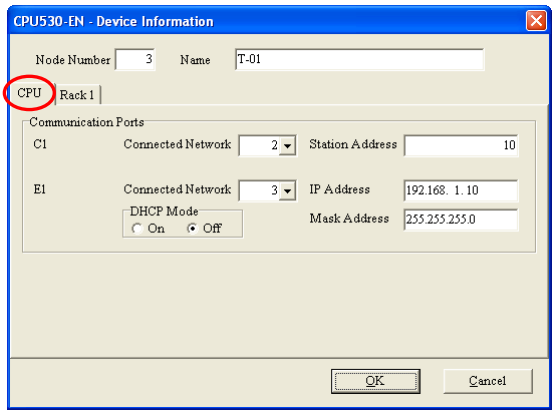
  There are two ways to open the **Device Information** window.

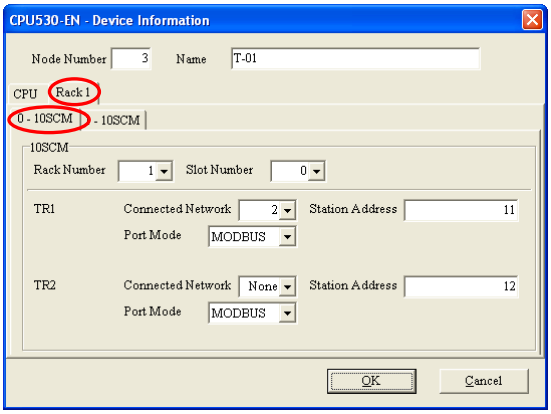
  (a) After users double-click a PLC, the **Device Information** window will appear. The users can also open the **Device Information** window by right-clicking the PLC, and clicking **Device Information** on the context menu.

(b) After users double click a module, the **Device Information** window will appear. The users can also open the **Device Information** window by right-clicking the module, and clicking **Module Information** on the context menu.



In the **Device Information** window, there are two tabs. The page displayed in the window depends on the device selected.

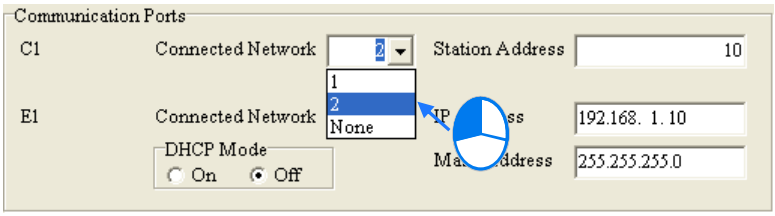




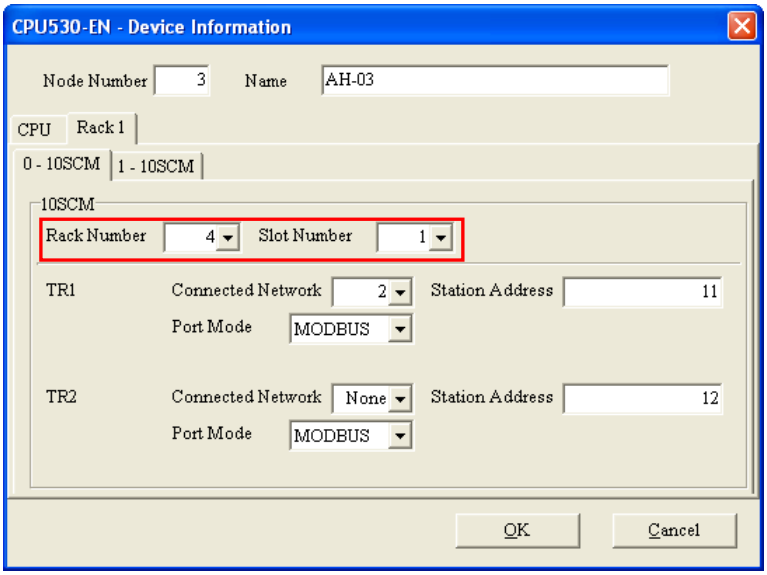
If the device selected is the AH500 series CPU module for which a project is created, or a module connected to the AH500 series CPU module for which a project is created, most boxes in the **Device Information** window are gray. The attributes of the device can only be modified by means of HWCONFIG. If the device selected is a DVP series PLC, a device which is added to the working area, the users can set the attributes of the device. Please refer to section 16.1.2 for more information about the meaning of attributes.

In the **Device Information** window, a port is related to a **Connected Network** drop-down list box. If a port can be connected to several networks, the network numbers assigned to these networks

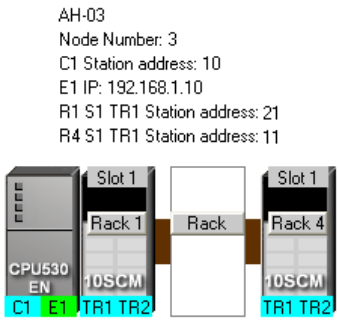
will be on the drop-down list which appears after the users click in the **Connected Network** drop-down list box related to the port. The users can select a network number on the drop-down list. If **None** is selected, the port will not connect to any network. This function is similar to the adjustment of a connection described in section 16.2.3.



In the page for a module, the users can select a rack number in the **Rack Number** drop-down list box, and a slot number in the **Slot Number** drop-down list box.

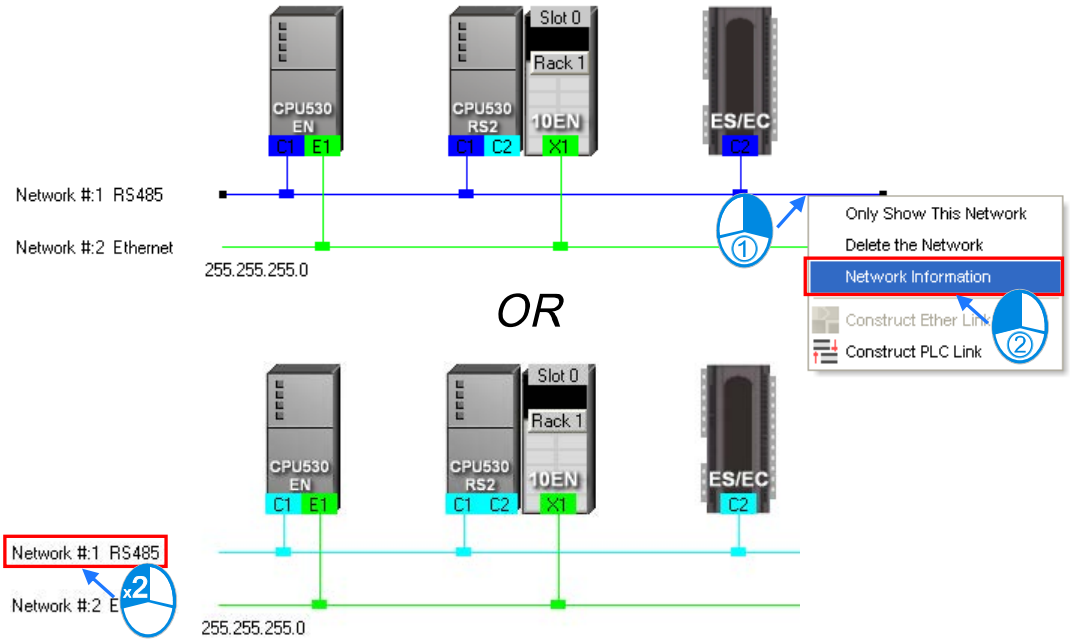


After the users complete the setting of the attributes of a node, and click **OK** in the **Device Information** window, the attributes of the node will be updated immediately.

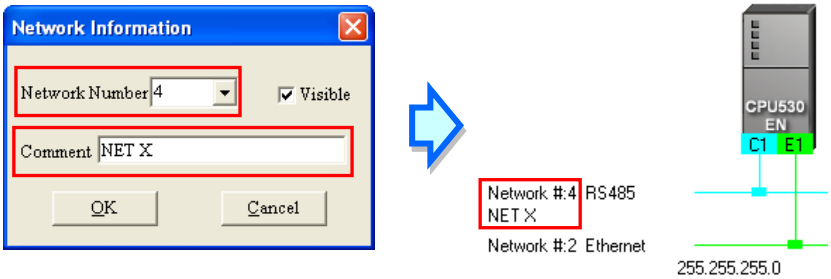


- **Setting the attributes of a network**

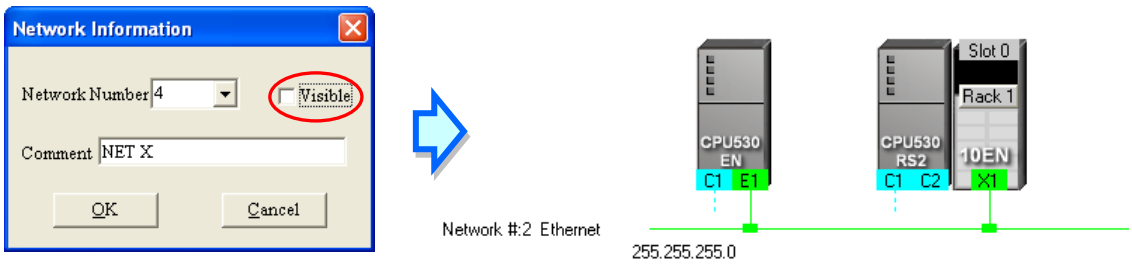
  After users double-click a network, the **Network Information** window will appear. The users can also open the **Network Information** window by right-clicking the network, and clicking **Network Information** on the context menu.



In the **Network Information** window, the users can select a network number which is not assigned to any network in the **Network Number** drop-down list box. Besides, the users can type a comment in the **Comment** box. After the users complete the setting of the attributes of the network, and click **OK** in the **Network Information** window, the attributes of the network will be updated immediately.

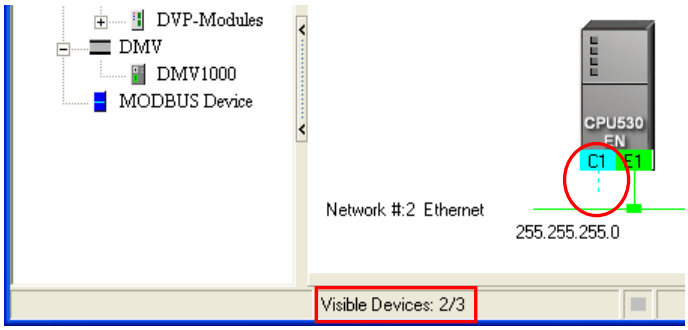


If the users unselect the **Visible** checkbox in the **Network Information** box, the network and the devices connected to the network will become invisible, the other devices connected to the other networks will still be displayed, and the lines connected to the network will become dotted lines. Please refer to section 16.2.5 for more information about hiding/displaying devices or networks.

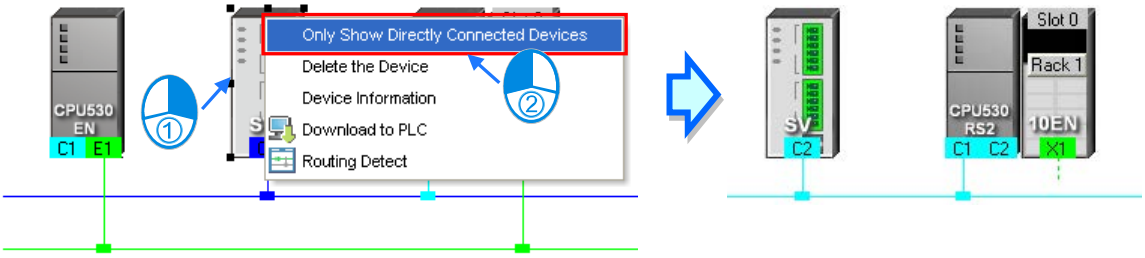


16.2.5 Hiding / Displaying Devices or Networks

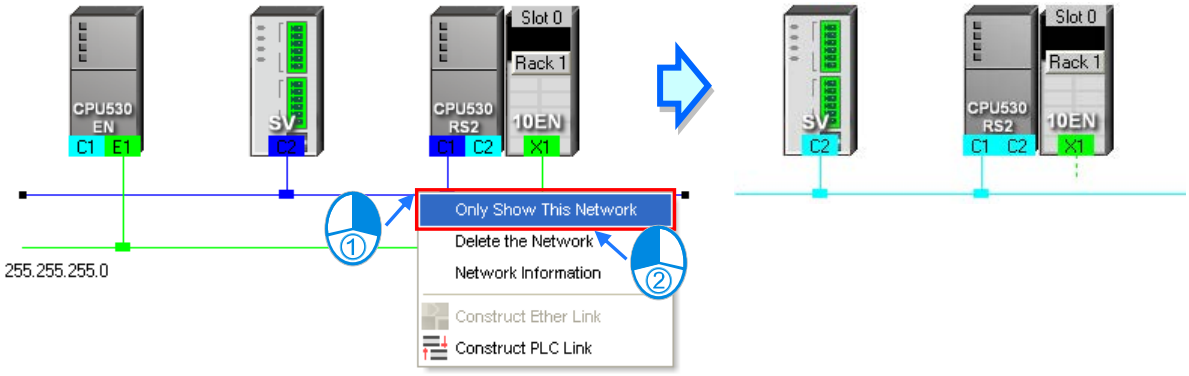
Users can hide/display devices or networks in the working area. The number of devices visible and the total number of devices are displayed in the status bar. Besides, if a dotted line is connected to a port of a device, the port is connected to an invisible network.



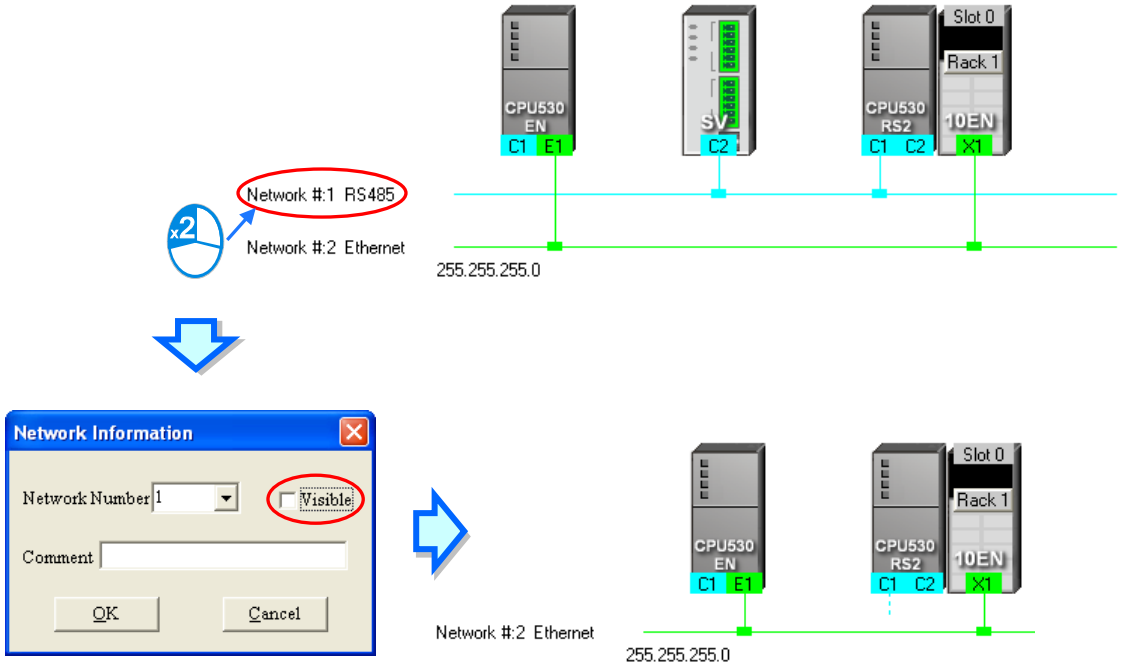
- **Only displaying the objects connected to a device**
  After users right-click a node, and click **Only Show Directly Connected Devices** on the context menu, only the network and the devices which are connected to the node will be displayed.



- **Only displaying the devices connected to a network**
  After users right-click a network, and click **Only Show This Network** on the context menu, only the devices connected to the network will be displayed.

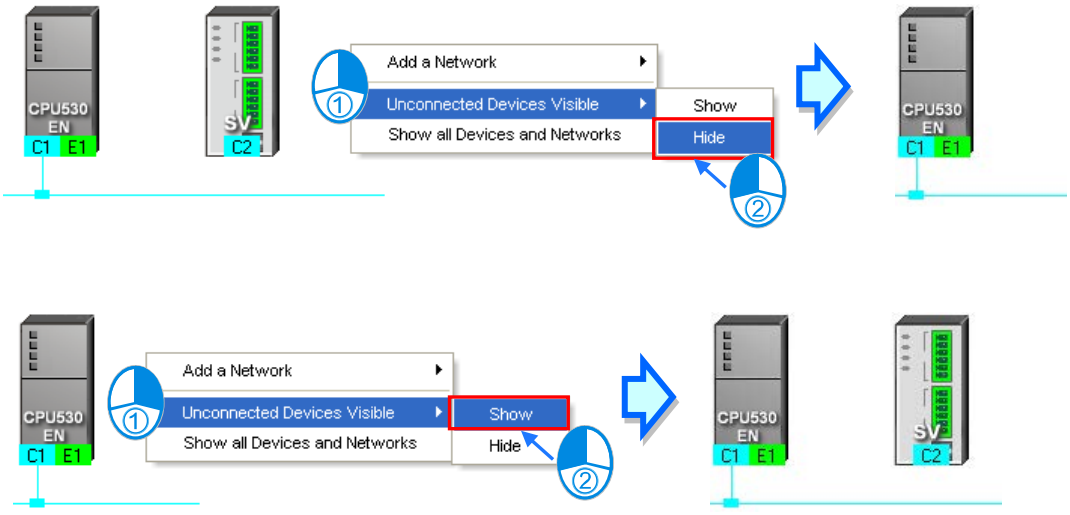


- **Hiding a network and the devices connected to the network**
  After users unselect the **Visible** checkbox in the **Network Information** box, the network and the devices connected to the network will become invisible, but the other devices connected to the other networks will still be displayed,



- **Hiding/Displaying the devices which are not connected to any networks**

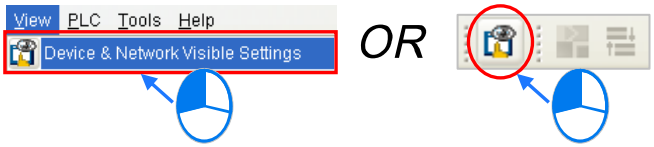
  If users want to hide/display the devices which are not connected to any networks, they can right-click the working area, point to **Unconnected Devices Visible** on the context menu, and click **Hide**/**Show**. This operation affects the devices in the present working area. It does not affect the devices which will be added latter.



- **Setting the display states of all the objects at the same time**

  Users can set the display states of all the devices at the same time.

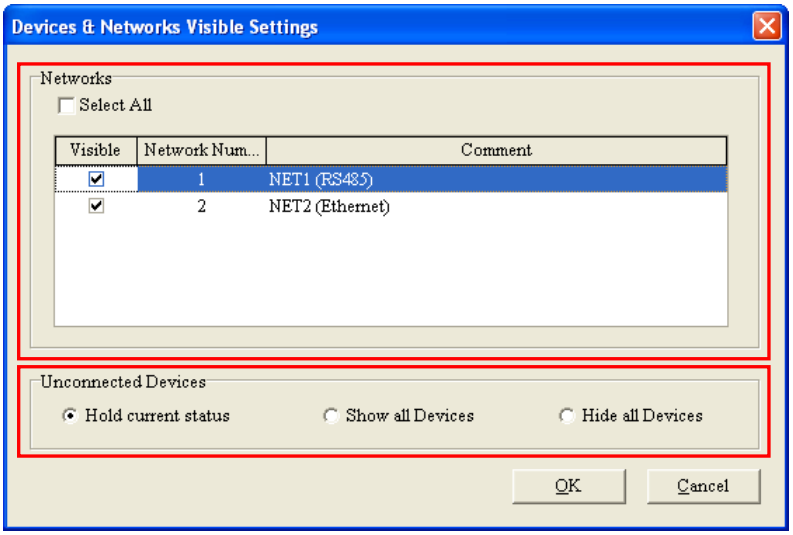
  Click **Device & Network Visible Settings** on the **View** Menu, or on the toolbar.



  Select networks which will be displayed in the **Networks** section in the **Devices & Networks Visible Settings** window. The networks which are not selected will not be displayed in the

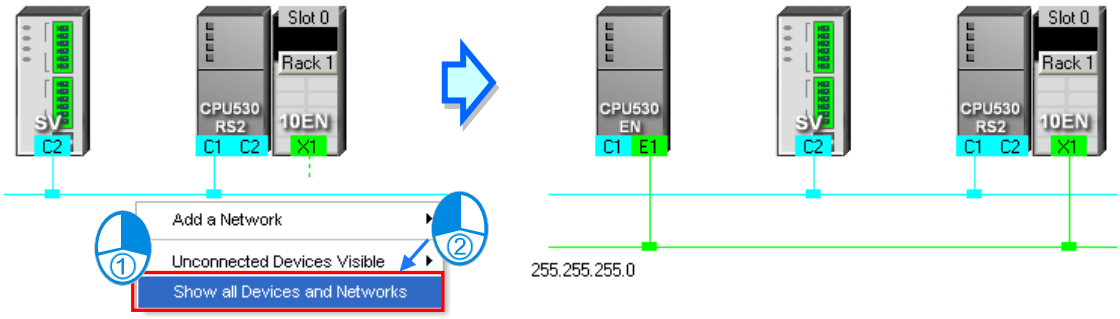
working area. If the users select the **Select All** checkbox, all the networks in the **Networks** section will be selected. If the users unselect the **Select All** checkbox, all the networks in the **Networks** section will be unselected.

In the **Unconnected Devices** section, the users can set the display states of the devices which are not connected to any networks. This operation affects the devices in the present working area. It does not affect the devices which will be added latter.



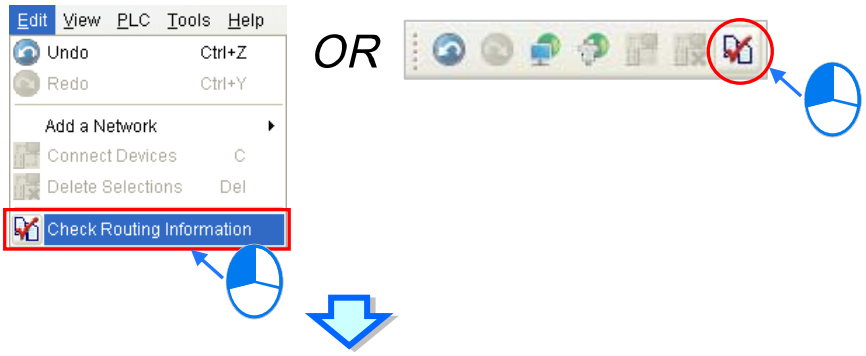
- **Displaying all the objects**

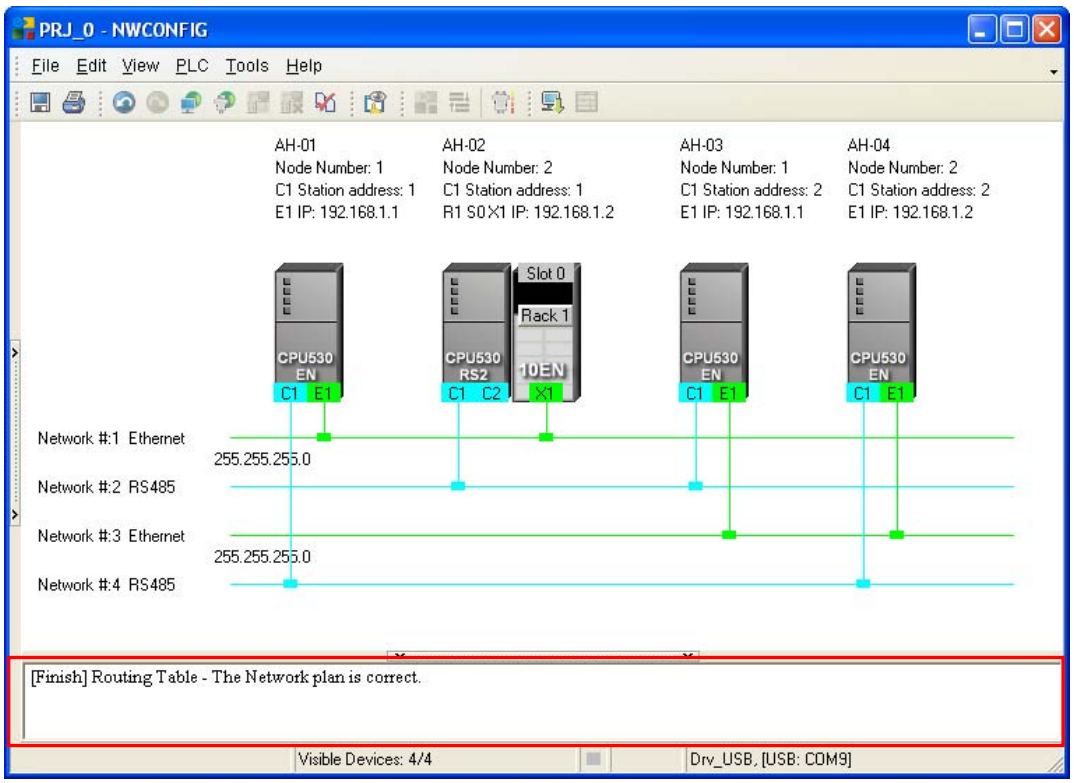
  After users right-click the blank in the working area, click **Show All Devices and Networks** on the context menu, the devices and the networks which are hidden will be displayed.



16.2.6 Correct Network Architecture

After users click **Check Routing Information** on the **Edit** menu, or on the toolbar, the system will check whether the network architecture the users create is correct, and the check result will be displayed in the message display area.





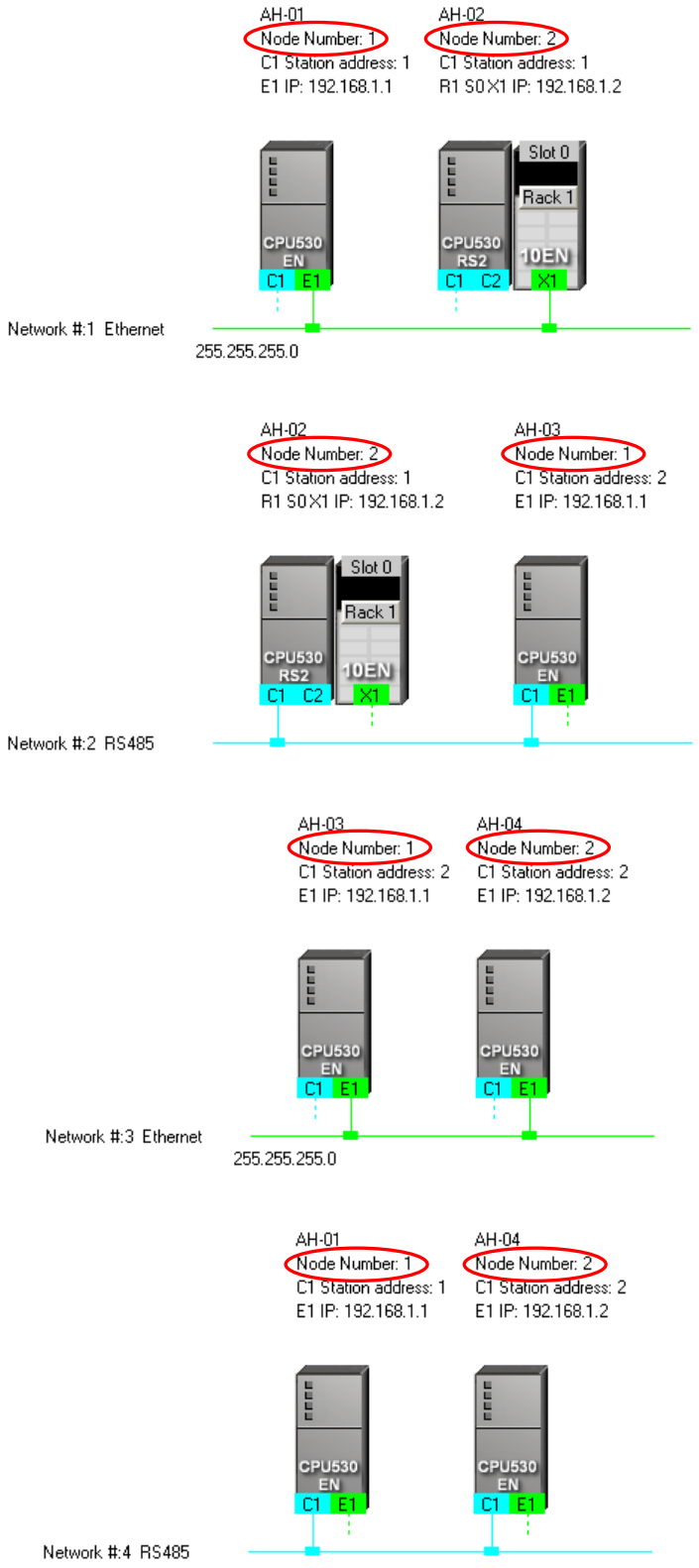
16

The system checks the ports which are connected to networks. It checks whether the information about the nodes and the information about the networks are correct. Specifically, it checks the node numbers, the RS-485 station addresses, and the IP addresses in the working area. At first glance, the network architecture in the figure above seems to be incorrect in that the node number assigned to AH-01 is the same as the node number assigned to AH-03, the node number assigned to AH-02 is the same as the node number assigned to AH-04, the RS-485 station address assigned to AH-01 is the same as the RS-485 station address assigned to AH-02, the RS-485 station address assigned to AH-03 is the same as the RS-485 station address assigned to AH-04, the IP address assigned to AH-01 is the same as the IP address assigned to AH-03, and the IP address assigned to AH-02 is the same as the IP address assigned to AH-04. The users can view a network at a time by means of a skill introduced in section 16.2.5. The users have to make sure that the node number, the RS-485 station address, and the IP address which are assigned to a node are not the same as the node number, the RS-485 station address, and the IP address which are assigned to another node. The message in the message display area in the figure above indicates that the network architecture in the working area is correct.

- **Node number**

  In principle, the node number assigned to a node in a network can not be the same as the node number assigned to another node in the network. If users view a network at a time, they can

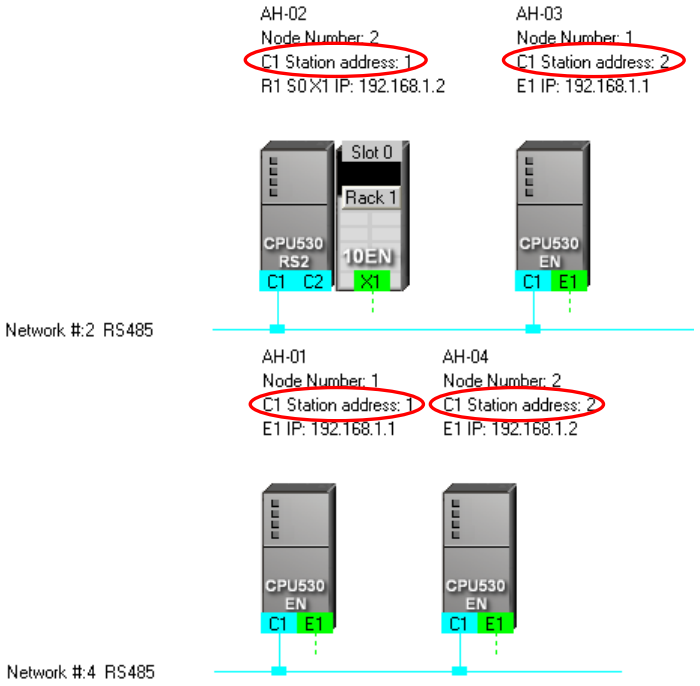
check whether the node number assigned to a node connected to a network is the same as the node number assigned to another node connected to the network.



- **RS-485 station address**

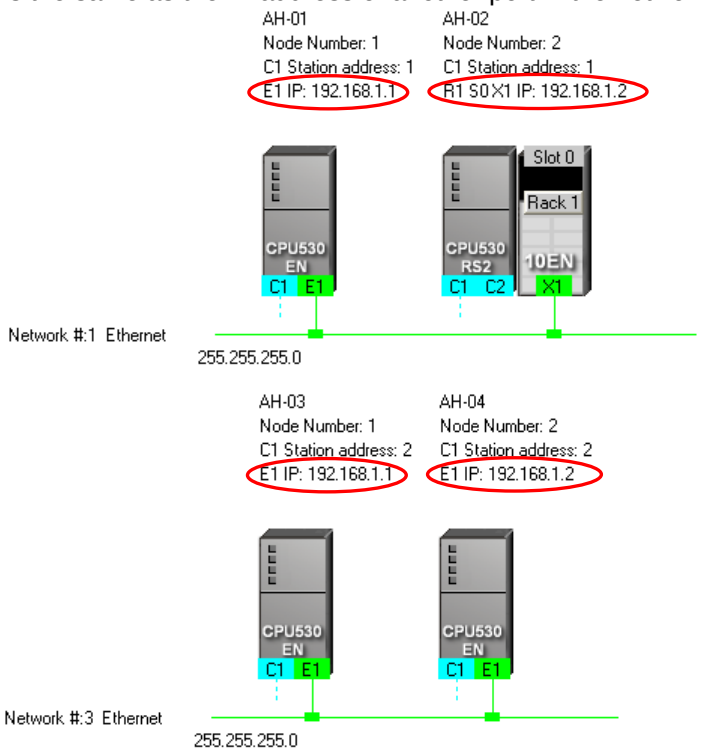
  In principle, the RS-485 station address of a port in a network can not be the same as the RS-485 station address of another port in the network. If users view an RS-485 network at a time, they can check whether the RS-485 station address of a port in a network is the same as the RS-485

station address of another port in the network.



- **IP address**

  The IP address of a port in a network can not be the same as the IP address of another port in the network. If users view an Ethernet network at a time, they can check whether the IP address of a port in a network is the same as the IP address of another port in the network.



16.2.7 Downloading Routing Tables

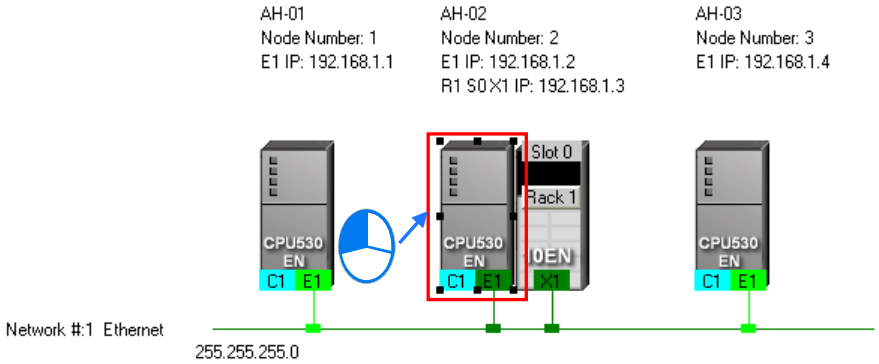
After users make sure that the network architecture they create is correct, they can download the routing tables produced to PLCs. The routing data stored in a PLC is data related to the PLC itself,

and therefore the routing tables downloaded to nodes are different. The users have to download the routing tables produced to nodes in the working area.

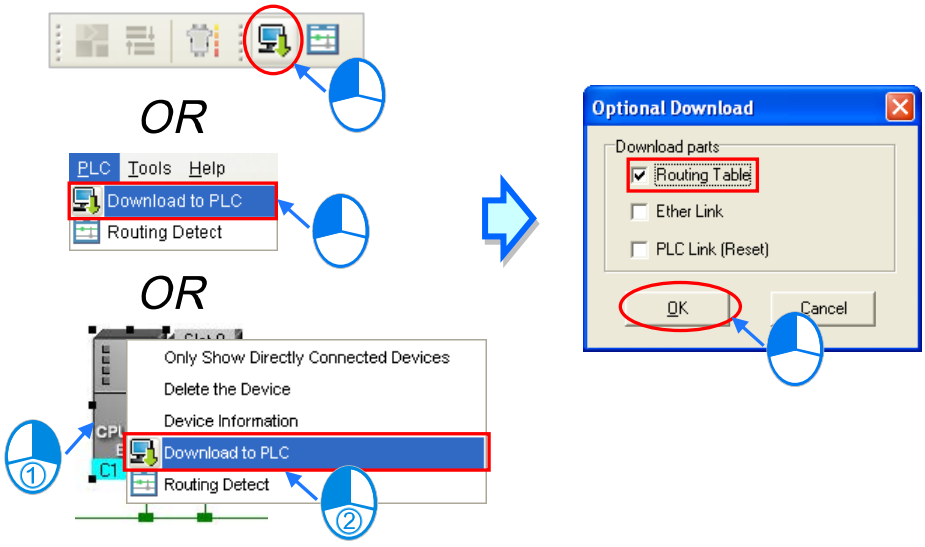
The users can download the routing tables produced to nodes one by one, or download the routing tables produced to nodes at the same time. If the users want to download the routing tables produced to nodes, the **Routing Mode** checkbox in the **Select a Driver** window must be unselected. Please refer to section 16.1.3 for more information.

- **Single node**

  The users have to select a node in the working area. Only AH500 series CPU modules support routing. If the users select a device which is not an AH500 series CPU module, a routing table can not be downloaded to the device.

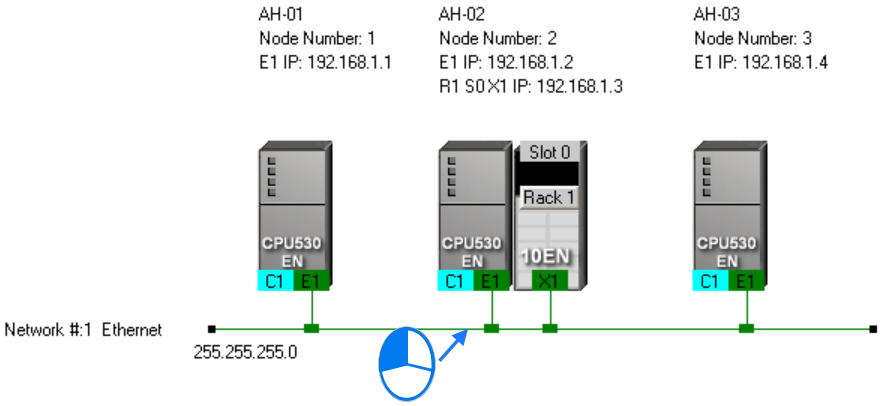


After the users click **Download to PLC** on the **PLC** menu, or on the toolbar, the **Optional Download** window will appear. The users can also open the **Optional Download** window by right-clicking the device they select, and clicking **Download to PLC** on the context menu. After the users select the **Routing Table** checkbox in the **Optional Download** window, and click **OK**, a routing table will be downloaded to the device.

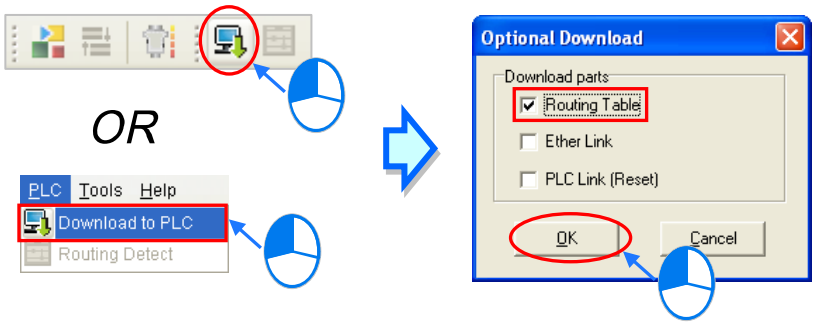


- **Multiple nodes**

  The users have to select an Ethernet network in the working area. If the actual connection is consistent with the setting in NWCONFIG, parameters can be downloaded to the nodes connected to the Ethernet network. If the users want to download the routing tables produced to multiple devices connected to the Ethernet network, the connection type that the driver selected in the **Driver Name** drop-down list box in the **Select a Driver** window uses must be Ethernet. Please refer to section 16.1.3 for more information.



After the users click **Download to PLC** on the **PLC** menu, or on the toolbar, the **Optional Download** window will appear. After the users select the **Routing Table** checkbox in the **Optional Download** window, and click **OK**, the routing tables produced will be downloaded to the nodes connected to the Ethernet network.



Additional remark

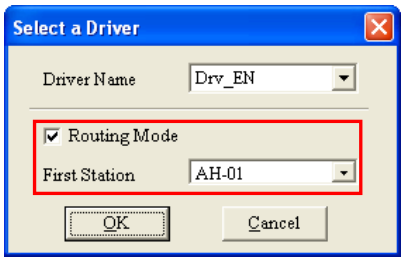
If the parameters related to an Ether Link or a PLC Link can be downloaded to the object selected, users can select the **Ether Link** checkbox or the **PLC Link (Reset)** checkbox in the **Optional Download** window. If a checkbox in the **Optional Download** window is gray, the checkbox can not be selected. Please refer to the following sections for more information about Ether Links and PLC Links.

Besides, the routing data stored in a PLC is data related to the PLC itself. The users can not upload the routing data stored in a PLC. The system does not provide the function of uploading routing data.

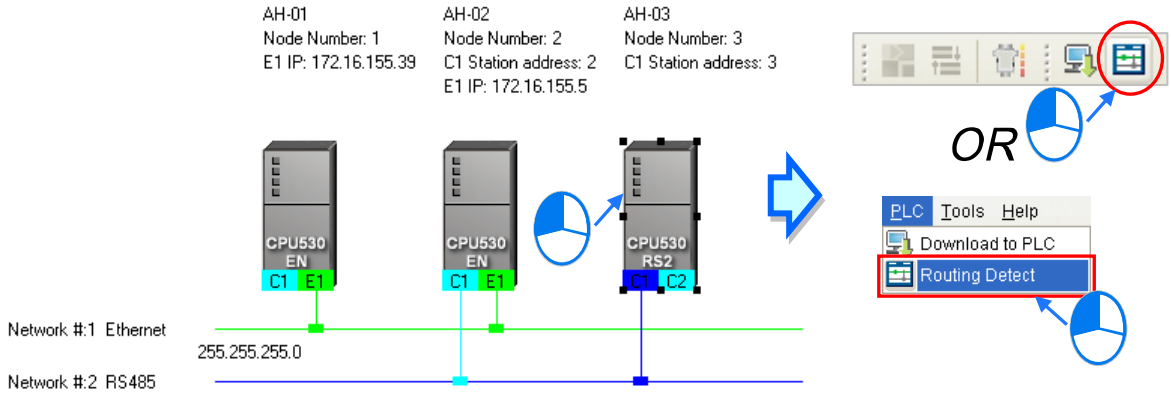
16.2.8 Testing Routing

After the routing tables produced are downloaded, users can test routing by means of a function provided by NWCONFIG. The steps of testing routing are as follows.

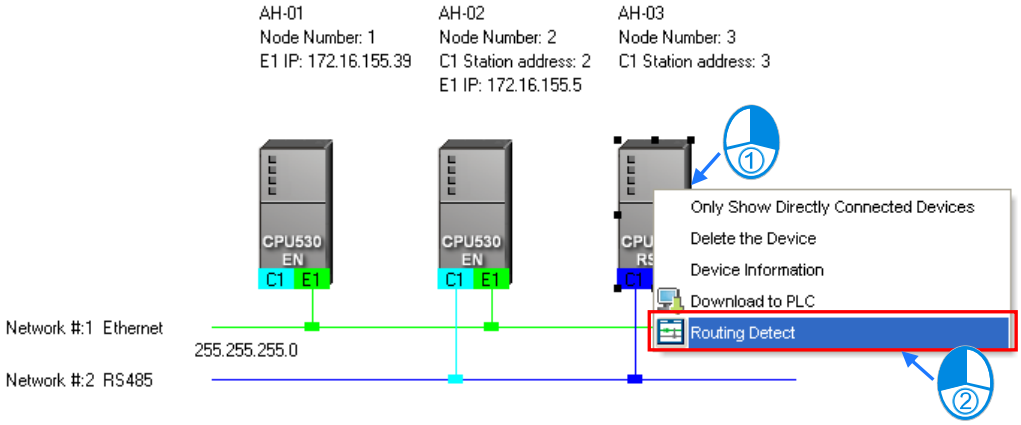
(1) The users have to make sure that all the nodes are wired according to the configuration in NWCONFIG, and operate normally. The setting of the devices has to be consistent with the setting in NWOCNIFG. In the **Select a Driver** window, the users have to select the **Routing Mode** checkbox, and select a device in the **First Station** drop-down list box. Please refer to section 16.1.3 for more information about setting communication.



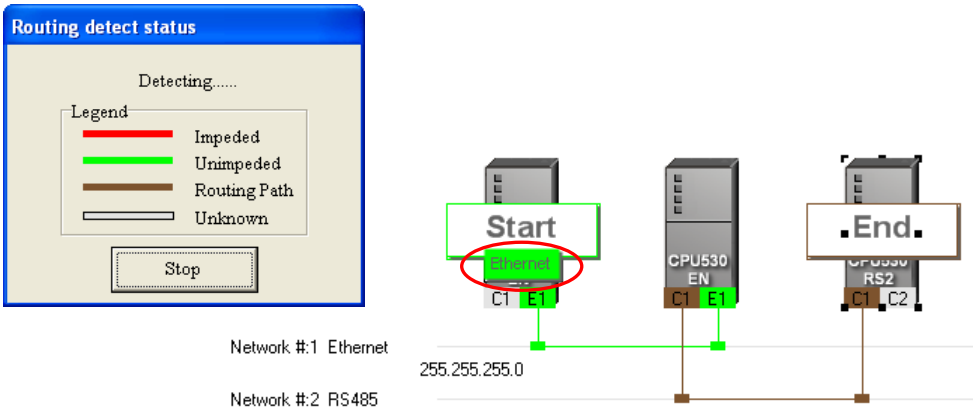
(2) After the users select the destination device toward which packets are relayed, they have to click on the toolbar, or **Routing Detect** on the **PLC** menu.



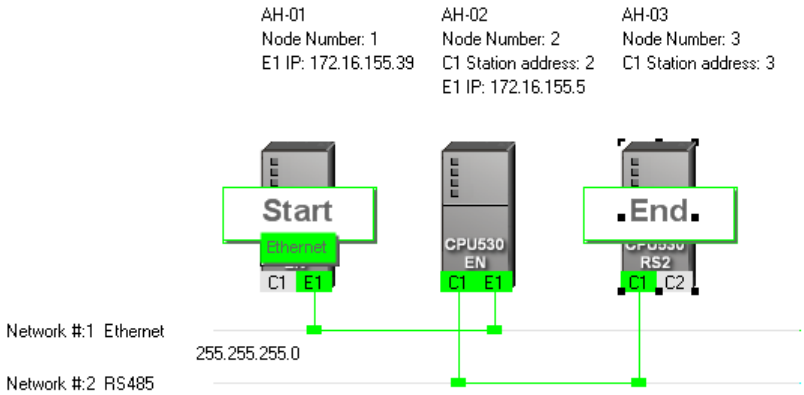
The users can also right-click the destination device, and then click **Routing Detect** on the context menu.



(3) After **Routing Detect** is clicked, the display of the network architecture in the working area will change, and the **Routing detect status** window will appear. The way in which the node which is designated as the first station is connected to the computer is also shown in the working area. Please see the red circle in the figure below.



(4) After the detection is complete, the detection result will be shown in the working area.



Additional remark

If the detection fails, the users have to make sure of the following points.

(a) The users have to make sure that the IP addresses and the station addresses which are assigned to the devices and the communication setting in the devices are consistent with the setting in NWCONFIG. If AH500 series CPU modules or AH500 series modules are used, the users have to make sure that the parameters in the AH500 series CPU modules in HWCONFIG or the parameters in the AH500 series modules in HWCONFIG are set correctly, and downloaded to the AH500 series CPU modules or the AH500 series modules successfully. If DVP series PLCs or DVP series modules are used, the users have to make sure that the communication parameters in the related registers are correct. If other devices are used, the users have to refer to manuals for the usage of these devices, and make sure that the communication parameters in these devices are correct.

(b) The users have to make sure that all the network connection is consistent with the setting in NWCONFIG. They also have to make sure that every node is connected to a network correctly, and operates normally.

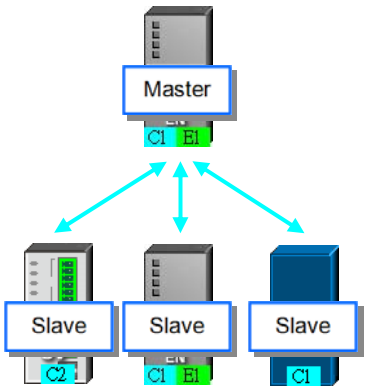
(c) Please refer to section 16.1.3, and make sure that the communication setting is correct.

16.3 Constructing a PLC Link

A PLC Link is a network mechanism for data exchange performed through an RS-485 cable. If there are several nodes in an RS-485 network, users can create a mechanism for data exchange in the network. If the parameters which are set are downloaded to the PLC which functions as a master station, the system of the PLC will perform data exchange through special relays and special registers when the PLC runs.

A PLC Link is a master/slave model. There is only one master station in an RS-485 network, and the other stations which are slave stations passively receive reading/writing commands from the master

station. The slave stations can not exchange data. They have to exchange data through the master station.

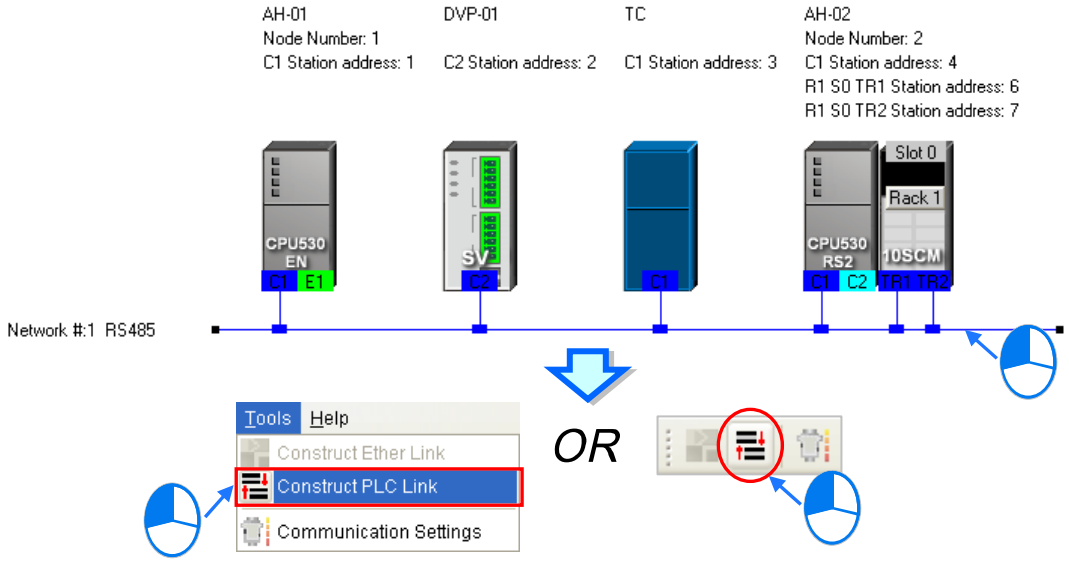


16.3.1 Opening the PLC Link Table Editor Window

Before creating a PLC Link, users have to make sure that all the network setting is correct. Please use one of the methods described below.

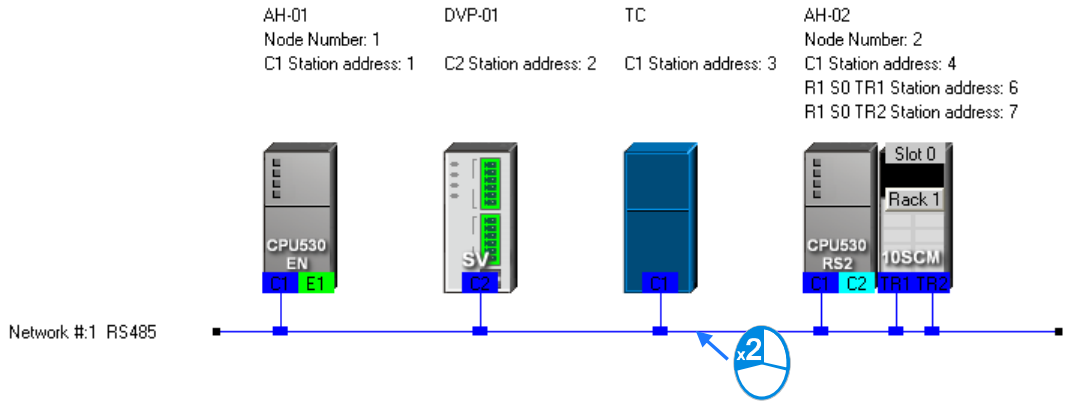
- **Method 1**

  After the users select a network, they have to click **Construct PLC Link** on the **Tools** menu, or on the toolbar.

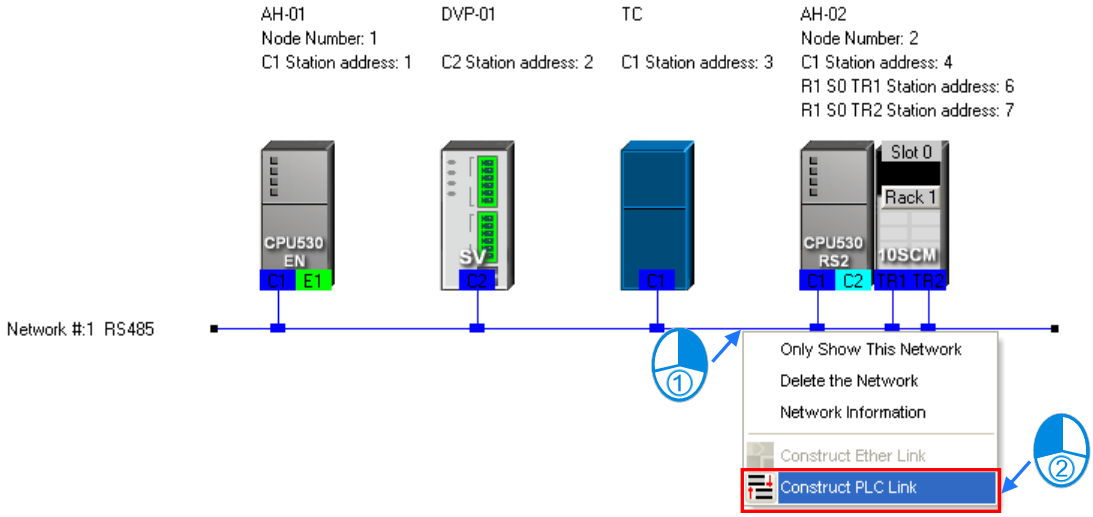


- **Method 2**

  Double-click network.



- **Method 3**
  Right-click a network, and then click **Construct PLC Link** on the context menu.

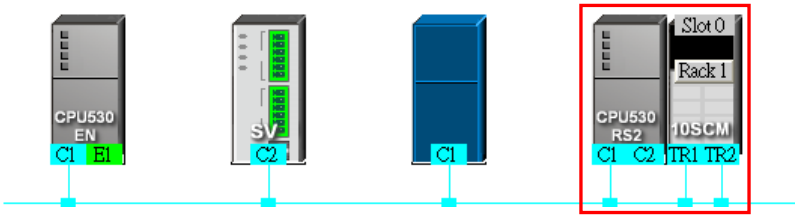


After the users use one of the methods described above, the **PLC Link Table Editor** window will be opened. The **PLC Link Table Editor** window leads the users to construct a PLC Link step by step. The steps of constructing a PLC Link are designating a port as a master station, setting communication parameters, and creating a data exchange table. The system leads the users to the operation screen displayed last time as soon as the **PLC Link Table Editor** window is opened. As a result, if the users construct a PLC Link for the first time, the screen displayed will lead the users to accomplish the first step.

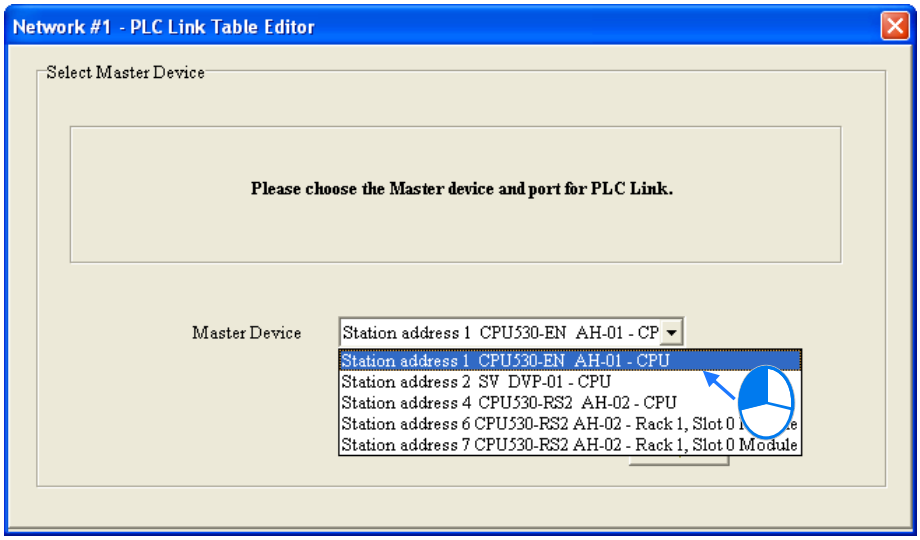
Besides, a PLC Link is executed through special relays and special registers. The construction tool introduced here is just a friendly user interface which helps users download the parameters related to a PLC Link to the related special relays and the related special registers. The exact execution of a PLC Link depends on the special relays and the special registers in a PLC. To ensure that the PLC Link constructed can operate normally, users have to make sure of the functions of the PLCs and the limitations of the PLCs with regard to the PLC Link.

16.3.2 Designating a Port as a Master Station (Step 1)

Only an AH500 series CPU module, an AH500 series module, a DVP series PLC, or a DVP series module can be designated as the master station. Not all ports of a PLC or a module can be designated as the master station, and therefore users have to read the usage of the PLC or the module before they designate a port as the master station. Besides, if some of the ports of a node can be designated as masters, the PLC which is a part of the node will execute the PLC Link no matter what port is designated as a master station.



(1) Select a port of a node in the **Master Device** drop-down list box. Only the ports which can be designated as master stations are listed.

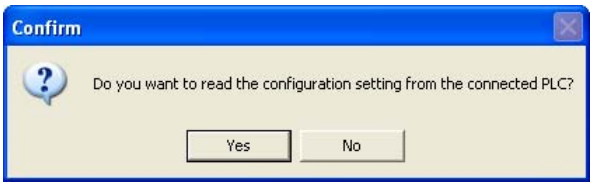


(2) After users click , the system will ask the users whether they want to upload the setting related to a PLC Link through the master station. If the users click **No**, they will be led to the second step. If the users click **Yes**, the setting related to a PLC Link in the PLC which is a part of the node will be uploaded through the master station, and the data uploaded will be displayed on the screen after the users are led to the third step. Before the users upload the data, they have to make sure of the following points.

(a) The users have to make sure that the computer and the port designated as a master station are connected by means of a communication cable.

(b) The users have to make sure that a driver has been created correctly, and the driver is started.

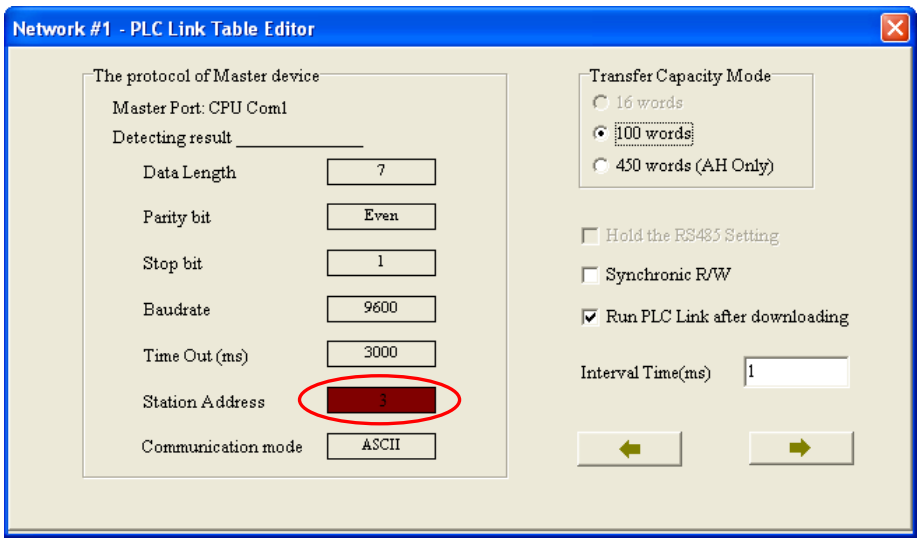
(c) The users have to make sure that they have completed the communication setting in NWCONFIG.

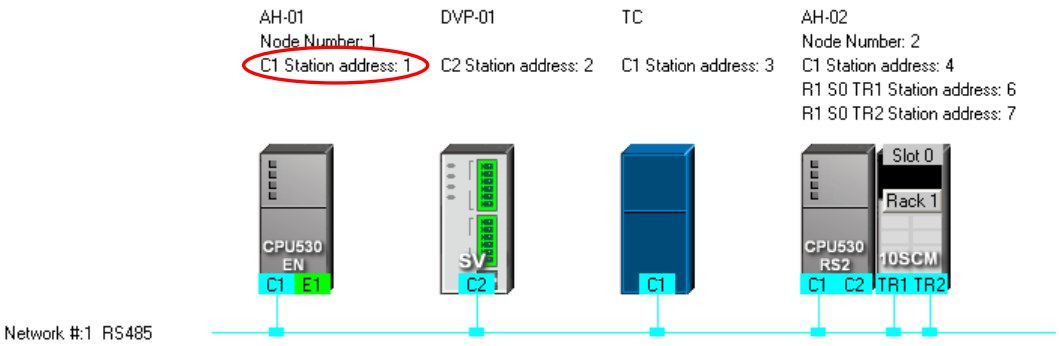


16.3.3 Setting Communication Parameters (Step 2)

After the system leads users to the second step, the users have to set the communication parameters in the **PLC Link Table Editor** window. The parameters uploaded through the master station are displayed at the left part of the window. The setting of the communication parameters of all the slave stations in the same network must be the same as the setting of the communication parameters of the master station. If no parameters are uploaded, “Unknown” will be shown in the

boxes at the left part of the window. If the station address uploaded is different form the station address assigned o the master station, the **Station Address** box will become red.



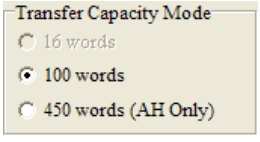


16

- **Transfer Capacity Mode**

  The users can set 16 data exchange groups or 32 data exchange groups, depending on the model selected. The users can select a maximum data length in the **Transfer Capacity Mode** section. Besides, the maximum data length which can be set varies with the PLC which is designated as a master station. Please refer to manuals for more information.

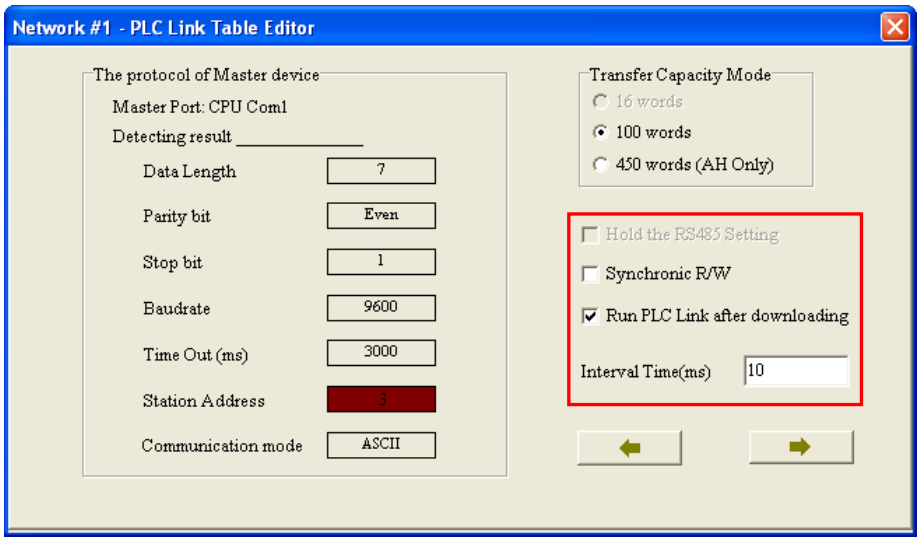
  If an AH500 series CPU module or an AH500 series module is designated as a master station, the **450 words (AH Only)** option button in the **Transfer Capacity Mode** section can be selected. Only AH500 series CPU modules allow 450-word data to be exchanged. As a result, if the **450 words (AH Only)** option button in the **Transfer Capacity Mode** section is selected, the DVP series PLCs and the other devices can not execute a PLC Link.



- **Hold the RS485 Setting**

  Generally speaking, the communication parameters in a DVP series PLC will be restored to the default values if the DVP series PLC is turned on after a power failure. However, if the **Hold the RS485 Setting** checkbox is selected, the communication parameters stored will be loaded again if a DVP series PLC runs after it is stopped. Please refer to manuals for more information about the communication parameters in DVP series PLCs.

- **Synchronic R/W**
  Generally speaking, a master station sends a writing command and a reading command to a slave station separately. If the **Synchronic R/W** checkbox is selected, the master station can complete reading and writing simultaneously by means of a specific Modbus function code (the hexadecimal code 17), and the efficiency of data exchange is increased. However, the users have to make sure that the devices involved in data exchange support the Modbus function code before they select the **Synchronic R/W** checkbox. If the devices do not support the Modbus function code, the Modbus code can not be identified, and the reading/writing of data will fail after they receive the commands from the master station.
- **Run PLC Link after downloading**
  If the **Run PLC Link after downloading** checkbox is selected, the PLC Link constructed will be enabled after the PLC Link constructed is downloaded to the PLC connected to the computer.
- **Interval Time (ms)**
  The users can specify how often the master station sends a command.

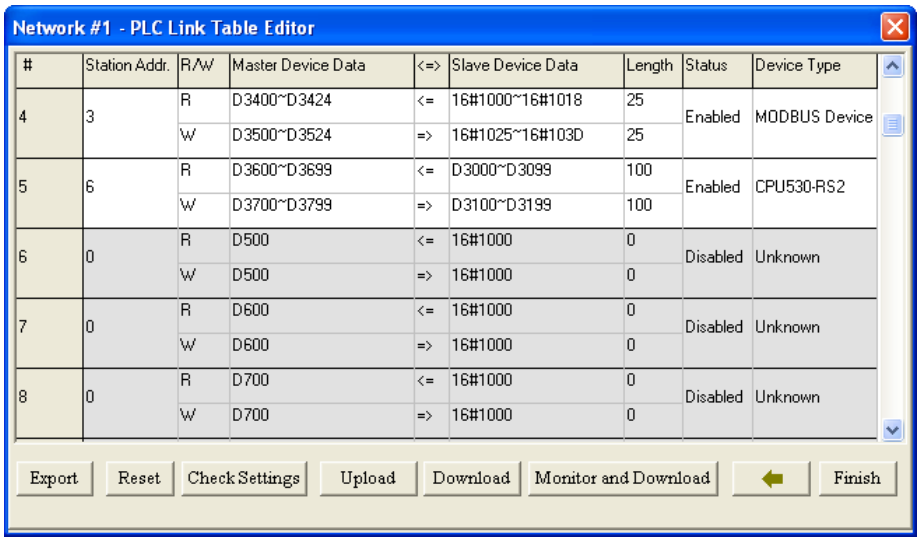


After the users click , the system will lead the users to the next step. If the users click , the system will lead the users to the previous step.

16.3.4 Creating a Data Exchange Table (Step 3)

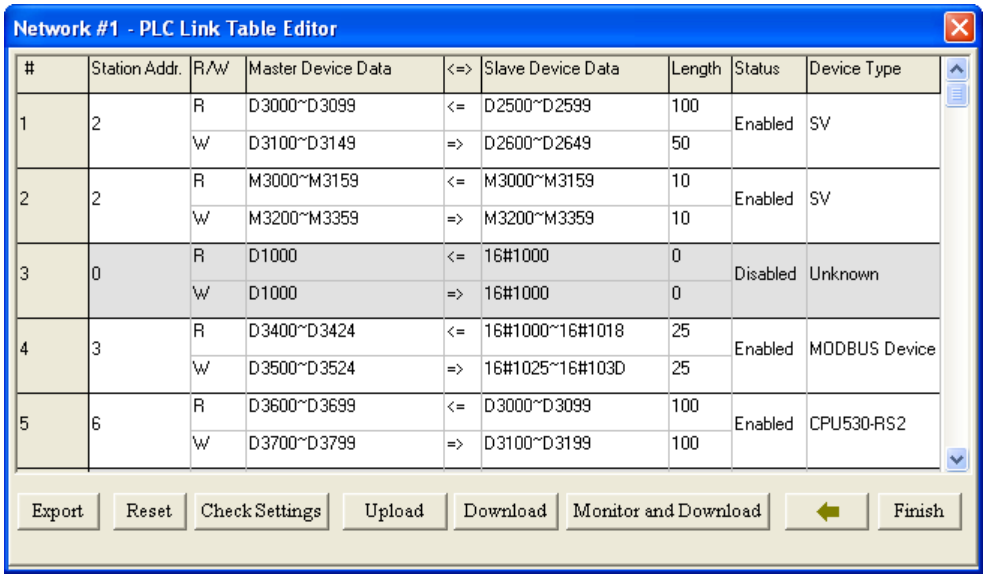
16.3.4.1 Introduction of a Data Exchange Table

The table below is a data exchange table. When a PLC Link is executed, the master station sends reading/writing commands to the slave stations according to the data exchange table created.



Column	Description
Serial Number (#)	The data exchange groups in the data exchange table in the **PLC Link Table Editor** window are numbered. Users can set 16 data exchange groups or 32 data exchange groups, depending on the model selected.
Station Addr.	The slave station address which belongs to a data exchange group is indicated. A station address can belong to several data exchange group. Besides, the station address 0 represents an undefined slave station, and is not a broadcast station address.
R/W	R: The master station reads the data in devices in a slave station. W: The master station writes data into devices in a slave station.
Master Device Data	The range of devices which are used in the master station is indicated.
<=>	<=: The master station reads the data in devices in a slave station. =>: The master station writes data into devices in a slave station.
Slave Device Data	The devices which are used in a slave station are indicated. If a slave station is a user-defined Modbus device, the devices used will be represented by hexadecimal addresses.
Length	A data length is indicated. A length indicates the number of devices used.
Status	Users can decide whether to involve a data exchange group in data exchange. If a data exchange group is set incorrectly, the data exchange group will not be involved in data exchange. If a data exchange group is not involved in data exchange, the related data will be on a gray ground.
Device Type	The device types shown are consistent with the device names shown in the working area in NWCONFIG. If a slave station address is assigned to a network module, the name of the PLC to which the network module is connected will be shown. Besides, if a slave station is undefined, it is unknown.

The figure below is an example of a data exchange table. Group #1, group #2, group #4, and group #5 are involved in data exchange. Please refer to the table below for more information.



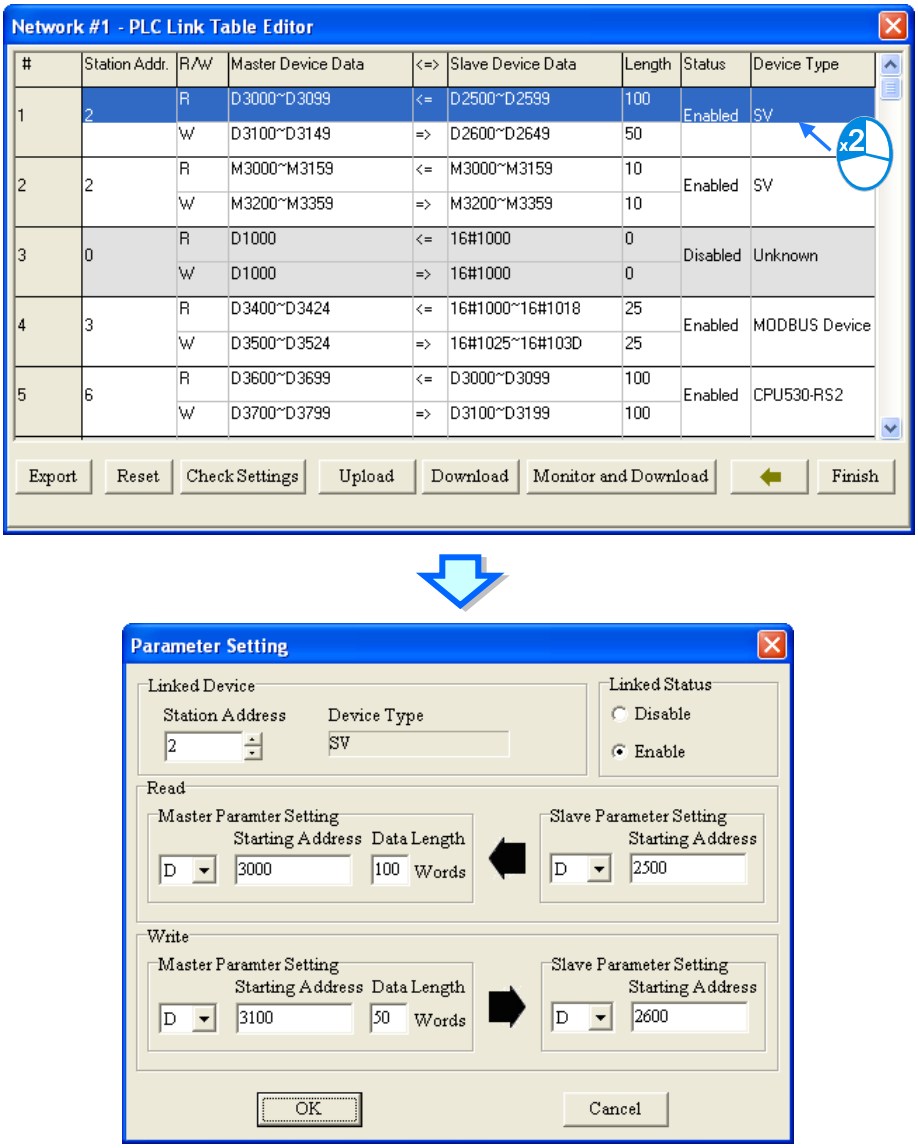
Serial number	Description
#1	The master station reads the data in D2500~D2599 in the slave station whose station number is 2, and stores the data in D3000~D3099 in itself. Meanwhile, the data in D3100~D3149 in the master station is written into D2600~D2649 in the slave station.
#2	The master station reads the data in M3000~M3159 in the slave station whose station number is 2, and stores the data in M3000~M3159 in itself. Meanwhile, the data in M3200~M3359 in the master station is written into M3200~M3359 in the slave station.
#4	The master station reads the data in 16#1000~16#1018 in the slave station whose station number is 3, and stores the data in D3400~D3424 in itself. Meanwhile, the data in D3500~D3524 in the master station is written into 16#1025~16#103D in the slave station.
#5	The master station reads the data in D3000~D3099 in the slave station whose station number is 6, and stores the data in D3600~D3699 in itself. Meanwhile, the data in D3700~D37999 in the master station is written into D3100~D3199 in the slave station.

Additional remark

If “Disabled” appears in the **Status** cell for a data exchange group, the data exchange group will not be involved in data exchange, and the system will ignore the setting in other cells for the data exchange group. Besides, if the setting of a data exchange group becomes incorrect after a modification, “Disabled” will appear in the **Status** cell for the data exchange group.

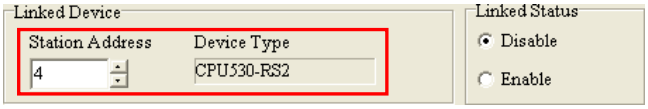
16.3.4.2 Setting a Data Exchange Group

If users want to set the parameters for a data exchange group, they can double-click the data exchange group in the data exchange table in the **PLC Link Table Editor** window.



- **Linked Device**

  The users can select a slave station address which belongs to this data exchange group. The device type displayed in the **Device Type** box is a model name displayed in the working area in NWCONFIG. If the slave station address selected is assigned to a port of a network module, the name of the PLC to which the network module is connected will be shown in the **Device Type** box. If the slave station address selected is undefined, or is not involved in the PLC Link constructed, “Unknown” will be displayed in the **Device Type** box. Besides, the station address assigned to the master station can not be selected, and a slave station address can belong to several data exchange groups.

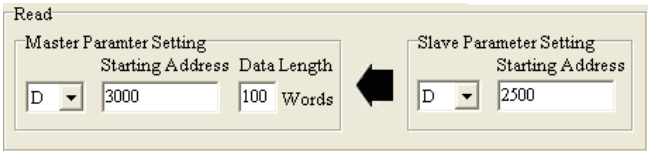


- **Linked Status**

  After the users select a slave station address which can be used, they can select the **Disable** option button, or the **Enable** option button in the **Linked Status** section.

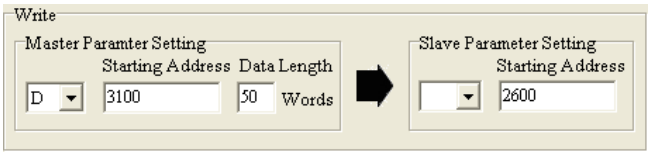
- **Read**

  The users can select a device type, type a starting address, and type a data length in the **Master Parameter Setting** section. They can select a device type, and type a stating address in the **Slave Parameter Setting** section. Generally speaking, the device types selected in the **Read** section must be the same, and do not have to be the same as the device types selected in the **Write** section.



- **Write**

  The users can select a device type, type a starting address, and type a data length in the **Master Parameter Setting** section. They can select a device type, and type a stating address in the **Slave Parameter Setting** section. Generally speaking, the device types selected in the **Write** section must be the same, and do not have to be the same as the device types selected in the **Read** section.



The data in a group of devices can be written into several different groups of devices, but the data in different groups of devices are not allowed to be stored in the same group of device. Take the figure above for example. The data in D3100~D3149 in the master station can be written into different slave stations, but D3000~D3099 in the master station can not receive the data in devices other than D2500~D2599 in the slave station whose station address is 2. Besides, the device types selected in the **Read**/**Write** section must be the same. However, the limitation on the setting of a data exchange group varies with the master station/slave station selected or the communication parameters selected.

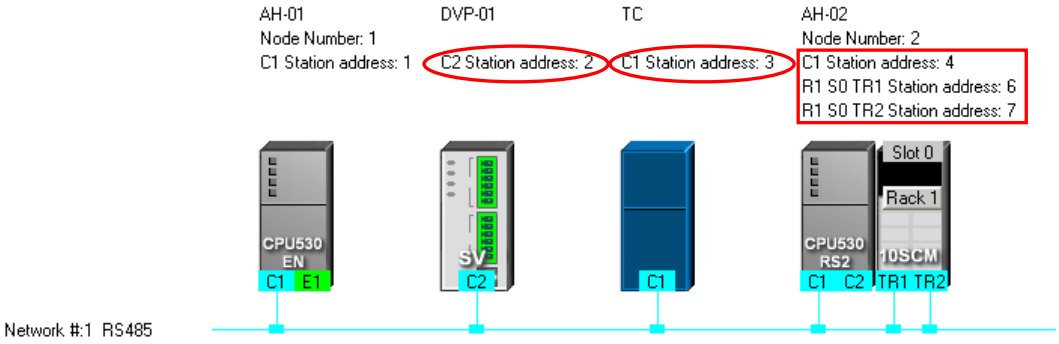
Related setting	Description
The **450 words (AH Only)** option button in the **Transfer Capacity Mode** section is selected.	Only AH500 series CPU modules allow 450-word data to be exchanged. As a result, if the **450 words (AH Only)** option button in the **Transfer Capacity Mode** section is selected, the DVP series PLCs and the other devices can not execute a PLC Link. The users can only select M devices, D devices, or L devices in the **Read**/**Write** section. The device types selected in the **Read**/**Write** section do not have to be the same.
The **16 words** option button in the **Transfer Capacity Mode** section is selected.	If the master station selected is a DVP series PLC, the **16 words** option button in the **Transfer Capacity Mode** section can be selected. If the **16 words** option button in the **Transfer Capacity Mode** section is selected, the starting device address in the **Read**/**Write** section will be a certain special register, and the users can only specify a data length.

Related setting	Description
The slave station selected is a user-define device.	If the slave station selected is a Modbus device, the devices used will be represented by hexadecimal addresses, and the users can select word devices or bit devices.
The **Synchronic R/W** checkbox is selected.	If the **450 words (AH Only)** option button is not selected, the users can only select D devices in the **Read/Write** section after the **Synchronic R/W** checkbox is selected.

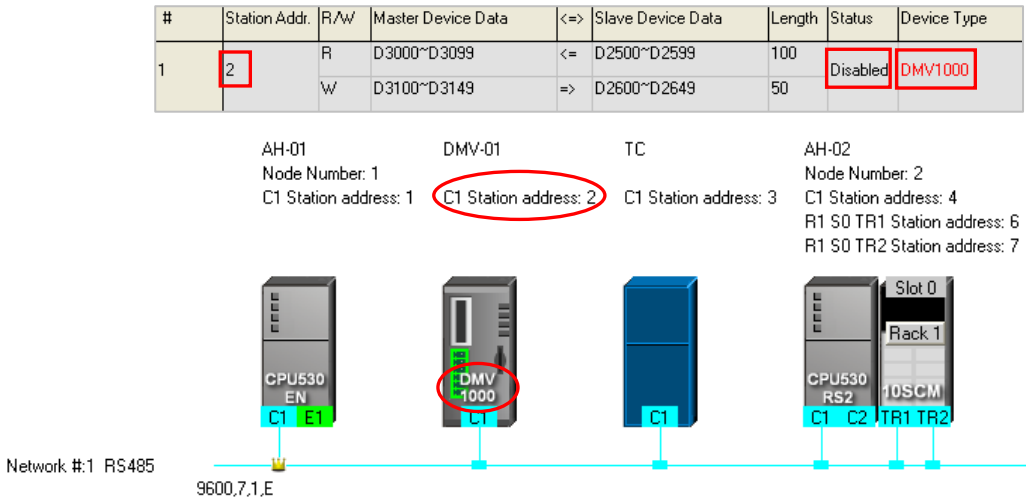
16.3.4.3 Consistency Between a Data Exchange Table and the Network Created in NWCONFIG

The device types shown in the data exchange table in the **PLC Link Table Editor** window are consistent with the device names shown in the working area in NWCONFIG. Besides, if a slave station address is assigned to a port of a network module, the name of the PLC to which the network module is connected will be shown.

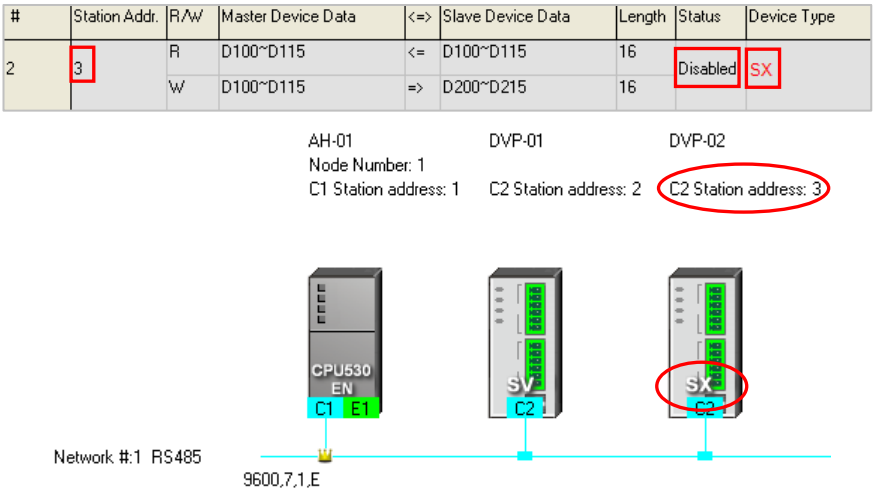
#	Station Addr.	R/W	Master Device Data	<=>	Slave Device Data	Length	Status	Device Type
1	2	R	D0~D99	<=	D100~D199	100	Enabled	SV
		W	D0~D99	=>	D200~D299	100		
2	3	R	D100~D199	<=	16#1000~16#1063	100	Enabled	MODBUS Device
		W	D100~D199	=>	16#1000~16#1063	100		
3	4	R	D200~D299	<=	D100~D199	100	Enabled	CPU530-RS2
		W	D200~D299	=>	D200~D299	100		
4	6	R	D300~D399	<=	D100~D199	100	Enabled	CPU530-RS2
		W	D300~D399	=>	D200~D299	100		
5	7	R	D400~D499	<=	D100~D199	100	Enabled	CPU530-RS2
		W	D400~D499	=>	D200~D299	100		



If users change the model to which a station address is assigned in NWCONFIG after a PLC Link is constructed, the new model name appearing in the **Device Type** cell for the station address will be in red, and “Disabled” will appear in the **Status** cell for the station address. The users have to check whether the setting is correct. If the setting is incorrect, the users have to modify the incorrect parts. After the users modify the incorrect parts, they can select the **Enable** option button in the **Linked Status** section.

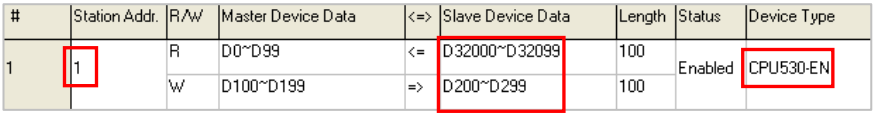


If the data uploaded from the master station which is an AH500 series CPU module is not consistent with the network created in NWCONFIG, the device types shown in the data exchange table in the **PLC Link Table Editor** window will still be consistent with the device names shown in the working area in NWCONFIG, the incorrect device types will be in red, and “Disabled” will appear in the **Status** cell for the incorrect device types. The users have to check whether the setting is correct. If the setting is incorrect, the users have to modify the incorrect parts. After the users modify the incorrect parts, they can select the **Enable** option button in the **Linked Status** section.



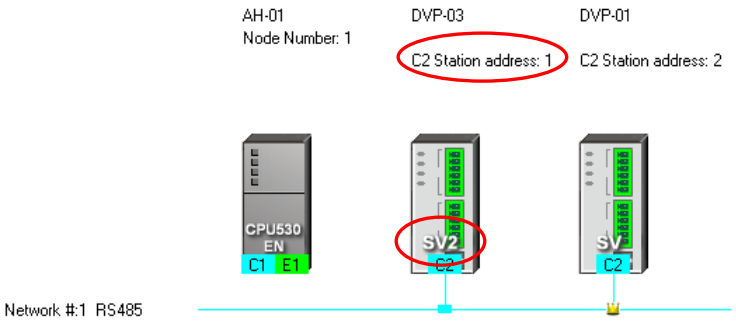
The device types in the data exchange table in the **PLC Link Table Editor** window can not be stored in a DVP series PLC, and therefore the users do not know whether the data uploaded from the DVP series PLC is consistent with the network created in NWCONIFG. The device types shown in the data exchange table in the **PLC Link Table Editor** window is consistent with the device names shown in the working area in NWCONFIG. If the device ranges set for a model is incorrect, “Disabled” will appear in the **Status** cell for the model. The users have to modify the incorrect parts. After the users modify the incorrect parts, they can select the **Enable** option button in the **Linked Status** section.

The figure below is a data exchange table which is downloaded to a DVP-SV series PLC. The slave station to which the station address 1 is assigned is AHCPU530-EN.



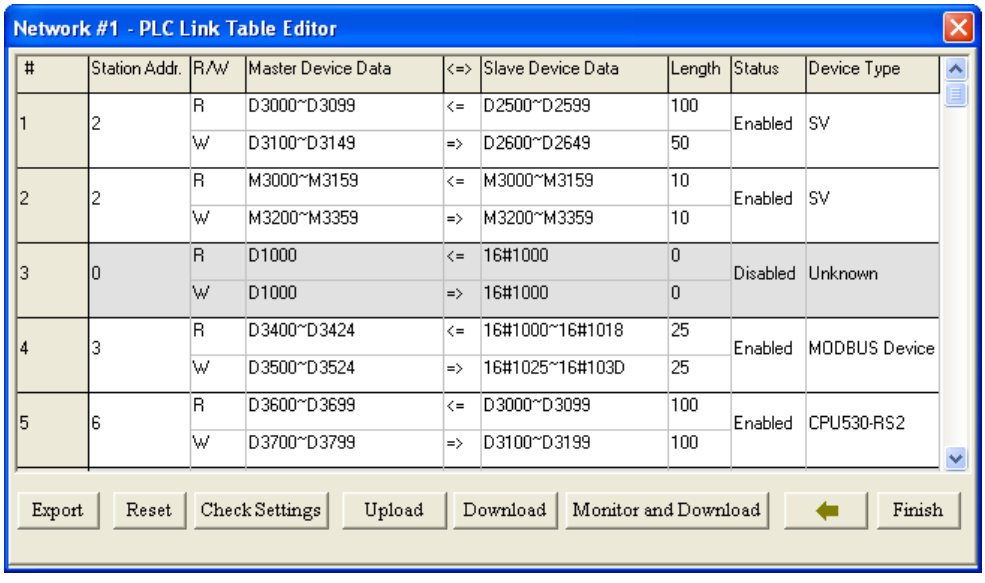
The present network created in NWCONFIG is shown below. The slave station to which the station address 1 is assigned is a DVP-SV2 series PLC. After the data exchange table in the DVP-SV series PLC is uploaded, the device types shown in the data exchange table in the **PLC Link Table Editor** window will be consistent with the device names shown in the working area in NWCONFIG. The device ranges set for the DVP-SV2 series PLC is not within the device range for DVP-SV2 series PLC, and therefore “Disabled” appears in the **Status** cell for the DVP-SV2 series PLC.

#	Station Addr.	R/W	Master Device Data	<=>	Slave Device Data	Length	Status	Device Type
1	1	R	D0~D99	<=	Address error	100	Disabled	SV2
		W	D100~D199	=>	Address error	100		



16.3.4.4 Managing a Data Exchange Table

There are buttons under the data exchange table in the **PLC Link Table Editor** window. Please refer to the table below for more information about the buttons. If users click , the system will lead the users to the previous step.



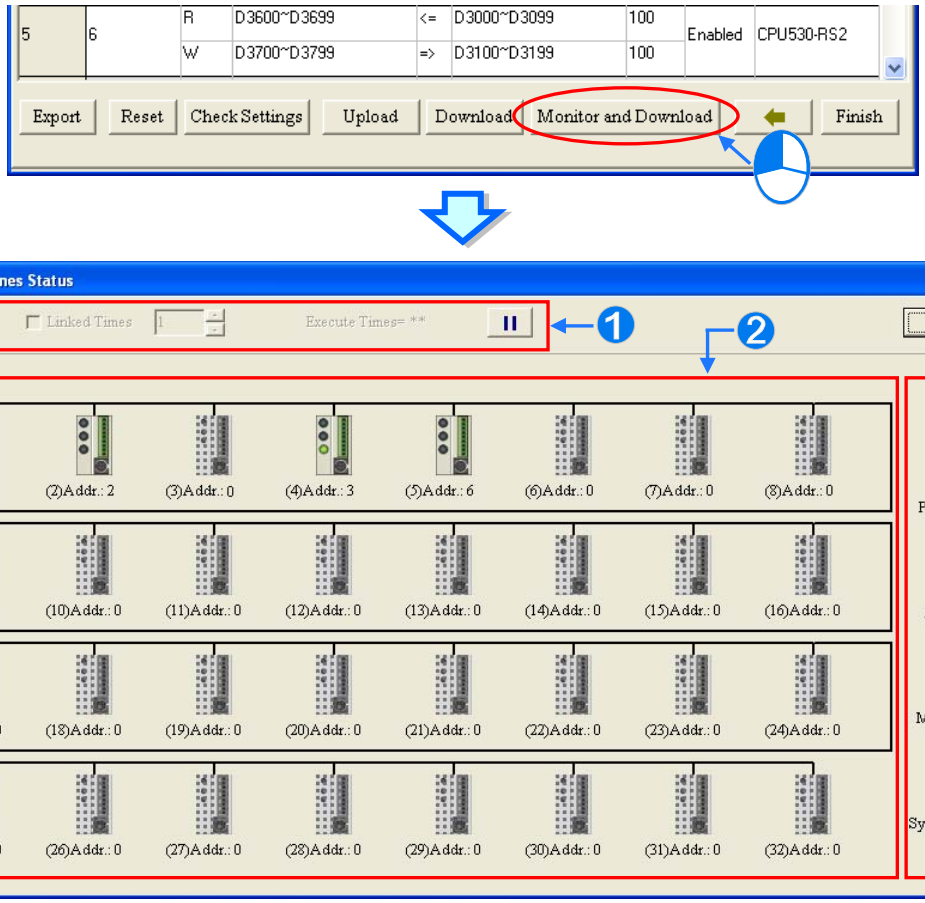
Item	Description
Export	The data in the data exchange table can be exported as a CSV file. The users can edit the CSV file through Microsoft Excel. The CSV file can also be used as reference material for another development work.
Reset	After the users click the button, the setting in the data exchange table will be restored to the initial setting.
Check Settings	After the users click the button, the data exchange table will be checked.

Item	Description
Upload	After the users click the button, the parameters related to a PLC Link in the master station will be uploaded.
Download	After the users make sure that the setting of a PLC Link is correct, they can download the setting to the master station by clicking the button.
Monitor and Download	After the users make sure that the setting of a PLC Link is correct, they can download the setting to the master station. After the setting is downloaded to the master station, the **Linked Machines Status** window will be opened automatically.
Finish	After the users click this button, the **PLC Link Table Editor** window will be closed. Before the **PLC Link Table Editor** window is closed, the system will ask the users whether they want to save the modifications they make.

During the execution of a PLC Link, the slave stations passively receive reading/writing commands from the master station. As a result, the parameters set in the data exchange table in the **PLC Link Table Editor** window are for the master station. If the users click **Upload**, the parameters related to a PLC Link in the master station will be uploaded. If the users click **Download**, the setting in the data exchange table in the **PLC Link Table Editor** window will be downloaded to the master station. Likewise, the master station will be monitored if the users click **Monitor and Download**. Before the users click **Upload**, **Download**, or **Monitor and Download**, they have to make sure that the device which is connected to the computer is the PLC which is designated as a master station, and the computer can communicate with the PLC normally. Please refer to section 16.1.3 for more information.

16.3.5 Monitoring a PLC Link

Before users monitor the PLC Link constructed, they have to make sure that the system is connect to the PLC which is designated as a master station normally. Please refer to section 16.1.3 for more information. After the users make sure that the PLC Link constructed is correct, they can click **Monitor and Download** under the data exchange table in the **PLC Link Table Editor** window to open the **Linked Machines Status** window. Besides, a PLC link can be executed normally only if the master station runs. As a result, the users have to make sure that the PLC which is designated as a master station runs before they monitor the PLC Link constructed.



❶ **Setting area**: The users can set the PLC Link constructed.

❷ **Monitoring area**: The status of the data exchange between the master station and the slave stations is displayed in this area. The users can also edit registers online in this area.

❸ **Indicators**: The LED indicators indicate the status of the PLC Link constructed. Please refer to the table below for more information.

Indicator	Description
PLC Run	When the PLC which is designated as a master station runs, the indicator is ON.
PLC Link Run	When the master station executes the PLC Link constructed, the indicator is ON.
Auto Mode	When the PLC Link constructed is executed automatically, the indicator is ON.
Manual mode	When the PLC Link constructed is executed manually, the indicator is ON. After the master station polls the slave stations a certain number of times, the indicator will be OFF.
Synchronic R/W	When the synchronic reading/writing function is enabled, the indicator is ON.

- **Setting area**

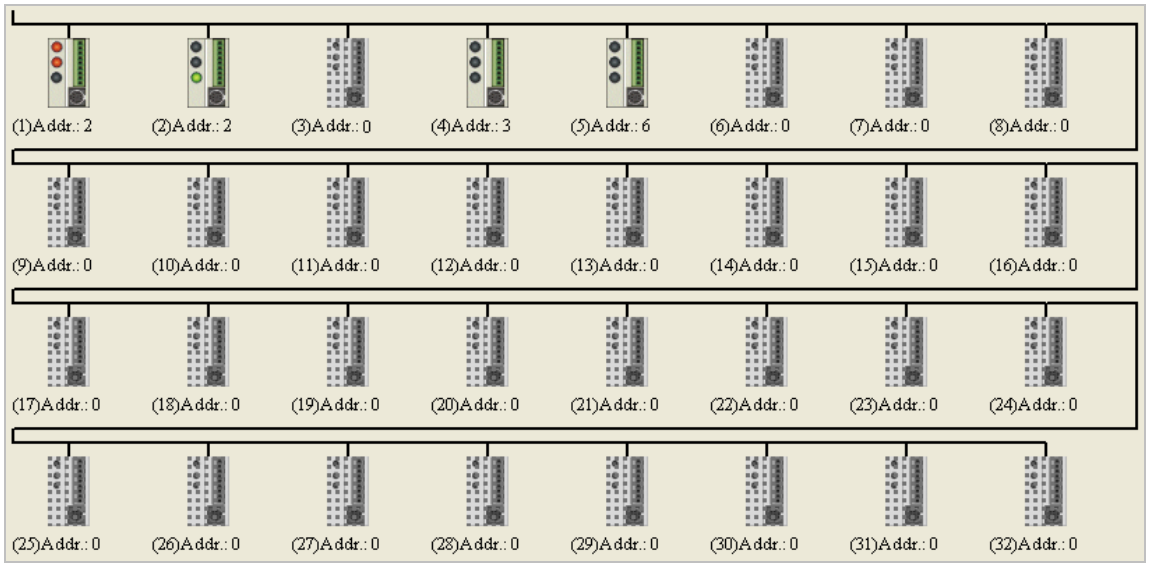
  The users can set the PLC Link constructed. If the users set the PLC Link constructed, the values in the related special registers in the master station and the states of the related special relays in the master station will be changed. After the users click **Close**, the status of the PLC Link will remain unchanged. As a result, the users have to make sure that the status of the PLC Link constructed is correct before they click **Close**.

1. When the computer is connected to the PLC which is designated as a master station, the indicator blinks.
2. If the **Linked Times** checkbox is selected, the PLC Link constructed will be executed manually. If the **Linked Times** checkbox is unselected, the PLC Link constructed will be executed automatically.
3. If the PLC Link constructed is executed manually, the users can set the number of times the master station polls the slave stations. The master station can poll the slave stations 65535 times at most. After the master station polls the slave stations a certain number of times, the PLC Link constructed will not be executed.
4. If the PLC Link constructed is executed manually, the number of times the PLC Link is executed will be displayed. If the master station exchanges data with all the slave stations once, the number of times the PLC Links constructed is executed will be one.
5. The users can start or stop the execution of the PLC Link constructed by clicking the button.

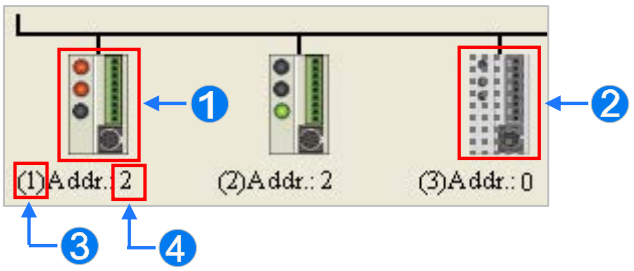
If the system is disconnect from the PLC which is designated as a master station while the PLC Link constructed is monitored, the **Continue to Monitor** button will appear in the upper right corner of the **Linked Machines Status** window. After the users eliminate the problem which results in the disconnection, they can click the **Continue to Monitor** button.

- **Monitoring area**

  When the PLC Link constructed is executed, the master station exchanges data with the slave stations. The status of the data exchange between the master station and the slave stations are displayed in this area.

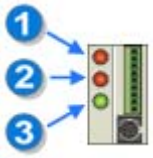
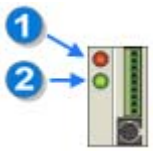


The small pictures in the monitoring area indicate the status of the main station rather than the statuses of the slave stations. The small pictures in the monitoring area are described below.

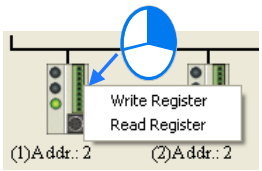


1. The indicators on the small picture indicate the status of the data exchange group represented by the small picture.
2. The data exchange group represented by the small picture is not involved in data exchange.
3. The number corresponds to a serial number in the data exchange table. It represents the serial number of the data exchange group represented by the small picture.
4. The number corresponds to a slave station address in the data exchange table. It represents the slave station address which belongs to the data exchange group represented by the small picture.

Besides, the PLC Link error flags in an AH500 series CPU module are slightly different from the PLC Link error flags in a DVP series PLC, and therefore the small picture representing an AH500 series CPU module is different from the small picture representing a DVP series PLC. Please refer to the following table for more information.

Master station	Small picture	Description
AH500 series CPU module		1. If an error occurs when data in a slave station is read, the indicator will be ON. 2. If an error occurs when data is written into a slave station, the indicator will be ON. 3. When data exchange is performed, the indicator will be ON.
DVP series PLC		1. If an error occurs when data in a slave station is read, or an error occurs when data is written into a slave station, the indicator will be ON. 2. When data exchange is performed, the indicator will be ON.

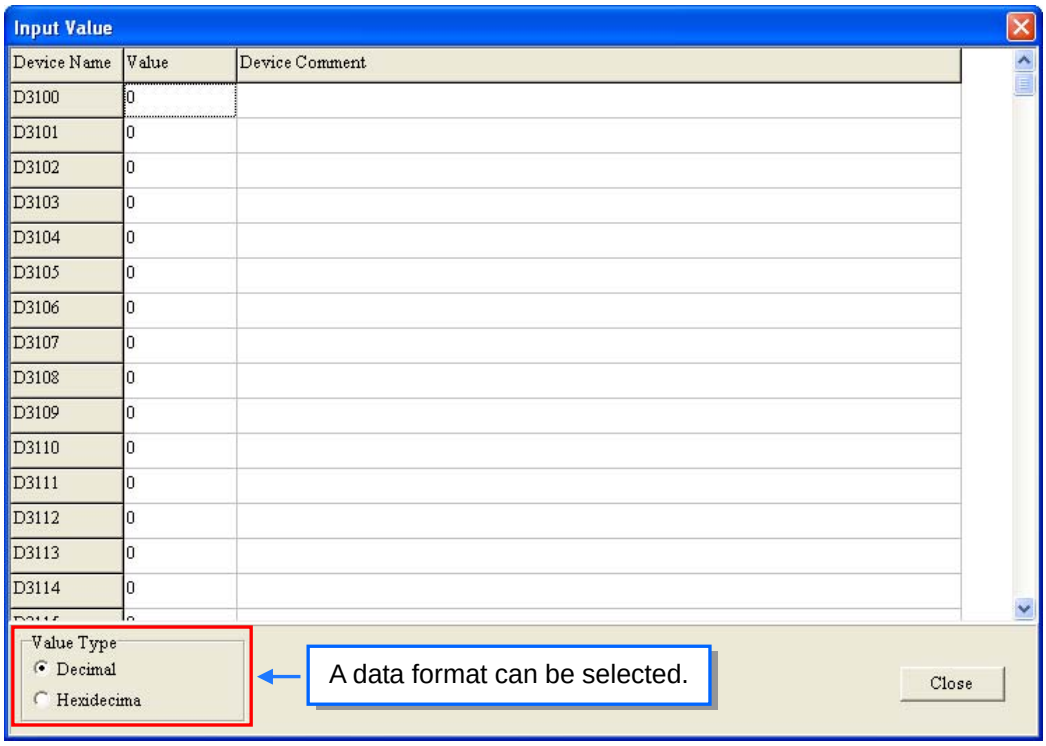
After the users select a small picture, and right-click the small picture, they can click **Write Register** or **Read Register** on the context menu.



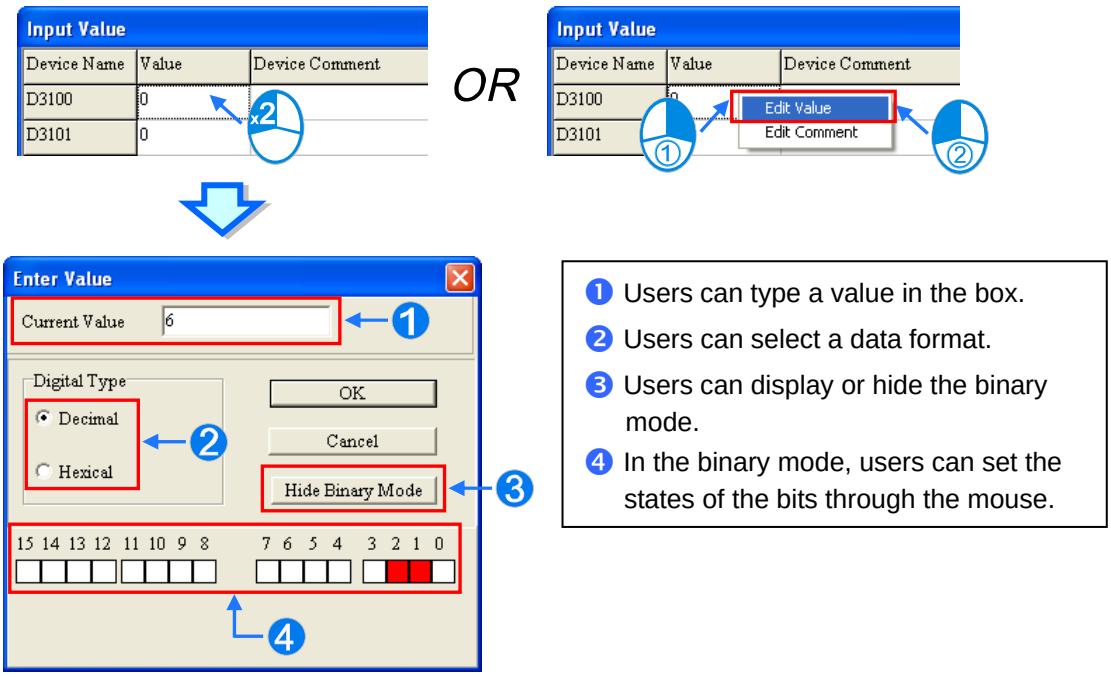
- **Write Register**: The data written into the slave station specified is stored in registers in the master station. For example, D3100~D3149 in the figure below are write registers.
- **Read Register**: The data **read** from the slave station specified is stored in registers in the master station. For example, D3000~D3099 in the figure below are read registers.

#	Station Addr.	R/W	Master Device Data	<=>	Slave Device Data	Length	Status	Device Type
1	2	R	D3000~D3099	<=	D2500~D2599	100	Enabled	SV
		W	D3100~D3149	=>	D2600~D2649	50		

After the users click **Write Register** or **Read Register** on the context menu, a correponding register monitoring table will appear.

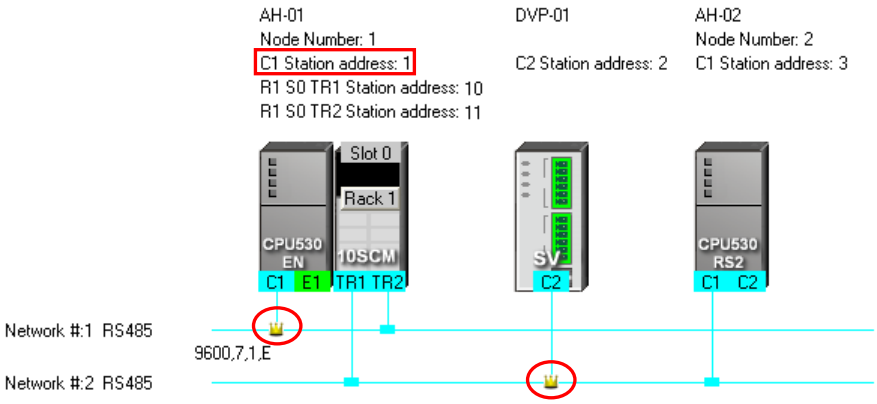


If the users double-click the **Value** cell for a device, or click **Edit Value** on the context menu after they right-click the **Value** cell, they can type a value in the **Enter Value** window. If the users double-click the **Device Comment** cell for a device, or click **Edit Comment** on the context menu after they right-click the **Device Comment** cell, they can make a comment on the device. The comments made are for the PLC Link constructed. They are not related to the comments on the devices in the program created in ISPSoft.

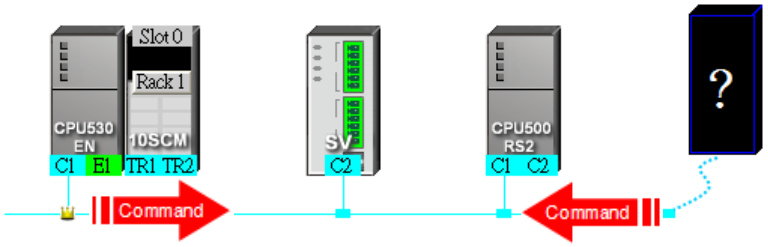


16.3.6 Importing Points About Constructing a PLC Link

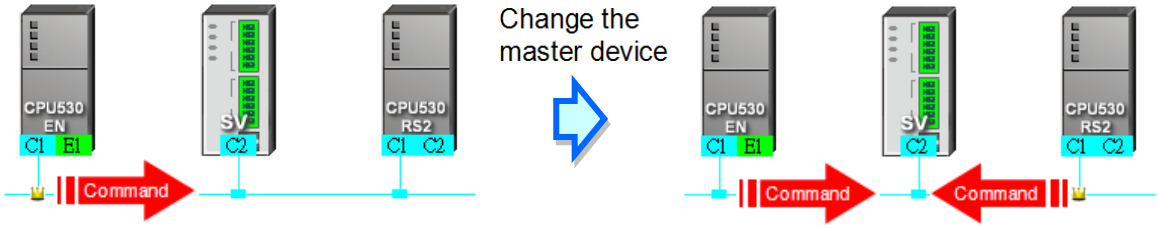
There is only one master station in an RS-485 network. If the node designated as a master station has several serial ports, the node can not be designated as a master station in another RS-485 network, but can be designated as a slave station in another RS-485 network. In the figure below, C1 on the node AH-01 is designated as a master station in network #1, and TR1 on the node AH-01 is designated as a slave station in network #2. Besides, C1 on AH-01 and TR2 on AH-01 are connected to network #1. The station addresses assigned to these two ports are different. As a result, TR2 is allowed to be designated as a slaved station in network #1.



When a PLC Link is executed, the master station sends reading/writing commands to the slave stations according to the data exchange table created. Another master station which can send commands is not allowed to exist. If there is an unknown device which does not appear in NWCONFIG and which can send commands in an RS-485 network, or there is a device which is designated as a slave station and which can send reading/writing commands in an RS-485 network, an error occurs when more than one device sends commands. As a result, after users create a network in NWCONIFG, they have to make sure that the actual connection is consistent with the network created in NWCONFIG.



Besides, if the users do not set the parameters related to a PLC Link in the original master station again before they designate another device as a master station, an error occurs when more than one device sends reading/writing commands. As a result, if the users want to designate another device as a master station, they have to make sure that the original master station does not execute the original PLC Link.



*** In addition to the conditions mentioned above, users have to prevent two devices or more than two devices from sending reading/writing commands simultaneously in any conditions.**

16.4 Constructing an Ether Link

16.4.1 Introduction of an Ether Link

An Ether Link is a network mechanism for data exchange performed through an Ethernet connection. If there are several nodes in an Ethernet network, users can create a mechanism for data exchange in the network, and select a start mode. If the parameters which are set are downloaded to the PLCs in the network, the systems of the PLCs perform data exchange according to the start mode selected when the PLCs run. The users do not have to write a redundant program. Besides, only AH500 series CPU modules support Ether Links.

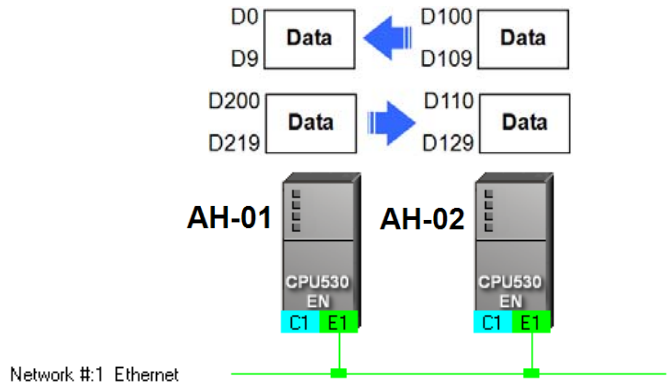
A PLC Link is a master/slave model. There is only one master station in an RS-485 network, and the other stations which are slave stations passively receive reading/writing commands from the master station. Compared with a PLC Link, an Ether Link adopts a safer data request mechanism. It is the data demanding nodes in an Ethernet network that execute an Ether Link.

An Ether Link is not a master/slave model. It allows a node to send reading commands which ask for data to other nodes. The nodes will send the data to the node after they receive the reading commands. Owing to the fact that a node can not send writing commands to other nodes, the use of an Ether Link is safer than the use of a PLC Link. Besides, all the nodes in an Ethernet network can send reading commands through TCP/IP, and the system automatically manages the transmission of packets through TCP/IP. Compared with a PLC Link, an Ether Link is more efficient.

Please refer to the example below for more information. If users want to create an Ether Link shown below, they have to create a data exchange table for the two data request nodes AH-01 and AH-02.

(a) AH-01 reads the data in D100~D109 in AH-02, and stores the data in D0~D9 in itself.

(b) AH-02 reads the data in D200~D219 in AH-01, and stores the data in D110~D129 in itself.



#	Device Name	Register	Starting Address	Range	<->	#	Device Na...	Register	Starting Address	Range	Size
1	AH-01	D	0	D0 ~ D9	<-	2	AH-02	D	100	D100 ~ D109	10
2	AH-02	D	110	D110 ~ D129	<-	1	AH-01	D	200	D200 ~ D219	20

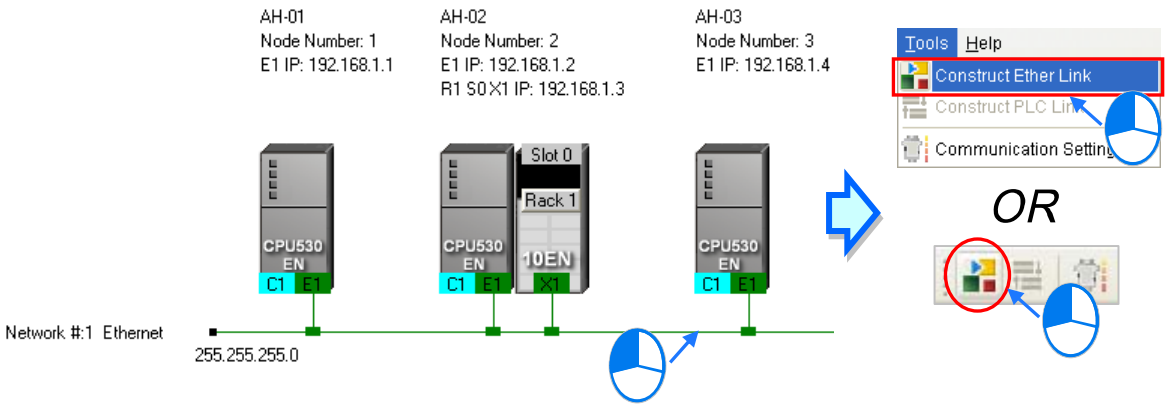
An Ether Link is based on a network. If a node is connected to several networks, users can set several groups of parameters related to Ether Links for the networks. As a result, there may be several groups of parameters related to Ether Links in a PLC. Owing to the fact that an Ether Link is not a master/slave model, each node in a network can be a data demanding node and a data supply node at the same time. In the first piece of data in the table above, AH-01 is a data requiring node, and AH-02 is a data providing node. In the second piece of data in the table above, AH-01 is a data providing node, and AH-02 is a data requiring node.

16.4.2 Opening the Ether Link Configuration Window

Before users construct an Ether Link, they have to make sure that a network is set correctly. There are three ways to open the **Ether Link Configuration** window.

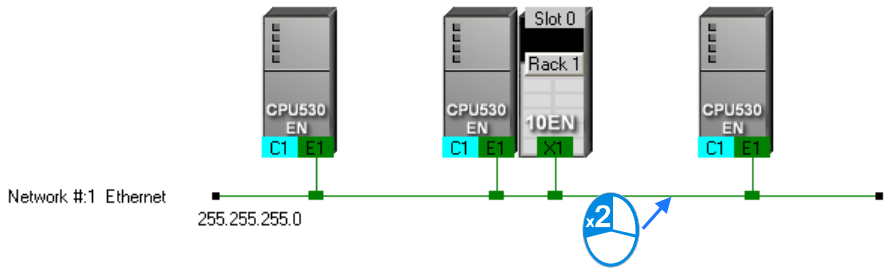
- **Method 1**

  After the users select a network, they have to click **Construct Ether Link** on the **Tools** menu, or on the toolbar.

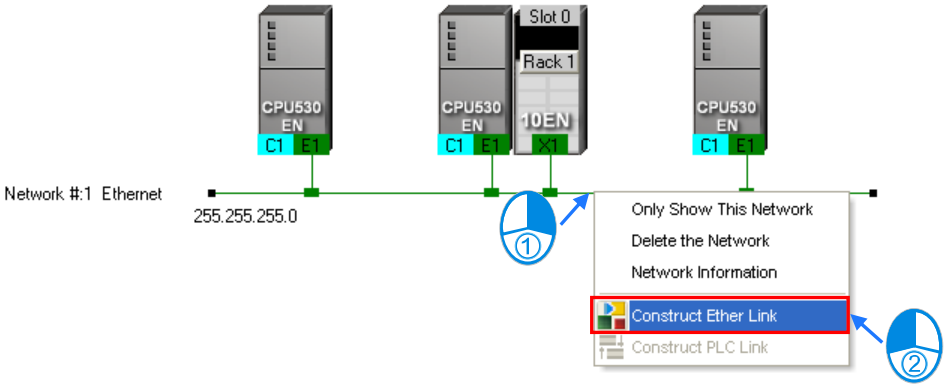


- **Method 2**

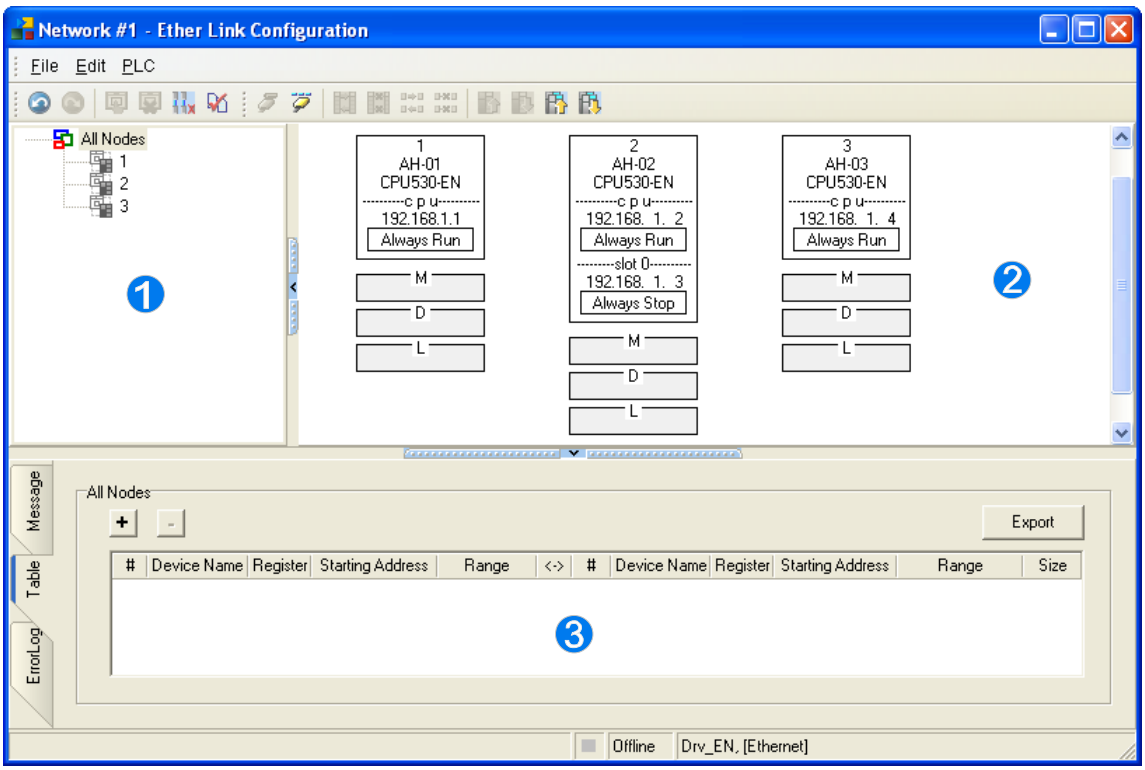
  Double-click a network.



- **Method 3**
  Right-click a network, and then click **Construct Ether Link** on the context menu.



The **Ether Link Configuration** window is shown below.

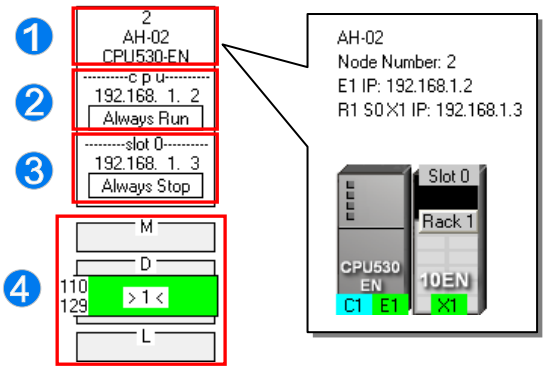


❶ **Node list**: After the users click a node, the node and the nodes of which the node demands data will be shown in the display area.

❷ **Display area**: The information about a node and the nodes of which the node demands data is displayed in this area.

❸ **Information area**: The users can click the **Message** tab, the **Table** tab, or the **Error Log** tab.

A node shown in the display area in the figure above is described below.



❶ The information is composed of a node number, a PLC name, and a model name.

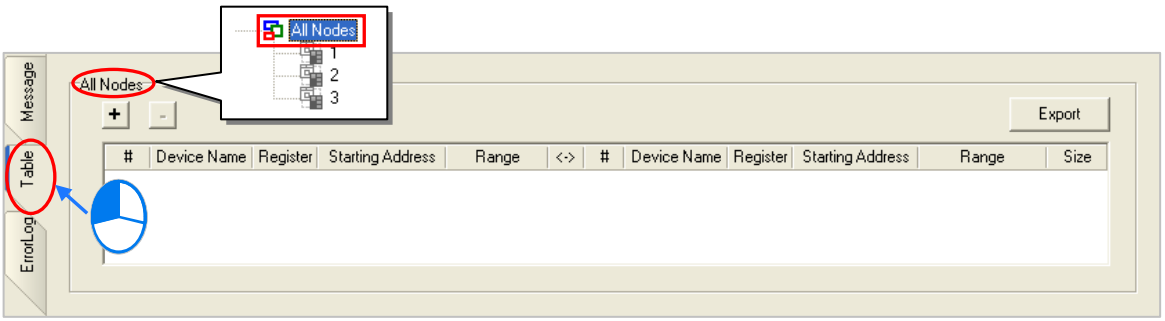
❷ The IP address assigned to AH500 series CPU module is 192.168.1.2. The start mode of the Ether Link constructed is **Always Run**.

❸ The IP address assigned to the module installed in slot 0 is 192.168.1.3. The start mode of the Ether Link constructed is **Always Stop**.

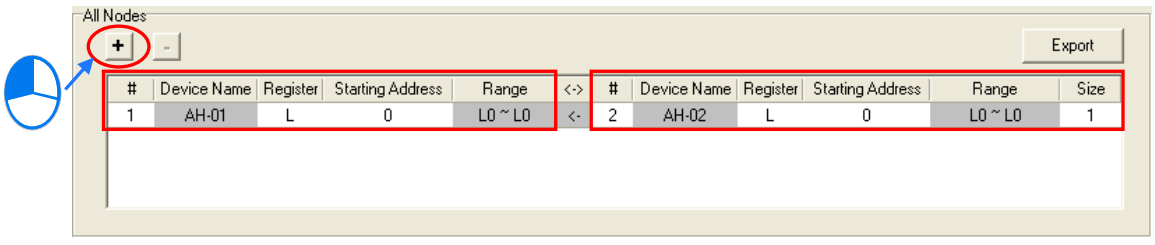
❹ >1< in the D block indicates that the D block demands data of node 1. The numbers at the left side of the D block indicate that the data demanded of node 1 will be stored in D110~D119 in node 2. The color assigned to the D block depends on the node number in the D block. Owing to the fact that there is not any information in the M block and the L block, the M block and the L block does not demand any data of other devices.

16.4.3 Creating and Managing a Data Exchange Table

If users want to create a data exchange table, they have to click the **Table** tab in the information area. The node which is selected on the node list is shown in the upper left corner of the information area. The data in the table is related to the node selected.

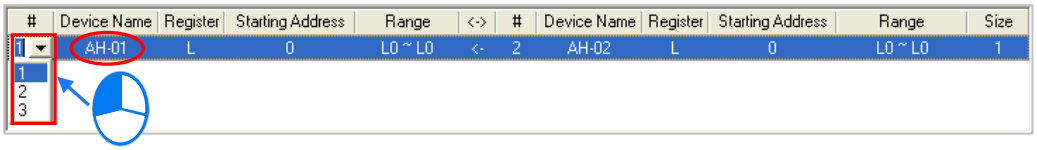


If the users click + , a new piece of data will be added to the table. A piece of data is composed of two parts. The left part of the data in the figure below indicates that the data demanding node AH-01 will store the data demanded in L0 in itself, and the right part of the data in the figure below indicates that the data supply node AH-02 will supply the data in L0 in itself.

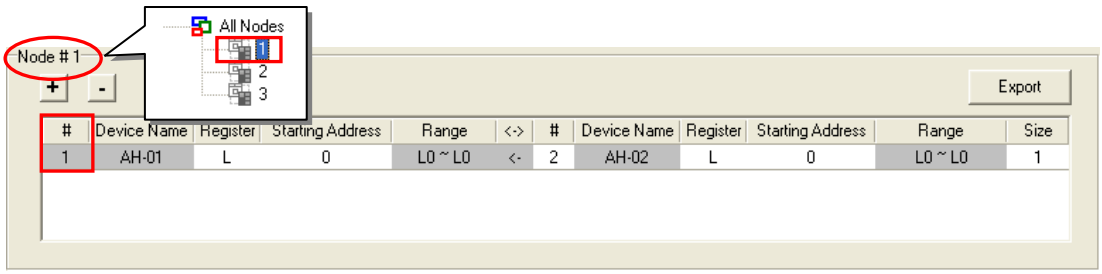


The steps of setting a data exchange group are as follows.

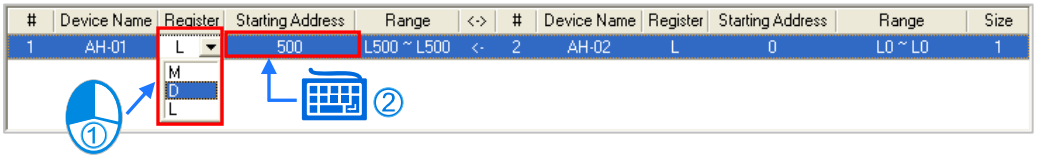
(1) Select a node number in the # cell for the data demanding node. After the users select a node number, the PLC name corresponds to the node number will be appear in the **Device Name** cell for the node number.



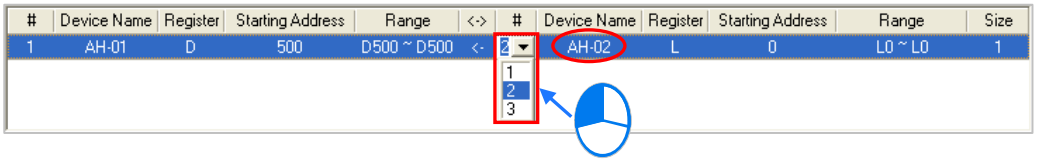
If the users select a specific node number rather than **All Nodes** on the node list, the data in the table will be related to the specific node number selected, and the fixed node number in the # cell for the data demanding node will be the specific node number selected on the node list.



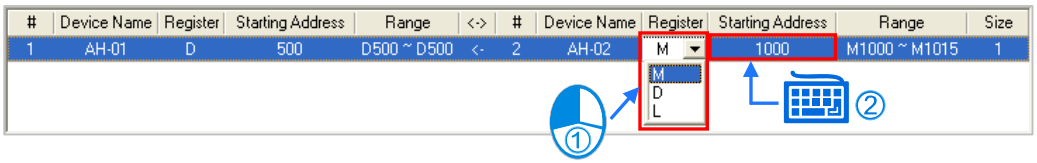
(2) Select a device type in the **Register** cell for the data demanding node, and type an address in the **Starting Address** cell for the data demanding node.



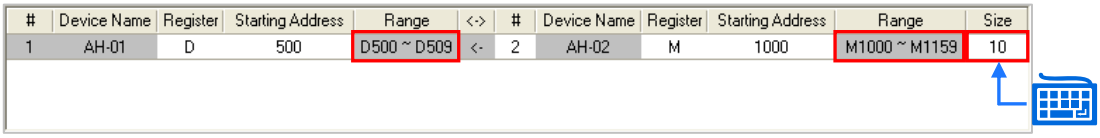
(3) Select a node number in the # cell for the data supply node. The node number in the # cell for the data demanding node can not be the same as the node number in the # cell for the data supply node. After the users select a node number, the PLC name corresponds to the node number will be appear in the **Device Name** cell for the node number.



(4) Select a device type in the **Register** cell for the data supply node, and type an address in the **Starting Address** cell for the data supply node. The device type selected in the **Register** cell for the data supply node does not have to be the same as the device type selected in the **Register** cell for the data demanding node.

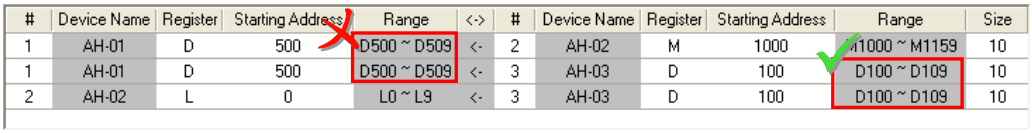


(5) Type a data length in the **Size** cell. A word is a unit. The maximum data length is 1900 words. After the users type a data length, the device ranges in the **Range** cells will change according to the data length typed.



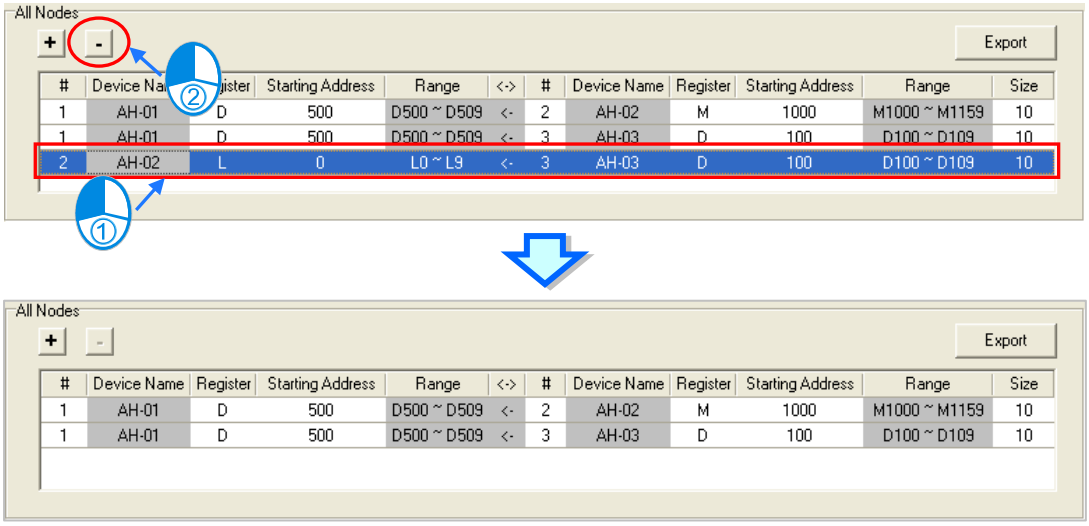
#	Device Name	Register	Starting Address	Range	<->	#	Device Name	Register	Starting Address	Range	Size
1	AH-01	D	500	D500 ~ D509	<-	2	AH-02	M	1000	M1000 ~ M1159	10

The users can create data exchange groups by following the steps described above. The device range in the **Range** cell for a data demanding node can not overlap the device range in the **Range** cell for another data demanding node whereas the device range in the **Range** cell for a data supply node can overlap the device range in the **Range** cell for another data supply node. In other words, different demanders can ask for the same data, but different data can not be store in the same block.



#	Device Name	Register	Starting Address	Range	<->	#	Device Name	Register	Starting Address	Range	Size
1	AH-01	D	500	D500 ~ D509	<-	2	AH-02	M	1000	M1000 ~ M1159	10
1	AH-01	D	500	D500 ~ D509	<-	3	AH-03	D	100	D100 ~ D109	10
2	AH-02	L	0	L0 ~ L9	<-	3	AH-03	D	100	D100 ~ D109	10

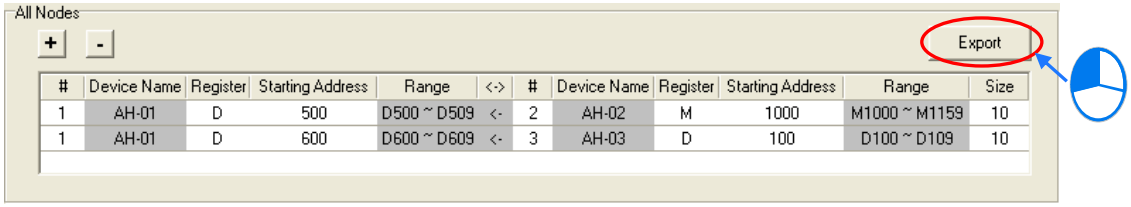
If the users want to delete a piece of data, they can click the piece of data, and click **-**.



#	Device Name	Register	Starting Address	Range	<->	#	Device Name	Register	Starting Address	Range	Size
1	AH-01	D	500	D500 ~ D509	<-	2	AH-02	M	1000	M1000 ~ M1159	10
1	AH-01	D	500	D500 ~ D509	<-	3	AH-03	D	100	D100 ~ D109	10
2	AH-02	L	0	L0 ~ L9	<-	3	AH-03	D	100	D100 ~ D109	10

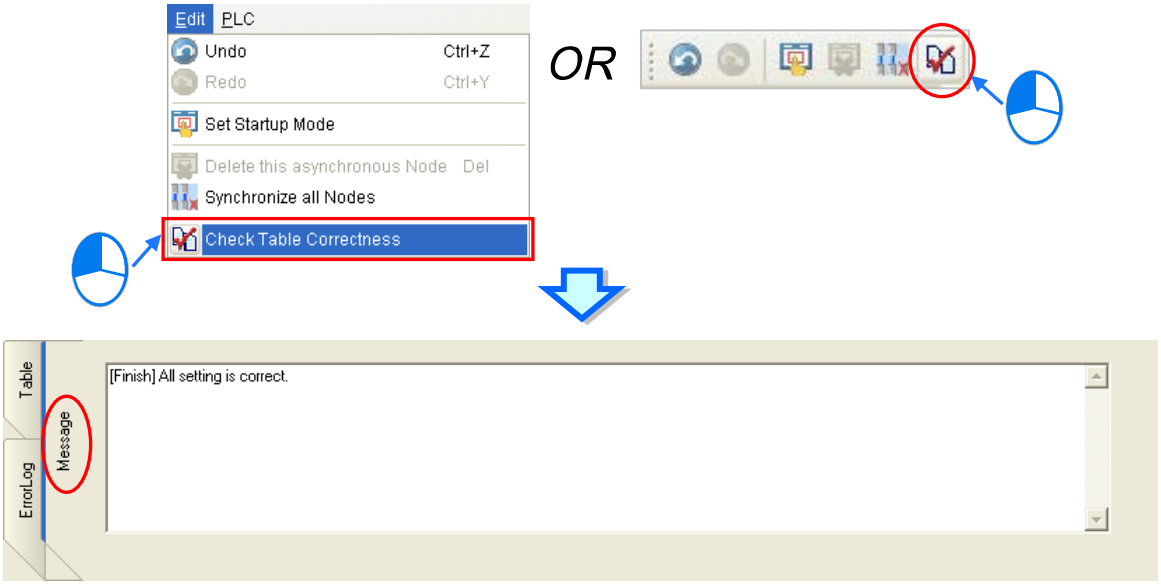
#	Device Name	Register	Starting Address	Range	<->	#	Device Name	Register	Starting Address	Range	Size
1	AH-01	D	500	D500 ~ D509	<-	2	AH-02	M	1000	M1000 ~ M1159	10
1	AH-01	D	500	D500 ~ D509	<-	3	AH-03	D	100	D100 ~ D109	10

After the users click **Export**, the data in the data exchange table can be exported as a CSV file. The users can edit the CSV file through Microsoft Excel. The CSV file can also be used as reference material for another development work.



#	Device Name	Register	Starting Address	Range	<->	#	Device Name	Register	Starting Address	Range	Size
1	AH-01	D	500	D500 ~ D509	<-	2	AH-02	M	1000	M1000 ~ M1159	10
1	AH-01	D	600	D600 ~ D609	<-	3	AH-03	D	100	D100 ~ D109	10

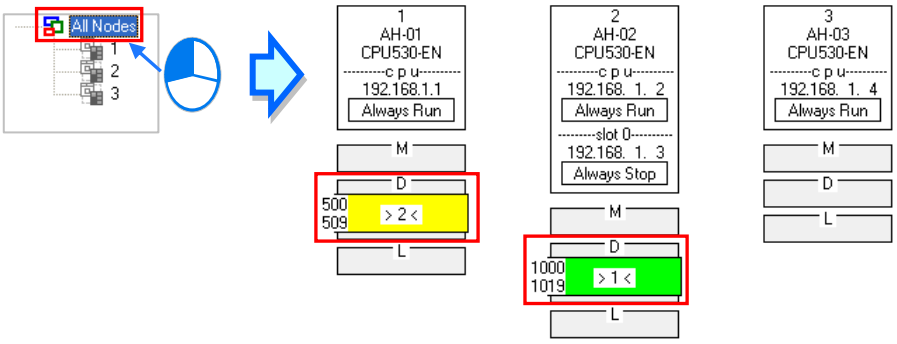
After the users create a data exchange table, they can click **Check Table Correctness** on the **Edit** menu or on the toolbar if they wan to check the data set in the table. The check result will be displayed in the **Message** page.



16.4.4 Node List and Display Area

If users click **All Node**s on the node list, all the nodes will be displayed in the display area, and all the data exchange groups set will be in the data exchange table under the display area. Besides, the devices in which the data demanded will be stored, and the data supply nodes are indicated in device blocks of the nodes in the display area.

>2< in the D block in node 1 indicates that the D block demands data of node 2, and the numbers at the left side of the D block indicate that the data demanded of node 2 will be stored in D500~D509 in node 1. Likewise, >1< in the D block in node 2 indicates that the D block demands data of node 1, and the numbers at the left side of the D block indicate that the data demanded of node 1 will be stored in D1000~D1019 in node 1.

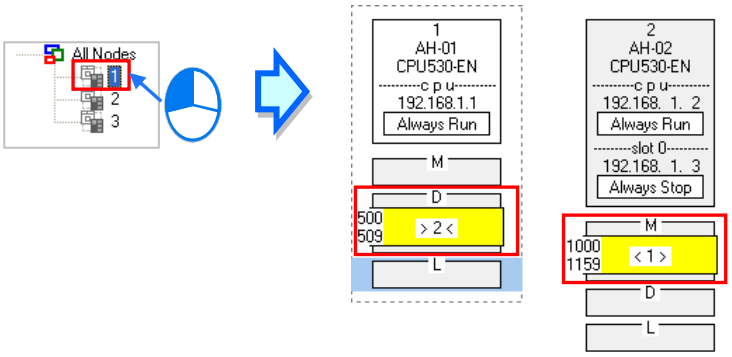


#	Device Name	Register	Starting Address	Range	<->	#	Device Name	Register	Starting Address	Range	Size
1	AH-01	D	500	D500 ~ D509	<-	2	AH-02	M	1000	M1000 ~ M1159	10
2	AH-02	D	1000	D1000 ~ D1019	<-	1	AH-01	L	0	L0 ~ L19	20

If the users click a specific node on the node list, the node and the nodes of which the node demands data will be shown in the display area, and the data in the data exchange table under the display will be related to the specific node selected.

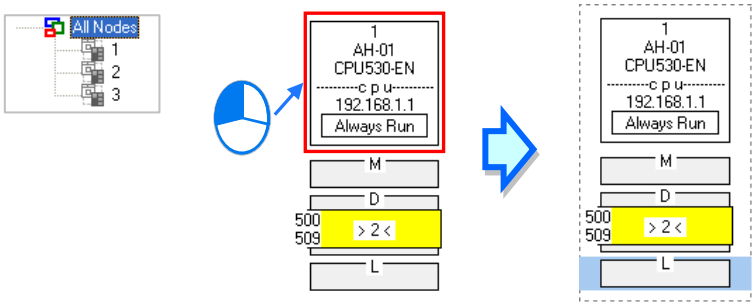
In the figure below, the dotted frame indicates that node 1 on the node list is selected, and the gray ground indicates that node 2 is a data supply node. Node 2 can not be selected, and the information in the M block in node 2 indicates the data which will be supplied to node 1. Owing to the fact that node 3 does not supply any data to node 1, node 3 is not displayed in the display area.

The numbers at the left side of the D block in node 1 indicates that the data demanded of node 2 will be stored in D500~D509 in node 1. <1> in the M block in node 2 indicates that the data in M1000~M1129 in node 2 will be supplied to 1.



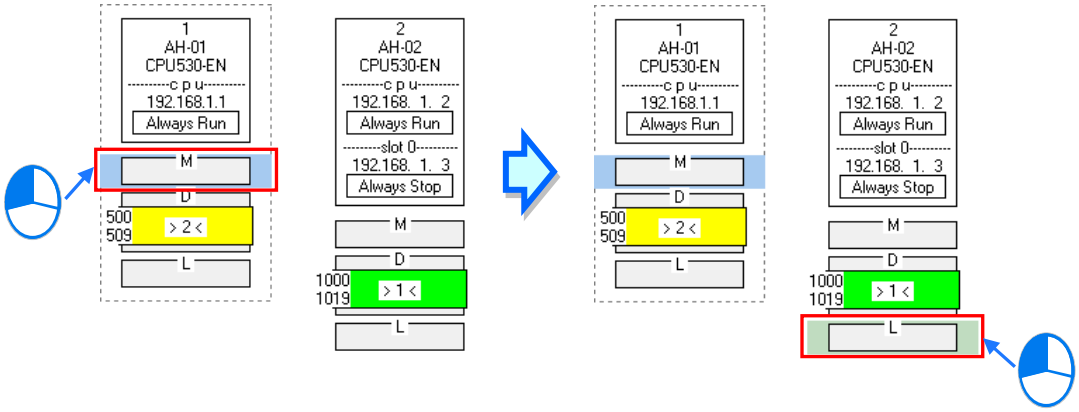
#	Device Name	Register	Starting Address	Range	<->	#	Device Name	Register	Starting Address	Range	Size
1	AH-01	D	500	D500 ~ D509	<-	2	AH-02	M	1000	M1000 ~ M1159	10

If a specific node on the node list is selected, the node in the display area will be selected. After the users select **All Nodes** on the node list, they can click the information about a node in the display area if they want to select the node. If the users click a device block in a node, the node will not be selected. After a node is selected, a dotted frame will appear.

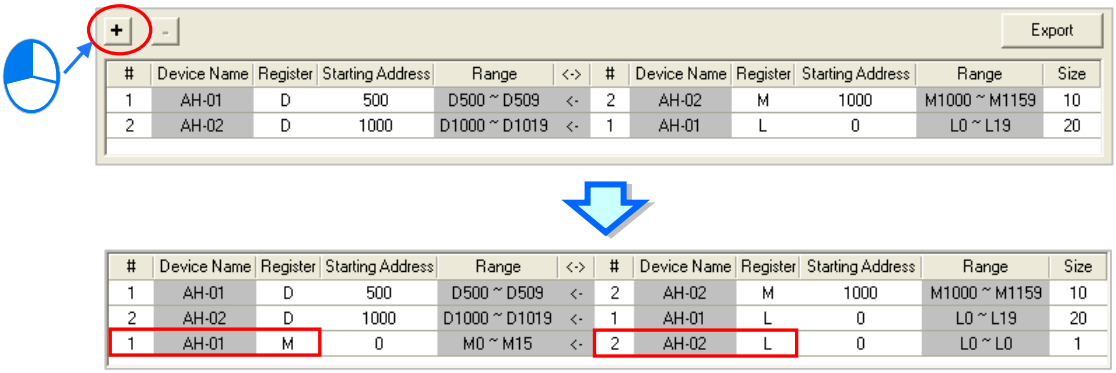


16

Once a node is selected, the node will be designated as a data demanding node. After the users click a device block in the node selected, a blue cursor will appear. After the users click a device block in another node (a data supply node), a green cursor will appear. The users can only click a device block in a data supply node. They can not select the node, otherwise the node will be designated as a data demanding node.



The users can click + to add a new piece of data to the data exchange table. The data includes the data demanding node, the node number assigned to the data demanding node, the data supply node, the node number assigned to the data supply node, and the device types selected. The users can refer to section 16.4.3, and set the other cells.



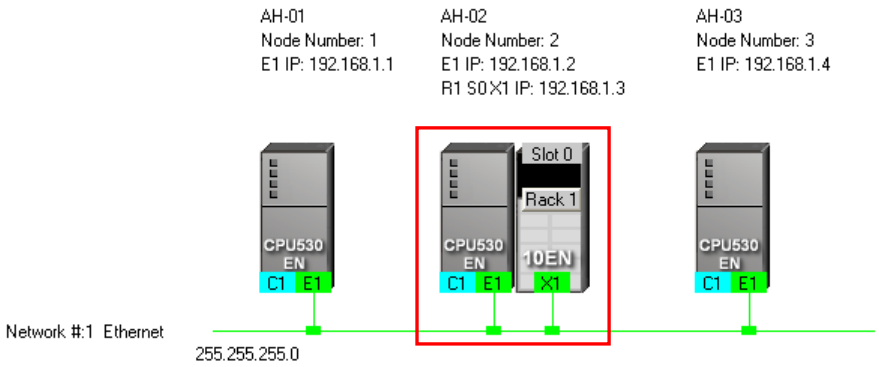
16.4.5 Start Mode of an Ether Link

There are three start modes. Please refer to the table below for more information.

Start mode	Description
Always Stop	During the operation of the PLC, no data exchange is performed.
Always Run	When the PLC runs, data exchange is performed.
SM Flag	The performance of data exchange depends on a special relay in the PLC. After users select **SM Flag**, they can set the initial state of the related special relay.

***. Please refer to manuals or technical documents for more information about special relays related to Ether Links.**

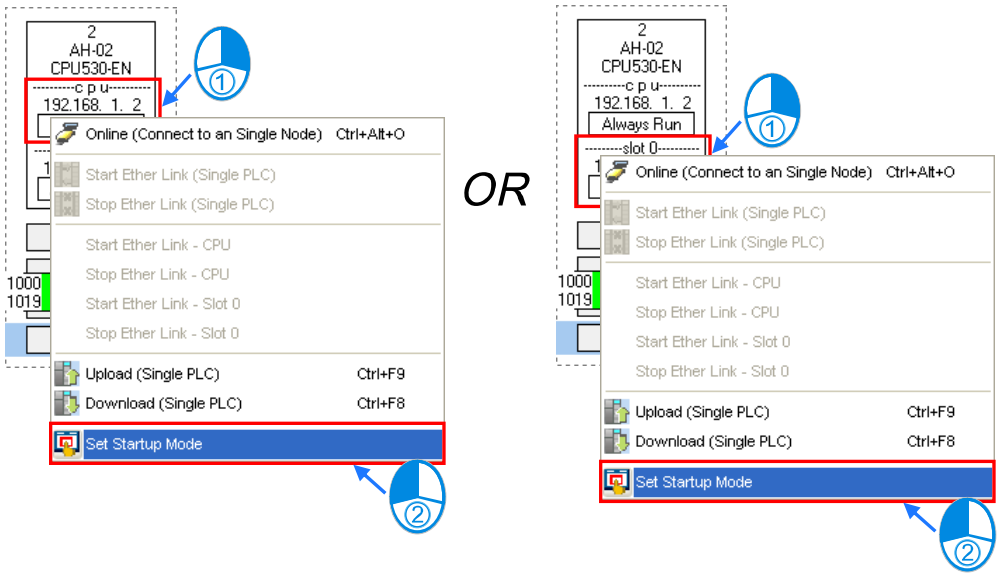
The execution of an Ether Link is based on the nodes in a network. If some of the ports that a node has are connected to a network, users can set the start modes of the ports separately. When the Ether Link constructed is executed, the system automatically distributes reading/writing work to the ports according to the start modes of the ports. If a port is disconnected, the system will pass the work belonging to the port to another port.



There are three ways to set the start mode of a node.

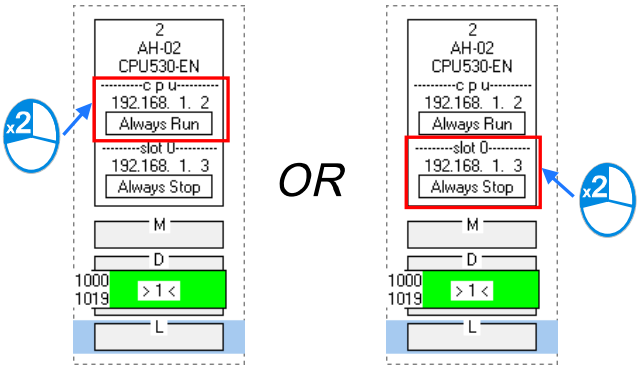
- **Method 1**

  Right-click CPU information or module information, and then click **Set Startup Mode** on the context menu.



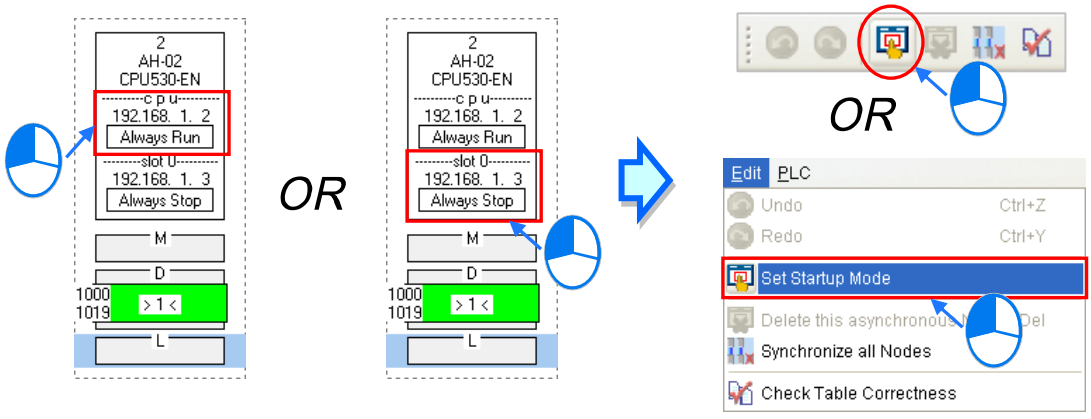
- **Method 2**

  Double-click CPU information or module information.



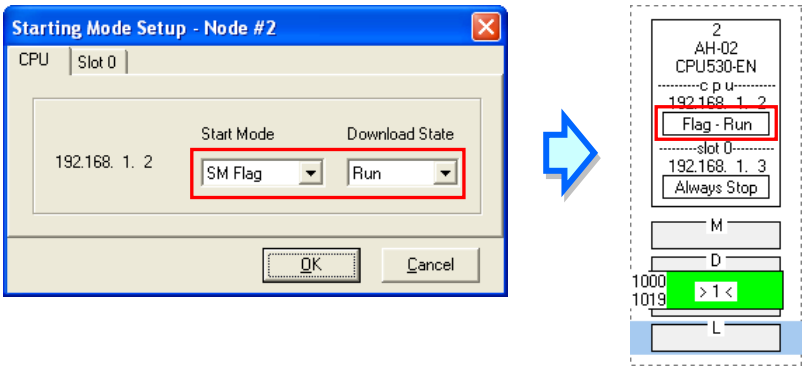
- **Method 3**

  After the users click CPU information or module information, they have to click on the toolbar, or **Set Startup Mode** on the **Edit** menu.



After the users use one of the methods described above, the Starting Mode Setup window will appear. The page displayed in the window varies with the information selected. The users can click the tabs in the window.

Select a mode in the **Start Mode** drop-down list box. If **SM Flag** is selected, the users can select an initial state in the **Download State** drop-down list box. After an initial state is selected, the users can click **OK**.



16.4.6 Downloading the Parameters Related to an Ether Link

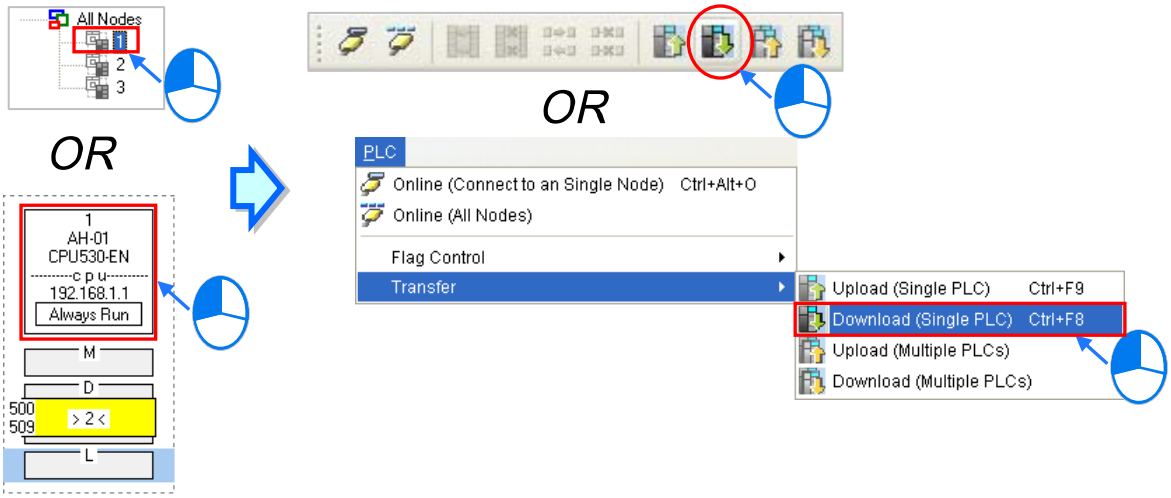
After the parameters related to an Ether Link are set, users have to download the parameters to PLCs. The PLCs can perform data exchange after the parameters are downloaded.

- **Single node**

  Only the data demanding setting related to the PLC selected, the start mode of the PLC selected, and the start modes of the modules connected to the PLC are downloaded. Before the users download the related parameters, they have to make sure that the system connects to the PLC normally, and they have completed the communication setting in NWCONFIG. Please refer to section 16.1.3 for more information.

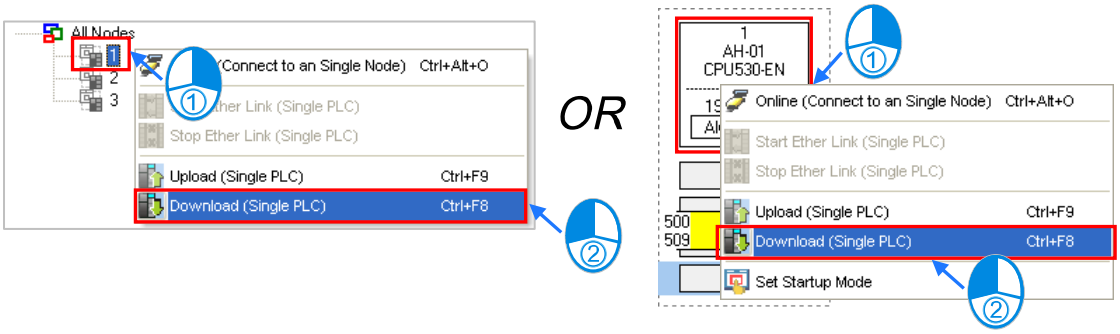
  ➢ **Method 1**

  Select a data demanding node, and then click on the toolbar. The users can also download the related parameters by selecting a data demanding node, clicking the **PLC** menu, pointing to **Transfer**, and clicking **Download (Single PLC)**.



  ➢ **Method 2**

  Select a data demanding node, right-click the data demanding node, and click **Download (Single PLC)** on the context menu.

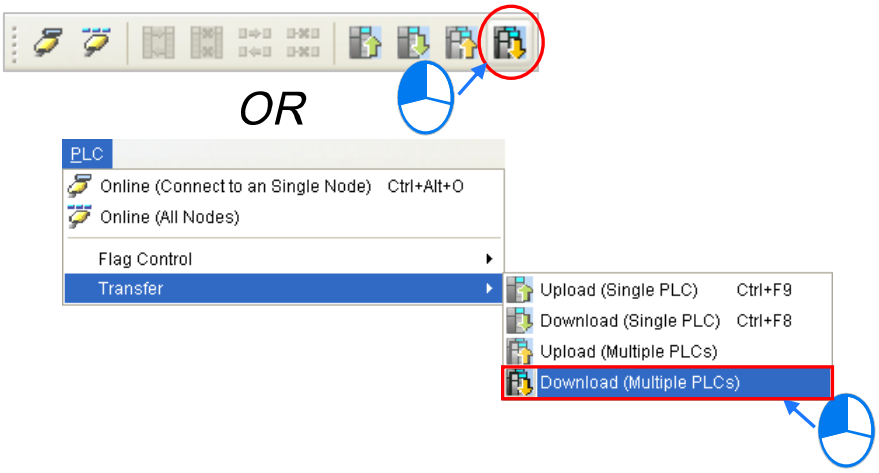


- **Multiple nodes**

  The data demanding setting related to all the nodes, and the start modes of all the nodes are downloaded. Before the users download the related parameters, they have to make sure that all the PLCs and all the modules are connected to an Ethernet network, and can connect to NWCONFIG through Ethernet. The connection type that the driver selected in the **Driver Name** drop-down list box in the **Select a Driver** window uses must be Ethernet, otherwise the related parameters can not be downloaded. Please refer to section 16.1.3 for more information.

  ➢ **Method 1**

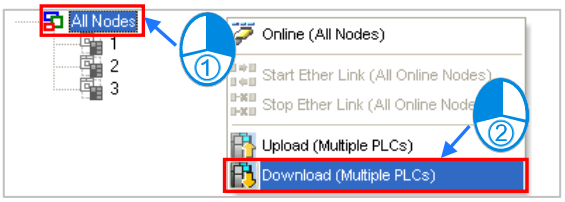
  Click the **PLC** menu, point to **Transfer**, and click **Download (Multiple PLCs)**. The users can also download the related parameters by clicking on the toolbar.



16

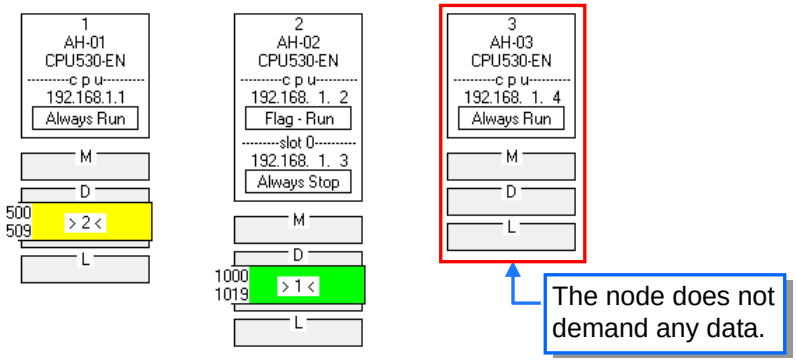
  ➢ **Method 2**

  Select **All Nodes** on the node list, right-click **All Nodes**, and click **Download (Multiple PLCs)** on the context menu.



Additional remark

If the parameters set include a node which does not demand any data, the node will not demand any data through the network specified after the parameters are downloaded to multiple nodes.



16.4.7 Uploading the Parameters Related to an Ether Link

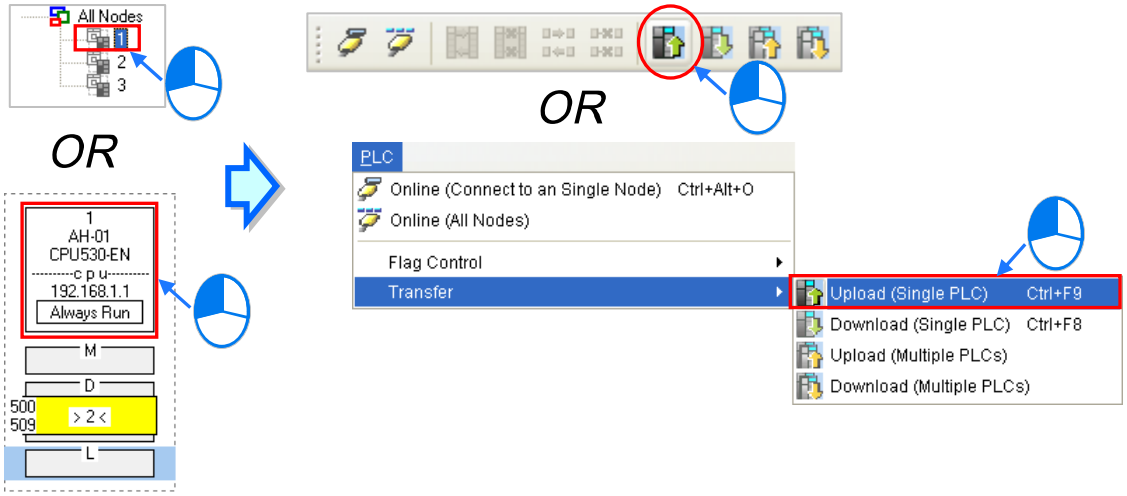
Users can upload the parameters related to an Ether Link in a PLC.

- **Single node**

  Only the parameters related to an Ether Link in the node selected are uploaded. Before the users upload the related parameters in a PLC, they have to make sure that the system connects to the PLC normally, and they have completed the communication setting in NWCONFIG. Please refer to section 16.1.3 for more information.

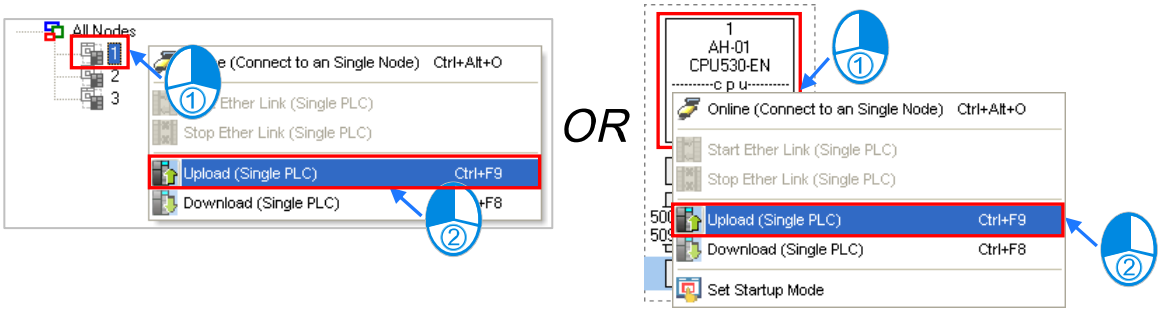
  ➢ **Method 1**

  Select a data demanding node, and then click on the toolbar. The users can also download the related parameters by selecting a data demanding node, clicking the **PLC** menu, pointing to **Transfer**, and clicking **Upload (Single PLC)**.



  ➢ **Method 2**

  Select a data demanding node, right-click the data demanding node, and click **Upload (Single PLC)** on the context menu.

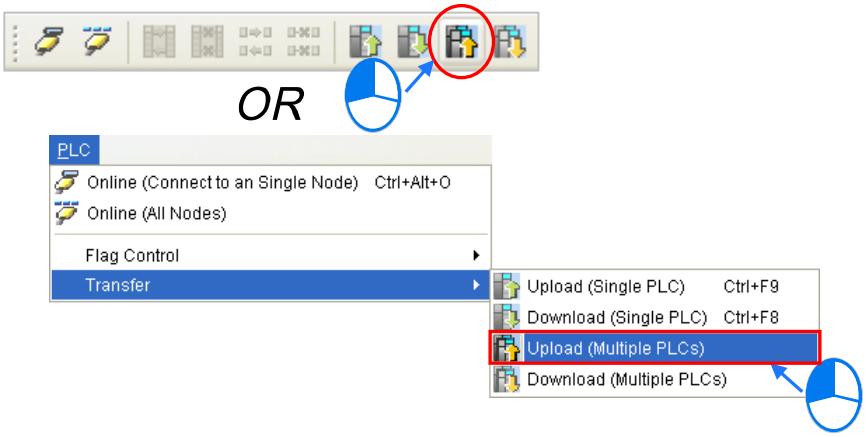


- **Multiple nodes**

  The parameters related to an Ether Link in all the nodes are uploaded. Before the users upload the related parameters, they have to make sure that all the PLCs and all the modules are connected to an Ethernet network, and can connect to NWCONFIG through Ethernet. The connection type that the driver selected in the **Driver Name** drop-down list box in the **Select a Driver** window uses must be Ethernet, otherwise the related parameters can not be uploaded. Please refer to section 16.1.3 for more information.

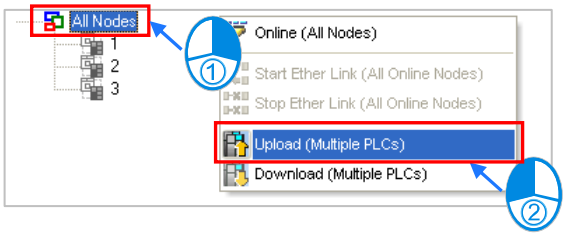
  ➢ **Method 1**

  Click the **PLC** menu, point to **Transfer**, and click **Upload (Multiple PLCs)**. The users can also download the related parameters by clicking on the toolbar.



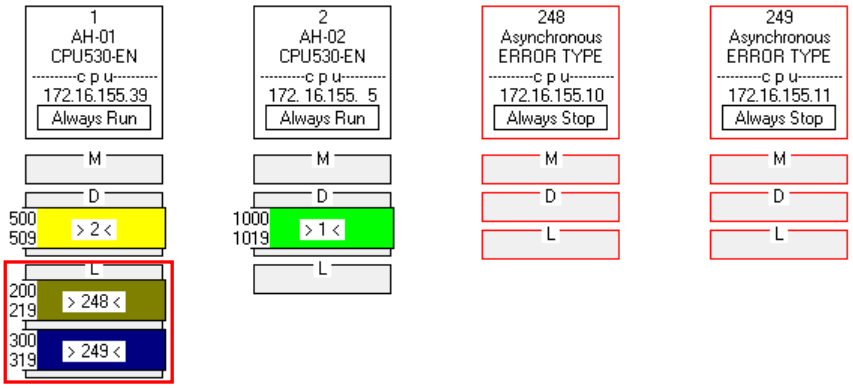
  ➢ **Method 2**

  Select **All Nodes** on the node list, right-click **All Nodes**, and click **Upload (Multiple PLCs)** on the context menu.



Additional remark

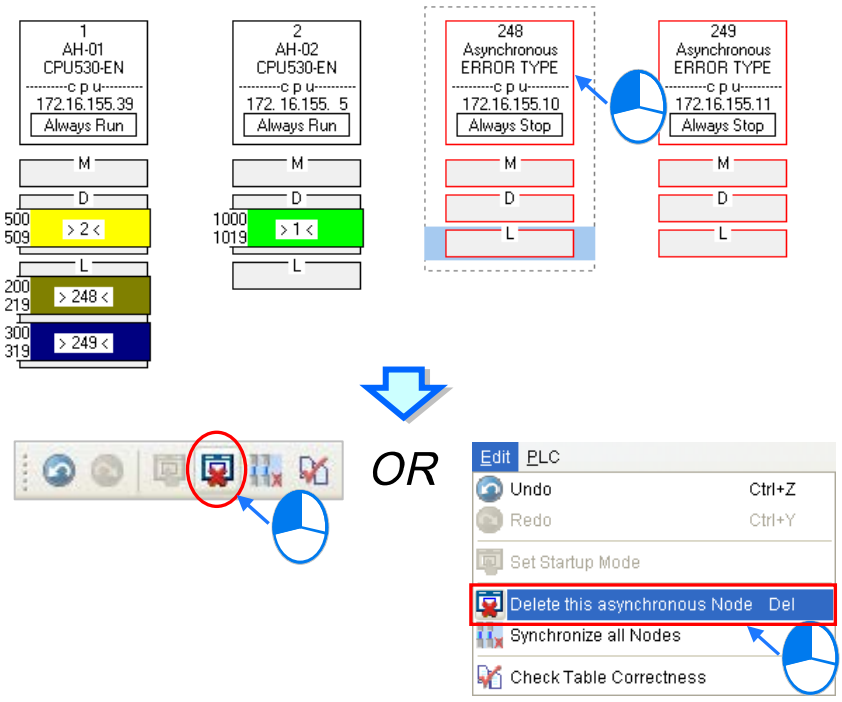
After the parameters which include a node not configured in NWCONFIG are uploaded, the node which is not configured in NWCONFIG will be called an asynchronous device, and will be in red. If the setting of an Ether Link includes an asynchronous device, the system does not allow the Ether Link to be monitored, and it does not allow the parameters related to the Ether Link to be downloaded.



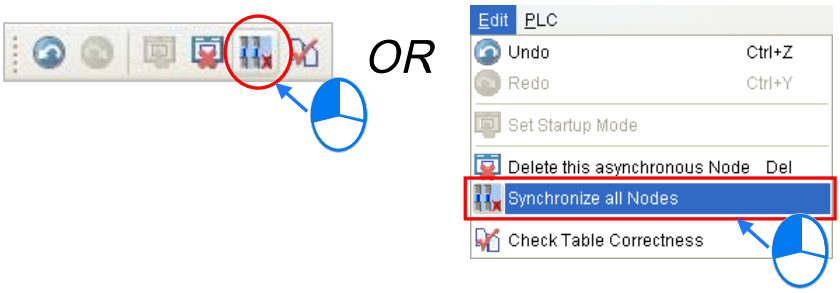
16.4.8 Deleting Asynchronous Device

If the setting of an Ether Link includes asynchronous devices, the system does not allow the Ether Link to be monitored, and it does not allow the parameters related to the Ether Link to be downloaded. Users have to find out the reason for the existence of the asynchronous devices. If the network configuration in NWCONFIG is incorrect, the users have to modify the network configuration, and upload the parameters related to the Ether Link again. If the parameters uploaded are not applicable to the current network configuration, the users can delete the asynchronous devices.

Select an asynchronous device which will be deleted, and then click **Delete This Asynchronous Node** on the **Edit** menu, or on the toolbar.



If the users want to delete all the asynchronous devices at the same time, they can click **Synchronize All Nodes** on the **Edit** menu, or on the toolbar.



16.4.9 Enabling / Disabling the Online Monitoring Function

In the **Ether Link Configuration** window, users can execute or test the Ether Link constructed by means of the online monitoring functions provided by NWCONFIG. The users can enable/disable the function of monitoring a single node/multiple nodes online.

Enabling/Disabling	Description	
Single node	Function	Enabling or disabling the function of monitoring the node selected online
	Condition	The users have to make sure that ISPSoft can connect to the PLC selected normally, and they have completed the communication setting in NWCONFIG.
Multiple nodes	Function	Enabling or disabling the function of monitoring all the nodes online
	Condition	The users have to make sure that all the nodes are connected to a network, and can connect to ISPSoft through Ethernet. The connection type that the driver selected in the **Driver Name** drop-down list box in the **Select a Driver** window uses must be Ethernet.

***. Please refer to section 16.1.3 for more information about the communication setting in NWCONFIG.**

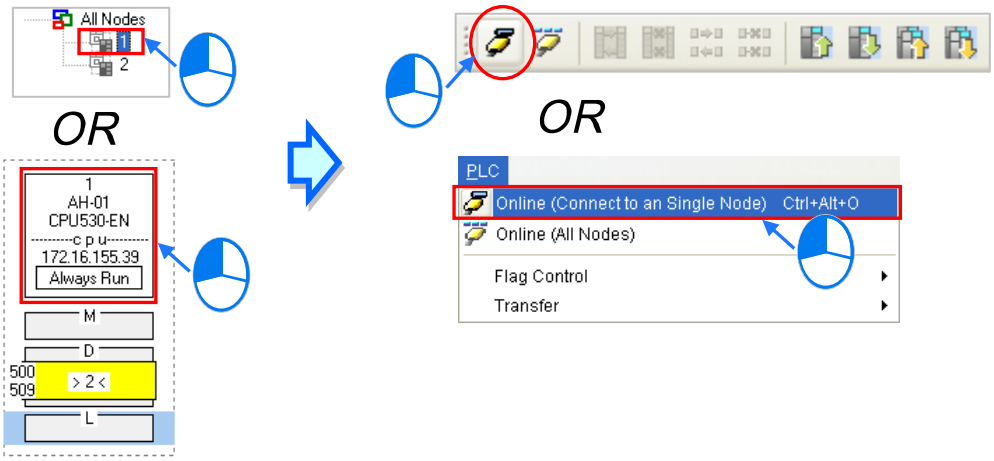
Before the users enable the online monitoring function, they have to make sure that all the nodes are connected according to the network framework created in NWCONFIG, and can operate normally.

(a) Every node has been connected to a network according to the network framework created in NWCONIFG.
(b) The users have set the parameters for Ethernet ports of the nodes by means of HWCOFNIG, and the parameters have been downloaded to the PLCs and the modules. The setting of the parameters must be consistent with the setting in NWCONFIG.
(c) The parameters related to an Ether Link have been downloaded to the PLC selected.
(d) Every node is powered up, and can operate normally.

16.4.9.1 Enabling a Monitoring Function

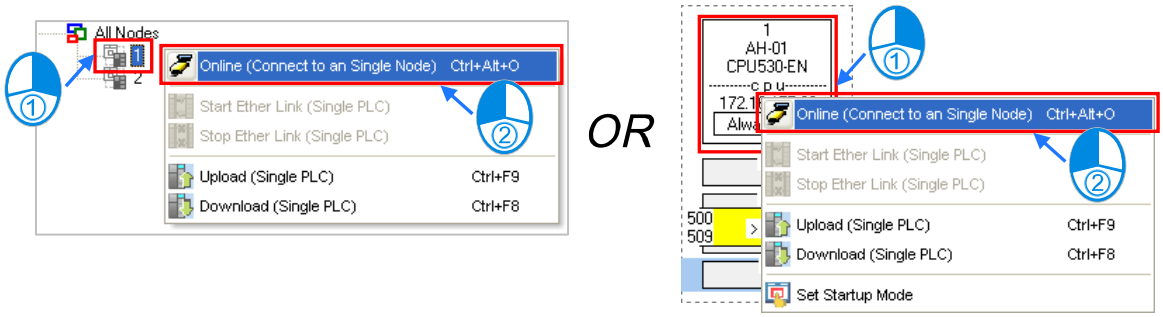
- **Enabling the function of monitoring a single node**
  - **Method 1**

    Select a data demanding node, and then click on the toolbar, or **Online (Connect to a Single node)** on the **PLC** menu. When the data demanding node is monitored, on the toolbar is pressed.



➢ **Method 2**

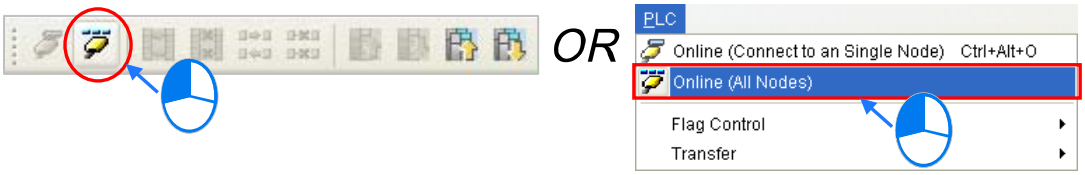
Select a data demanding node, right-click the data demanding node, and click **Online (Connect to a Single Node)** on the context menu. When the data demanding node is monitored, on the toolbar is pressed.



● **Enabling the function of monitoring multiple nodes**

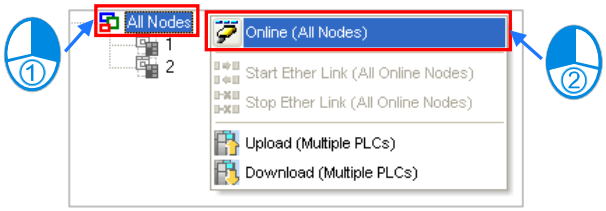
➢ **Method 1**

Click **Online (All Nodes)** on the **PLC** menu, or on the toolbar.



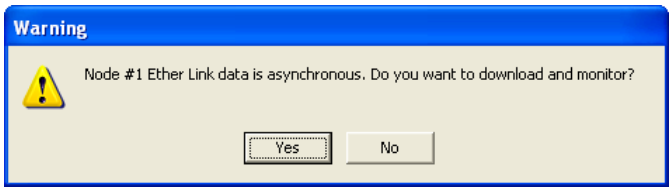
➢ **Method 2**

Select **All Nodes** on the node list, right-click **All Nodes**, and click **Online (All Nodes)** on the context menu.



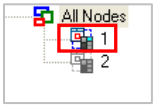
Additional remark

Before the system enters a monitoring mode, it checks whether the Ether Link constructed in the **Ether Link Configuration** window is consistent with the setting in the PLCs. If the Ether Link constructed in the **Ether Link Configuration** window is not consistent with the setting in the PLCs, the system will ask the users to download the related parameters again.

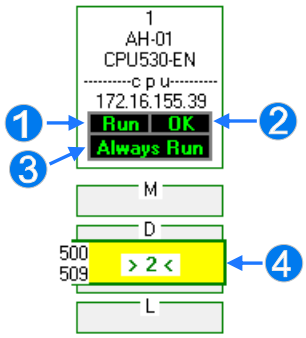


16.4.9.2 Monitoring Statuses

After the system enters a monitoring mode, the node which is monitored will be colored on the node list.



Besides, the words and the pictures in the display area will indicate the execution status of the current Ether Link after the system enters a monitoring mode.

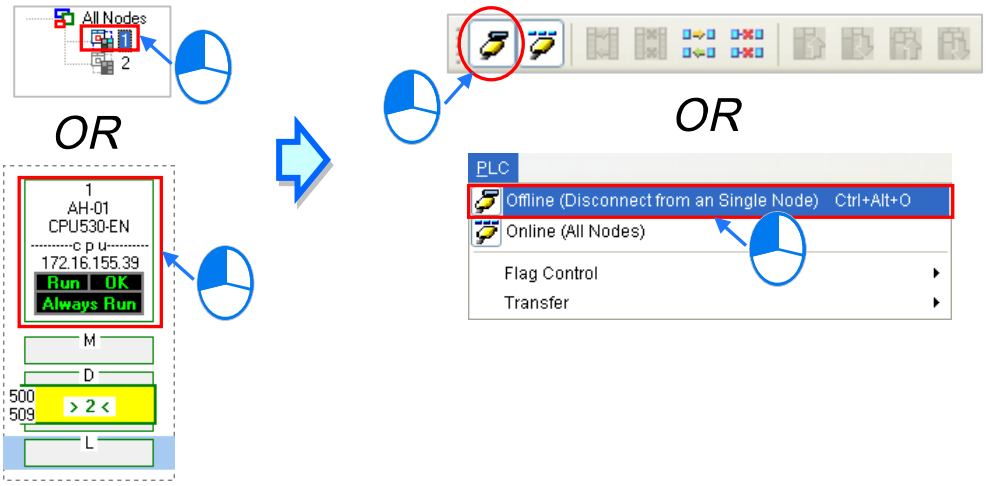


Status	Description
❶ Running/Stopping	Run: The Ether Link constructed is executed.
	Stop: The Ether Link is not executed.
❷ Operating status	OK: The Ether Link constructed is executed normally.
	Error: The Ether Link constructed is not executed normally.
❸ Start mode	Always Stop: **Always Stop**
	Always Run: **Always Run**
	Flag Mode: **SM Flag**
❹ Data block	500 509 > 2 <: The data exchange is being performed.
	500 509 ≠ 2 ≠: The performance of the data exchange stops.

16.4.9.3 Disabling a Monitoring Function

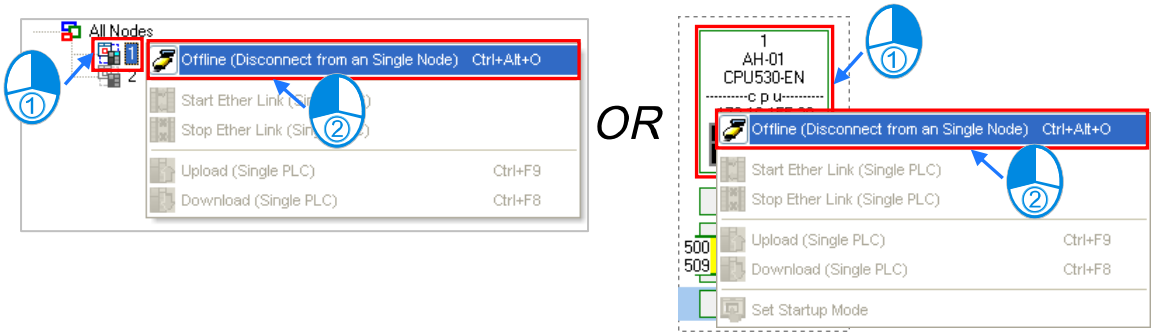
- **Disabling the function of monitoring a single node**
  - ➢ **Method 1**

    Select a data demanding node, and then click on the toolbar, or **Offline (Disconnect from a Single node)** on the **PLC** menu.

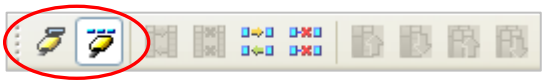


➢ **Method 2**

Select a data demanding node, right-click the data demanding node, and click **Offline (Disconnect from a Single Node)** on the context menu.



When the data demanding node selected is not monitored, on the toolbar is not pressed. If another node is monitored, will be pressed.

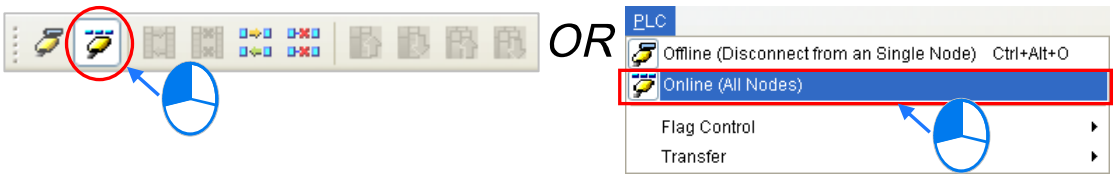


16

- **Disabling the function of monitoring multiple nodes**

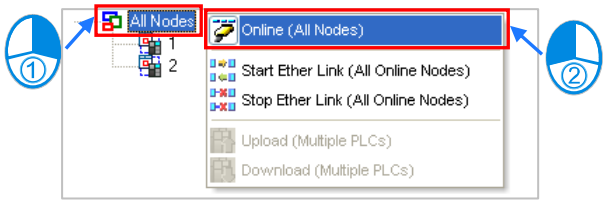
➢ **Method 1**

Click **Online (All Nodes)** on the **PLC** menu, or on the toolbar.



➢ **Method 2**

Select **All Nodes** on the node list, right-click **All Nodes**, and click **Online (All Nodes)** on the context menu.



16.4.10 Starting / Stopping the Execution of an Ether Link Online

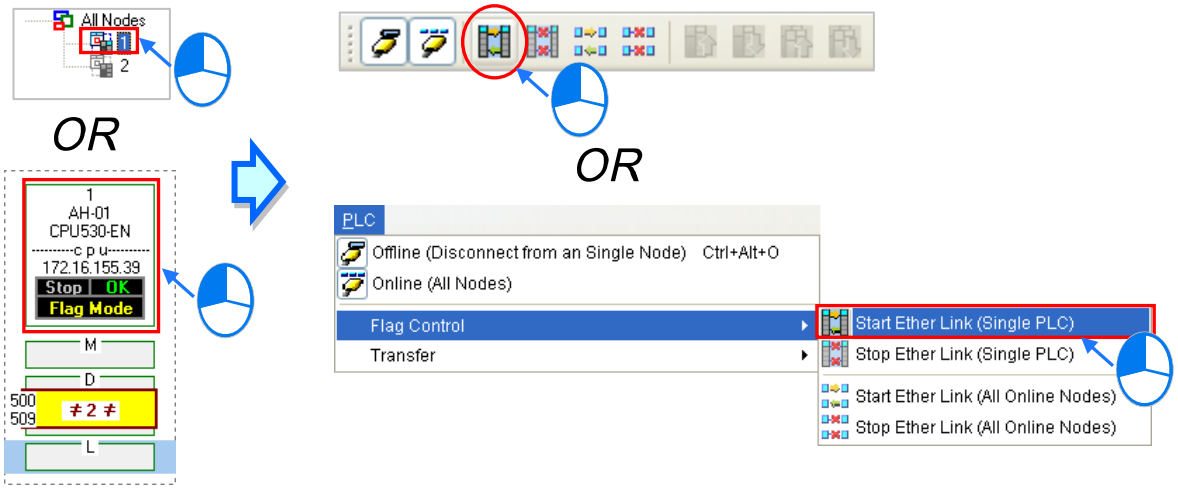
If the start mode of a node is **SM Flag**, users can make the node start or stop the execution of the Ether Link constructed by means of controlling the state of the related flag when the node is monitored online. If the node is not monitored online, the users can not make the node start or stop the execution of the Ether Link constructed by means of controlling the state of the related flag.

Users can make a single node/multiple nodes start or stop the execution of the Ether Link constructed. The conditions for making a single node/multiple nodes start or stop the execution of the Ether Link constructed are the same as the conditions for enabling/disabling the function of monitoring a single node/multiple nodes online. Please refer to section 16.4.9 for more information.

16.4.10.1 Starting the Execution of an Ether Link

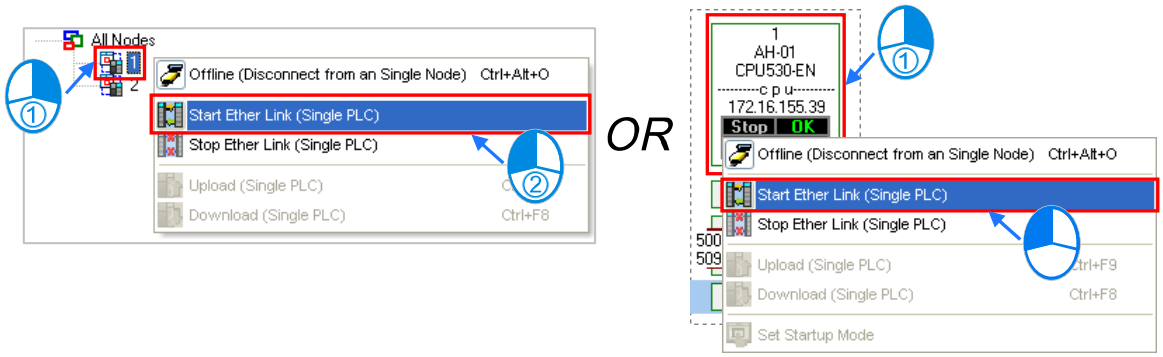
- **Making a single node start the execution of an Ether Link**
  - **Method 1**

    If users want to make a node start the execution of the Ether Link constructed, they have to select the node, and click on the toolbar. They can also make the node start the execution of the Ether Link constructed by selecting the node, clicking the **PLC** menu, pointing to **Flag Control**, and clicking **Start Ether Link (Single PLC)**.



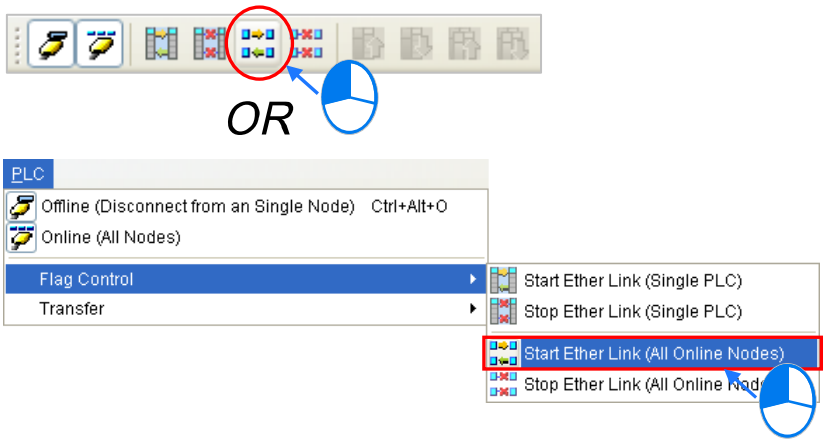
  - **Method 2**

    Select a node, right-click the node, and click **Start Ether Link (Single PLC)** on the context menu.



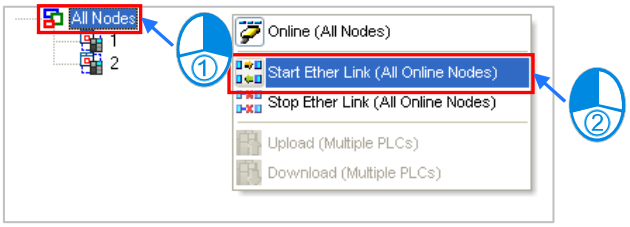
- **Making multiple nodes start the execution of an Ether Link**
  - **Method 1**

    If users want to make all the nodes start the execution of the Ether Link constructed, they have to click the **PLC** menu, point to **Flag Control**, and click **Start Ether Link (All Online Nodes)**. The users can also make all the nodes start the execution of the Ether Link constructed by clicking on the toolbar.



  - **Method 2**

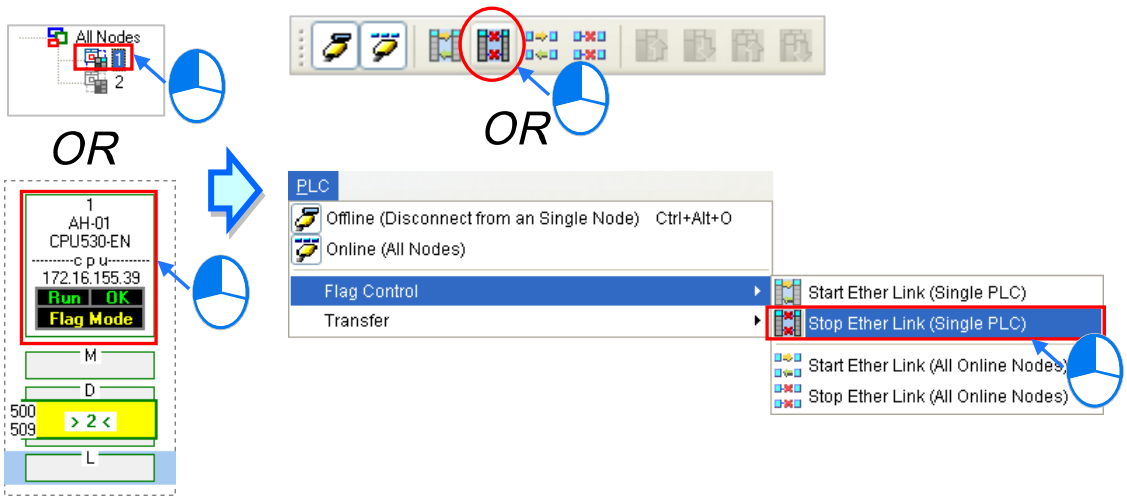
    Select **All Nodes** on the node list, right-click **All Nodes**, and click **Start Ether Link (All Online Nodes)** on the context menu.



16.4.10.2 Stopping the Execution of an Ether Link

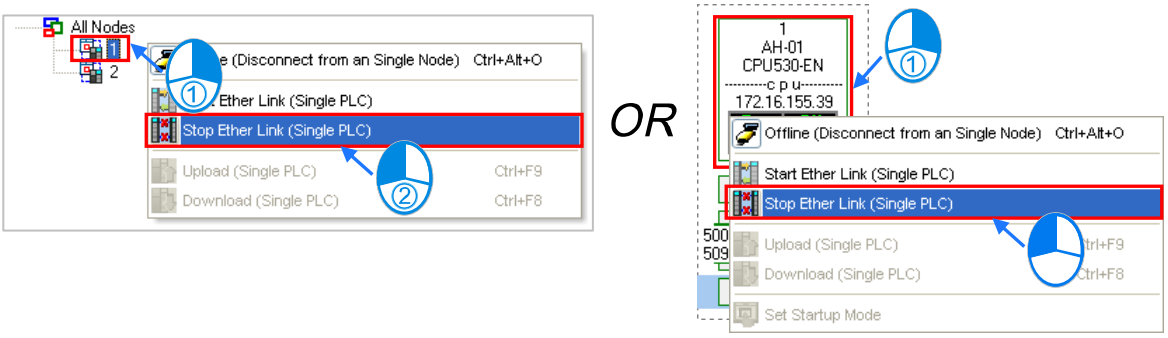
- **Making a single node stop the execution of an Ether Link**
  - **Method 1**

    If users want to make a node stop the execution of the Ether Link constructed, they have to select the node, and click on the toolbar. They can also make the node stop the execution of the Ether Link constructed by selecting the node, clicking the **PLC** menu, pointing to **Flag Control**, and clicking **Stop Ether Link (Single PLC)**.



➢ **Method 2**

Select a node, right-click the node, and click **Stop Ether Link (Single PLC)** on the context menu.

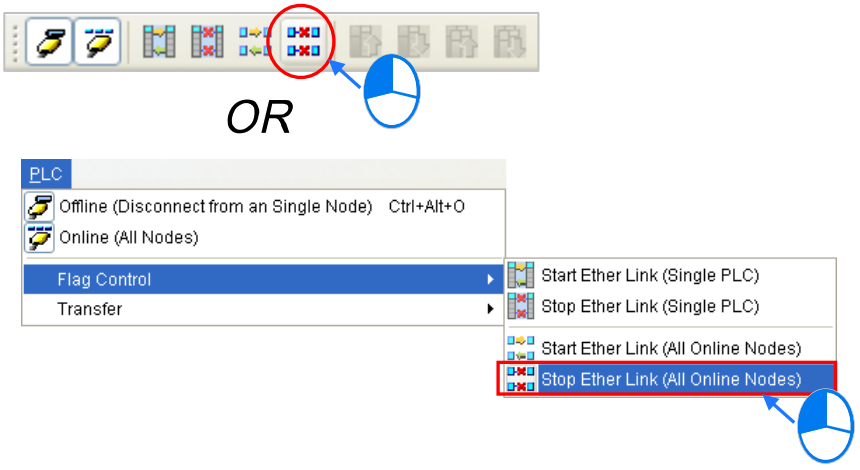


16

- **Making multiple nodes stop the execution of an Ether Link**

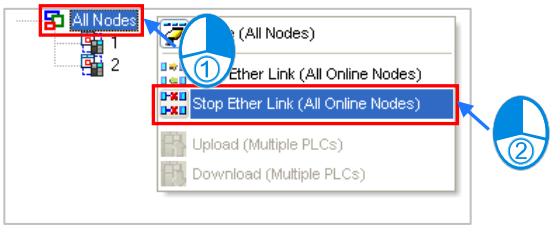
➢ **Method 1**

If users want to make all the nodes stop the execution of the Ether Link constructed, they have to click the **PLC** menu, point to **Flag Control**, and click **Stop Ether Link (All Online Nodes)**. The users can also make all the nodes stop the execution of the Ether Link constructed by clicking on the toolbar.



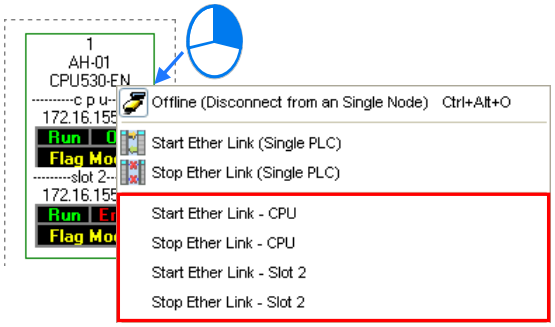
➢ **Method 2**

Select **All Nodes** on the node list, right-click **All Nodes**, and click **Stop Ether Link (All Online Nodes)** on the context menu.



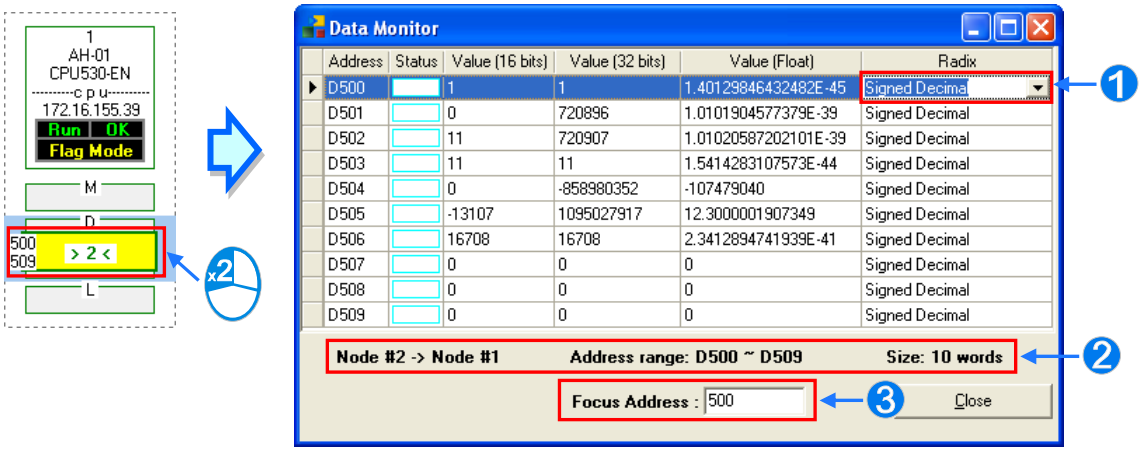
Additional remark

If a node in the display area has several Ethernet ports, users can make a port start/stop the execution of the Ether Link constructed by means of clicking an item on the context menu which appears after they right-click the node. The execution of an Ether Link is based on the nodes in a network. If users make a port that a node has stop the execution of the Ether Link constructed, another node that the node has can still execute the Ether Link constructed.



16.4.11 Monitoring Table and Error Log

After users double-click a data block, the **Data Monitor** window will appear. The users can view the values in the window. They can not change the values in the window.

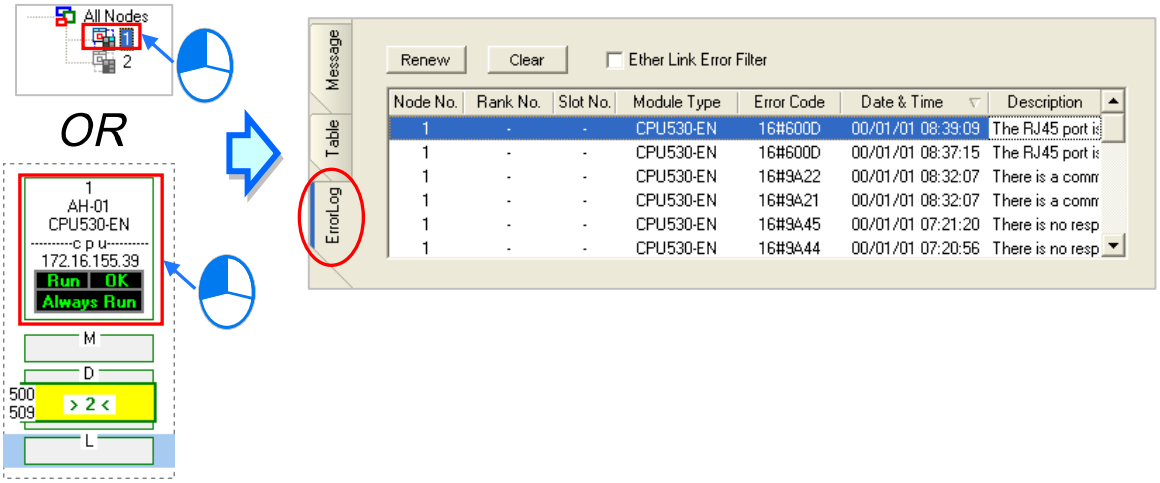


❶ After users click the **Radix** cell for a device, they can select a method of representing the value in the device on the drop-down list.

❷ The information about the data block which is monitored is displayed.

❸ Users can type a device address in this box. After the users press **Enter** on the keyboard, they can easily view the device address in the window.

After a node is selected, the error log in the node will be displayed in the **Error Log** page. If the users select the **Ether Link Error Filter** checkbox, only the error log related to the Ether Link constructed will be listed. Besides, after the users click **Renew**, the error log in the node will be retrieved, and the error log retrieved will be displayed in the **Error Log** page. After the users click **Clear**, the error log in the **Error Log** page and the error log in the node will be cleared.



If the node selected consists of a CPU module and a module, the error log in the **Error Log** page will be composed of the errors occurring in the CPU module and the errors occurring in the module. If an error code in the **Error Log** page corresponds to an error occurring in the module, the model name of the module will be displayed in the **Module Type** cell for the error code, the rack on which the module is installed will be indicated by the number in the **Rack No.** cell for the error code, and the slot in which the module is installed will be indicated by the number in the **Slot No.** cell for the error code. If an error code in the **Error Log** page corresponds to an error occurring in the CPU module, there will be no numbers in the **Rack No.** cell for the error code and the **Slot No.** cell for the error code.

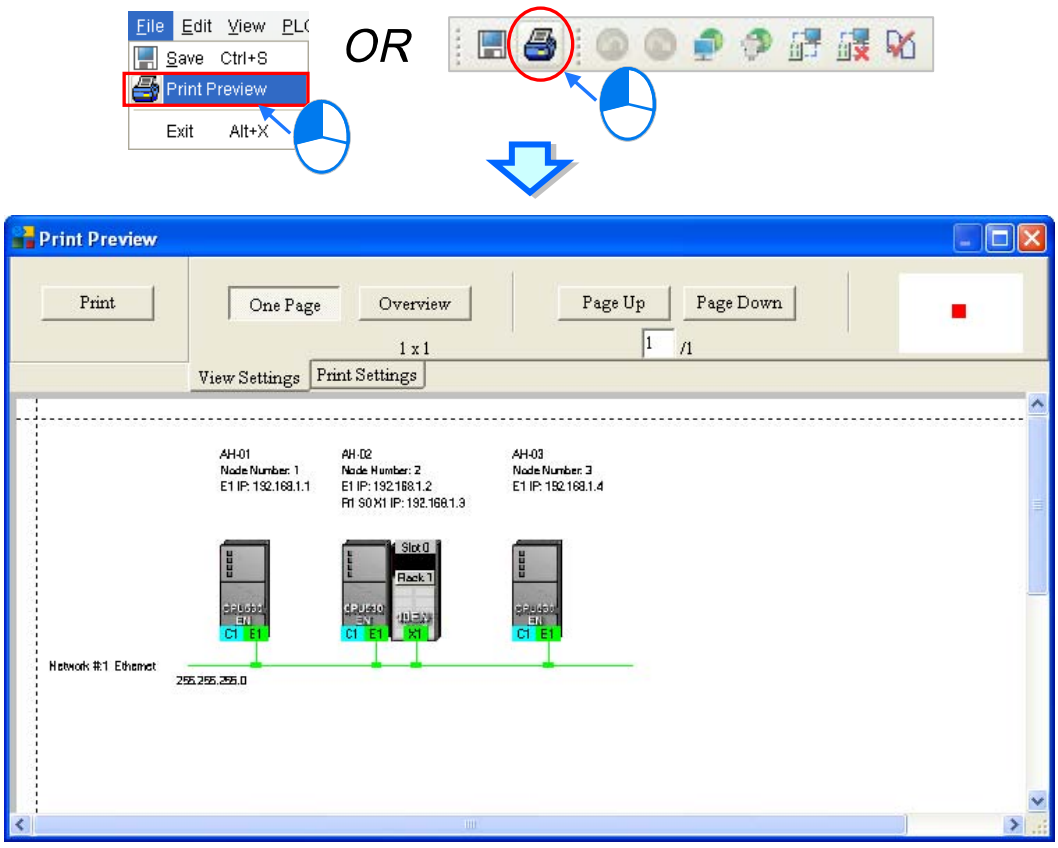
16.5 Managing and Applying NWCONFIG

16

16.5.1 Saving Parameters and Printing a Network Framework

If users want to save the parameters set in NWCONFIG, they can click **Save** on the **File** menu, or on the toolbar. The parameters which can be saved are the network framework created in NWCONFIG, the parameters related to the PLC Links constructed, and the parameters related to the Ether Links constructed. After the saving of the parameters set in NWCONFIG is complete, an nw file whose primary filename is the project name/group name, and an nwsd file whose primary filename is the project name/group name will appear in the folder in which the project/group of projects is/are saved.

After the users click **Print Preview** on the **File** menu, or on the toolbar, the system will automatically open the **Print Preview** window, and the network framework that the users create in NWCONFIG will be displayed in the **Print Preview** window. Please refer to appendix C for more information.



Before the users print the data related to a PLC Link or an Ether Link, they have to export the data as a CSV file. After the CSV file is opened in Microsoft Excel, they can print the data in the CSV file.

16.5.2 Downloading Parameters

In the working area in NWCONFIG, users can download the routing tables produced to PLCs, the parameters related to the PLC Links constructed, the parameters related to the Ether Links constructed to the nodes.

16

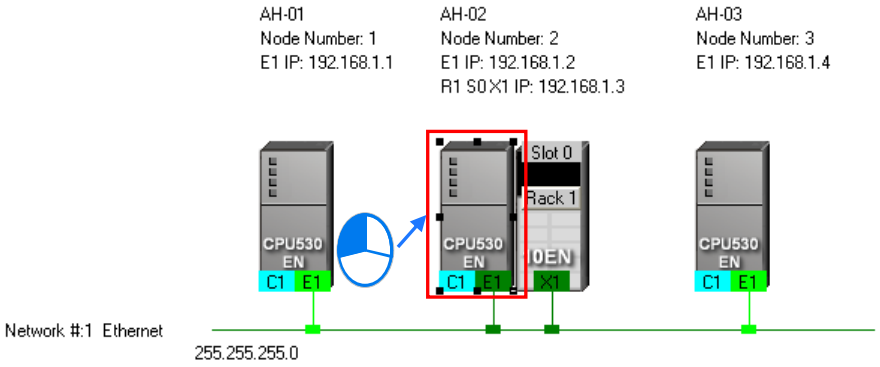
16.5.2.1 Introduction of Parameters

- **Routing table**
  The routing data stored in a PLC is data related to the PLC itself, and therefore the routing tables downloaded to nodes are different. Users have to download the routing tables produced to nodes in the working area.
- **Parameters related to a PLC Link**
  The parameters related to a PLC Link can only be downloaded to the PLC designated as a master station. If the parameters related to a PLC Link are downloaded to a slave station, the related special relays and the related special registers in the slave station will be restored to the default setting. There is only one master station in a network. If users are not sure whether a device was designated as a master station, and whether the parameters related to a PLC Link was downloaded to the device, they have to download the parameters related to a PLC Link to the device.
- **Parameters related to an Ether Link**
  The parameters related to an Ether Link must be downloaded to the data demanding nodes. However, the data exchange table stored in a PLC is a table related to the PLC itself. Besides, if the parameters set include a node which does not demand any data, the data in the node will be cleared after the parameter are downloaded to the node, and the start mode of the node will depend on the parameters after the parameters are downloaded to the node.

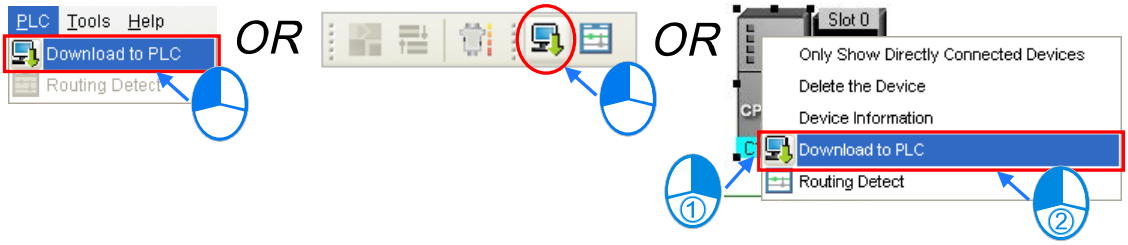
16.5.2.2 Description of Downloading Parameters

If the users want to download parameters, the **Routing Mode** checkbox in the **Select a Driver** window must be unselected. Please refer to section 16.1.3 for more information.

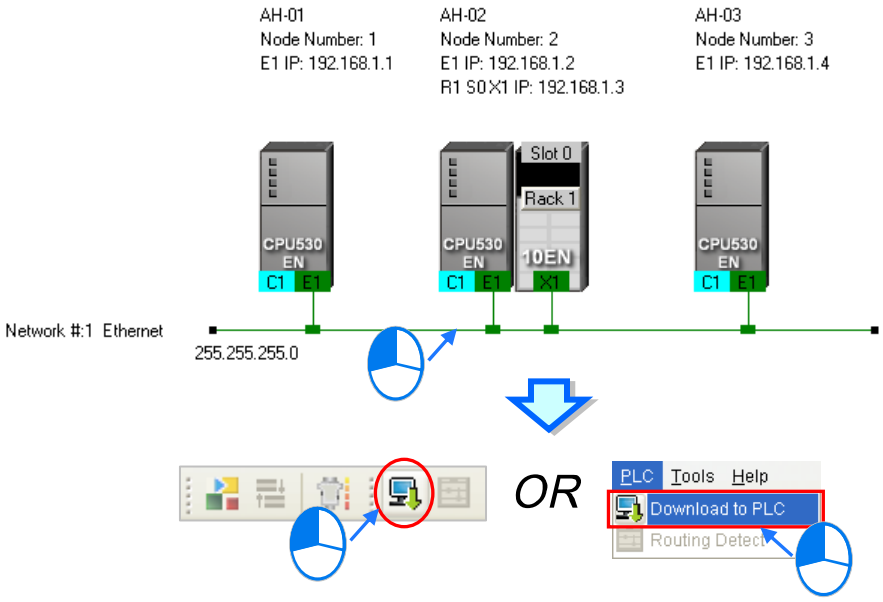
- **Single node**
  The users have to select a node in the working area.



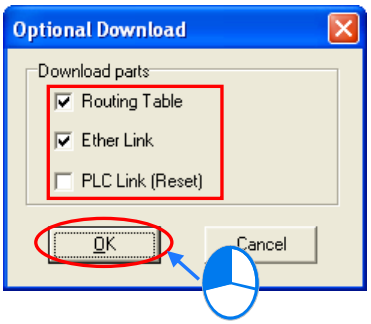
After the users click **Download to PLC** on the **PLC** menu, or on the toolbar, the **Optional Download** window will appear. The users can also open the **Optional Download** window by right-clicking the device they select, and clicking **Download to PLC** on the context menu.



- **Multiple nodes**
  The users have to select an Ethernet network in the working area. After the users click **Download to PLC** on the **PLC** menu, or on the toolbar, the **Optional Download** window will appear. If the actual connection is consistent with the setting in NWCONFIG, parameters can be downloaded to the nodes connected to the Ethernet network.



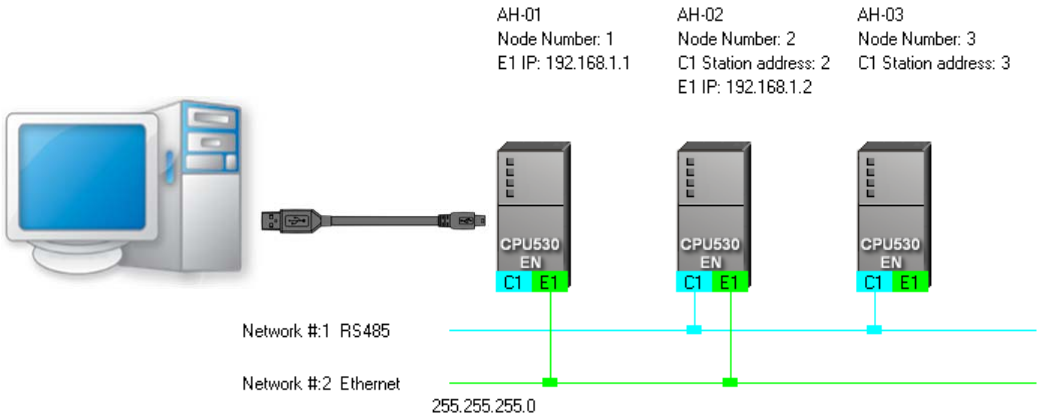
After the users follow the steps described above, the **Optional Download** window will appear. If a checkbox in the **Optional Download** window is gray, the checkbox can not be selected. After the users select checkboxes in the **Optional Download** window, they can click **OK**.



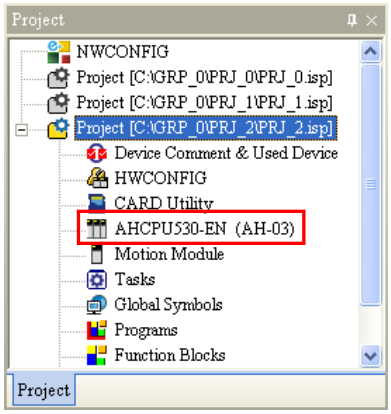
16.5.3 Using Routing in ISPSoft

After users create a network architecture in NWCONFIG, and download the routing tables produced to PLCs, they can download data to a device which is not directly connected to ISPSoft through routing, upload data from a device which is not directly connected to ISPSoft through routing, or monitor data in a device which is not directly connected to ISPSoft through routing. If the users want to use routing in ISPSoft, the devices used must be AH500 series CPU modules.

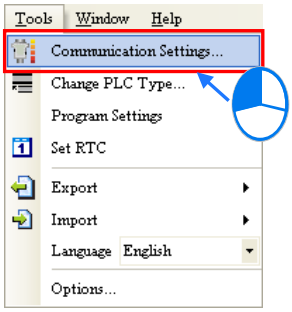
In the figure below, the PLC which actually connects to the computer is AH-01. AH-03 can be monitored through routing.



(1) Users have to activate the project for AH-03.



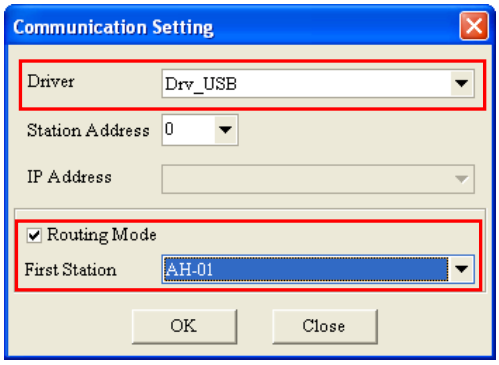
(2) Click the **Tools** menu, and then click **Communication Settings…**.



(3) In the **Communication Setting** window, the users have to select the **Routing Mode** checkbox, and select a device in the **First Station** drop-down list box.

Generally speaking, the device which actually connects to the computer is the first station. If the computer connects to several devices, or connect to devices through Ethernet, the users have to designate a device as the first station according to the network framework created in NWCONFIG. Besides, if the **Routing Mode** checkbox is selected, the driver selected in the **Driver** drop-down list box must be a driver which can connect to the first station.

After the users click **OK** in the **Communication Setting** window, AH-03 can be monitored through AH-01.



Additional remark

Before users use routing, they have to create projects, configure hardware, and configure a network. Please refer to section 16.1.4 for more information. The users can also refer to section 16.1.3 for more information about the communication setting in NWCONFIG.

Chapter 17 Data Backup and Data Restoration

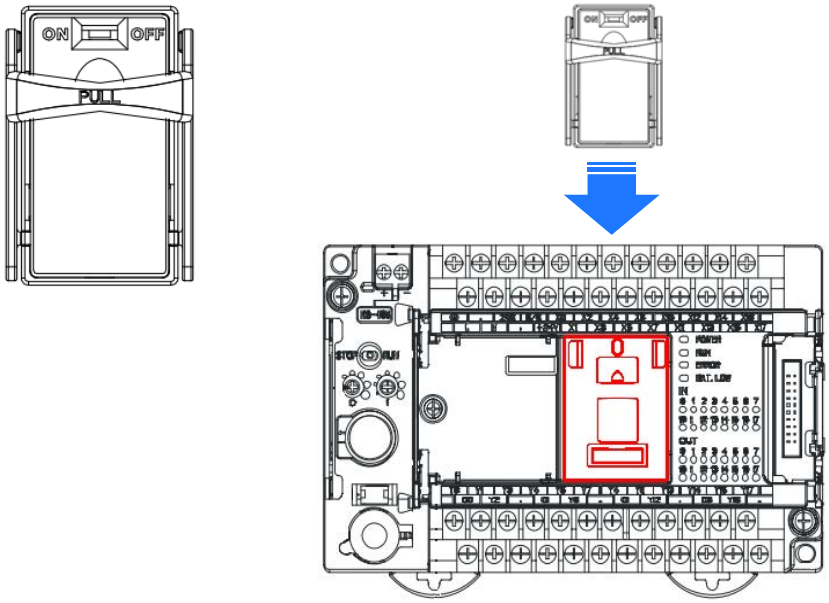
Table of Contents

17.1 Data Backup Memory Cards 17-2
17.1.1 Introduction of Data Backup Memory Cards 17-2
17.1.2 Using a Data Backup Memory Card 17-2
17.2 Permanent Data Backup 17-3
17.3 CARD Utility 17-4
17.3.1 Introduction of CARD Utility 17-4
17.3.2 Backup 17-5
17.3.3 Restoration 17-9

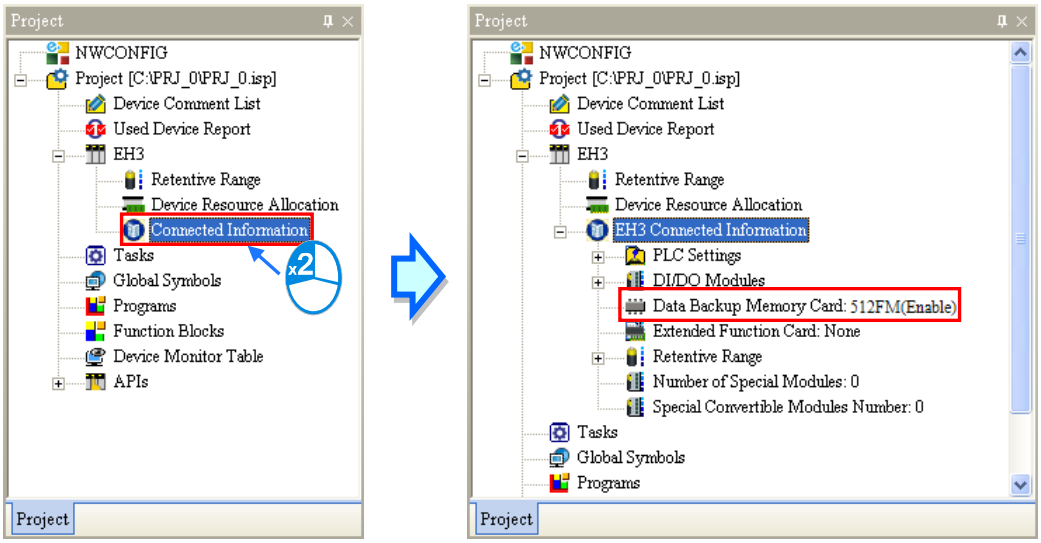
17.1 Data Backup Memory Cards

17.1.1 Introduction of Data Backup Memory Cards

A DVP-EH2 series PLC can be used with DVP-256FM, a data backup memory card with a milky cover. A DVP-EH3 series PLC can be used with DVP-512FM, a data backup memory card with a black cover.



Install a memory card in a PLC, power the PLC up, unfold the PLC section in the project management area in ISPSoft, and double-click **Connected Information**. If ISPSoft is connected to the PLC normally, the system information about the PLC will be under the **Connected Information** section. The **Data Backup Memory Card** item indicates whether the memory card is enabled.

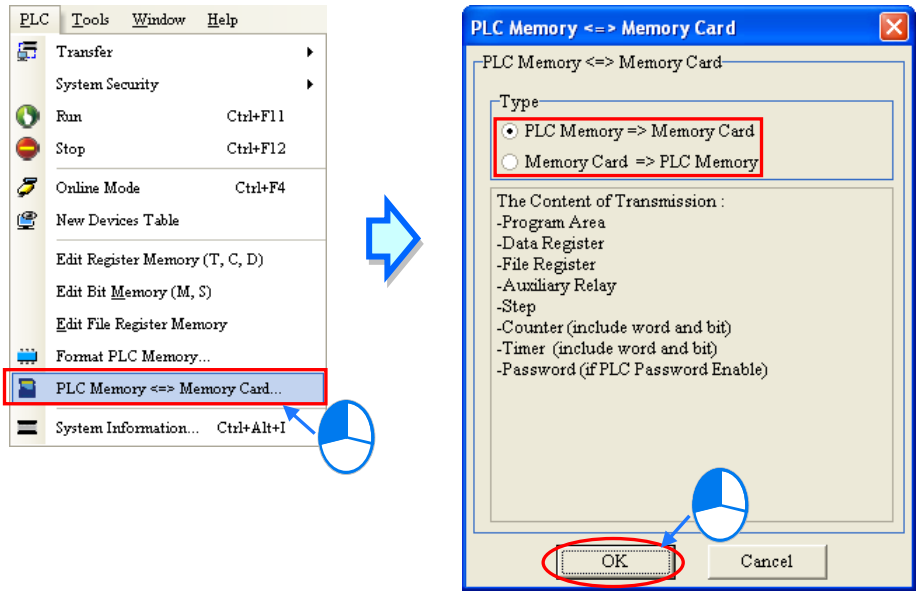


***. Please refer to section 2.4 for more information about communication setting.**

17.1.2 Using a Data Backup Memory Card

Users have to make sure that ISPSoft is connected to a PLC before they use the data backup memory card installed in the PLC. Click **PLC Memory<=>Memory Card…** on the **PLC** menu, select

the **PLC Memory=>Memory Card** option button or the **Memory Card=>PLC Memory** in the **Type** section in the **PLC Memory<=>Memory Card** window, and click **OK**.

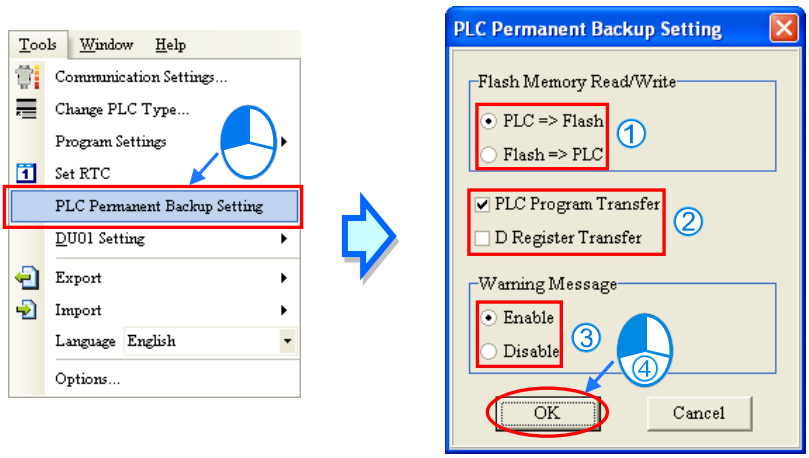


***. Please refer to the related user manuals or technical documents for more information about data backup memory cards, the models which are supported, and the firmware versions which are supported.**

17.2 Permanent Data Backup

There is flash memory in a DVP-EH2/DVP-EH2-L/DVP-EH3/DVP-EH3-L/DVP-SV/DVP-SV2 series PLC. The flash memory in a PLC is used to back up the data in the PLC. Please refer to manuals or documents for more information about the functions of flash memory.

Users have to make sure that ISPSoft is connected to a PLC before they perform a data backup or a data restoration through ISPSoft. Click **PLC Permanent Backup Setting** on the **Tools** menu. In the **PLC Permanent Backup Setting** window, the users have to select the **PLC=>Flash** option button or the **Flash=>PLC** option button in the **Flash Memory Read/Write section**, select contents which will be transferred, select the **Enable** option button or the **Disable** option button in the **Warning Message** section. (A data restoration is automatically performed when the flat battery in a PLC causes a data loss. If the **Enable** option button in the **Warning Message** section is selected, a warning message appears when a data restoration is automatically performed.) After the setting in the **PLC Permanent Backup Setting** window is complete, the users can click **OK**.



17.3 CARD Utility

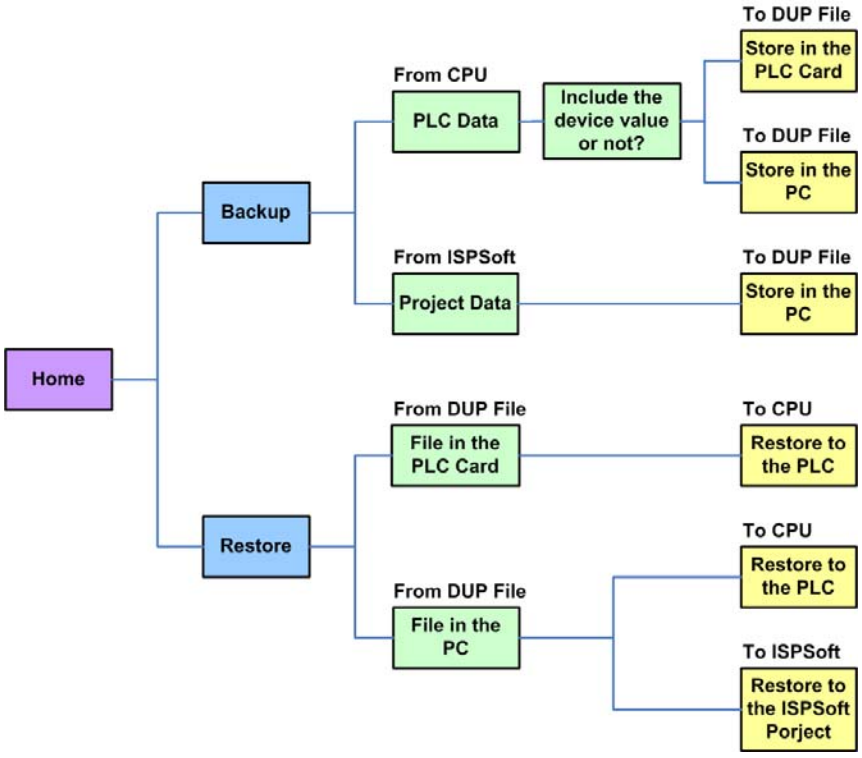
17.3.1 Introduction of CARD Utility

There are SD slots on AH500 series CPU modules. Users can back up/restore data in an AH500 series CPU module through the use of a memory card. Besides, ISPSoft provides CARD Utility for AH500 series CPU modules. The users can back up/restore data in an AH500 series CPU module or an ISPSoft project through a wizard. The program code, the parameter setting, the hardware configuration, and the network configuration in an AH500 series CPU module or an ISPSoft project can be backed up. The values in the devices in an AH500 series CPU module can also be backed up. Please refer to operation manuals or technical documents for more information about the specifications of the SD cards which can be inserted into AH500 series CPU modules, and the usage of the SD cards.

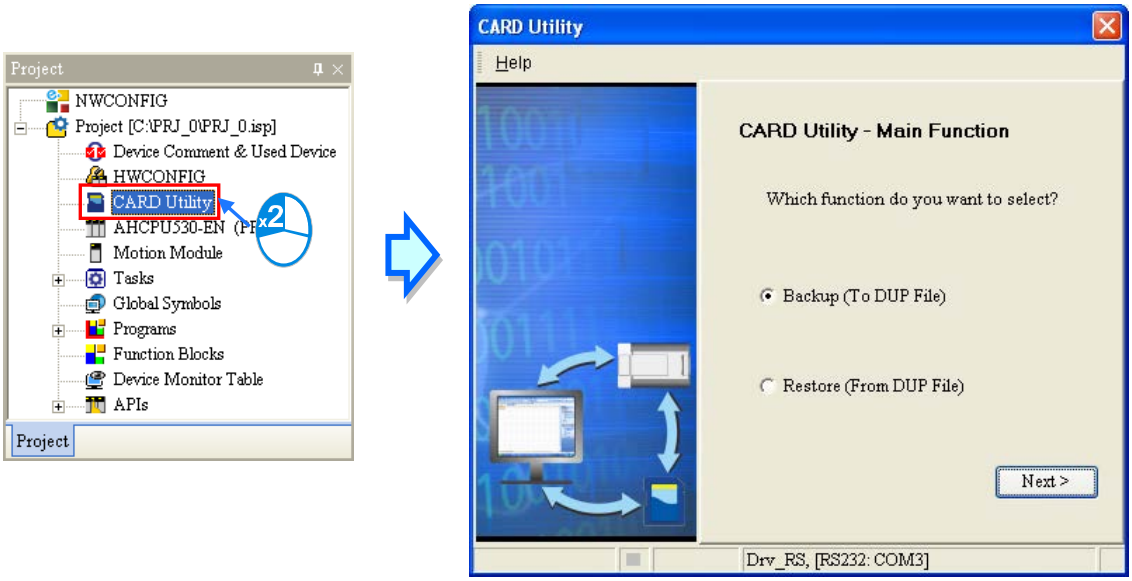
The hardware configuration stored in an AH500 series CPU module is data which is only related to the AH500 series CPU module itself. If users want to back up a hardware configuration, only the part of the network configuration which is related to the AH500 series CPU module selected will be backed up. The part of the network configuration backed up consists of a routing table and an Ether Link. Likewise, if the users want to restore data backed up to an ISPSoft project, there will be no network configuration in the ISPSoft project. Please refer to chapter 16 for more information about a network configuration.

The functions supported by CARD Utility are described below. The diagram below is a flowchart.

- If users export data in an AH500 series CPU module as a backup file (*.dup), the data exported can be saved in the memory card inserted in the AH500 series CPU module, or a folder in the computer. The users can decide whether to back up the values in the devices in the AH500 series CPU module.
- If users export an ISPSoft project for an AH500 series CPU module as a backup file (*.dup), the ISPSoft project exported can only be saved in a folder in the computer, and the values in the devices in the AH500 series CPU module are not backed up.
- Users can put the backup file saved in the memory card inserted in an AH500 series CPU module into the AH500 series CPU module.
- Users can put a backup file (*.dup) saved in a computer into the AH500 series CPU module connected to the computer, or restore the backup file to an ISPSoft project. If the users choose to restore the backup file to an ISPSoft project, the system will automatically skip the values in the devices and the hardware configuration in the backup file.



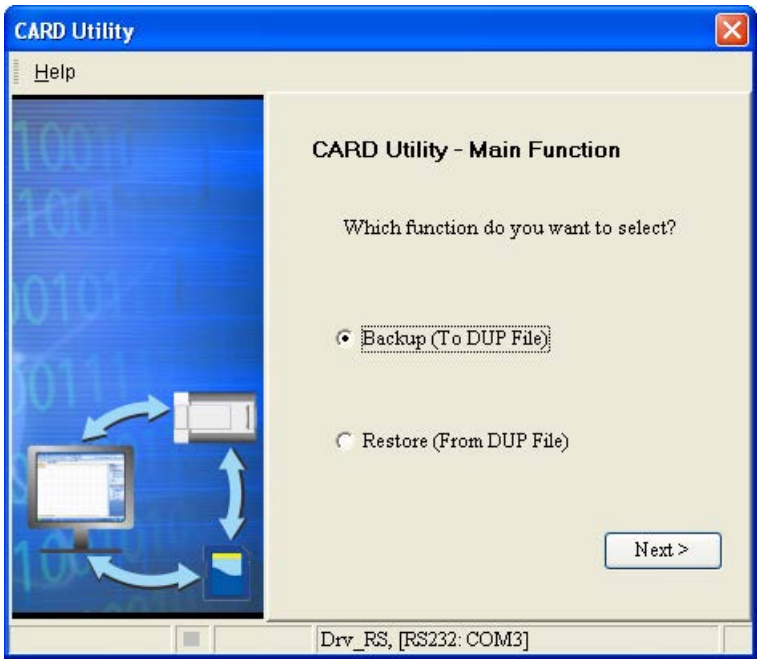
After users double-click **CARD Utility** in the project management area, the system will open the **CARD Utility** window.



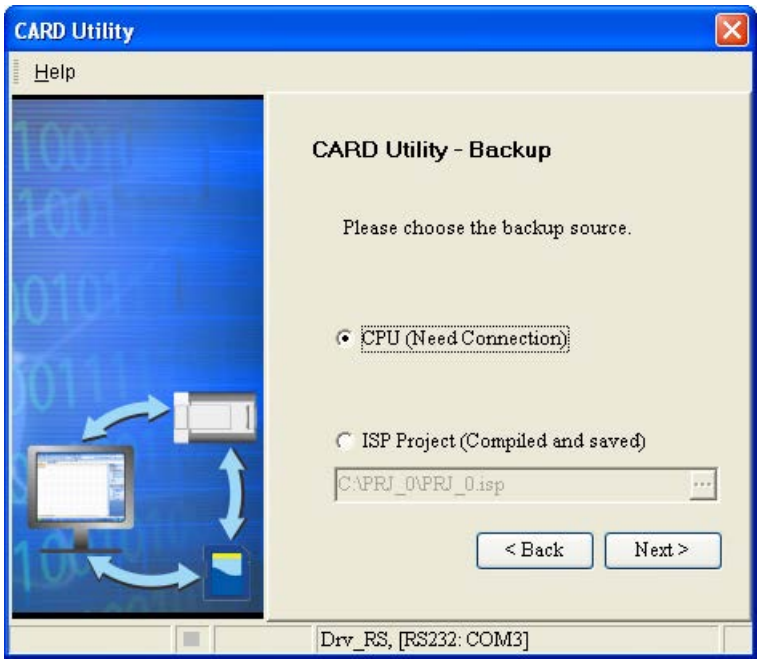
17.3.2 Backup

If the backup source/backup destination is an AH500 series CPU module or the memory card inserted in an AH500 series CPU module, users have to make sure that ISPSoft is connected to the AH500 series CPU module normally. Please refer to section 2.4 for more information.

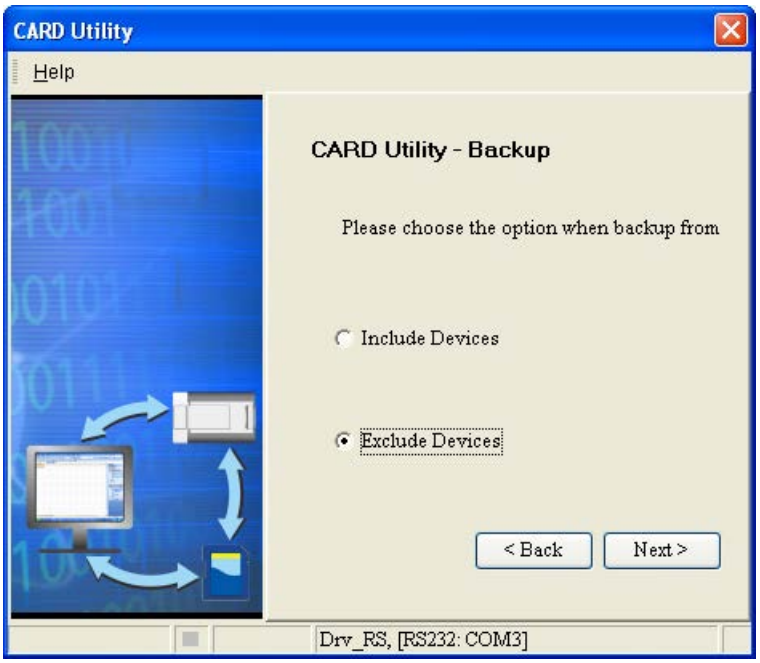
(1) Select the **Backup (To DUP File)** option button in the **CARD Utility** window, and then click **Next**.



(2) Select a backup source, and then click **Next**.
After the users select the **ISP Project (Compiled and saved)** option button, they have to click ..., and select an isp file in the **Open** window. If the program in the isp file selected is not compiled, a message appears when the isp file is backed up. Open the isp file with ISPSoft, compile the program in the isp file, and save the isp file. After the program in the isp file is compiled, the users can back up the isp file.



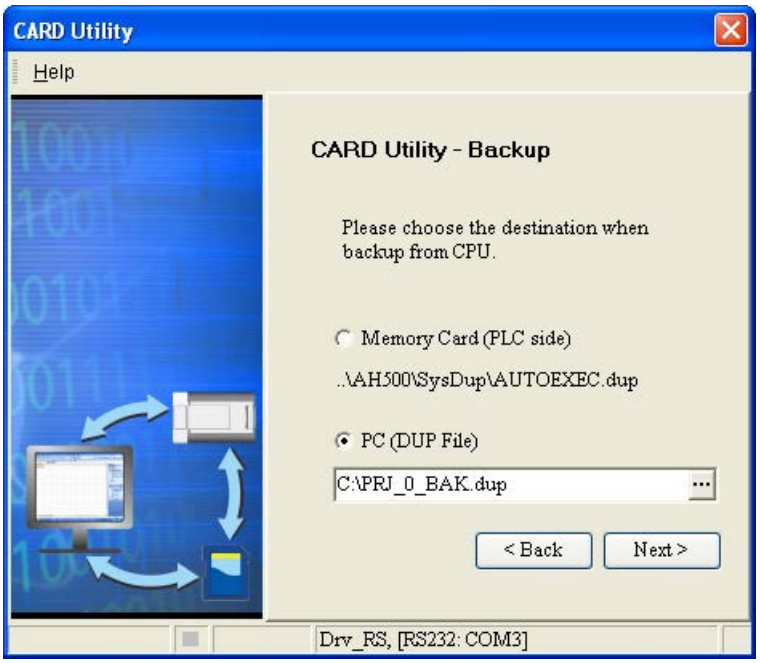
(3) After the users select the **CPU (Need Connection)** option button, they have to decide whether to back up the values in the devices in the AH500 series CPU module which is connected to ISPSoft.



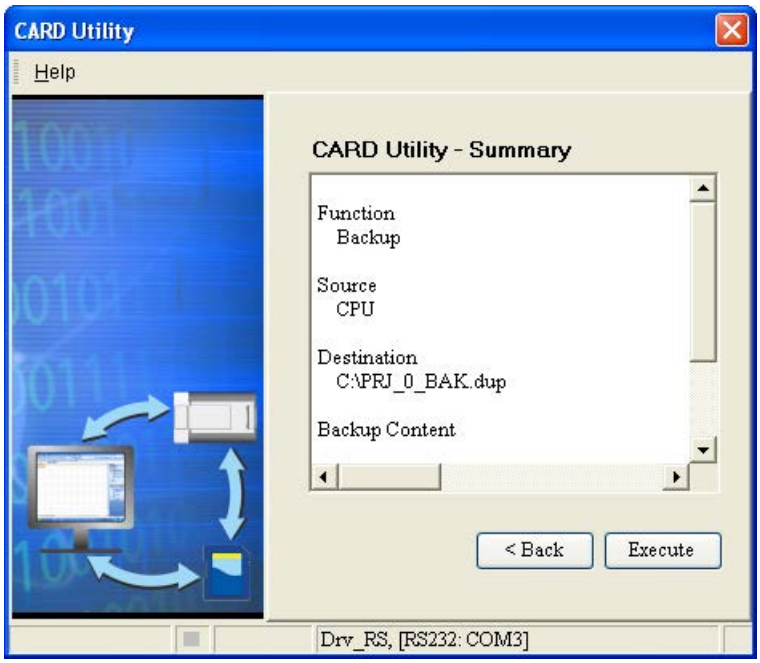
(4) Select a backup destination. If the backup source is an ISPSoft project, the backup destination must be a computer.

a. If the **Memory Card (PLC Side)** option button is selected, the filename of the backup file which will be produced will be **AUTOEXEC.dup**, and the path which points to the backup file will be **Root directory of the memory card\AH500\SysDup\AUTOEXEC.dup**.

b. If the **PC (DUP File)** option button is selected, the users have to click ···, select a folder in the **Save in** drop-down list box in the **Save As** window, and type a filename in the **File name** box.

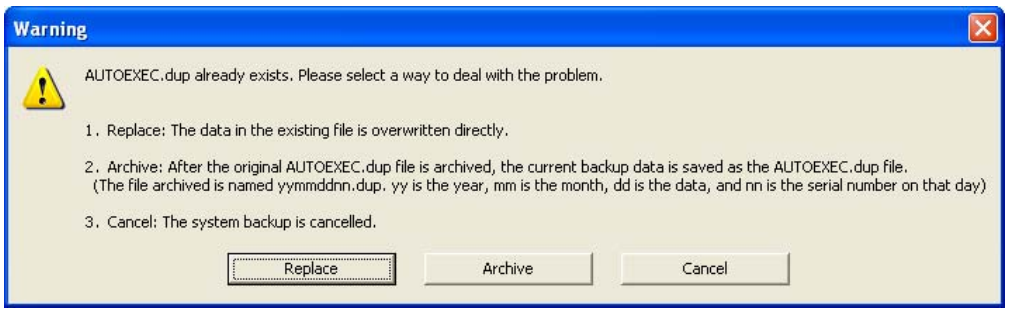


(5) After the users make sure that the summary in the **CARD Utility** window is consistent with the data backup which will be performed, they can click **Execute**.



Even if the users click **Cancel** to stop ISPSoft from performing the data backup in the process of backing up data in the AH500 series CPU module onto the memory card inserted in the AH500 series CPU module, the AH500 series CPU module will still performs the data backup. The users can turn off the AH500 series CPU module to stop the data backup from being performed. However, the backup file produced is not a complete backup file. As a result, the users have to delete the backup file from the memory card.

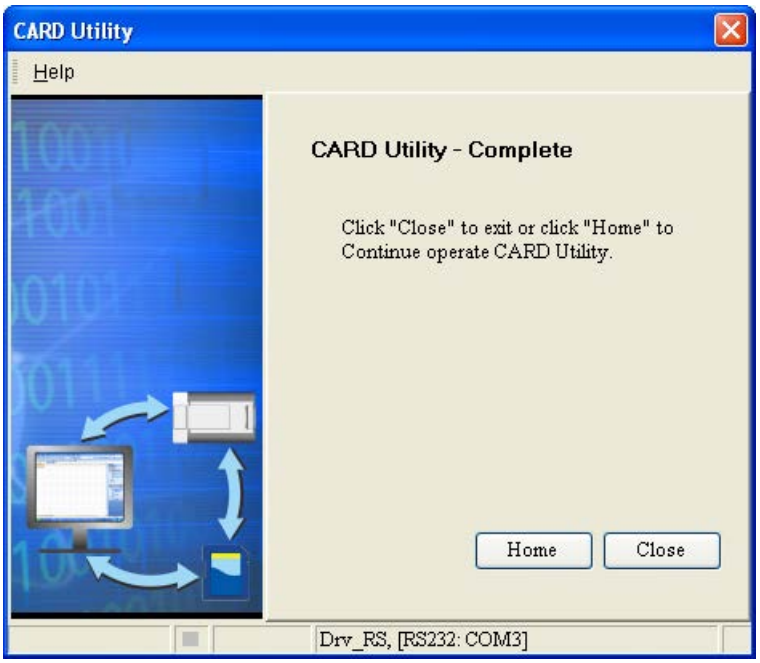
If the **Memory Card (PLC Side)** option button is selected, the filename of the backup file which will be produced will be **AUTOEXEC.dup**, and the path which points to the backup file will be **Root directory of the memory card\AH500\SysDup\AUTOEXEC.dup**. If there is an old backup file in the memory card inserted in the AH500 series CPU module which is connected to ISPSoft, the **Warning** window will appear. The users have to click **Replace**, Archive, or **Cancel** in the Warning window according to the message in the window.



If the data backed up is protected by passwords, these passwords will also be backed up.

Data backup	Description
CPU module→Memory card	The data backed up includes the PLC ID and the PLC password set in the CPU module.
CPU module→Computer	The system asks users to type a PLC ID and a PLC password. If the PLC ID and the PLC password typed are correct, the data backup will be performed. The data backed up includes the PLC ID and the PLC password.
ISPSoft project→Computer	The data backed up includes the program ID and the project password set in the ISPSoft project.

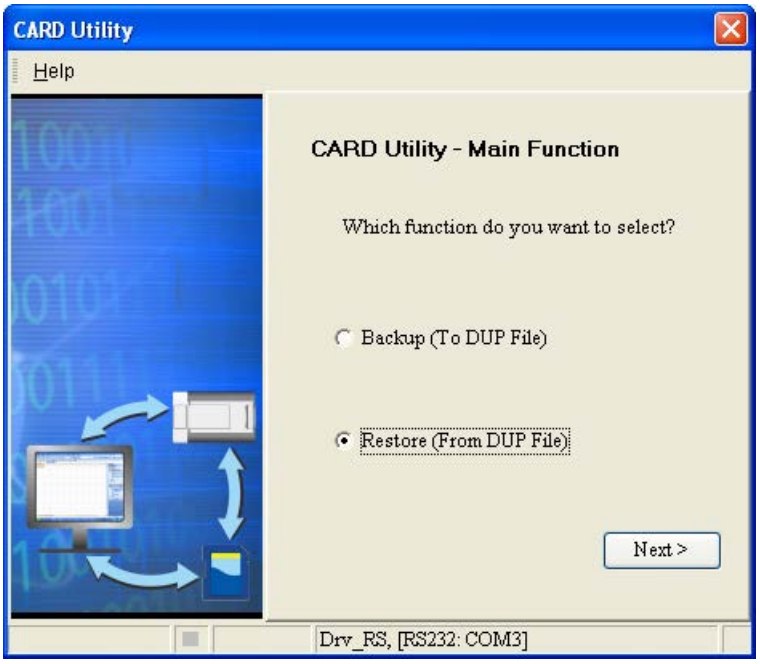
(6) After the data backup is performed, the users can click **Home** or **Close** in the **CARD Utility** window.



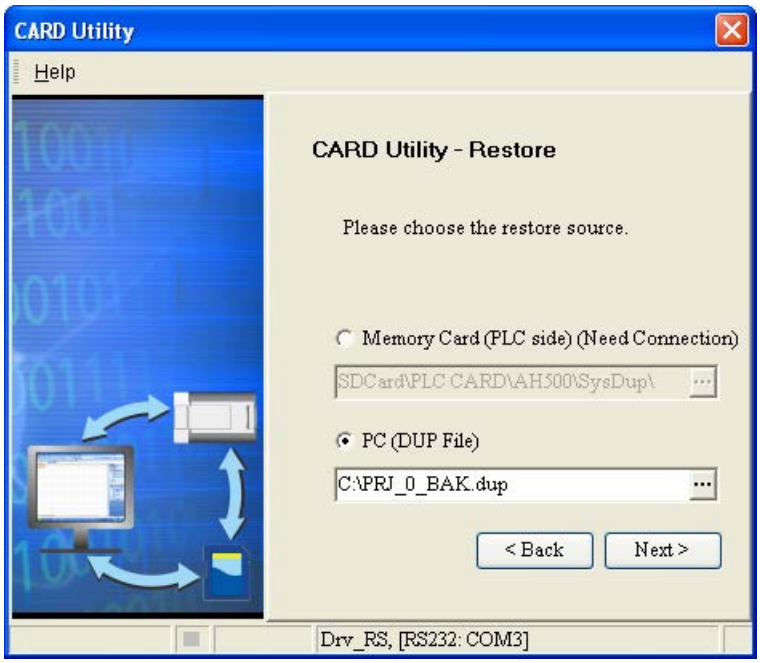
17.3.3 Restoration

If the restoration source/restoration destination is an AH500 series CPU module or the memory card inserted in an AH500 series CPU module, users have to make sure that ISPSoft is connected to the AH500 series CPU module normally. Please refer to section 2.4 for more information.

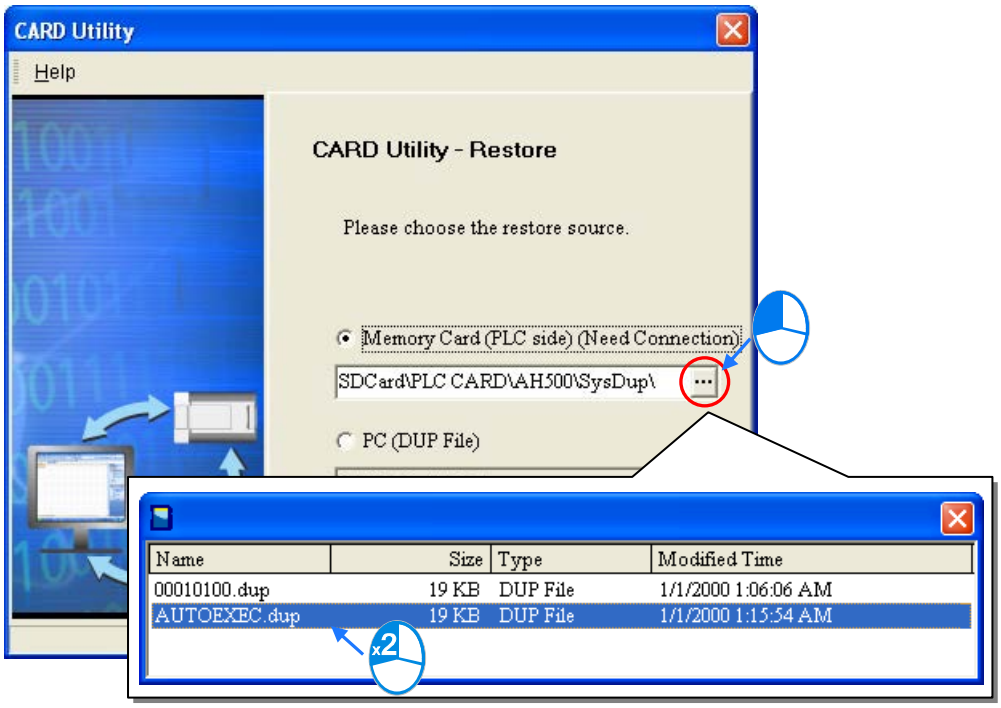
(1) Select the **Restore (From DUP File)** option button in the **CARD Utility** window, and then click **Next**.



(2) Select a restoration source, click ···, and select a backup file.



If the **Memory Card (PLC side) (Need Connection)** option button is selected, the backup files in the memory card inserted in the AH500 series connected to ISPSoft will be displayed in a window after ··· is clicked. The users have to double-click a backup file in the window.

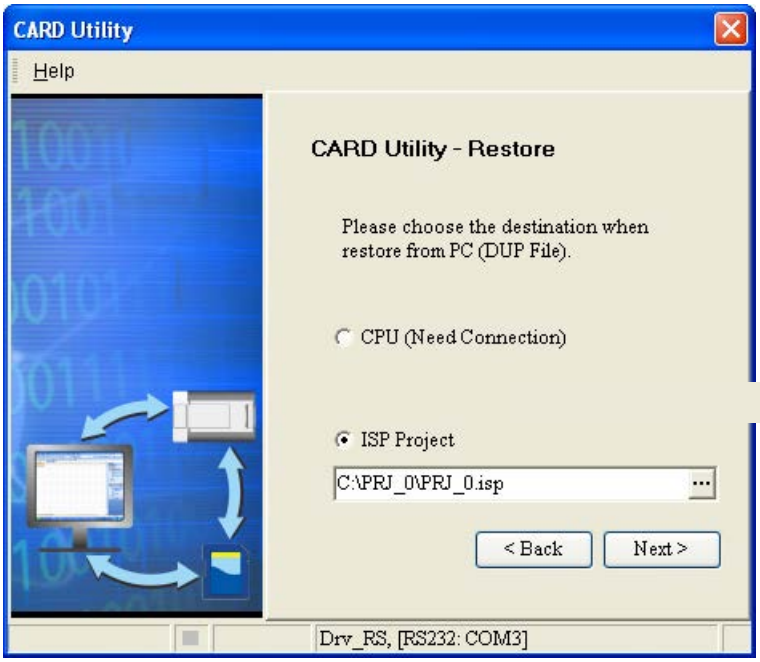


(3) Select a restoration destination, and then click **Next**.

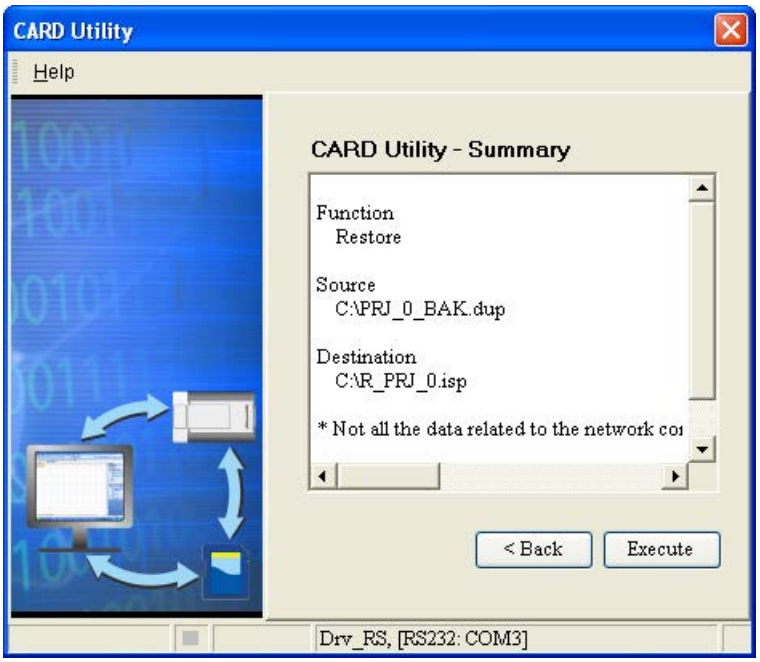
a. If the users want to put the backup file selected into the AH500 series CPU module which is connected to ISPSoft, they have to select the **CPU (Need Connection)** option button. If the restoration source is the memory card inserted in the AH500 series CPU module connected to ISPSoft, the restoration destination must be the AH500 series CPU module.

b. If the **ISP Project** option button is selected, the users have to click ···. After the users click ···, they have to specify a filename and a path. If the path specified point to a file

which exists in the computer, the file will be overwritten after the data restoration is peformed.



(4) After the users make sure that the summary in the **CARD Utility** window is consistent with the data restoration which will be performed, they can click **Execute**.



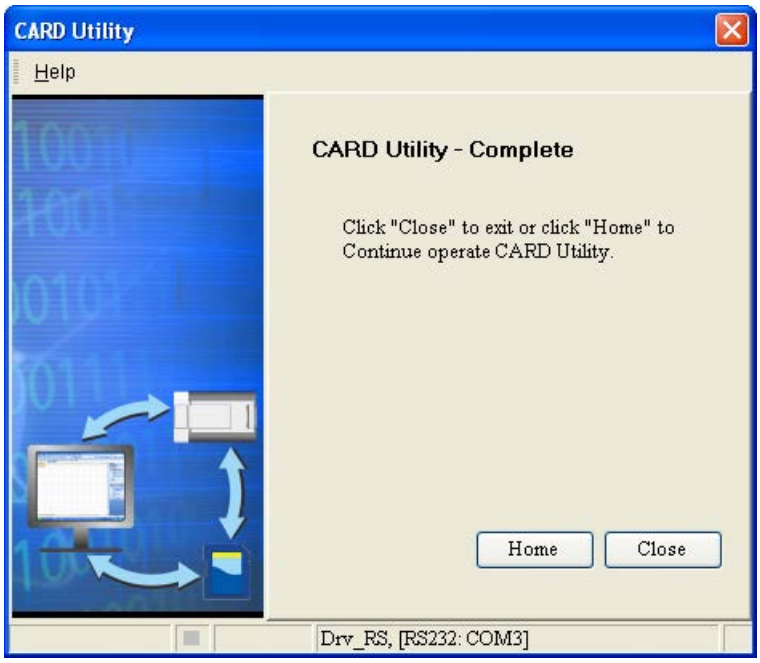
If the users click **Cancel** in the process of restoring data to the AH500 series CPU module, the data will not be completely restored. To prevent the AH500 series CPU module from operating incorrectly, the users have to restore the AH500 series CPU module to the factory setting if they do not perform the data restoration again. Besides, the AH500 series CPU module will still performs the data restoration even if the users click **Cancel** in the process of restoring a backup file in the memory card inserted in the AH500 series CPU module. The users can turn off the AH500 series CPU module to stop the data restoration from being performed.

If restoration source/restoration destination contains a password and an ID, the password and

the ID will be processed as follows.

Data restoration	Description
Memory card→CPU module	a. The ID in the backup file must be the same as the ID in the CPU module, otherwise the data restoration will not be performed. b. If there is a PLC password in the CPU module, the password in the backup file must be the same as the PLC password in the CPU module. Otherwise the data restoration will not be performed. c. If there is no PLC password in the CPU module, and there is a password in the backup file, the system will perform the data restoration, and the password in the backup file will become the PLC password in the CPU.
Computer→CPU module	a. The ID in the backup file must be the same as the ID in the CPU module, otherwise the data restoration will not be performed. b. If there is a PLC password in the CPU module, the password in the backup file must be the same as the PLC password in the CPU module. Otherwise the data restoration will not be performed, and a message will appear. c. If there is no PLC password in the CPU module, and there is a password in the backup file, the system will perform the data restoration, and the password in the backup file will become the PLC password in the CPU.
Computer→ ISPSoft project	The ID and the password in the backup file will become the program ID and the project password in an ISPSoft project.

(7) After the data restoration is performed, the users can click **Home** or **Close** in the **CARD Utility** window.



Appendix A USB Connection

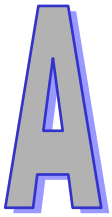


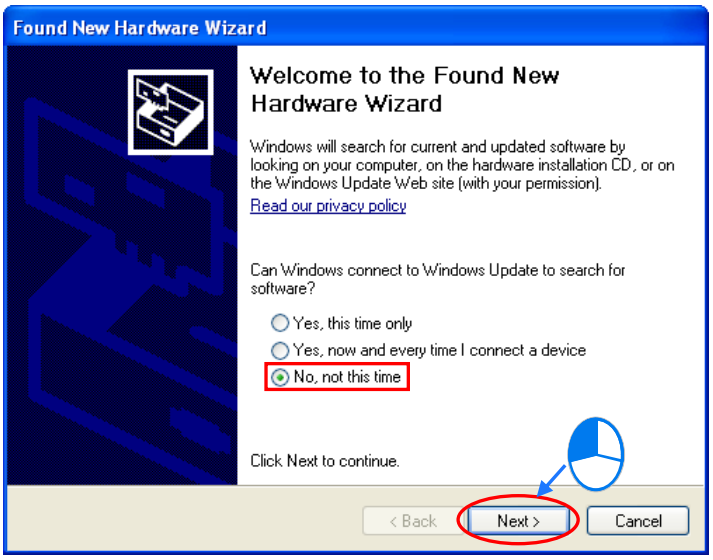
Table of Contents

A.1 Installing the USB Driver for a PLC A-2
A.2 Creating a Driver Whose Connection Type is USB in COMMGR A-5
A.3 Setting the USB Port on a DVP-SX2 Series PLC A-7

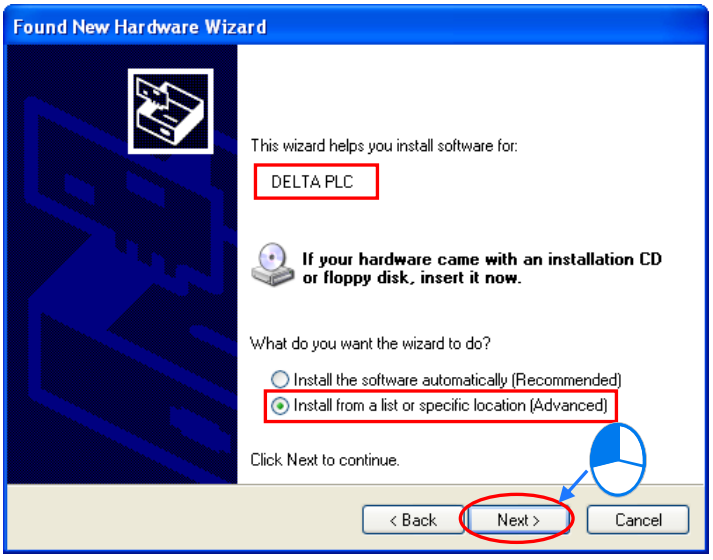
A.1 Installing the USB Driver for a PLC

The installation of the USB driver for a PLC on Windows XP is introduced below. If users want to install the USB driver for a PLC on another operating system, they have to refer to the instructions in the operating system for more information about the installation of new hardware.

(1) Make sure that the PLC is supplied with power normally. Connect the PLC to a USB port on the computer with a USB cable. Select the **No, not this time** option button in the **Found New Hardware Wizard** window, and then click **Next**.



(2) The name of the USB device detected is displayed in the window. The device name shown in the figure below is the name of an AH500 series CPU module. Different models have different names. Please select the **Install from a lost or specific location (Advanced)** option button.

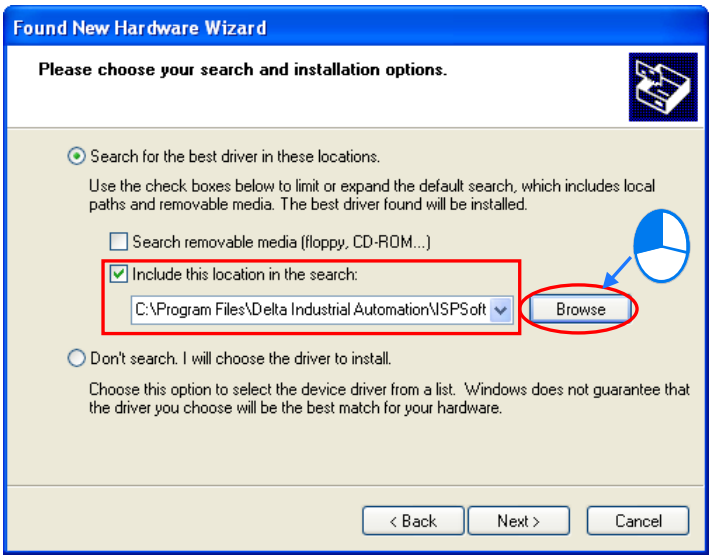


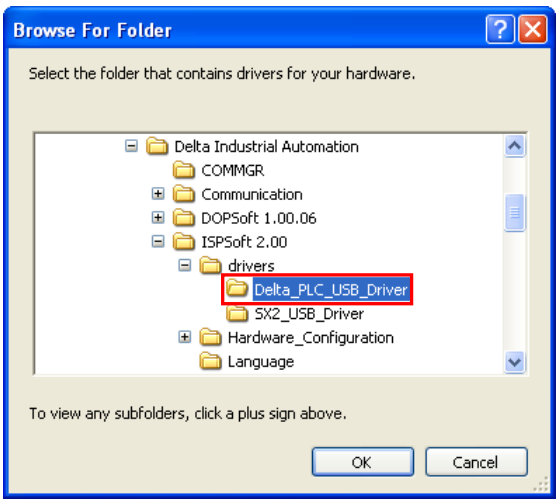
(3) The Delta PLCs which are equipped with built-in USB interfaces are listed in the tabled below. After ISPSoft version 2.00 or above is installed, the drivers for DVP-SX2 series PLCs, DVP-SE series PLCs, and AH500 series CPU modules will be in the folders denoted by the paths in the table.

Model	Path
DVP-SX2	Installation path of ISPSoft\drivers\SX2_USB_Driver\
DVP-SE	Installation path of ISPSoft \drivers\Delta_PLC_USB_Driver\
AH500	Installation path of ISPSoft \drivers\Delta_PLC_USB_Driver\

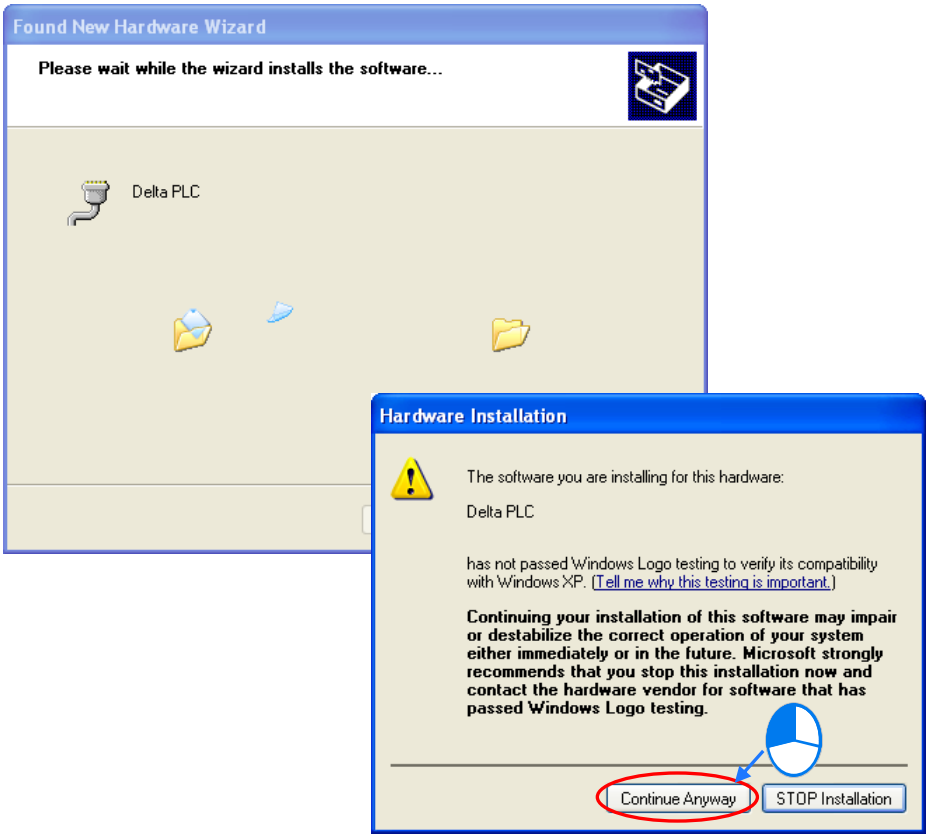
***. The default installation path of ISPSoft is C:\Program Files\Delta Industrial Automation\ISPSoftx.xx.**

Specify a path according to the PLC which is connected. If the driver for a PLC is gotten in another way, users have to specify the corresponding path. Click **Next** to carry on the installation.

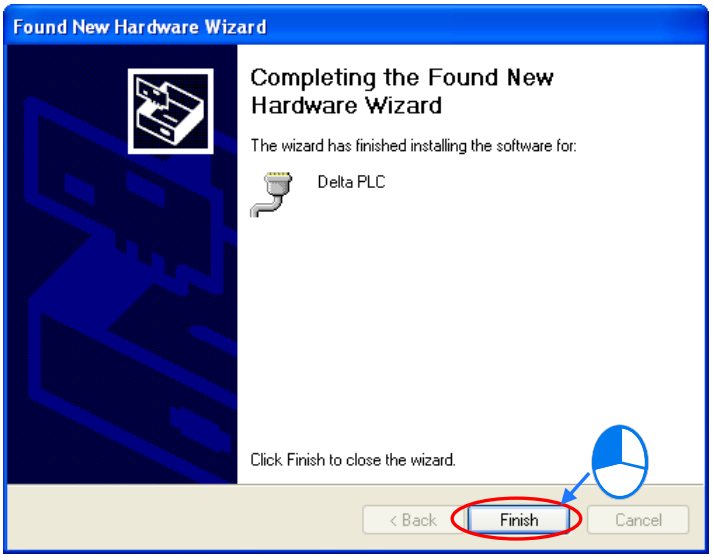




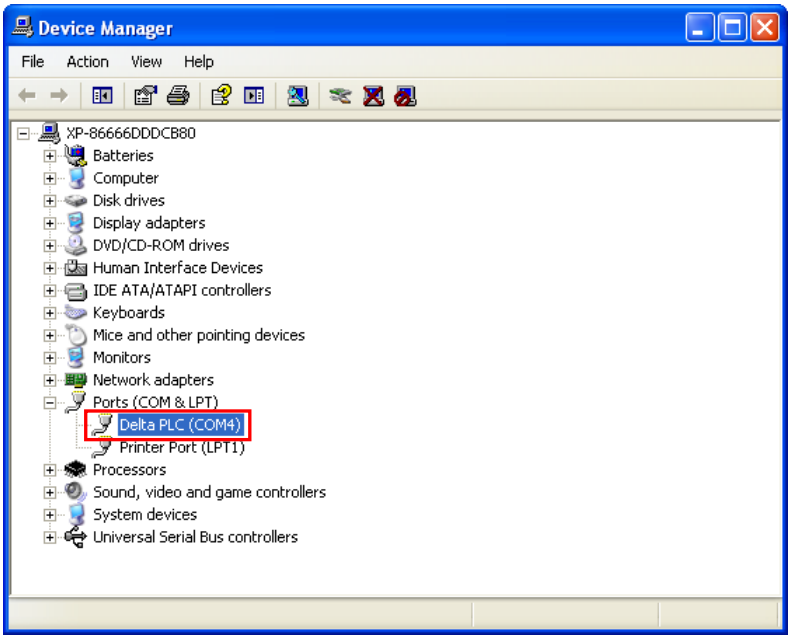
(4) After the correct driver is found in the folder denoted by the path, the system will install the driver. If the **Hardware Installation** window appears during the installation, please click **Continue Anyway**.



(5) Click **Finish** after the installation is finished.



(6) Open the **Device Manager** window after the installation is finished. If the name of the USB device connected is under **Ports (COM&LPT)**, the installation of the driver is successful. The operating system assigns a communication port number to the USB device.



***. The device name shown in the figure above is the name of an AH500 series CPU module. Different models have different names.**

Additional remark

- If the PLC is connected to another USB port on the computer, the system may ask users to install the driver again. The users can follow the steps above, and install the driver again. After the driver is installed, the communication port number that the operating system assigns to the USB device may be different.
- If Windows XP SP3 has not been installed on the computer, an error message will appear during the installation. Users can deal with the problem in either way below.

  (a) Cancel the installation, install Windows XP SP3, and reinstall the driver according to the steps above.

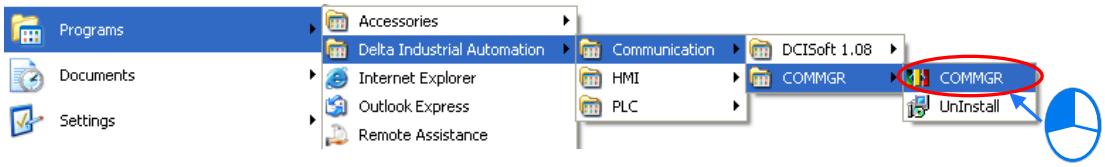
  (b) Get the file needed, and specify the path pointing to the file in the **Files Needed** window.

A.2 Creating a Driver Whose Connection Type is USB in COMMGR

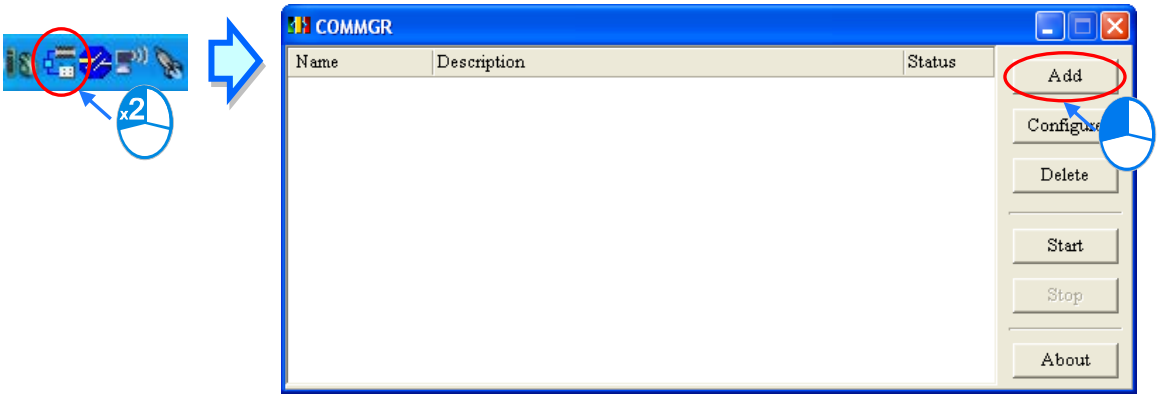
The steps of creating a driver whose connection type is USB in COMMGR are as follows.

(1) Make sure that the PLC is supplied with power normally. Connect the PLC to a USB port on the computer with a USB cable.

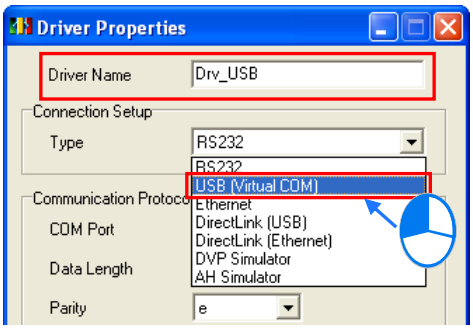
(2) Make sure that COMMGR is started, and the icon representing COMMGR is displayed on the system tray. If the icon representing COMMGR is not displayed on the system tray, users can start COMMGR by clicking the shortcut on the **Start** menu (**Start**>**Programs**>**Delta Industrial Automation**>**Communication**>**COMMGR)**.



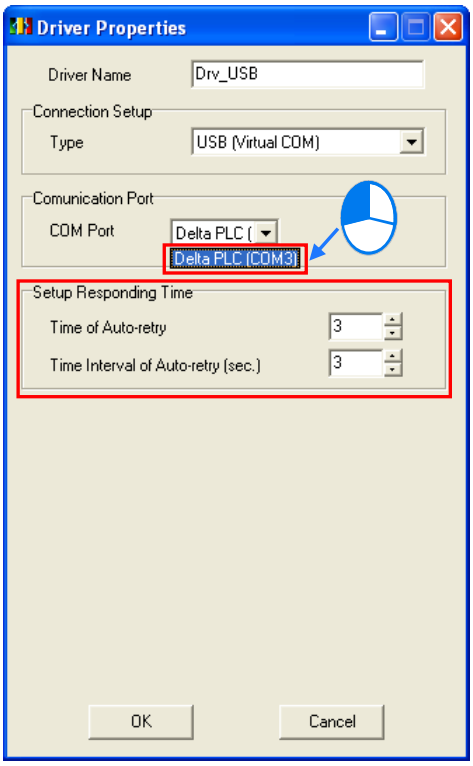
(3) Double-click the icon representing COMMGR on the system tray, and then click **Add** in the **COMMGR** window.



(4) Type a driver name in the **Driver Name** box, and then select **USB (Virtual COM)** in the **Type** drop-down list box in the **Connection Setup** section. If the PLC connected is a DVP-SX2 series PLC, **RS232** in the **Type** drop-down list box must be selected. Please refer to the following section for more information.

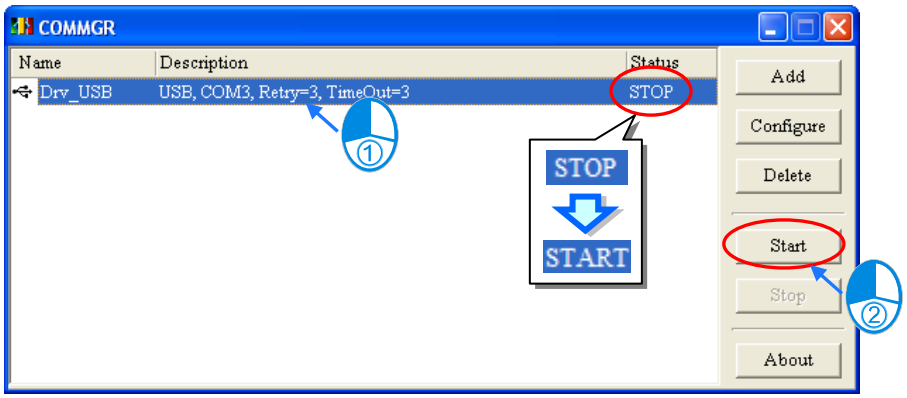


(5) Select a device name which corresponds to the PLC connected in the **COM Port** drop-down list box in the **Communication Port** section, select the number of times the sending of a command is retried if a connection error occurs in the **Time of Auto-retry** box, and select an interval of retrying the sending of a command in the **Time Interval of Auto-retry** box. Click **OK**.



If the device name corresponding to the PLC connected does not appear in the **COM Port** drop-down list box, users have to check whether the device name of the PLC connected is under **Ports (COM&LPT)**. If the device name of the PLC connected is not under **Ports (COM&LPT)**, the users have to check whether the PLC is correctly connected to the computer with the USB cable, and check whether the driver for the PLC is corrected installed on the computer. After the users make the checks, they have to close the **Driver Properties** window, and open the window again.

(6) Click the driver created in the **COMMGR** window, and then click **Start**.

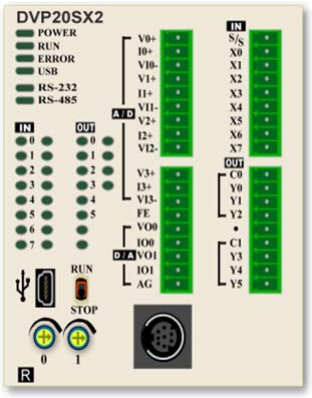


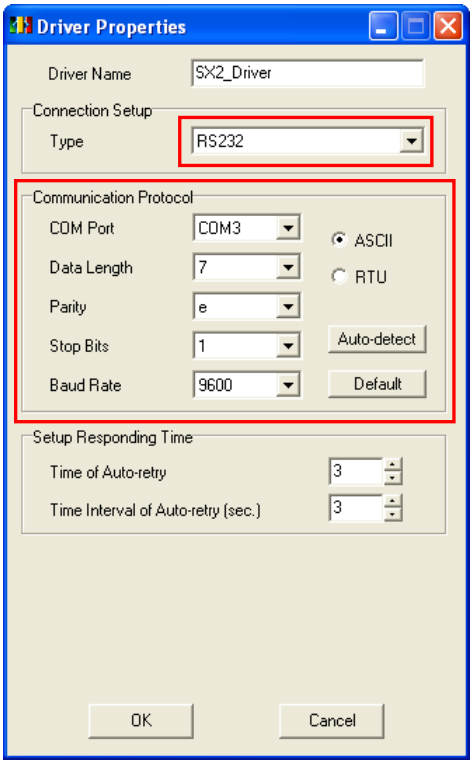
***. Please refer to section 2.4 for more information about subsequent tests and the operation of COMMFR.**

A.3 Setting the USB Port on a DVP-SX2 Series PLC

The operation of the USB port on a DVP-SX2 series PLC differs from the operation of the USB ports on other models in that a circuit which converts USB to RS-232 is installed in the DVP-SX2 series PLC. As a result, the operation inside the DVP-SX2 series PLC adopts RS-232 although the port on the PLC is a USB port. Users must select **RS232** in the **Type** drop-down list box in the **Driver Properties** window if they want to create a driver in COMMGR. The setting of the communication protocol for exchanging data through the USB port is the same as the setting of the communication protocol for exchanging data through a general RS-232 port.

Users can set the communication protocol for exchanging data through the USB port on a DVP-SX2 PLC by means of writing a value into D1109. Please refer to the programming manual for DVP-SX2 series PLCs for more information. If users want to create the driver in COMMGR, they have to make sure that the communication protocol for exchanging data through the driver is the same as the communication protocol for exchanging data through the USB port on a DVP-SX2.


DVP20SX2
POWER
RUN
ERROR
USB
RS-232
RS-485
IN
OUT
RUN
STOP
R


Driver Properties
Driver Name
SX2_Driver
Connection Setup
Type
RS232
Communication Protocol
COM Port
COM3
Data Length
7
Parity
e
Stop Bits
1
Baud Rate
9600
ASCII
RTU
Auto-detect
Default
Setup Responding Time
Time of Auto-retry
3
Time Interval of Auto-retry (sec.)
3
OK
Cancel

Appendix B Important Points Related to AH500 Series CPU Modules

Table of Contents

B.1 Device Addresses in AH500 Series CPU Modules B-2
B.1.1 Devices in AH500 series CPU Modules B-2
B.1.2 Manipulating the Bits in the X/Y/D/L Devices B-3
B.1.3 Updating Input/Output Directly B-4
B.2 Device Resources in AH500 Series CPU Modules B-5
B.2.1 AHCPU500-EN/AHCPU500-RS2 ..... B-5
B.2.2 AHCPU510-EN/AHCPU510-RS2 ..... B-6
B.2.3 AHCPU520-EN/AHCPU520-RS2 ..... B-6
B.2.4 AHCPU530-EN/AHCPU530-RS2 ..... B-6

AH500 series PLCs are medium-sized PLCs released by Delta Electronics, Inc. in 2012. They are slightly different from DVP series PLCs. Users have to read the manuals for different models before they develop projects. The points to which users have to pay attention when they develop a project for an AH500 series CPU module are described in appendix B.

B.1 Device Addresses in AH500 Series CPU Modules

B.1.1 Devices in AH500 series CPU Modules

The devices in AH500 series CPU modules are listed in the following table.

Device type	Access property	Description
X	Bit/Word	Input relay It represents the state of a digital input. (*1)
Y	Bit/Word	Output relay It represents the state of a digital output. (*1)
M	Bit	Auxiliary relay The state of a bit is stored in an auxiliary relay.
S	Bit	Stepping relay It is used as a state flag for a step in a sequential function chart.
D	Bit/Word	Data register Operation data is stored in a data register. (*1)
L	Bit/Word	Link register A link register is used for data exchange. (*1) (*2)
T	Bit	Timer If a timer is used as a bit device, the timer is ON when the time interval which is measured conforms to the setting value, and the timer is OFF when the time interval which is measured does not conform to the setting value.
	Word	Timer If a timer is used as a word device, the value stored in the timer is the present value of the timer.
C	Bit	Counter If a counter is used as a bit device, the counter is ON when the number of times a particular event or process has occurred conforms to the setting value, and the counter is OFF when the number of times a particular event or process has occurred conforms to the setting value.
	Word	Counter If a counter is used as a word device, the value stored in the counter is the present value of the counter.
HC	Bit	32-bit counter If a 32-bit counter is used as a bit device, the 32-bit counter is ON when the number of times a particular event or process has occurred conforms to the setting value, and the 32-bit counter is OFF when the number of times a particular event or process has occurred conforms to the setting value.
	Word	32-bit counter If a 32-bit counter is used as a word device, the value stored in the 32-bit counter is the present value of the 32-bit counter.

Device type	Access property	Description
E	Word	Index register The value stored in an index register indicates the offset for a device modified. (*3)
SM	Bit	Special auxiliary relay It is used as a state flag for a special function of the system. (*4)
SR	Word	Special data register It is used as a data register for a special function of the system. (*4)

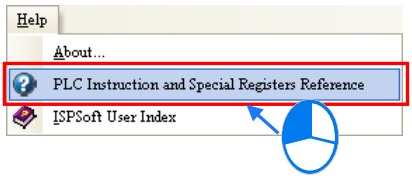
***1. AH500 series CPU modules support the manipulation of the bits in the input relays, the output relays, the data registers, and the link registers. For example, X0.1 indicates that bit 1 in X0 is manipulated, Y0.1 indicates that bit 1 in Y0 is manipulates, D0.1 indicates that bit 1 in D0 is manipulates, and L0.1 indicates that bit 1 in L0 is manipulated. Please refer to the following sections for more information.**

***2. A link register is used for data exchange, but it can also be used as a general data register.**

***3. The index registers that AH500 series CPU modules have are E devices. AH500 series CPU modules do not have F devices.**

***4. AH500 series PLCs are different from DVP series PLCs in that the special auxiliary relays and the special data registers in AH500 series PLCs are called SM devices and SR devices.**

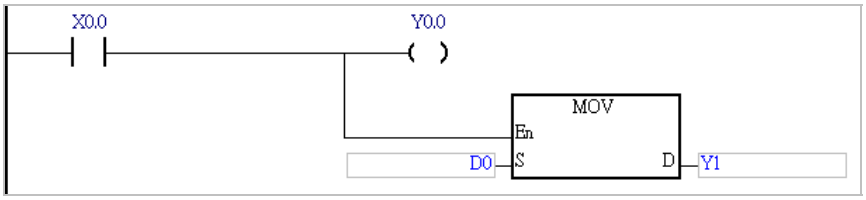
Please refer to AH500 Programming Manual or **PLC Instructions and Special Registers Reference** in a project for an AH500 series CPU module in ISPSoft for more information about the devices in AH500 series CPU modules.



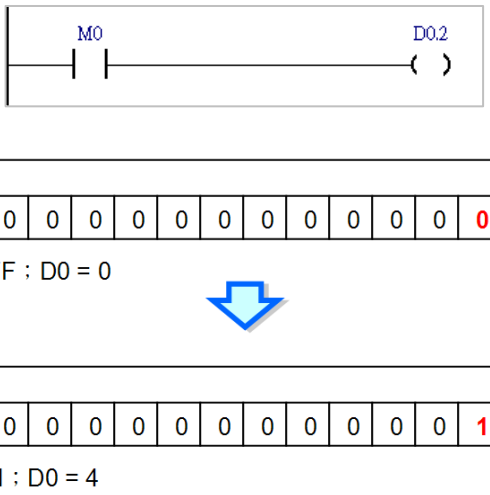
B.1.2 Manipulating the Bits in the X/Y/D/L Devices

The X/Y devices in DVP series PLCs are represented by X0, Y1, and etc. The X/Y devices which are used as bit devices in AH500 series CPU modules are represented by X0.0, Y0.0, and etc. Besides, the X/Y devices which are used as word devices in AH500 series CPU modules are represented by X0, Y0, and etc. X0.0 represents the first bit (the lowest bit) in X0, and X0.15 represents the last bit (the highest bit) in X0.

After the value in D0 is moved to Y1, the states of Y1.0~Y1.15 will change.



The bits in the X/Y devices in AH500 series CPU modules can be manipulated. The bits in the D/L devices in AH500 series CPU modules can also be manipulated in the same way. Please refer to the following example. The state of M0 is sent to D0.2. Suppose the initial value in D0 is 0. When M0 is ON, the value in D0 becomes 4.

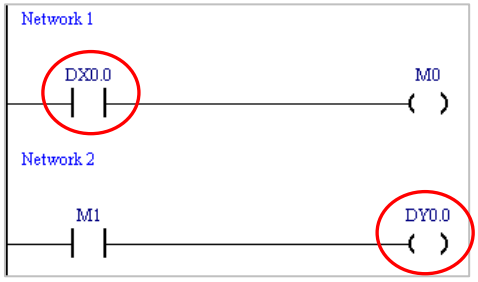


If the bits in the X/Y/D/L devices which are word devices in an AH500 series CPU module are manipulated, the memory blocks that the X/Y/D/L devices occupy are manipulated. However, the memory blocks that the T/C/HC devices used as bit devices occupy are different from the memory blocks that the T/C/HC devices used as word devices occupy. If a timer is used as a bit device, the timer is ON when the time interval which is measured conforms to the setting value, and the timer is OFF when the time interval which is measured does not conform to the setting value. If a timer is used as a word device, the value stored in the timer is the present value of the timer. If a counter is used as a bit device, the counter is ON when the number of times a particular event or process has occurred conforms to the setting value, and the counter is OFF when the number of times a particular event or process has occurred conforms to the setting value. If a counter is used as a word device, the value stored in the counter is the present value of the counter.

B.1.3 Updating Input/Output Directly

The way in which the input/output in an AH500 series CPU module is updated is similar to the way in which the input/output in a DVP series PLC is updated in that the input/output in either model will be updated after the program in the model is scanned. However, an AH500 series CPU module supports a special modification. If the X device or the Y device in the network in a program is modified by “D”, the system updates the state of the device when the network is executed. Only X devices and Y devices can be modified by “D”. Symbols can not be modified by “D”.

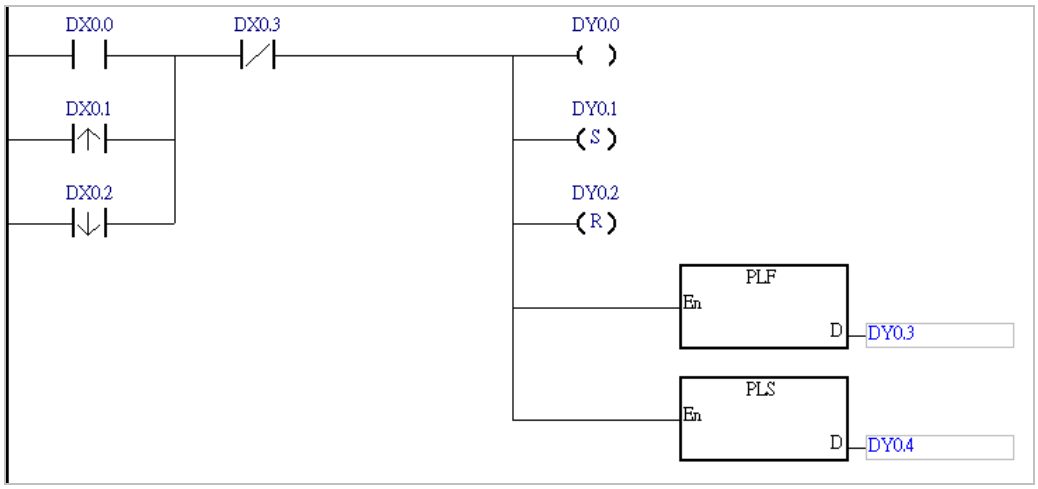
When network 1 is executed, the system reads the state of X0.0. Likewise, the system sends the state of M1 to Y0.0 when network 2 is executed.



The instructions which support the modification by “D” are listed in the following table. If an X device which is modified by “D” or a Y device which is modified by “D” is used in an instruction which is not listed in the table below, the system will regard it as incorrect syntax, and an error will occur after the program is compiled. Besides, only the bits in the X devices which are word devices and the bits in the Y devices which are word devices can be modified by “D”, and the operands in the applied instructions can not be modified by “D”.

Device type	Instruction
X	LD/LDI/LDP/LDF/OR/ORI/ORP/ORF/AND/ANI/ANDP/ANDF
Y	OUT/SET/RST/PLS/PLF

The figure below is an example of applying the modification by “D” legally.



The figure below is an example of applying the modification by “D” illegally. The Y device is modified by “D”. It can not be used as an input contact.

B.2 Device Resources in AH500 Series CPU Modules

B.2.1 AHCPU500-EN / AHCPU500-RS2

Device type	Address range	
X	Word: X0~X63	Bit: X0.0~X63.15
Y	Word: Y0~Y63	Bit: Y0.0~Y63.15
D	Word: D0~D16383	Bit: D0.0~D16383.15
L	Word: L0~L16383	Bit: L0.0~L16383.15
M	M0~M8191	
S	S0~S2047	
C	C0~C2047	
HC	HC0~HC63	
T	T0~T2047	
E	E0~E31	
SR	SR0~SR2047	
SM	SM0~SM2047	
Number of memory blocks which can be assigned to function block instances	64	

B.2.2 AHCPU510-EN / AHCPU510-RS2

Device type	Address range	
X	Word: X0~X127	Bit: X0.0~X127.15
Y	Word: Y0~Y127	Bit: Y0.0~Y127.15
D	Word: D0~D32767	Bit: D0.0~D32767.15
L	Word: L0~L32767	Bit: L0.0~L32767.15
M	M0~M8191	
S	S0~S2047	
C	C0~C2047	
HC	HC0~HC63	
T	T0~T2047	
E	E0~E31	
SR	SR0~SR2047	
SM	SM0~SM2047	
Number of memory blocks which can be assigned to function block instances	256	

B.2.3 AHCPU520-EN / AHCPU520-RS2

Device type	Address range	
X	Word: X0~X255	Bit: X0.0~X255.15
Y	Word: Y0~Y255	Bit: Y0.0~Y255.15
D	Word: D0~D65535	Bit: D0.0~D65535.15
L	Word: L0~L65535	Bit: L0.0~L65535.15
M	M0~M8191	
S	S0~S2047	
C	C0~C2047	
HC	HC0~HC63	
T	T0~T2047	
E	E0~E31	
SR	SR0~SR2047	
SM	SM0~SM2047	
Number of memory blocks which can be assigned to function block instances	512	

B.2.4 AHCPU530-EN / AHCPU530-RS2

Device type	Address range	
X	Word: X0~X511	Bit: X0.0~X511.15
Y	Word: Y0~Y511	Bit: Y0.0~Y511.15
D	Word: D0~D65535	Bit: D0.0~D65535.15
L	Word: L0~L65535	Bit: L0.0~L65535.15
M	M0~M8191	
S	S0~S2047	
C	C0~C2047	
HC	HC0~HC63	
T	T0~T2047	
E	E0~E31	

Device type	Address range
SR	SR0~SR2047
SM	SM0~SM2047
Number of memory blocks which can be assigned to function block instances	1024

MEMO

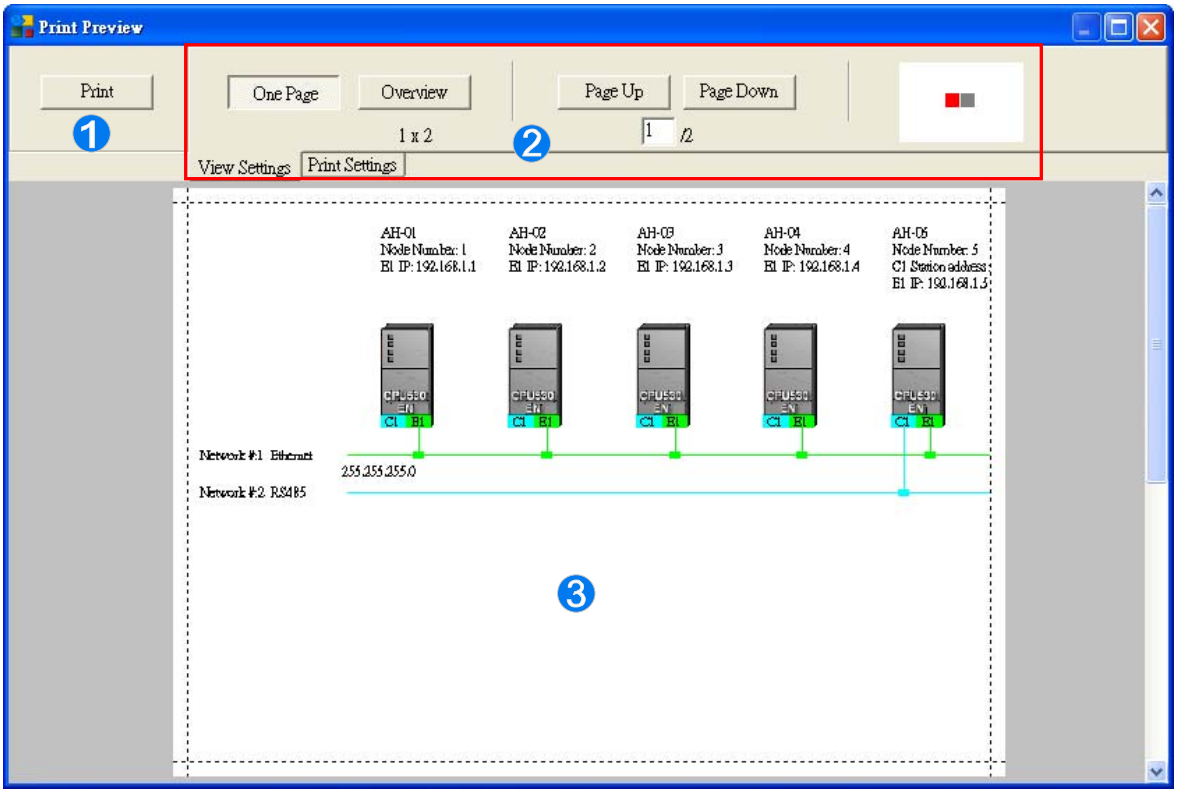
Appendix C Print Management Tool

Table of Contents

C.1 Introduction of the Environment C-2
C.2 Introduction of the Setting Area C-3

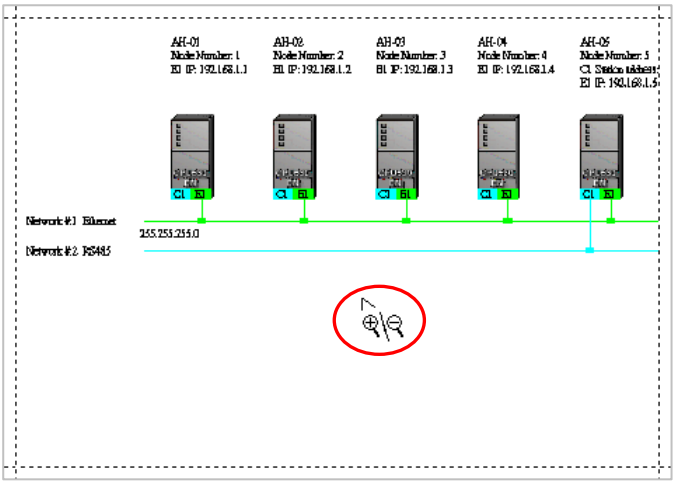
C.1 Introduction of the Environment

The print management tool provides a convenient operation interface. Users can set a print format, and preview the document which will be printed in the **Print Preview** window.



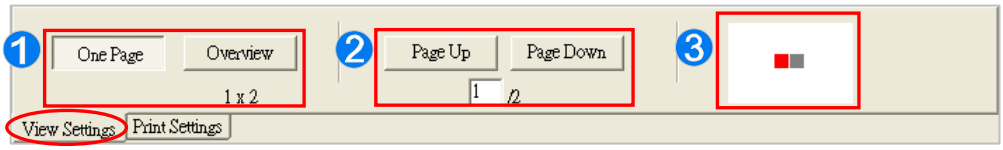
1. **Print button**: After the setting is complete, users can click the **Print** button to start the printing of the document.
2. **Setting area**: There are two tabs in the setting area. One is the **View Settings** tab, and the other is the **Print Settings** tab.
3. **Preview area**: The document which will be printed is displayed in this area.

If users move the mouse cursor to the preview area and click the left mouse button, the pointer will become a magnifying glass. After the left mouse button is clicked, ISPSoft will zoom in on the preview area. After the right mouse button is clicked, ISPSoft will zoom out on the preview area.



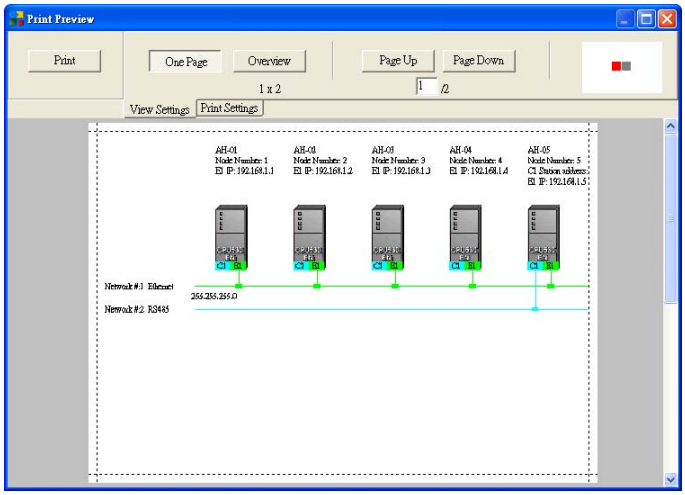
C.2 Introduction of the Setting Area

After users click the **View Settings** tab, they can adjust the display state of the preview area.

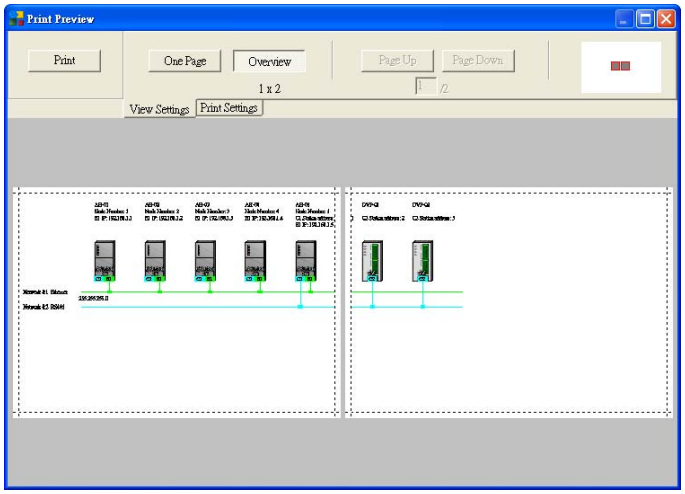


1. If the print area does not fit the printed page, the system will divide the print area into several pages. After **One Page** is clicked, one page will be displayed. After **Overview** is clicked, all the pages will be displayed. The number of pages into which the print area is divided is under the **Overview** button.
2. Users can select a page which will be displayed by clicking **Page Up** or **Page Down**. The present page number and the total number of pages are under the buttons.
3. The graphic representations show the relative positions of all the pages. The red block represents the present page.

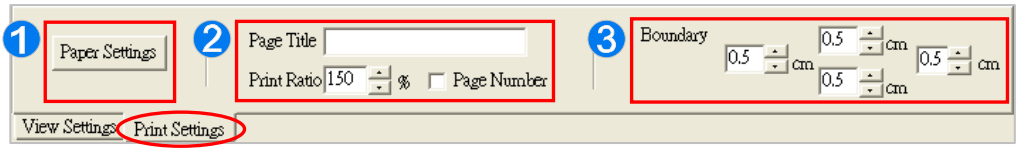
- **One page**



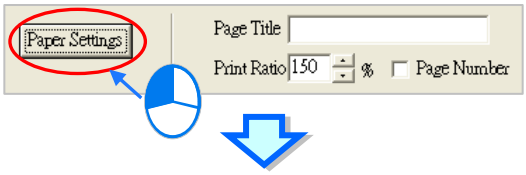
- **Overview**

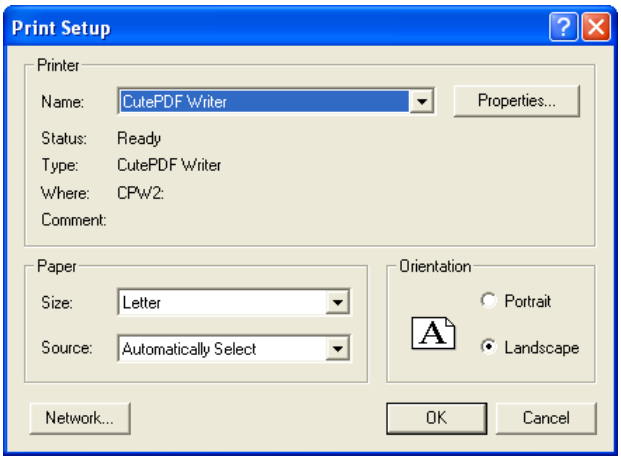
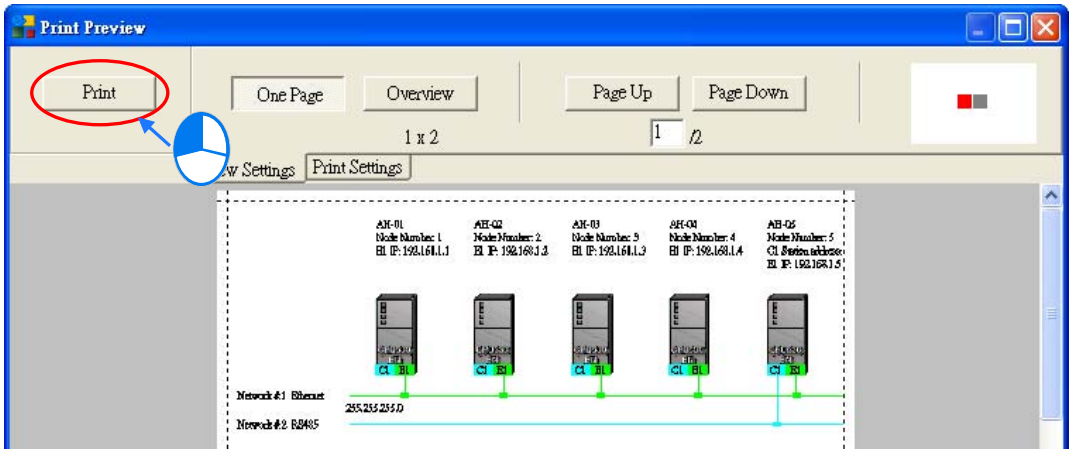


After users click the **Print Setting** tab, they can set the print format.



1. After **Paper Settings** is clicked, the **Print Setup** window will appear.
2. Users can scale the document by selecting a percentage in the **Print Ratio** box. The users can also insert page numbers and a page title in the document.
3. User can set document margins they want.





After the setting is complete, users can click the **Print** button to start the printing of the document.